

Nouveautés de yacs Martin

Version du 24 février 2010

Table des matières

1. Améliorations visibles.....	6
1.1. Nouvelles possibilités de mise en page.....	6
1.2. Amélioration des calendriers.....	6
1.3. Navigation depuis un téléphone mobile.....	7
1.4. Mise en évidence du thème courant.....	7
1.5. Construction à la souris de thème visuel.....	7
1.6. Organisation flexible des composants de page.....	8
1.7. Alignement des règles de style.....	8
1.8. Regroupement des images de référence.....	8
1.9. Illustration des commandes.....	9
1.10. Améliorations du panneau de contrôle.....	9
2. Améliorations de contenu.....	10
2.1. Les groupes, les blogs, et les pages privées.....	10
2.2. Duplication de section.....	10
2.3. Les commentaires « à la Facebook ».....	10
2.4. Les commentaires « à la Google Mail ».....	10
2.5. Signature dynamique des commentaires.....	11
2.6. Amélioration des étiquettes.....	11
2.7. Amélioration du suivi des problèmes.....	11
2.8. Affichage des profils dans une catégorie.....	12
2.9. Informations publiques, communautaires, privées.....	12
2.10. Propriétaire de section, d'article.....	13
2.11. Affichage des pages récentes dans une section	13
2.12. Partage sur les réseaux sociaux.....	13
2.13. Liens de voisinage.....	14
2.14. Visualisation de type wiki.....	14
2.15. Formulaire simplifié de saisie de contenu.....	14
2.16. Extension de publication.....	14
2.17. Nouveaux codes yacs.....	15
2.18. Intégration d'Open Flash Chart.....	15
2.19. Gestion des images.....	16
2.20. Améliorations dans la gestion des fichiers.....	16

2.21. Visibilité des notifications.....	17
2.22. Simplification de la gestion des notifications.....	17
2.23. Notifications et mode brouillon.....	17
2.24. Messages aux participants.....	17
2.25. Messages aux membres d'une catégories.....	18
3. Autres améliorations.....	19
3.1. Centralisation de la gestion des droits.....	19
3.2. Nouvelles fonctions liées à la gestion des droits.....	19
3.3. Liens canoniques.....	20
3.4. Permanence des fils de nouvelles.....	20
3.5. Optimisation du code d'affichage des pages.....	21
3.6. Extension de l'interface overlay.....	21
3.7. Ouverture à des scripts fournis par des tiers.....	22
3.8. Sauvegarde de paramètres dans les conteneurs.....	22
3.9. Héritage des options.....	22
3.10. Moins de transactions réseau.....	22
3.11. Moins de requêtes dans la base de données.....	22
3.12. Extension des scripts inclus.....	23
3.13. Cache dans des fichiers.....	23
3.14. Taille maximum des fichiers.....	23
3.15. Support de PHP 5.....	23
3.16. Amélioration des tests système.....	23
3.17. Automatisation de la mise à jour.....	24
4. Fichiers supprimés.....	25
5. Fonctions supprimées ou changées.....	26
5.1. are_allowed(), dans actions/actions.php.....	26
5.2. are_allowed(), dans articles/articles.php.....	26
5.3. are_allowed(), dans categories/categories.php.....	26
5.4. are_allowed(), dans comments/comments.php.....	26
5.5. are_allowed(), dans files/files.php.....	26
5.6. are_allowed(), dans images/images.php.....	26
5.7. are_allowed(), dans links/links.php.....	26
5.8. are_allowed(), dans locations/locations.php.....	26
5.9. are_allowed(), dans sections/sections.php.....	26
5.10. are_editable(), dans comments/comments.php.....	26

5.11. build_user_contact(), dans shared/surfer.php.....	27
5.12. build_user_contact(), dans skins/skin_skeleton.php.....	27
5.13. check_default_editor(), dans shared/surfer.php.....	27
5.14. get_action_label(), dans shared/anchors.php.....	27
5.15. is_editable(), dans articles/article.php.....	27
5.16. is_interactive(), dans articles/article.php.....	27
5.17. list_articles_for_member_by(), dans shared/members.php.....	27
5.18. list_editors_by_name_for_member(), dans shared/members.php.....	27
5.19. list_sections_for_user(), dans shared/members.php.....	27
5.20. personal_sections(), dans shared/surfer.php.....	28
5.21. stat_for_author(), dans articles/articles.php.....	28
5.22. stat_articles_for_anchor(), dans shared/members.php.....	28
5.23. stat_sections_for_anchor(), dans shared/members.php.....	28
5.24. stat_users_for_anchor(), dans shared/members.php.....	28

A propos de l'auteur

Bernard Paques est le créateur initial du système yacs, l'architecte du logiciel, et son principal développeur. Il est aussi webmestre de plusieurs serveurs, tous construits avec yacs, sur plate-forme Linux. En véritable praticien des plate-forme web dynamiques, il a formalisé son expérience dans ce guide, ensemble de bonnes pratiques développées sur le terrain à travers de multiples projets.

Pour en savoir plus sur l'auteur : <http://www.linkedin.com/in/webnetworker>

Remerciements

Ce document n'aurait pas vu le jour sans la communauté yacsienne. Merci aux muses, aux relecteurs, et à tous ceux qui ont pris un peu de leur temps pour améliorer ce guide.

Pour rejoindre la communauté : <http://www.yacs.fr>

Copyright et licence d'utilisation

Copyright © 2010 Bernard Paques

Ce guide est distribué selon les termes de la licence Creative Commons by-nc-sa 3.0. En conséquence, vous êtes autorisé à copier, modifier, et distribuer publiquement ce document, aux conditions suivantes :

- Attribution – Vous devez reconnaître le travail de l'auteur, en préservant les marques de copyright placées dans le document.
- Non-commercial – Vous ne pouvez pas valoriser ce travail à travers une activité commerciale sans l'autorisation de l'auteur.
- Partage similaire – Si vous modifiez ou réutilisez ce travail, vous pouvez distribuer le document obtenu, mais seulement sous la même licence ou sous une licence similaire.

1. Améliorations visibles

Cette section présente les nouveautés qui intéresseront surtout les infographistes et les web designers. Ceux qui s'intéressent avant tout à l'identité visuelle de leur site.

1.1. Nouvelles possibilités de mise en page

Comme tout gestionnaire de contenu qui se respecte, yacs fournit un grand nombre d'options pour affiner la mise en page des informations. Plusieurs layouts (c'est le nom des mises en page dans yacs) ont été introduits avec yacs Martin pour enrichir la gestion des pages :

- `accordion` – Une seule page est visible à la fois. En cliquant sur un titre on fait apparaître son contenu, et le contenu de la page affichée précédemment est caché.
- `carrousel` – Avec un contrôleur en Flash qui fait tourner les images à l'écran
- `iui` – Cette mise en page spéciale est réservée aux accès depuis un mobile, genre iPhone
- `newspaper` – A la façon d'un journal, pour mettre en valeur les dernières nouvelles
- `rights` – Mise en page spéciale, utilisée en interne par yacs pour afficher les rôles d'une personne (observateur, éditeur, propriétaire)
- `spray` – Une mise en page indissociable de l'overlay issue, pour construire un outil de suivi des problèmes
- `tagged` – Avec les étiquettes affichées prêt des titres, pour les informations classifiées
- `titles` – Les titres et les vignettes de page, et puis c'est tout !

Les sections ne sont pas en reste, avec les nouveaux layouts suivants :

- `accordion` – Une seule sous-sections est visible à la fois. En cliquant sur un titre on fait apparaître son contenu, et le contenu de la section affichée précédemment est caché.
- `carrousel` – Avec un contrôleur en Flash qui fait tourner les images à l'écran
- `comma` – Titres séparés par des virgules
- `rights` – Mise en page spéciale, utilisée en interne par yacs pour afficher les rôles d'une personne (observateur, éditeur, propriétaire)
- `slashdot` – L'équivalent d'une mise en page existante pour les articles, mais pour les sections

Les mises en page qui existaient précédemment ont toutes été revues et optimisées, soit pour les performances, soit pour le rendu visuel, soit pour préciser l'usage des overlays, soit pour une combinaison de ces améliorations.

Les mises en page spécifiques à la page d'accueil du site ont aussi été retouchées. Le layout `boxesandarrows` a été remplacé par le layout `hardboiled`, beaucoup plus performant pour afficher les nouvelles que son prédécesseur.

1.2. Amélioration des calendriers

Les vignettes des pages, si elles existent, sont incluses dans les calendriers dynamiques. Créer un événement attractif devient aussi facile que d'ajouter une page pour chaque date, et de doter chaque page

d'une image qui lui est propre. Yacs Martin fait le reste !

La construction des calendriers est aussi plus solide qu'avec yacs Macnana. Par exemple les requêtes tiennent mieux compte des sections et des pages privées, ce qui permet de gérer des calendriers particuliers à un groupe de personnes.

Les événements affichent la date dans le titre, grâce aux nouvelles possibilités de l'interface overlay. Il n'y a plus besoin de répéter la date dans le titre manuellement, ni de visiter la page d'un événement pour en connaître la date. L'overlay `day` fourni avec yacs Martin combine le titre et la date en fonction du point de vue sur l'information.

1.3. Navigation depuis un téléphone mobile

Oui, vous avez bien lu, yacs Martin facilite la navigation d'un serveur yacs depuis un téléphone intelligent comme ceux fournis par Apple ou Google !

Le logiciel embarque une librairie Javascript, disponible sous forme de logiciel libre bien sûr, qui sait construire des interfaces visuelles « à la iPhone ». Et un peu de code a été ajouté à yacs pour détecter les mobiles, et pour mettre en oeuvre cette nouvelle librairie. Essayez de consulter <http://www.yacs.fr> depuis votre mobile, et voyez par vous-même.

La page d'accueil est remplacée par une liste navigable à l'aide du pouce, qui contient les mêmes articles que ceux fournis dans les fils de nouvelles RSS et ATOM. En sélectionnant un article on affiche son contenu dans un format adapté aux petits écrans.

Sur le plan technique, le script de construction de la page d'accueil détecte les mobiles et se débranche vers un script spécialisé appelé `index_on_mobile.php`. Il en va de même pour le script d'affichage des pages, qui débranche vers `articles/view_on_mobile.php`. Un nouveau thème spécial, appelé `skins/_mobile`, a aussi été introduit pour faire le lien avec la librairie Javascript iui, qui fait tout le boulot côté terminal. Et voilà !

Bien sûr, tout ceci reste encore expérimental, et l'interface permet seulement la consultation à ce stade du développement. A quand la prochaine version majeure, qui permettra d'ajouter un commentaire ou de bloguer facilement directement depuis son mobile ? Pour cela, il faudra se faire reconnaître, et donc implémenter un formulaire d'authentification adapté. Il y a encore du travail, et les volontaires ayant de bonnes connaissances sur les mobiles sont les bienvenus.

1.4. Mise en évidence du thème courant

L'index des thèmes (géré par `skins/index.php`) présente l'ensemble des thèmes disponibles sur le serveur. La nouveauté, c'est que la vignette du style courant est encadrée d'un trait épais. Auparavant, tous les thèmes apparaissaient de la même façon.

1.5. Construction à la souris de thème visuel

La construction d'un thème visuel est une opération complexe, parce qu'elle requière de nombreuses briques techniques. Difficile d'arriver à ses fins lorsque l'on a un peu de mal avec les fichiers CSS, les images associés, et l'inspection de page avec Firebug. La boîte à outil existe, elle est très puissante, mais elle est aussi peu maniable pour un non-initié. Le plus souvent, les web designers dérivent un thème à partir d'un thème existant, puis retouchent progressivement les fichiers.

Tout change avec yacs Martin, avec l'introduction du thème flexible. L'originalité de ce thème est qu'il est doté d'un panneau de configuration, où se fait le paramétrage du rendu visuel. Lorsqu'un associé du site change les paramètres dans le panneau de configuration, plusieurs fichiers sont modifiés, dont une feuille de styles et un fichier de paramètres. Ce sont ces fichiers qui sont utilisés par la suite pour le rendu effectif

des pages.

Quelques-unes des fonctionnalités du thème flexible :

- largeur fixe ou variable de la mise en page
- deux ou trois colonnes, placées à gauche ou à droite
- téléversement d'images pour compléter la librairie de base
- choix parmi une trentaine de styles pour les onglets
- configuration fine des en-têtes de page et des bas de page
- grand choix d'images de fond pour les différents blocs dynamiques générés par yacs
- palette visuelle de choix de couleurs pour les liens, les titres, etc (jscolor)

Le thème flexible est fourni avec une bibliothèque d'images qui permet de réaliser des sites très différents les uns des autres. Le panneau de configuration gère 179 paramètres différents, c'est dire la richesse des combinaisons possibles.

1.6. Organisation flexible des composants de page

Les composants de page sont à présent rangés en mémoire avant la construction des pages dynamiques. Un panneau de configuration permet de modifier l'ordre d'apparition des composants dans les différentes zones de la page. Il est même possible d'ajouter une page nommée dans la liste des composants de page, si besoin.

Sur le plan technique, yacs Macnana plaçait les boîtes de côté dans la variable `$context['aside']`. Avec yacs Martin, tous les composants sont placés dans la variable `$context['components']` et l'assemblage final est fait dans `skins/page.php`.

L'assemblage se règle dans le panneau de configuration du moteur de rendu visuel, implémenté dans `skins/configure.php`.

1.7. Alignement des règles de style

Les dénominations des règles de style sous-jacentes à yacs, et des images, ont été harmonisées pour faciliter le travail de design. Lorsque cela est possible, les règles de style sont nommées d'après les mises en page qui les utilisent.

De plus, le fichier CSS principal de yacs, c'est-à-dire le fichier `skins/_reference/yacs.css`, contient à présent l'essentiel des styles générés par yacs. Cette consolidation est destinée autant à simplifier les évolutions futures qu'à limiter au strict minimum le contenu des thèmes développés par les web designers.

Les tables gérées par la feuille de style de référence de yacs sont dotées de bordure par défaut, il n'est plus besoin d'ajouter manuellement la classe `grid` aux tables créées dans TinyMCE comme précédemment. Une nouvelle classe `table.layout` est définie dans `skins/_reference/yacs.css` pour les tables sans bordure.

1.8. Regroupement des images de référence

Initialement, yacs utilisait les images placées dans le répertoire du thème en cours. La seule solution pour personnaliser le logiciel était de dupliquer entièrement un thème existant pour, ensuite, changer quelques fichiers. A un certain point, un assez grand nombre d'images furent regroupées dans le répertoire `skins/images`. Puis, plus récemment, et avec l'introduction d'AJAX, de nouvelles images partagées furent

placées dans le répertoire `skins/_reference`. Bref, yacs Macnana offrait de grandes possibilités de personnalisation, mais au prix d'une complexité importante de l'organisation des images.

Tout ceci change radicalement avec yacs Martin, puisque toutes les images de référence sont rangées dans l'un des sous-répertoires de `skins/_reference`. Le répertoire `skins/images` est supprimé, et la plupart des images qui restaient dans les répertoires des thèmes fournis avec yacs ont été transférées vers `skins/_reference`.

La standardisation du répertoire de référence s'accompagne d'une innovation majeure, puisqu'elle s'intègre à la fonction de chargement d'image. Concrètement, cela signifie que lorsqu'il doit charger une image, yacs regarde d'abord dans le répertoire du thème courant. S'il ne trouve rien, il charge l'image de référence. Voir la fonction `define_img()` dans `skins/skin_skeleton.php` pour analyser le fonctionnement.

Le résultat, c'est qu'il est possible de créer un thème à partir d'un répertoire à peu près vide, avec juste un fichier `template.php` pour finaliser les pages, un fichier `skin.php` minimal, et une feuille de style pour charger `skins/_reference/yacs.css`. A partir de là, tout est possible. Le graphiste peut créer, dans le répertoire du thème, les quelques images qui lui permettront de personnaliser le rendu visuel. Où il peut répliquer l'intégralité du répertoire de référence dans le répertoire du thème, puis changer les images. La démarche de création et de personnalisation de l'interface devient beaucoup plus progressive et logique.

Bien sûr, la mutualisation des images de référence s'accompagne d'une réduction très significative du nombre de fichiers fournis dans l'archive de yacs.

1.9. Illustration des commandes

Les commandes mises à disposition dans la boîte à outils, et dans d'autres boîtes de côté, se sont étoffées. Sur le plan ergonomique, chaque lien textuel a été doté d'une petite image (16 pixels par 16 pixels) qui facilite la compréhension et l'accès aux commandes. Ces images sont toutes extraites de la fameuse librairie FamFamFam créée et diffusée gratuitement par Mark James.

1.10. Améliorations du panneau de contrôle

L'onglet qui conduit aux panneaux de configuration est affiché par défaut, puisqu'il semble que ce soit le plus utilisé par les associés. Un clic de gagné, parce que le temps des webmestres, c'est de l'argent !

Une nouvelle commande est ajoutée pour faciliter le téléversement d'un fichier au format `.csv`, par exemple pour peupler le contenu de l'une des tables de la base de données. Cette fonction d'importation simple existait déjà avec yacs Macnana, mais elle n'était pas encore intégrée dans le panneau de contrôle. C'est à présent chose faite.

2. Améliorations de contenu

2.1. *Les groupes, les blogs, et les pages privées*

Chaque type de contenu est placé dans une section séparée. Yacs Martin crée dans la carte du site une section intitulée 'Groupes', une autre appelée 'Blogs', et une autre nommée 'Discussions'.

La création d'un nouveau groupe, ou d'un nouveau blog, se fait par un utilisateur depuis son propre profil, à condition que cette fonction ait été autorisée dans le panneau de configuration des utilisateurs.

L'utilisateur doit indiquer s'il souhaite créer un espace public ou privé, indiquer le titre et le champ d'introduction, et le script `sections/new.php` automatise le reste des étapes de création. Par exemple, un groupe est structuré autour d'un « mur » de commentaires, et d'un forum de discussion. Une page « collante » est ajoutée dans le forum pour décrire les règles de fonctionnement du groupe.

Note : la section nommée 'Pages privées' n'est plus utilisée, et il est possible de la recycler en changeant son titre en 'Discussions', en modifiant son nom de 'private' à 'threads', et en ajoutant l'overlay `thread` aux pages qui y sont créés. L'overlay `thread` sert à afficher la liste des participants à une page privée, c'est lui qui indique sur votre profil que, pour la page XYZ, vous êtes « avec Pierre, Paul et Jacques »

2.2. *Duplication de section*

Cette fonction était attendue, elle est implémentée dans le fichier `sections/sections.php`, et elle servira notamment aux propriétaires de section à préparer des modèles de sous-section, y compris le contenu-type (pages, fichiers, etc) et l'organisation des droits d'accès (assignation d'éditeurs)

2.3. *Les commentaires « à la Facebook »*

La nouvelle version de yacs s'inspire partiellement des innovations introduites par le célèbre réseau social, notamment l'organisation des contributions sur un « mur ».

Plus précisément, yacs Martin fournit une option '`comments_as_wall`' qui, lorsqu'elle est activée sur une page, modifie la gestion des commentaires comme suit :

- les commentaires sont listés dans l'ordre inverse de leur création (le plus récent en haut, et le plus ancien en bas),
- un champ de saisie d'un nouveau commentaire est inséré en tête de liste, avec possibilité d'attacher un fichier.

Cette option est particulièrement utile pour les fils de discussion privés, parce qu'elle facilite l'accès aux contributions les plus récentes, même sur des pages constituées de plusieurs centaines de commentaires. De plus, le formulaire est particulièrement engageant, ce qui favorise les interactions entre membres.

2.4. *Les commentaires « à la Google Mail »*

Le layout de mise en page compact des commentaires affiche à présent les premiers mots du contenu du commentaire. Bien sûr, ceci s'inspire de ce que fait Google Mail pour afficher les messages dans la boîte aux lettres. Cet apport n'a l'air de rien, mais il facilite grandement le repérage d'une contribution dans un flot important d'informations. Voir `comments/layout_comments_as_compact.php`

Cette façon de procéder a aussi été intégrée dans les notifications de nouveaux commentaires. Si le commentaire est vraiment très court, il est contenu tout entier dans la notification. S'il est plus long, les

premiers mots donnent une bonne idée de la teneur du commentaire, ce qui facilite grandement sa gestion adaptée.

2.5. Signature dynamique des commentaires

Voici une autre demande populaire qui est comblée avec yacs Martin.

Précédemment, la signature était placée à la fin du commentaire, et le tout était sauvé dans la base de données, au moment de la création du commentaire. Maintenant, yacs ajoute dynamiquement les signatures au moment où il construit la liste des commentaires.

Cette façon de faire est certes plus coûteuse en performance, mais elle offre l'immense avantage de dynamiser considérablement l'usage des signatures. Changer sa signature devient un acte de communication important, puisque tous vos commentaires vont se mettre à l'afficher immédiatement après le changement. Si vous avez un message à faire passer, vous savez à présent comment vous y prendre.

Les signatures dynamiques simplifient aussi la maintenance des liens et des images utilisées. Précédemment, il était plutôt difficile de revenir manuellement sur les commentaires signés pour corriger un lien devenu obsolète, ou pour empêcher le chargement d'une image disparue depuis longtemps. Maintenant, il suffit de modifier son profil pour régler ce genre de problèmes en quelques secondes.

2.6. Amélioration des étiquettes

Les étiquettes ont été introduites avec succès dans yacs Macnana, comme un moyen simple de faire participer les usagers à la classification de l'information. Pour chaque étiquette, le logiciel crée si besoin un lien vers la catégorie associée.

Avec yacs Martin, la modification d'une étiquette entraîne aussi la suppression du lien vers l'ancienne catégorie. Ce mécanisme simple a pour but, bien sûr, d'éviter l'accumulation de catégorisations obsolètes.

L'étiquetage a été étendu aux sections.

Avec yacs Martin, les catégories et les étiquettes peuvent être colorisées ! Choisissez une couleur dans le formulaire de modification d'une catégorie, et elle sera affichée à l'arrière-plan de tous les liens associés. Cette fonction est particulièrement utile pour attirer l'attention des internautes sur un sujet particulier, en distinguant les étiquettes associées.

2.7. Amélioration du suivi des problèmes

L'overlay `issue` avait été introduit dans une version précédente de yacs pour structurer la capture et le traitement des incidents. C'est l'outil utilisé pour gérer les remontées de bogues sur www.yacs.fr par exemple. Voir `overlays/issue.php` pour plus de détail.

Ce script a été profondément remanié avec yacs Martin pour augmenter ses fonctionnalités. Ceci a été rendu possible grâce à l'évolution de l'interface overlay, et par ajout de l'overlay `articles/layout_articles_as_spray.php`.

Cette combinaison fournit un véritable système de gestion et de suivi des problèmes, offrant des caractéristiques assez avancées :

- catégorisation initiale du problème : incident, maintenance, amélioration, développement
- la catégorisation peut être revue et modifiée par la suite
- le responsable du problème est aussi le propriétaire de la page
- le responsable du problème peut changer à tout moment, et cette action est aussi enregistrée sous forme de commentaire

- la saisie du nom du nouveau responsable est facilitée par l'auto-completion (liste progressive des membres qui correspondent à la saisie)
- les champs de suivi du problème sont disponibles dans un onglet séparé du formulaire de modification de page
- lorsque les champs de suivi sont modifiés, ceci est mémorisé automatiquement dans un commentaire
- les commentaires manuels restent possibles, bien entendu
- l'ensemble des commentaires manuels et automatiques constituent l'historique complet du traitement de l'incident
- le résumé du suivi, le nom du responsable, et l'avancement du traitement sont résumés en début de page
- le titre de chaque page mentionne l'état d'avancement du traitement (en cours, résolu, ...)
- une barre de progression visualise l'avancement du traitement (de 0 à 100%) dans la liste des problèmes

Pour se rendre compte de l'efficacité obtenue, le mieux est de consulter un exemple concret de résolution d'un bug :

<http://www.yacs.fr/article-6844-destinataires-d-une-lettre-d-information>

Comment créer un système de gestion des problèmes (« Bug Tracking System » en anglais) sur votre site ? Créez une section en demandant que les pages soient basées sur l'overlay `issue`, et listées avec la mise en page `spray`. Assignez les responsables du suivi à cette section, et c'est tout !

Et puisque « charité bien ordonnée commence par soi-même », nous avons utilisé ces nouvelles possibilités pour gérer les bugs résiduels dans yacs Martin RC5 :

<http://www.yacs.fr/section-489-rc5-livree-test-de-yacs-10.2.12-alias-yacs-martin>

2.8. Affichage des profils dans une catégorie

Les catégories servent à constituer des listes de sections, de pages, ou d'utilisateurs. Avec yacs Martin, il devient possible de choisir un layout particulier pour lister les personnes assignées à une catégorie. Avec un peu de développement PHP, on pourra adapter le rendu visuel des catégories à toutes sortes de besoins pour réaliser des annuaires à la carte, avec ou sans photo, avec plus ou moins d'informations sur les personnes, etc.

2.9. Informations publiques, communautaires, privées

Ces libellés ont été généralisés dans tout le code pour simplifier la gestion des droits d'accès. En résumé :

- Public – tout internaute, et les moteurs de recherche
- Communautaire – pour les personnes authentifiées (membres et souscripteurs)
- Privé – pour les personnes (membres et souscripteurs) assignées explicitement, et les associés

Les restrictions d'accès se règlent indépendamment sur chaque section, chaque article et chaque fichier. Par

défaut, les informations sont en accès public. Toutefois, les restrictions sont aussi transmises au contenu, ce qui signifie que yacs combine le réglage d'accès d'un élément avec celui du conteneur, et retient le plus restrictif. Avec yacs Martin, le formulaire de modification précise ces différents cas dans les messages d'aide à la saisie, pour faciliter la compréhension des utilisateurs.

Grâce à l'héritage des restrictions d'accès, il n'y a pas besoin de gérer explicitement les droits pour chaque information. Généralement, il suffit de préparer quelques sections publiques, plus quelques sections communautaires ou privées, pour arriver à ses fins.

Un autre problème commun est l'élargissement des accès à une page placée dans une section privée. Si vous pouvez définir explicitement la liste des ayants droits, alors le mieux est d'utiliser la fonction 'Inviter des participants' directement. Sinon, si vous devez ouvrir l'accès à tous les membres, voir à tous les internautes, il faudra déplacer l'information dans une section communautaire ou publique.

2.10. Propriétaire de section, d'article

Un nouveau champ a été ajouté aux sections et aux articles pour mémoriser le propriétaire de l'information. Par défaut, le propriétaire d'une information est son créateur. Ce droit peut être délégué à tout moment à une autre personne par le propriétaire lui-même, ou par un associé.

Pour changer le propriétaire d'une section ou d'une page il faut passer par le formulaire de modification, puis sélectionner l'onglet Options. Yacs affiche le propriétaire en cours, et propose un bouton pour le modifier. Le changement est géré par le script `sections/own.php` pour les sections et par le script `articles/own.php` pour les articles.

Le propriétaire d'une section dispose aussi des droits de propriété sur l'ensemble du contenu, y compris les sous-sections et les articles. Ainsi, il est très courant que plusieurs personnes différentes puissent agir en tant que propriétaire d'une page donnée, et cette approche collective doit être prise en compte dans la gestion des droits mise en place sur un serveur.

Par exemple, le responsable d'un département pourra créer une section pour ses équipes, puis créer une sous-section pour chacune des équipes qu'il gère. A leur tour, les responsables d'équipe seront propriétaires de leur espace d'équipe, et pourront s'ils le souhaitent déléguer la responsabilité de certaines pages à un de leurs équipiers. Grâce à la gestion des droits de propriété, yacs martin permet une transposition facile des responsabilités organisationnelles dans l'espace web.

Par rapport aux versions précédentes, certains droits donnés aux éditeurs sont à présent dévolus aux seuls propriétaires. Par exemple, les éditeurs ne peuvent plus inviter de nouveaux participants, seuls les propriétaires peuvent le faire.

2.11. Affichage des pages récentes dans une section

Par défaut, les versions précédentes affichaient, sur une section, les pages les plus récentes postées dans les sous-sections. L'option '`no_new_articles`' devait être positionnée pour empêcher cet affichage.

Avec yacs Martin, le comportement est inversé, et la liste des pages récentes est affichée seulement si l'option '`with_deep_news`' est positionnée.

2.12. Partage sur les réseaux sociaux

Pour les sites visibles depuis Internet, yacs Martin offre de partager une page, une section, un profil ou une catégorie sur les réseaux sociaux du moment : Facebook, Twitter et LinkedIn. Les visiteurs sont invités à faire le buzz des informations publiques postées sur votre site web !

De plus, il est possible d'indiquer son compte twitter sur son profil, et yacs ajoutera l'icône Twitter juste en-dessous de votre photo sur tous les commentaires que vous ferez.

2.13. Liens de voisinage

Dans les versions précédentes de yacs, l'affichage d'une page provoquait systématiquement l'intégration de liens vers les pages précédente et suivante dans la même section. Pour empêcher ce comportement il fallait ajouter l'option `'no_neighbours'` au niveau de la section.

Avec yacs Martin ce comportement est inversé. Les liens de navigation vers les pages précédente et suivante sont affichées seulement dans les sections qui ont l'option `'with_neighbours'`.

De plus, les liens de voisinage sont placés au début et à la fin de la zone d'affichage principale, au milieu de la page. Précédemment, ces liens étaient placés le plus souvent dans une boîte de côté.

La nouvelle approche est, bien sûr, destinée à faciliter la mise en place de « livres électroniques » constitués de pages à lire en séquence.

2.14. Visualisation de type wiki

Une nouvelle option `'view_as_wiki'` fait son apparition pour faciliter la collaboration de plusieurs intervenants sur la même page web. L'idée est d'utiliser les commentaires attachés à la page pour synchroniser les contributeurs, mais indépendamment du contenu « utile » de la page.

Cette visualisation, implémentée dans le script `articles/view_as_wiki.php`, s'inspire de celle mise en oeuvre sur Wikipedia. Elle se caractérise par deux onglets placés tout en haut de la page, avant le titre. Le premier onglet, qui est celui affiché par défaut, présente le contenu de la page : le titre, l'introduction, la description, les fichiers attachés. Le deuxième onglet, lui, est réservé aux commentaires. Le fil de discussion est donc séparé du contenu, il sert uniquement pour demander des modifications, pour gérer les apports, ou pour documenter les actions effectuées.

2.15. Formulaire simplifié de saisie de contenu

Les versions précédentes de yacs permettaient d'intégrer tout type de script externe pour la visualisation de pages, par le biais des options de configuration. Par exemple, sur l'option `'view_as_tabs'`, la visualisation ordinaire est remplacée par le chargement du script `articles/view_as_tabs.php`. Dans le cadre d'un projet particulier, il est possible d'utiliser l'option `'view_as_truc'`, à condition de fournir à yacs un script `articles/view_as_truc.php`.

Ce mécanisme d'extension simple et puissant a été étendu avec yacs Martin aux formulaire de saisie de l'information. L'option à mettre en oeuvre doit être préfixée par `edit_as_`. Par exemple, si vous souhaitez simplifier la saisie d'informations, vous pourrez utiliser l'option `'edit_as_simple'`, qui chargera le script `articles/edit_as_simple.php` en lieu et place du script usuel. Comme son nom l'indique, ce script, fourni avec yacs Martin, propose un formulaire simplifié, sans les onglets de tabulation ni les options compliquées. Vous pourrez vous inspirer de ce script pour créer toutes sortes de formulaires de saisie différents.

En combinant les options `views_as_` et `edit_as_`, vous contrôlez les scripts chargés par yacs pour la saisie et pour la restitution d'informations. Il devient extrêmement facile d'adapter le serveur aux besoins spécifiques d'une application client, d'autant plus que vous conservez la puissance des overlays et des layouts si nécessaire.

2.16. Extension de publication

Dans les versions précédentes de yacs l'extension de publication était appelée seulement lors de la publication explicite d'une page. Avec yacs Martin cette extension est aussi invoquée lors de la création d'une page en mode d'auto-publication (option `auto_publish`).

Ce changement est important, parce qu'il aide votre serveur à diffuser les nouvelles pages sur Internet. Précédemment, seules les publications explicites donnaient lieu à des requêtes 'ping'. À présent toute page publiée sera annoncée sur les serveurs que vous aurez paramétrés.

De plus, lors de la création d'une page publiée, yacs indique la liste des serveurs avertis de votre contribution. Vous pouvez donc vérifier, en temps réel, que la diffusion des informations publiées s'effectue correctement.

2.17. Nouveaux codes yacs

Vous vouliez intégrer plus facilement des fils de nouvelles externes dans un serveur yacs ? Vous vouliez pouvoir intégrer un flux Twitter dans une page ? Si c'est le cas, vous allez adorer yacs martin et ses nouveaux codes.

Une nouvelle page d'aide pour les widgets a été ajoutée avec le script `codes/widgets.php` et c'est là que seront décrits les widgets qui viendront avec les prochaines versions de yacs.

L'intégration de fils RSS ou ATOM dans yacs s'appuie à présent sur la librairie SimplePie, écrite en PHP. L'embryon de code qui faisait cela, mais moins bien, a été retiré. La fiabilité de la nouvelle librairie est une nécessité avec toutes les variantes XML placées par les uns et par les autres dans leurs fils de nouvelles. En particulier, yacs Martin est beaucoup plus solide que son prédécesseur pour l'intégration de fils ATOM. La librairie SimplePie est placée dans le fichier `include/simplepie.inc` de l'archive standard de yacs.

Liste des codes introduits dans yacs Martin :

- `[form=123]` – lien vers un formulaire
- `[table.json]` – formater un objet table au format JSON
- `[newsfeed]` – inclusion dynamique de flux RSS ou ATOM par requête côté client (AJAX)
- `[newsfeed.embed]` – inclusion dynamique de flux RSS ou ATOM par requête côté serveur (PHP)
- `[twitter]` – inclusion dynamique (AJAX) du flux d'un abonné Twitter
- `[tsearch]` – inclusion dynamique de flux de recherche Twitter (sur mot-clé)
- `[retweet]` – bouton incitant à propager le buzz sur Twitter
- `[chart]` – affichage d'un diagramme construit avec Open Chart Flash

2.18. Intégration d'Open Flash Chart

Open Flash Chart est un composant, écrit en Flash, qui permet de réaliser toutes sortes de diagrammes à l'écran. Il est distribué avec une licence compatible avec celle de yacs, et donc nous l'avons intégré dans la distribution standard de yacs Martin.

Vous trouverez quelques exemples de ce que l'on peut faire avec Open Flash Chart à l'adresse suivante :

<http://teethgrinder.co.uk/open-flash-chart/>

De plus, un nouveau code a été introduit dans yacs pour activer le composant Open Flash Chart et pour lui transmettre les données à visualiser. Il est donc possible, avec cet outillage, de créer des diagrammes spectaculaires, à partir du formulaire de saisie de page existant. Cette fonction reste complexe, donc réservée aux usagers avertis. Mais elle permet, si besoin, de réaliser des tableaux de bord attractifs sans ajout de briques logicielles externes.

Bien sûr, comme pour tous les codes importants de yacs, cette fonction est documentée dans une page d'aide, en l'occurrence `codes/misc.php`, et vous pouvez consulter celle du site www.yacs.fr à l'adresse suivante :

`http://www.yacs.fr/codes/misc.php`

2.19. Gestion des images

Les formulaires de saisie des informations ont été modifiés pour faciliter la gestion des différents médias, y compris les images. Par défaut, yacs intègre un code de rendu d'image dans le champ de description à chaque fois qu'une image est ajoutée à une page. En modifiant la page, on peut accéder à l'onglet Médias pour gérer la liste des images, et pour obtenir les codes d'intégration disponibles.

Une nouvelle fonction est proposée pour téléverser plusieurs images d'un seul coup. Le script `images/upload.php` sait recevoir et traiter une archive au format `.zip` contenant plusieurs fichiers. Ce script extrait les fichiers, les retaille si nécessaire, crée une petite vignette, et enfin intègre les codes d'appels dans la page cible. Comme son usage est a priori assez rare, il faut passer par la commande 'Ajouter une image' pour y accéder.

Les sections et les catégories ne proposent plus de gérer d'image de puce. Cette fonction était très rarement utilisée, et peut-être remplacée par un layout adapté, en cas de besoin.

La première image ajoutée à une page devient la vignette de cette page. Bien sûr, ceci peut être repris par la suite en modifiant la page.

2.20. Améliorations dans la gestion des fichiers

La réservation des fichiers, soit pour revue, soit pour modification, est devenue nettement plus simple et efficace dans yacs Martin. L'interface est plus explicite, même pour les débutants en informatique, et l'efficacité du travail collaboratif en est améliorée d'autant.

Le champ de description qui existait précédemment devient un historique des modifications effectuées sur un fichier. A chaque téléversement, il est possible d'indiquer en quelques mots la raison du changement, et cette information alimente l'historique. L'historique est restitué sous forme d'une boîte pliante, pour être à disposition tout en minimisant l'espace occupé à l'écran. Ainsi, même si yacs Martin continue de gérer une seule version de chaque fichier, il parvient, en pratique, à se rapprocher d'un petit système de gestion documentaire.

Lorsque le fichier est une vidéo ou un fichier MP3, yacs Martin sait intégrer un composant Flash pour le visualiser ou le jouer. Le partage de documents multimédia en est, bien sûr, considérablement facilité. Dans un fil de discussion, un contributeur peut ajouter un commentaire avec une petite vidéo attachée. Les autres utilisateurs cliqueront sur le composant pour lancer le film à l'écran, sans avoir besoin de le télécharger. Bien sûr, cette fonction est inspirée de YouTube, même si elle reste beaucoup moins sophistiquée. En particulier, seuls quelques formats de fichiers sont pris en compte, et le serveur ne sait pas transcoder les codecs. En pratique, si vos utilisateurs ont des Macs, ou s'ils savent générer du Flash vidéo et des `.mp3`, ils devraient pouvoir se faire plaisir. Voir la nouvelle fonction `Files::interact()` dans `files/files.php`

Lorsque le fichier téléversé est une image, yacs crée automatiquement une vignette réduite et l'affiche à côté du nom du fichier, à la place de l'icône standard.

Et pour tous les types de fichier, les traitements effectués après téléversement ont été consolidés dans une seule fonction `Files::upload()` dans `files/files.php`. L'occasion, encore une fois, de rationaliser le code et de le mutualiser autant que possible.

Les types de fichiers .docm, .docx, .pptm, .pptx, .xslm, .xlsx sont reconnus par yacs Martin.

Le contrôle d'accès sur les fichiers sécurisés est beaucoup plus strict que précédemment. En particulier, il n'est plus possible d'attacher un fichier privé et d'en donner l'accès grâce un code [download] ou [embed]. Pensez-y si certains utilisateurs se plaignent de ne plus pouvoir charger certains documents après la migration vers yacs Martin.

Dans les collections, les images affichées dans le slideshow sont triées par ordre alphabétique.

2.21. Visibilité des notifications

Les notifications ont été ajoutée au logiciel depuis longtemps. La nouveauté, c'est que yacs Martin indique au contributeur la liste des personnes notifiées après un ajout. Cette fonction donne beaucoup de visibilité aux auteurs, qui peuvent ainsi déterminer s'ils doivent inviter de nouveaux participants, etc.

2.22. Simplification de la gestion des notifications

Pour aider les usagers bombardés de notifications yacs Martin offre plusieurs mécanismes complémentaires :

Les message 'Oublier cette page' et 'Oublier cette section' ont été remplacés par un simple 'Arrêter les notifications', c'est plus clair !

2.23. Notifications et mode brouillon

yacs n'envoie plus de notifications pour les ajouts effectués sur les pages en mode brouillon. En revanche, la publication est à présent notifiée à tous les observateurs.

Il est donc possible de créer une nouvelle page, en mode brouillon, et de la peaufiner progressivement, y compris en ajoutant des images ou des pièces jointes, sans provoquer une seule notification. L'étape de publication sera aussi celle de la notification aux observateurs.

2.24. Messages aux participants

La version yacs Martin offre deux mécanismes distincts pour l'envoi de messages liés à une section ou à une page.

Le premier mécanisme sert à inviter de nouveaux participants à une section ou à une page. Un message est préparé par un propriétaire ou par un associé, à destination de certains membres de la communauté, ou à destination de personnes extérieures. Les membres sont aussi ajoutés à la liste des observateurs de la section ou de la page cible, ce qui signifie qu'ils seront automatiquement notifiés des ajouts effectués par la suite. Les invitations pour les articles sont gérées dans le script `articles/invite.php`, et celles pour les sections dans le script `sections/invite.php`.

Le deuxième mécanisme sert à envoyer un message électronique aux participants d'une section ou d'une page, en complément des notifications automatiques. Par exemple, le propriétaire d'une section pourra utiliser cette fonction pour rappeler l'échéance d'un travail collectif, ou un membre pour demander de l'aide aux autres membres. De plus, certains destinataires peuvent être désélectionnés en cas de besoin, ce qui permet un routage assez précis des messages, en fonction des besoins. Tout ceci est géré dans le script `articles/mail.php` pour les articles, et dans `sections/mail.php`.

Les invités peuvent utiliser la commande 'Arrêter les notifications' pour réguler les messages qu'ils reçoivent. Les éditeurs disposent aussi d'une commande 'Quitter cette page' pour renoncer à leur assignation.

2.25. Messages aux membres d'une catégories

Les catégories sont un moyen simple de créer des listes de personnes, comme par exemple tous les comptables de l'entreprise, ou les personnels travaillant de nuit, ou encore le pool des techniciens d'après-vente. Pour mettre à profit ces regroupements, yacs Martin permet d'envoyer un message aux personnes rattachées à une catégorie. Toutes les adresses sont listées dans le formulaire d'envoi de message, géré par `categories/mail.php`, et il est possible de déselectionner certaines d'entre elles en cas de besoin. Pour l'instant, cette fonction est réservée aux associés, et il est probable qu'elle soit étendue à d'autres rôles dans une prochaine version.

3. Autres améliorations

3.1. Centralisation de la gestion des droits

La gestion des droits de création et de modification a été centralisée dans quelques fonctions pour la plupart des objets gérés par yacs.

Types d'objets	Gestion des droits
Actions (dans <code>actions/actions.php</code>)	<code>Actions::allow_creation()</code>
Articles (dans <code>articles/articles.php</code>)	<code>Articles::allow_creation()</code> <code>Articles::allow_deletion()</code> <code>Articles::allow_display()</code> <code>Articles::allow_message()</code> <code>Articles::allow_modification()</code> <code>Articles::allow_publication()</code>
Catégories (dans <code>categories/categories.php</code>)	<code>Categories::allow_creation()</code>
Commentaires (dans <code>comments/comments.php</code>)	<code>Comments::allow_creation()</code> <code>Comments::allow_modification()</code>
Fichiers (dans <code>files/files.php</code>)	<code>Files::allow_creation()</code> <code>Files::allow_modification()</code>
Images (dans <code>images/images.php</code>)	<code>Images::allow_creation()</code>
Liens (dans <code>links/links.php</code>)	<code>Links::allow_creation()</code> <code>Links::allow_trackback()</code>
Emplacements (dans <code>locations/locations.php</code>)	<code>Locations::allow_creation()</code>
Sections (dans <code>sections/sections.php</code>)	<code>Sections::allow_creation()</code> <code>Sections::allow_message()</code> <code>Sections::allow_modification()</code>

3.2. Nouvelles fonctions liées à la gestion des droits

Les requêtes effectuées dans la base de données prennent systématiquement en compte les droits des membres sur les sections et les articles qui leur sont attribués.

De nouvelles fonctions sont introduites pour évaluer les droits sur un article : `is_assigned()`, `is_editable()`, `is_owned()` et `is_watched()`. Voir le fichier `articles/articles.php` pour plus de détails. L'équivalent existe pour les sections.

La classe statique `Surfer` est dotée de deux fonctions pour lister toutes les sections et tous les articles assignés à un utilisateur : `Surfer::assigned_articles()` et `Surfer::assigned_section()`.

Un document d'une trentaine de pages a été publiée récemment pour servir de référence sur la gestion des

droits de yacs Martin. De plus, la plupart des requêtes émises pas yacs vers la base de données ont été vérifiées et affinées pour s'assurer de l'implémentation correcte des principes édictés.

3.3. Liens canoniques

L'accès à l'information est flexible, puisque yacs affiche une page dès qu'il en connaît le type et le numéro. Cette flexibilité permet de s'accommoder à la fois des changements de titre, et des imprécisions dans les liens fournis par les internautes.

Pour yacs, tous les liens suivants conduisent à la même page :

- <http://www.yacs.fr/article-6783-en-2010-yacs-relance-la-matrice->.
- <http://www.yacs.fr/article-6783-en-2010>
- <http://www.yacs.fr/article-6783-ancien-titre-eventuel>
- <http://www.yacs.fr/article-6783>
- <http://www.yacs.fr/articles/view.php/6783>
- <http://www.yacs.fr/articles/view.php?id=6783>

Cette façon de procéder convient aux êtres humains, imparfaits et fantasques, que nous sommes. Mais pour les ordinateurs, et notamment pour les moteurs de recherche, cette souplesse induit beaucoup de confusion. En particulier, ils n'aiment pas recevoir le même contenu par des liens différents. Chez Google comme chez d'autres, la duplication des informations peut nuire à la visibilité d'un site.

La version yacs Martin ajoute une nouvelle bonne pratique à la souplesse précédente. Si le lien reçu n'est pas le lien canonique d'une page, yacs redirige le visiteur vers le bon lien. Un saut de puce qui restera indolore pour la plupart des internautes, mais qui fera toute la différence pour les moteurs de recherche.

Pour reprendre l'exemple précédent, le lien canonique est :

<http://www.yacs.fr/article-6783-en-2010-yacs-relance-la-matrice->.

Si l'un des liens suivants est fourni à yacs, l'internaute sera redirigé vers l'adresse canonique :

- <http://www.yacs.fr/article-6783-en-2010>
- <http://www.yacs.fr/article-6783-ancien-titre-eventuel>
- <http://www.yacs.fr/article-6783>
- <http://www.yacs.fr/articles/view.php/6783>
- <http://www.yacs.fr/articles/view.php?id=6783>

La redirection s'effectue par le biais d'un code HTTP 301, pour s'assurer que les moteurs de recherche mémorisent seulement les liens canoniques.

De plus, les liens canoniques sont fournis sous forme de méta-information à destination des moteurs de recherche, à l'intérieur des pages, conformément aux spécifications de Google sur le sujet.

3.4. Permanence des fils de nouvelles

Dans les versions précédentes de yacs les scripts utilisés pour construire les fils de nouvelles comportaient, dans leur nom, la version du protocole mis en oeuvre. Par exemple : `feeds/rss_2.0.php` pour le fil au format RSS.

Avec yacs Martin ces scripts ont été renommés pour des raisons de simplification, et aussi pour faciliter l'évolution vers de nouvelles versions en cas de besoin. L'exemple précédent est donc devenu : `feeds/rss.php` tout simplement. Au passage, le format RSS version 0.92 n'est plus supporté, et le script correspondant a été supprimé.

Le fichier `.htaccess` construit par yacs tient compte de cette modification, et il convient de rafraichir ce fichier après la migration vers yacs Martin pour être sûr de pointer vers les nouveaux scripts.

Par sécurité, le fichier `.htaccess` de yacs Martin redirige les requêtes des anciens scripts vers les nouveaux. Votre serveur continuera de servir les demandes vers `feed/rss_2.0.php` après la migration, si votre serveur autorise la ré-écriture d'URL.

Le tableau ci-dessous rappelle les liens à utiliser pour les fils de nouvelles lorsque la ré-écriture d'URL est disponible, ainsi que les scripts mis en oeuvre.

Format	Adresse ré-écrite	Script
ATOM	<code>feeds/atom</code>	<code>feeds/atom.php</code>
RSS	<code>feeds/rss</code>	<code>feeds/rss.php</code>
RDF	<code>feeds/rdf</code>	<code>feeds/rdf.php</code>
OPML	<code>feeds/opml</code>	<code>feeds/opml.php</code>
Contenu des articles récents, en RSS	<code>feeds/articles</code>	<code>articles/feed.php</code>
Liste des commentaires récents, en RSS	<code>feeds/comments</code>	<code>comments/feed.php</code>
Liste des fichiers récents, en RSS	<code>feeds/files</code>	<code>files/feed.php</code>

3.5. Optimisation du code d'affichage des pages

Le débranchement du script de visualisation ordinaire vers un autre, disponible depuis yacs Macnana, permet toutes sortes d'extensions intéressantes. Yacs l'utilise pour lui-même, avec les options `'view_as_tabs'` et `'view_as_chat'`. Il est bien sûr possible d'ajouter son propre script, écrit sur mesure, (par exemple : `articles/view_as_foo_bar.php`) et de le mettre en oeuvre en indiquant l'option associée (dans l'exemple : `'view_as_foo_bar'`).

Avec yacs Martin, ce mécanisme est optimisé pour éviter les redondances de code dans tous ces scripts de visualisation. Le script `articles/view.php` a été réorganisé pour effectuer plusieurs traitements préliminaires avant de passer la main aux scripts de débranchement. Du coup, ces scripts sont considérablement simplifiés. Par exemple, `articles/view_as_chat.php` est passé de 41.8 koctets en version 9.5 à 21.2 koctets en version 10.2, soit une réduction de 50% !

3.6. Extension de l'interface overlay

L'usage de la fonction membre `get_text()` a été étendue aux champs d'introduction et de description. Ceci permet par exemple de compléter un titre de page en fonction des données contenues dans l'overlay, comme implémenté dans l'overlay `issue` (dans le fichier `overlays/issue.php`). L'appel des fonctions de

L'interface overlay a aussi été systématisé dans les différents script de yacs Martin.

L'overlay d'un article peut déclarer la fonction Javascript `validateOnSubmit()` qui, si elle existe, sera appelée automatiquement par yacs pour valider les données saisies avant qu'elles ne soient transmises au serveur. Pratique pour améliorer l'interactivité des contrôles de saisie, non ? Un exemple concret est visible dans le script `overlays/day.php`, qui refuse l'envoi des données tant qu'une date n'a pas été saisie.

L'interface overlay elle-même a évolué, et offre deux nouvelles possibilités :

- passage d'options à toutes les fonctions de restitution de contenu
- nouvelle fonction membre `get_value()` pour exposer l'état de l'overlay (par exemple, dans un layout)

3.7. Ouverture à des scripts fournis par des tiers.

Les fonctions `get_url()` s'accommodent de variantes extérieures à yacs. Auparavant, ces fonctions renvoyaient systématiquement à l'un des scripts de yacs, et l'intégration de scripts spécifiques était plus compliquée.

3.8. Sauvegarde de paramètres dans les conteneurs

Une nouvelle fonction membre `set_values()` a été ajoutée aux objets de type `Anchor`. Voir par exemple le fichier `articles/article.php` pour l'implémentation de cette fonction pour les articles.

3.9. Héritage des options

Une nouvelle fonction `Articles::has_option()` a été introduite dans `articles/articles.php` pour déterminer si une option est positionnée soit au niveau d'un article, soit de l'un de ses conteneurs. Par exemple :

```
if(Articles::has_option('links_by_title', $anchor, $item))  
    ...
```

3.10. Moins de transactions réseau

AJAX est utilisé à grande échelle dans yacs Martin pour améliorer l'interactivité d'un site, mais il faut bien reconnaître que quelquefois ceci se fait au dépend des performances.

Après une analyse minutieuse du comportement de yacs Macnana, les interrogations effectuées en tâche de fond lors de l'affichage d'une page ont été supprimées. Dans yacs Martin, il n'y a plus de recherche des modifications à intervalles périodiques comme dans la version précédente.

3.11. Moins de requêtes dans la base de données

Le logiciel utilise des « petites » requêtes SQL pour déterminer le nombre de fichiers, de commentaires, ou de liens attachés à une page. Même si chaque requête reste simple, leur multiplication engendre une surcharge de travail pour le serveur.

L'analyse du code a montré qu'une partie des informations collectées dans ces requêtes n'était pas exploitée. Autrement dit, le serveur consacrait une partie de son activité à faire des choses parfaitement

inutiles.

Avec yacs Martin, les fonctions correspondant à ces requêtes ont été simplifiées, ainsi que les requêtes SQL mises en oeuvre. Par exemple, la date de dernière modification n'est plus recherchée systématiquement, comme c'était le cas dans la version précédente. Elle n'est pas proposée non plus dans les nouvelles fonctions de comptage. Voir le chapitre se rapportant aux fonctions remplacées.

3.12. *Extension des scripts inclus*

Les scripts suivants sont maintenant inclus systématiquement dans `shared/global.php` :

- `categories/categories.php`
- `files/files.php`
- `shared/mailer.php`

3.13. *Cache dans des fichiers*

Le cache dans la base de données est toujours disponible, quoique moins utilisé que dans les versions précédentes de yacs à cause du niveau de personnalisation atteint pour la génération des pages dynamiques.

De plus, yacs martin s'est doté d'un cache statique dans des fichiers, placés dans le répertoire `temporary`. Le but spécifique de ce cache est de gérer les informations quasi-stables, en particulier les fils de nouvelles RSS et ATOM, y compris ceux générés par yacs que ceux provenant de l'extérieur.

3.14. *Taille maximum des fichiers*

La taille maximum définie dans `PHP.INI` est à présent évaluée une seule fois dans `shared/global.php`, et placée dans la variable `$context['file_maximum_size']`. Le code redondant qui faisait ces évaluations en a été simplifié d'autant, bien sûr...

3.15. *Support de PHP 5*

La version yacs Martin est développée en environnement PHP 5, et tous les messages d'avertissement spécifiques à cette version du langage ont été pris en compte. Le logiciel fonctionne sans erreur d'incompatibilité sur les systèmes PHP 5 courants.

La fonction `html_entity_decode()` ne fonctionne pas correctement dans tous les environnements PHP. Pour cette raison, elle a été encapsulée dans le script `shared/safe.php` et en appelant `Safe::html_entity_decode()` vous n'aurez plus de problèmes de décodage.

Note : Le positionnement de l'option 'STRICT' dans PHP 5 continue de créer des messages d'avertissement, qui sont liés à de nouveaux mots-clés à ajouter dans les modèles de programmation objet. Ces raffinements seront pris en compte lorsque la communauté aura formellement décidé d'abandonner complètement les environnements d'exécution PHP 4, ce qui devrait être le cas pour la prochaine version majeure.

3.16. *Amélioration des tests système*

A l'usage le script de test du système s'est avéré trop fragile dans certains environnements. A quoi sert un script de test s'il n'est pas capable de s'exécuter jusqu'au bout ? En conséquence, le script a été simplifié pour le rendre plus insensible aux variations de PHP comme d'Apache. Voir le fichier `control/test.php` pour plus d'information.

3.17. Automatisation de la mise à jour

La mise à jour des scripts s'appuie sur un mécanisme de signature des fichiers mis en place presque dès l'origine de yacs. L'algorithme de signature a été considérablement simplifié et optimisé avec yacs Martin, avec une très grande accélération des calculs et, en même temps, une grande économie de mémoire. Voir la fonction `hash()` dans `scripts/scripts.php`. Cette nouvelle efficacité devrait être mise à profit sur les serveurs sous-dimensionnés, pour lesquels la mise à jour automatique se plantait faute de ressources CPU et mémoire suffisantes.

15 scripts à exécution unique ont été préparés pour faciliter la migration des sites existants vers yacs Martin. Les actions effectuées sont assez variées :

- déplacement des images dans le répertoire de référence
- mise à jour des fichiers non-PHP, notamment les feuilles de styles cascadées et les fichiers javascript
- actions sur la base de données pour tenir compte de l'évolution des tables (ajout des propriétaires)

4. Fichiers supprimés

Fichiers supprimés	Commentaires
agents/browsers.php agents/browsers_hook.php	Les calculs statistiques ne sont plus effectués par yacs. L'alternative recommandée est piwik, ou Google Analytics.
articles/fetch_for_palm.php	Qui utilise encore un Palm ? On pourra le remettre si de nombreuses voix le demandent, bien sûr.
articles/layout_articles_as_boxes_and_arrows.php articles/layout_articles_as_ids.php articles/layout_articles_as_rpc.php articles/layout_articles_as_slides.php articles/layout_articles_as_thread.php articles/layout_articles_as_thumbnails.php	Non utilisés, ni proposés, par yacs Martin
feeds/atom_0.3.php	Remplacé par feeds/atom.php
feeds/rss_0.92.php	
feeds/rss_1.0.php	Remplacé par feeds/rdf.php
feeds/rss_2.0.php	Remplacé par feeds/rss.php
images/set_as_bullet.php	
included/browser/freemindbrowser.jar	Seul le composant d'affichage Flash reste supporté
included/php-pdb*	La conversion vers le format Palm n'est plus nécessaire
sections/layout_sections_as_boxesandarrows.php	
sections/slideshow.php	Le composant de présentation en ligne S5 n'est plus disponible
services/slashdot_codec.php	Le format XML pour les nouvelles de Slashdot n'est plus supporté
De nombreux fichiers ont été déplacés ou supprimés dans le répertoire skins dans le cadre de la centralisation des ressources de référence dans skins/_reference	

5. Fonctions supprimées ou changées

5.1. *are_allowed()*, dans *actions/actions.php*

Cette fonction est remplacée par `Actions::allow_creation()`, dans le même fichier. Les paramètres n'ont pas besoin d'être changés.

5.2. *are_allowed()*, dans *articles/articles.php*

Cette fonction est remplacée par `Articles::allow_creation()`, dans le même fichier. Attention, l'ordre des paramètres est inversé.

5.3. *are_allowed()*, dans *categories/categories.php*

Cette fonction est remplacée par `Categories::allow_creation()`, dans le même fichier. Les paramètres n'ont pas besoin d'être changés.

5.4. *are_allowed()*, dans *comments/comments.php*

Cette fonction est remplacée par `Comments::allow_creation()`, dans le même fichier. Attention, le troisième paramètre joue un rôle différent.

5.5. *are_allowed()*, dans *files/files.php*

Cette fonction est remplacée par `Files::allow_creation()`, dans le même fichier. Attention, le troisième paramètre joue un rôle différent.

5.6. *are_allowed()*, dans *images/images.php*

Cette fonction est remplacée par `Images::allow_creation()`, dans le même fichier. Les paramètres n'ont pas besoin d'être changés.

5.7. *are_allowed()*, dans *links/links.php*

Cette fonction est remplacée par `Links::allow_creation()`, dans le même fichier. Attention, le troisième paramètre joue un rôle différent.

5.8. *are_allowed()*, dans *locations/locations.php*

Cette fonction est remplacée par `Locations::allow_creation()`, dans le même fichier. Les paramètres n'ont pas besoin d'être changés.

5.9. *are_allowed()*, dans *sections/sections.php*

Cette fonction est remplacée par `Sections::allow_creation()`, dans le même fichier. Attention, les deux paramètres sont inversés.

5.10. *are_editable()*, dans *comments/comments.php*

Cette fonction est remplacée par `Comments::allow_modification()`, dans le même fichier. Les

paramètres sont identiques aux précédents.

5.11. *build_user_contact()*, dans *shared/surfer.php*

Cette fonction est supprimée.

5.12. *build_user_contact()*, dans *skins/skin_skeleton.php*

Cette fonction est supprimée.

5.13. *check_default_editor()*, dans *shared/surfer.php*

L'appel de la fonction `Surfer::check_default_editor()` a changé. Dans les versions précédentes, on pouvait écrire :

```
$fields = Surfer::check_default_editor(array());
```

Maintenant, il faut écrire à la place :

```
$fields = array();  
Surfer::check_default_editor($fields);
```

5.14. *get_action_label()*, dans *shared/anchors.php*

Avant, il suffisait d'appeler `get_action_label()`. Avec yacs martin, le code de cette fonction a été déplacé dans le fichier `shared/anchors.php`, et il convient d'employer `anchors::get_action_label()` à la place. Les paramètres d'appel restent inchangés.

5.15. *is_editable()*, dans *articles/article.php*

Cette fonction est remplacée par `is_assigned()` dans le même fichier. Attention, cette nouvelle fonction est plus complète que `is_editable()`. Voir aussi `Articles::is_editable()` dans `articles/articles.php`.

5.16. *is_interactive()*, dans *articles/article.php*

Cette fonction est déplacée dans `shared/anchor.php`.

5.17. *list_articles_for_member_by()*, dans *shared/members.php*

Cette fonction est supprimée. Sa fonction est partiellement remplacée par `Articles::list_for_user_by()` dans `articles/articles.php`

5.18. *list_editors_by_name_for_member()*, dans *shared/members.php*

Le nom de cette fonction devient `list_editors_for_member()`.

5.19. *list_sections_for_user()*, dans *shared/members.php*

Cette fonction est remplacée par `Sections::list_by_date_for_user()` dans

sections/sections.php

5.20. *personal_sections()*, dans *shared/surfer.php*

Cette fonction devient `assigned_sections()`, dans le même fichier.

5.21. *stat_for_author()*, dans *articles/articles.php*

Cette fonction est remplacée par `count_for_user()` dans le même fichier. Attention, la structure de l'information fournie est différente.

5.22. *stat_articles_for_anchor()*, dans *shared/members.php*

Cette fonction est remplacée par `count_articles_for_anchor()` dans le même fichier. Attention, la structure de l'information fournie est différente.

5.23. *stat_sections_for_anchor()*, dans *shared/members.php*

Cette fonction est remplacée par `count_sections_for_anchor()` dans le même fichier. Attention, la structure de l'information fournie est différente.

5.24. *stat_users_for_anchor()*, dans *shared/members.php*

Cette fonction est remplacée par `count_users_for_anchor()` dans le même fichier. Attention, la structure de l'information fournie est différente.