

Système de gestion de documents Borges

Autodocumentation

Publié par
Camille Bégnis

Joël Pomerleau
`joel@mandrakesoft.com`

Christian Roy
`croy@mandrakesoft.com`

Fabian Mandelbaum
`fabman@mandrakesoft.com`

Peter Pingus
`pp@mandrakesoft.com`

Jerry Huynh-Tot
`jerry@mandrakesoft.com`

John Rye
`jrt746@clear.net.nz`

Système de gestion de documents Borges: Autodocumentation

Publié 2002-04-19

Copyright © 2002 MandrakeSoft SA

Publié par et Camille Bégnis, Joël Pomerleau, Christian Roy, Fabian Mandelbaum, Peter Pingus, Jerry Huynh-Tot, et John Rye

Table des matières

| | |
|------------------------------------------------------|-----------|
| Préface | i |
| 1. Notice légale | i |
| 2. A propos de la documentation de Borges..... | i |
| 1. Un Concept révolutionnaire | ?? |
| 1.1. Qu'est-ce que Borges ? | 1 |
| 1.1.1. Caractéristiques..... | ?? |
| 1.2. Choisir Borges | 1 |
| 1.2.1. En ai-je besoin ? | ?? |
| 1.2.2. Borges est-il fait pour moi ? | ?? |
| 1.3. Un peu de vocabulaire..... | 3 |
| 2. Guide de démarrage rapide | ?? |
| 2.1. Installation..... | 7 |
| 2.1.1. Où se le procurer ? | ?? |
| 2.1.2. Comment l'installer ? | ?? |
| 2.1.3. Dépendances | 7 |
| 2.2. First Steps..... | ?? |
| 2.3. Begin your own Project..... | ?? |
| 3. Manuel de référence utilisateur | ?? |
| 3.1. Documents writing..... | ?? |
| 3.2. Output Generation | ?? |
| 3.3. Output Style Customizations | ?? |
| 3.4. Revision Management | ?? |
| 4. Features for the Project Manager | ?? |
| 5. Borges and XML Editors (Emacs Rules) | 13 |
| 6. Borges and CVS Integration | ?? |
| 7. Programmer's Reference manual..... | ?? |
| 8. Getting Help | ?? |
| 9. Module échantillon pour tests..... | 21 |
| A. Borges Commands Reminder | 23 |
| B. GFDL..... | ?? |

Préface

1. Notice légale

Copyright © 2002 par MandrakeSoft S.A.

Ce manuel est la propriété intellectuelle de MandrakeSoft. Ce manuel peut être librement copié, distribué et/ou modifié selon les termes de la Licence GNU Free Documentation License, version 1.1 ou ultérieure publiée par la Free Software Foundation ; sans section inaltérable, les textes de couverture, et sans texte de dos de couverture. Une copie complète de la licence se trouve à la section .

Linux est une marque déposée de Linus Torvalds. Toutes les autres marques déposées et copyrights appartiennent à leurs propriétaires respectifs.

2. A propos de la documentation de Borges

Borges est un système de gestion de documents extensible basé sur XML, utilisant la puissance des technologies du logiciel ouvert. Il a été conçu pour faciliter la gestion de langues multiples, la réutilisation du contenu et le travail d'équipe.

Ce manuel est destiné aux utilisateurs de Borges. Vous y apprendrez comment installer Borges, créer un nouveau projet et l'utiliser. Ce manuel inclut également un *Guide de Référence Utilisateur* qui explique plus loin les fonctions centrales telles que les fichiers `conf /`, etc. En dernier, le *Manuel de Référence du Programmeur* couvrira le travail interne de l'application tel que la création de feuilles de style/DTD personnalisées et l'ajout de nouveaux modules.

Chapitre 1. Un Concept révolutionnaire

1.1. Qu'est-ce que Borges ?

Borges est un système de gestion de documents extensible et libre (*open source*) destiné à faciliter des projets de documentation en format XML. Il a été conçu dans un souci d'internationalisation et de réutilisation des contenus, de flexibilité, de travail collaboratif, etc.

Le but de Borges est de fournir un outil convivial :

- pour les débutants : grâce à une interface très simple pour compiler les documents XML et DocBook et les exporter vers divers formats ;
- pour les utilisateurs avancés : en leur offrant la possibilité de personnaliser n'importe quel aspect du système comme les formats d'exportation, la mise en pages et d'autres commandes personnalisées, etc.
- pour les chargés de projets : en leur fournissant un puissant outil de suivi de projets permettant de « jongler » avec les auteurs et les traducteurs, les échéances, etc.

1.1.1. Caractéristiques

Les DTDs prises en charge sont DocBook TDB(Training DocBook), un dérivé de la DTD DocBook utilisé pour rédiger les supports de cours de MandrakeSoft. L'ajout de DTD externes est très facile, même si le système de contrôle des révisions ne fonctionne pas encore avec des DTD autres que DocBook et TDB.

Actuellement, le système permet de :

- compiler des fichiers source en PDF, PS et (X)HTML;
- gérer différentes versions d'un document en définissant des versions dérivées à partir de balises ;
- chaque module est attribué à une équipe de contributeurs : auteurs, traducteurs et relecteurs, chacun étant responsable d'une des étapes de création d'un module. Chaque contributeur peut prendre connaissance de ses tâches sur une page Web et peut recevoir des courriels contenant sa liste des tâches ;
- suivre l'évolution du projet dans sa globalité (à partir de divers manuels) ou à partir d'éléments basiques (paragraphe) et leur traduction ;
- suivre l'évolution d'un module selon 6 étapes prédéfinies (de l'écriture à la relecture finale). Lorsque chaque tâche est accomplie, l'étape correspondante est validée pour que le module passe à la suivante.

Aperçu rapide des fonctionnalités de Borges :

- gestion automatique des images aux formats EPS, PNG, DIA, XFig ;
- gestion automatique globale ou locale (par manuel) des éléments externes ;
- gestion automatique des modules en tant qu'éléments externes.

1.2. Choisir Borges

1.2.1. En ai-je besoin ?

Au lieu d'en présenter les caractéristiques techniques, cette section définit les besoins auxquels peut répondre *Borges*.

Une version constamment revue et disponible en plusieurs langues

Si vous gérez ou éditez des ouvrages qui ont constamment besoin d'être mis à jour dans plusieurs langues, alors *Borges* est fait pour vous. Il vous permettra de contrôler les modifications au niveau d'un paragraphe ou d'un bloc de texte, en optimisant le temps de lecture du traducteur ou du relecteur.

Gestionnaire d'équipe

Si vous dirigez une équipe d'auteurs, même dispersée sur Internet, *Borges* vous simplifiera considérablement la vie, grâce à l'intégration CVS, à la gestion des tâches, des révisions et des langues.

Des contenus réutilisables

Si les contenus que vous éditez sont réutilisables, *Borges* est fait pour vous. Par exemple, vous rédigez un guide de voyage sur les États-Unis et vous désirez éditer des ouvrages sur chacun des états américains sans avoir à manipuler votre document. Vous pouvez également éditer un seul livre parmi un nombre illimité.

Éditer sous de multiples formats

Dans le monde d'Internet aujourd'hui, le format que vous choisissiez pour publier vos travaux est susceptible d'évoluer. Par ailleurs, dans la perspective de la gestion de la relation client (*Customer Relationship Management*), il est de rigueur de mettre à disposition de vos utilisateurs des contenus dans le format qui leur sera le plus approprié. Cela peut être un livre, un site Internet, un fichier PDF téléchargeable... *Borges*, s'appuyant sur les formats XML et DocBook, a été spécifiquement conçu pour répondre à ce type de besoin... Vous pouvez définir des mises en pages pour tous ces formats et adopter une approche d'éditeur de contenus.

1.2.2. Borges est-il fait pour moi ?

Borges ne vous sera d'aucune utilité si vous :

- rédigez rarement des documents de plus de 2 pages ;
- avez rarement à traduire vos documents ;
- ne voulez pas travailler pas sur un système d'exploitation autre que Windows ;
- êtes effrayé à la vue d'un texte en mode console.

Utilisez *Borges* si vous :

- êtes amené à gérer des documents volumineux ;
- devez traduire ces documents dans plusieurs langues ;
- dirigez une équipe de personnes impliquées dans la rédaction de ces documents ;
- vous arrachez les cheveux car des éléments de vos documents changent tous les jours ;
- pouvez compter sur une personne à l'aise sous *GNU/Linux*;
- désirez migrer vos documents et votre équipe vers une nouvelle génération de documentation technique en XML et DocBook.

En clair, *Borges* vous apportera une solution pour gérer de façon efficace de gros projets de documentation tout en y apportant une plus grande qualité et en réduisant les temps de travail. La seule contrepartie sera de consacrer du temps pour lire la documentation et pour s'habituer au système. L'installation du système sous *GNU/Linux* est indispensable. Si vous ne connaissiez pas le format DocBook, il vous faudra aussi l'assimiler.

Toujours intéressé par la bête ? Félicitations ! Continuez votre lecture et bonne chance pour la suite, vous ne le regretterez pas !

1.3. Un peu de vocabulaire

Nous allons expliquer tous les termes utilisés dans la documentation de *Borges*: projet, auteur, initiales d'auteur, document, sous-document, module, état d'un module, atome, révision d'un atome, etc.

Note : Ces termes ne sont pas présentés dans un ordre particulier.

auteur

Un auteur peut être le rédacteur, le traducteur ou le relecteur d'un module. Généralement, le concept d'« auteur » est lié au créateur (dans ce cas précis, le rédacteur) de quelque chose, mais *Borges* considère les traducteurs et les relecteurs comme des auteurs.

Voir aussi : initiales d'auteur, module.

initiales d'auteur

Borges identifie les différents auteurs qui participent à un projet par leurs initiales. Cela restreint les auteurs à l'usage d'initiales *uniques*.

Si votre projet comprend un petit groupe d'auteurs, des initiales de deux lettres seront suffisantes. Cependant, plus de lettres peuvent être utilisées si les initiales sont uniques.

Voir aussi : auteur, projet.

projet

Un projet est un document ou un ensemble de documents que vous gérez avec *Borges*. En général, un projet comporte une série de documents.

Voir aussi : document.

super-document

Ce terme désigne un ensemble de modules, structuré de façon à former un livre, un article, un guide d'utilisation, soit tout bloc d'information exhaustive concernant un sujet particulier.

Le super-document est le document « maître » à partir duquel différents documents peuvent être générés. La structure du super-document est définie dans le fichier `master.top.xml`.

Un super-document peut contenir mutuellement des informations exclusives qui seront triées en divers documents spécialisés.

Voir aussi : document.

document

Un document est une compilation d'un super-document avec pour résultat un fichier PDF ou (X)HTML. Vous pouvez choisir de compiler la totalité du super-document ou

seulement quelques éléments de ce dernier. Les documents peuvent être des livres entiers, des articles, des pages de référence, des lettres, des manuels, etc.

Voir aussi : compilation, super-document.

compilation

La compilation est le procédé par lequel une série de fichiers source écrits en XML sont « transformés » en document PDF ou (X)HTML.

module

Les modules sont les parties qui composent un document. Habituellement, un super-document est divisé en petits éléments appelés modules afin de simplifier la rédaction, la traduction, la gestion et la réutilisation du contenu. Les chapitres, sections, annexes et glossaires sont de bons candidats pour devenir des modules.

En fait, *Borges* requiert que tout élément structuré soit placé dans un module pour pouvoir être traduit, et afin de tirer partie des caractéristiques du système de gestion des révisions.

Les modules peuvent être marqués, à l'aide de l'attribut `condition=`, afin d'être exclus lors de certaines compilations. Cela vous permettra de créer plusieurs types de document à partir d'un ensemble unique de modules, tout en les améliorant grâce aux fonctions de réutilisation des contenus de *Borges*.

Voir aussi : document, super-document, projet.

module d'origine

Ceci est utilisé pour préciser quel module a été écrit par un rédacteur. Les traducteurs utiliseront le module d'origine comme référence pour toutes les traductions.

Voir aussi : module, module traduit.

module traduit

Cette expression désigne un module qui n'est pas celui d'origine, mais une traduction du module d'origine.

Voir aussi : module, module d'origine.

état d'avancement d'un module

Les modules passent par différentes étapes durant leur cycle de vie. Chaque « étape » détermine l'état d'avancement du module.

Afin de passer d'une étape à une autre, quelques opérations doivent être exécutées sur le module, par exemple : la rédaction, la traduction, la correction orthographique, la relecture, etc.

Voir aussi : cycle de vie.

atome

Les atomes sont les éléments XML servant aux modifications de contrôle au sein d'un module. Ce sont les plus petits éléments contenant du texte. Les atomes typiques en *DocBook* sont `<title>` et `<para>`.

Voir aussi : révision d'atome.

révision d'atome

Les atomes possèdent un numéro de révision lequel est utilisé par le système de gestion de révision de *Borges* dans le but de retracer les changements faits dans les modules sur une « échelle atomique ».

Voir aussi : atome.

cycle de vie

Le cycle de vie d'un module est fait de diverses étapes (ou états) au travers desquelles le module doit passer pour être considéré comme prêt à être diffusé. Actuellement, Borges fonctionne selon un seul cycle de vie fixe, lequel est détaillé dans .

Voir aussi : module, état d'avancement d'un module.

Chapitre 2. Guide de démarrage rapide

2.1. Installation

Jusqu'à maintenant, *Borges* a seulement été testé sur Mandrake Linux 8.2. Il devrait fonctionner sur n'importe quel système Linux pourvu que les dépendances nécessaires soient installées. Merci de nous informer des succès ou échecs que vous aurez rencontré sur n'importe quel autre système.

2.1.1. Où se le procurer ?

Les versions courantes sont disponibles dans le répertoire *Borges* FTP¹. Vous y trouverez différents paquetages :

- si vous utilisez un système basé sur des paquetages en mode RPM, installez les paquetages *Borges* et *Borges-DocBook* noarch ;
- Vous pouvez également choisir de vous procurer le *tarball* (*Borges-*.tar.bz2*) ;

Finalement, si vous aimez vivre dangereusement, vous pouvez vous procurer la CVS version actuelle avec les paramètres suivants : `CVS_RSH=ssh` et `CVSROOT=:ext:anoncvs@cvs.mandrakesoft.com:/cooker`. Ensuite, vous pouvez obtenir le module *Borges* avec le mot de passe `cvs`.

2.1.2. Comment l'installer ?

Installez simplement les paquetages RPM, ou lisez les instructions dans le *tarball*.

Note : *Borges* s'installe par défaut dans `/usr/share/Borges/`.

2.1.3. Dépendances

Si vous n'installez pas *Borges* depuis des paquetages RPM, vous aurez à vérifier que les programmes ou les bibliothèques suivantes sont disponibles sur votre système :

- `make`
- `libxslt-proc` à partir de `libxslt` projetGnome ;
- `perl` ;
- les bibliothèques `perl-XML-Twig`, `perl-DateManip` et `perl-XML-LibXML` ;
- l'éditeur de diagrammes `dia` si vous souhaitez travailler avec des diagrammes `dia` ;
- l'éditeur de diagrammes `xfig`, si vous souhaitez travailler avec des diagrammes `xfig` ;
- `XFree86-Xvfb` est nécessaire pour la transformation de diagrammes ;
- DocBook DTD XML version 4.1.2 dans le répertoire `/usr/share/sgml/docbook/xml-dtd-4.1.2/` ;
- la feuille de style DocBook DSSSL dans le répertoire `/usr/share/sgml/docbook/dsssl-stylesheets/` ;
- feuilles de style DocBook XSL dans le répertoire `/usr/share/sgml/docbook/xsl-stylesheets/` ;
- `openjade` ;
- `tetex-latex` ;
- `jadetex`.

1. <ftp://ftp.mandrakesoft.com/camille/pub/Borges/>

Astuce : Si quelque chose se passe mal pendant que vous essayez d'installer `Borges`, assurez-vous que ces applications soient installées correctement.

2.2. First Steps

first-steps-borges-sect1

Give basic commands to configure, compile PDF and html outputs, generate the report; for sample document.

2.3. Begin your own Project

first-project-borges-sect1

Tell how to configure the system for the new manual/languages. Then begin a step-by-step example:

1. Define your manual by writing the `master.top.xml` file;
2. Define local entities for titles notably;
3. Test `master.top.html` compilation;
4. Build module templates;
5. Populate templates;
6. Test doc compilation.

Chapitre 3. Manuel de référence utilisateur

3.1. Documents writing

borges-writing-features-sect1

At first, review the format of configuration files: in conf/ but also conf.xml. For master.top.xml detail all the required elements for the revision system to work.

Then Explain in details all functionalities, those already seen above, plus global entities, images, stylesheet driver files, sub_manuals, validation, ...

3.2. Output Generation

borges-compile-features-sect1

Detail:

1. Available output formats;
2. How to generate a single manual;
3. Where to customize them;
4. How to associate a special stylesheet to a special document;
5. How to generate a set of manuals/languages documents.

3.3. Output Style Customizations

borges-style-custom-sect1

Tell how to customize stylesheets, rcreate new layers and use them..

3.4. Revision Management

borges-revision-features-sect1

First introduce the revision system in details with its various levels.

Enter in the details from top to bottom: whole project report, manual, module, element. For each, tell the way to make it work in the sources, the make target associated, and where to check the result.

Chapitre 4. Features for the Project Manager

borges-manager-features-sect1

Move here the mail section from previous chapter.

Chapitre 5. Borges and XML Editors (Emacs Rules)

borges-editors-chapter

In introduction tell that any editor should be OK provided it does not interfere with Borges requirements.

Then add a section for Emacs+PSGML which Borges fully supports..

Chapitre 6. Borges and CVS Integration

borges-cvs-chapter

Tell that Borges is designed to work fine on a CVS. Tell what to put on CVS and what not. Warn about the problems caused by module templates automatically generated that must be added to the CVS at the same time the modified master.top.xml is updated. Same when adding new languages/manuals.

Chapitre 7. Programer's Reference manual

programmer-borges-chapter

Enter in the details of implementations, how to add a new module for other DTDs/stylesheets...

Chapitre 8. Getting Help

borges-help-chapter

List all associated resources: website, ML, SF, IRC, what they are for (not only help, but bugreports, RFEs, etc.).

Chapitre 9. Module échantillon pour tests

passwords

```
root# head -c 6 /dev/urandom | mimencode
```

Cela imprimera cinq caractères aléatoires sur la console, et est utilisables pour la génération automatique de mots de passe. Vous trouverez **mimencode** dans le paquetage `metamailer`.

Annexe A. Borges Commands Reminder

command-reminder-appendix

This appendix will list all available make commands under Borges, sorted by topic: compilation, revision management, reports generation, etc.

Annexe B. GFDL

gfdl-appendix

