

# MISE EN PLACE D'UN SITE INTRANET

## AVEC ZOPE ET PLONE

AUTEUR : AURÉLIE CAMBOURIAN  
aurelie.cambourian@free.fr

PREMIÈRE ÉDITION LE : 02 MARS 2003  
ÉDITION COURANTE : 15 SEPTEMBRE 2003

Version 0.99

# LE SOMMAIRE

<b>I</b>	<b>PAR OÙ COMMENCER</b>	<b>9</b>
I.1	Ajout d'un Plone Site . . . . .	9
I.1.1	Explications . . . . .	9
I.1.2	Rubriques du document utilisées . . . . .	9
I.2	Créer les rubriques . . . . .	9
I.2.1	Explications . . . . .	9
I.2.2	Rubriques du document utilisées . . . . .	10
I.3	Gérer les utilisateurs . . . . .	10
I.3.1	Explications . . . . .	10
I.3.2	Rubriques du document utilisées . . . . .	10
I.4	Implémenter les types supplémentaires . . . . .	10
I.4.1	Explications . . . . .	10
I.4.2	Rubriques du document utilisées . . . . .	10
I.5	Organiser la navigation . . . . .	10
I.5.1	Explications . . . . .	10
I.5.2	Rubriques du document utilisées . . . . .	11
I.6	Le design du site : dernière étape . . . . .	11
I.6.1	Explications . . . . .	11
I.6.2	Rubriques du document utilisées . . . . .	11
I.7	Approfondissement . . . . .	11
I.7.1	Explications . . . . .	11
I.7.2	Rubriques du document utilisées . . . . .	11
<b>II</b>	<b>INSTALLATIONS</b>	<b>12</b>
II.1	Zope, CMF et Plone . . . . .	12
II.1.1	Zope . . . . .	12
II.1.2	CMF . . . . .	13
II.1.3	Plone version 1.0.X ou précédente . . . . .	13
II.1.4	Plone version 1.1.X ou 2.X . . . . .	14
II.2	Les autres produits Zope utilisés . . . . .	14
II.2.1	Localizer et TranslationService . . . . .	14
II.2.2	CMFForum . . . . .	14
II.2.3	ZPhotoSlides et LocalFS . . . . .	14
II.2.4	CMFTypes . . . . .	15

<b>III L'OBJET PLONE SITE</b>	<b>16</b>
III.1 Ajout de l'objet de type Plone Site	16
III.2 Fonctionnement de Plone - Contenu de l'objet Plone Site	16
III.2.1 MailHost	16
III.2.2 Members (Members)	16
III.2.3 acl_users (User Folder)	16
III.2.4 content_type_registry	17
III.2.5 cookie_authentication	17
III.2.6 index_html(Welcome to Plone)	17
III.2.7 plone_utils (Various Plone Utility methods)	17
III.2.8 portal_actions (Contains custom tabs and buttons)	17
III.2.9 portal_calendar (Controls how Events are shown)	17
III.2.10 portal_catalog (Indexes all content in the site)	17
III.2.11 portal_discussion(Control how discussions are stored by default on content)	18
III.2.12 portal_factory (Responsible for the creation of content objects)	18
III.2.13 portal_form (Used together with templates to do validation and navigation)	19
III.2.14 portal_form_validation (Deprecated, not in use)	19
III.2.15 portal_memberdata (Handles the available properties on Members)	19
III.2.16 portal_membership	19
III.2.17 portal_metadata	19
III.2.18 portal_migration	20
III.2.19 portal_navigation	20
III.2.20 portal_properties	20
III.2.21 portal_registration (Handles registration of new users)	21
III.2.22 portal_skins (Control skin behaviour (search order etc))	21
III.2.23 portal_syndication (Generate RSS folders)	24
III.2.24 portal_types (Controls the available Content Types in your portal)	25
III.2.25 portal_undo (Defines actions and functionality related to undo)	25
III.2.26 portal_url (Methods to anchor you to the root of your Plone site)	25
III.2.27 portal_workflow (Contains workflow definitions for your portal)	25
<b>IV INSTALLATION ET INTÉGRATION DES PRODUITS ZOPE DANS PLONE</b>	<b>26</b>
IV.1 CMFForum	26
IV.2 CMTypes	26
IV.3 ZPhotoSlides	26
IV.3.1 La css de présentation des albums	27
IV.4 CMFCollector	29
IV.4.1 Installation du Produit :	29
IV.4.2 Installation à la racine du Site Plone considéré :	29
IV.4.3 Installation du Workflow pour CMFCollector	29
IV.5 External Editor	30
IV.5.1 Installer External Editor	30
IV.5.2 Configurer External Editor	30
IV.6 Localizer et TranslationService	31
IV.6.1 prérequis	31
IV.6.2 Via la ZMI, à la racine de notre site Plone	31
IV.6.3 Pour que le langage utilisé soit celui du fichier que l'on a téléchargé	31

IV.6.4	Ajouter le support des langues dans le MessageCatalog . . . . .	31
IV.6.5	Activer l'utilisation du Catalogue . . . . .	31
IV.6.6	Supporter plusieurs langues . . . . .	31
IV.6.7	Caractères accentués . . . . .	32
<b>V</b>	<b>EXPORTER PUIS IMPORTER SON SITE</b>	<b>33</b>
V.1	Le bouton import/export . . . . .	33
V.1.1	Exporter son site . . . . .	33
V.1.2	Importer son site . . . . .	33
V.2	Après l'import, on va avoir l'erreur de type <code>__getitem__</code> . . . . .	34
V.2.1	Pourquoi ? . . . . .	34
V.2.2	Corriger . . . . .	34
<b>VI</b>	<b>L'ORGANISATION DE LA NAVIGATION</b>	<b>35</b>
VI.1	Les dossiers . . . . .	35
VI.1.1	Créer une rubrique (onglet par exemple) . . . . .	35
VI.1.2	Droits sur les dossiers : . . . . .	35
VI.2	Les slots . . . . .	36
VI.2.1	Créer un slot . . . . .	36
VI.2.2	Organiser les slots . . . . .	37
VI.3	Header et Footer . . . . .	38
VI.3.1	Le header . . . . .	38
VI.3.2	Le footer . . . . .	43
VI.4	Avoir un slot de navigation (à la racine) en images . . . . .	44
VI.4.1	le code . . . . .	45
VI.4.2	les remarques . . . . .	45
<b>VII</b>	<b>LES MÉTADONNÉES</b>	<b>46</b>
VII.1	Utilité et Présentation . . . . .	46
VII.2	Les métadonnées dans Plone . . . . .	46
VII.3	Validator ou comment obliger l'entrée des métadonnées . . . . .	47
VII.3.1	Script validant la création d'un dossier . . . . .	47
VII.3.2	L'aspect du formulaire d'édition doit montrer que le champ est obligatoire . . . . .	48
<b>VIII</b>	<b>RÉFLEXION SUR LES MÉTADONNÉES</b>	<b>49</b>
VIII.1	Le concept . . . . .	49
VIII.1.1	Le contexte . . . . .	49
VIII.1.2	Design de ce contexte . . . . .	49
VIII.2	Les détails de la mise en oeuvre . . . . .	50
VIII.2.1	Installations . . . . .	50
VIII.2.2	Adaptation de l'architecture du site Plone . . . . .	50
VIII.2.3	Paramétrage des métadonnées . . . . .	50
VIII.2.4	Ecrire les pages templates pour le classement . . . . .	50
<b>IX</b>	<b>LE WORKFLOW DANS PLONE</b>	<b>54</b>
IX.1	Ce qu'est un workflow . . . . .	54
IX.2	DCWorkflow, le workflow dans Plone . . . . .	54
IX.2.1	DCWorkflow, description . . . . .	55

IX.2.2	D'autres Workflow pour Plone . . . . .	55
IX.3	Un exemple de personnalisation du Workflow dans Plone . . . . .	55
IX.3.1	Objectif visé . . . . .	55
IX.3.2	Schéma du nouveau workflow . . . . .	56
IX.3.3	Un nouveau rôle dans le site Plone . . . . .	56
IX.3.4	Créer et configurer ce nouveau workflow . . . . .	56
IX.3.5	Configurer ce nouveau workflow . . . . .	57
IX.3.6	Prévenir les gens ayant le rôle de Reviewerun de contenus en attente de validation . . . . .	58
IX.3.7	Affecter ce nouveau workflow à des types de contenus . . . . .	58
IX.3.8	Vérification . . . . .	58
<b>X</b>	<b>GRAPHISME DU SITE</b>	<b>60</b>
X.1	Les images . . . . .	60
X.1.1	Modifier une image . . . . .	60
X.1.2	Insérer une nouvelle image . . . . .	60
X.1.3	Une organisation stratégique des images . . . . .	61
X.2	Les css . . . . .	61
X.2.1	Présentation des feuilles de styles . . . . .	61
X.2.2	Modifier les css de Plone . . . . .	63
<b>A</b>	<b>GNU Free Documentation License</b>	<b>65</b>

L'écriture de ce texte et les schémas ont été entièrement réalisés avec des logiciels libres à savoir,  $\LaTeX$  et xfig sur un ordinateur possédant un système d'exploitation complètement libre Debian GNU/Linux.



Les erreurs de ce manuel sont les miennes et en aucun cas ne peuvent être attribuées à une autre personne physique ou morale.

Copyright (c) Aurélie Cambourian.

Permission is granted to copy, distribute this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation ;

with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts.

A copy of the license is included in the section entitled "GNU Free Documentation License".

**Au 28 novembre 2003 :** Mise un peu à jour de la documentation en introduisant la version de Plone 2.0. (on retrouve ces modifications et ajouts à divers endroits du site, dans les descriptions du contenu de l'objet PloneSite en particulier). Ajout de section produits : CMFCollector [IV.4](#) et External Editor [IV.5](#). Ajout d'un slot de navigation en image [VI.4](#).

**Au 10 juillet 2003 :** Merci à Isa, j'avais fait une faute dans le workflow. L'état prepending doit avoir comme transitions prepublish, retract, reject et hide...

**Au 06 juillet 2003 :**

Dans cette version : rajout d'un chapitre sur le workflow [IX](#). Pour comprendre ces flux de travail, je donne un exemple pour ajouter une pré-relecture des contenus, afin d'épurer le travail des relecteurs finaux (cela s'applique bien à certaines hiérarchies). Comme toujours n'hésitez pas à m'écrire en cas de problème, j'essaierai du mieux que je peux de vous aider.

**Au 19 juin 2003 :**

Après plusieurs présentations de mon travail, rajout de la réflexion sur les métadonnées. Cette réflexion a permis de déterminer toute une philosophie d'utilisation de la lecture et l'affichage d'informations. Ajout donc du concept, et des scripts et templates permettant de classer les objets de son site, dans un nouveau chapitre [VIII](#) sur la réflexion sur les métadonnées.

**Au 29 avril 2003 :**

En fonction des questions posées sur le site [www.zopera.org](http://www.zopera.org) et reçues par e-mail, rajout d'une explication sur les onglets (category=portal\_tabs) dans le chapitre sur la navigation.

**Au 22 mars 2003 :**

En fonction des questions posées sur le site [www.zopera.org](http://www.zopera.org) et reçues par e-mail ajout de compléments d'informations sur les rubriques dans Plone, le graphisme, les cookies...

**Au 11 mars 2003 :**

Ajout des commentaires de Yves aux endroits correspondants de la documentation (installation avec un environnement Windows®, calendrier...)

**Au 02 mars 2003 :**

Ce document n'est pas entièrement rempli mais est dans une version utilisable pour aider à la compréhension de Plone.

Sébastien Chaumat a relu et proposé les corrections de ce document, ainsi que des idées telles que la section "par où commencer" avec Plone.

Pour SEb qui avait bien raison de me parler de Zope avec enthousiasme,

# Un site intranet

## Ce qu'est un site intranet, ses objectifs, son rôle, ses services...

Un site intranet est un ensemble de pages de même type que celles que l'on trouve sur Internet (liées entre elles par des liens hypertextes), publiées dans un navigateur Internet (comme Internet Explorer, Netscape, Mozilla). Un site intranet est un site qui correspond et appartient à l'entreprise qui le met en place.

On retrouvera ainsi dans l'intranet informatif les aspects les plus révolutionnaires et sympathiques de l'internet (tout au moins à ses débuts) :

- \* Accéder facilement à des informations toujours plus fiables et réactives (accélération de la mise à jour des documents pour tous)
- \* Réduire les coûts (déplacements, stockage, impression et diffusion d'information, etc...).
- \* Disposer de panneaux d'affichage collectif consultables sans se déplacer.
- \* Elargir le champ d'information à tous les utilisateurs du réseau
- \* Travailler avec des outils plus ergonomiques et plus conviviaux
- \* Disposer d'aides en ligne.
- \* Mise à disposition de documents techniques
- \* Moteur de recherche de documentations
- \* Un échange de données entre collaborateurs
- \* Messagerie électronique
- \* Forums de discussion, listes de diffusions
- \* Portail vers internet
- \* Avoir des "points dialogues" avec les acteurs de l'entreprise
- \* Disposer de notifications d'alerte sur les informations nouvelles
- \* Utiliser d'une manière active, contribuer, animer et partager
- \* Repérer facilement "qui fait quoi" dans les services
- \* Capitaliser les expériences
- \* Mettre en place une mémoire collective.

Un site intranet dans une entreprise permet de mettre facilement à la disposition des employés des documents divers et variés ( des textes, des images, des vidéos, des sons, des liens internet, des adresses de messagerie et même des applications...).

Dans un groupe scolaire comme dans une entreprise, on peut étendre les informations à tous les employés ainsi qu'à tous les élèves et étudiants, en prenant en compte l'intérêt et la confidentialité des documents publiés. Pour cela l'administrateur du site autorise ou non l'accès à certaines pages, comme lorsque sur le réseau il permet l'accès à telle ou telle ressource.

# Chapitre I

## PAR OÙ COMMENCER

L'organisation de ce document est très descriptive dans le sens où j'y décris les différents objets d'un site Plone ainsi que quelques produits utilisés pour la création d'un site Intranet (réalisé pour un groupe scolaire).

Ainsi sur des conseils avisés, j'ai mis en place cette page qui devrait vous aider à utiliser ce document pour créer un site Intranet en utilisant Zope et Plone.

S'il fallait recommencer et créer un nouveau site je pense que je suivrais les étapes suivantes :

### I.1 Ajout d'un Plone Site

#### I.1.1 Explications

Dans la ZMI (Zope Management Interface), ajouter un Plone Site, entrer un court identifiant, choisir un portail avec un dossier d'utilisateurs et un site accessible aux anonymes.

Tout de suite après, franciser le site Plone.

#### I.1.2 Rubriques du document utilisées

Le chapitre II sur les installations de Zope, CMF, Plone.

La sous-section III.1 sur l'ajout d'un site Plone.

La section II.2 sur l'installation des autres produits Zope utilisés.

La section IV.6 sur la localisation (francisation) du site Plone.

### I.2 Créer les rubriques

#### I.2.1 Explications

Puis créer des rubriques (dossiers) et remplir la page index\_html.

Pourquoi ? car Plone est déjà fonctionnel et qu'en fait ce qui compte en premier sont les services offerts et non pas l'optimisation de l'accès aux ressources, déjà bien avancé dans un site Plone.

## I.2.2 Rubriques du document utilisées

La section VI.1 sur la création de rubriques.

La sous-section III.2.6 sur index\_html.

## I.3 Gérer les utilisateurs

### I.3.1 Explications

Qui va se connecter à ce site ?

Remplir le dossier des membres et managers du portail. Ajouter ces utilisateurs aux rubriques déjà créées sachant qui est le manager de quelle rubrique (dossier dans mon cas) le cas échéant.

### I.3.2 Rubriques du document utilisées

La sous-section III.2.3 sur acl\_users.

## I.4 Implémenter les types supplémentaires

### I.4.1 Explications

Les objets créables par les utilisateurs sont par défaut dans Plone : Image, News, document (texte), fichier downloadable, événement, favori, dossier, lien, commentaire (si autorisé).

On peut ajouter un métatype pouvant inclure un objet dans le site correspondant un document ayant une extension en .doc contenant des tableaux, des titres, des paragraphes (provenant de AbiWord, StarOffice, MS-Office, OpenOffice...). Des types de contenus auxquels on peut attacher des autres contenus... Des systèmes de sondages, de synchronisation...

Et aussi ajouter un métatype "album de photographies" ainsi que "forum".

### I.4.2 Rubriques du document utilisées

La section II.2 sur l'installation des autres produits Zope utilisés.

Le chapitre IV sur l'intégration des produits CMF à Plone.

La sous-section III.2.24 sur portal\_types.

## I.5 Organiser la navigation

### I.5.1 Explications

Ensuite il faudrait organiser la navigation en choisissant quels slots à utiliser, au besoin en créer (pas vraiment nécessaire), puis rajouter dans le header une barre de navigation imagée, et enfin, configurer les onglets pour qu'ils présentent les endroits clés du site.

Peut-être que ce serait alors le moment de modifier le header en fonction des besoins (la modification de la personnal-bar par exemple, présentant les actions réalisables par l'utilisateur connecté).

## **I.5.2 Rubriques du document utilisées**

Le chapitre VI sur l'organisation de la navigation.

La sous-section III.2.16 sur "portal\_membership" sur les actions réalisables par les utilisateurs (de la personal bar et des onglets).

La sous-section III.2.8 sur portal\_actions (onglets).

## **I.6 Le design du site : dernière étape**

### **I.6.1 Explications**

A modifier en premier le logo et les images de navigation d'abord, puis celles d'agréments et enfin les css.

### **I.6.2 Rubriques du document utilisées**

La sous-section III.2.22 portal\_skins.

Le chapitre X sur le design (images et css).

## **I.7 Approfondissement**

### **I.7.1 Explications**

Utiliser les métadonnées pour améliorer la classification des documents (important) et modéliser un workflow personnalisé. On peut aussi se pencher sur les systèmes de caches.

### **I.7.2 Rubriques du document utilisées**

Le chapitre VII sur les métadonnées et le chapitre VIII sur la réflexion sur les métadonnées . Le chapitre IX sur le workflow dans Plone. Le chapitre ?? sur les systèmes de caches.

# Chapitre II

## INSTALLATIONS

### II.1 Zope, CMF et Plone

#### II.1.1 Zope

Pour plus de renseignements sur Zope, voir le site de Zope : [zope.org](http://zope.org).  
Ces indications d'installations ne remplacent pas la lecture de Zope/docs/install.txt

#### Indications d'installation :

Extraire Zope de son archive dans le dossier où on souhaite l'installer avec par exemple, la commande :  
tar -xovzf Zope-2.6.x-linux2-x86.tar.gz

Lancer la commande ./install située dans le dossier Zope-2.6.x-linux2-x86, à la fin du script d'installation apparait le nom 'admin' suivi d'un mot de passe qui sera celui initial du serveur Zope.

Aller dans le répertoire du serveur Zope, puis dans bin/ , là, renommer le dossier "python" en "python.old" par exemple (mv python python.old). Puis, faire un lien symbolique depuis le python 2.1 installé via apt-get (si debian) (ln -s /usr/bin/python2.1 python).

Pour lancer Zope on lance simplement le script 'start' situé dans le dossier de Zope. (via la commande './start' si on se situe dans le répertoire de Zope.)

Si on a perdu le mot de passe donné à la fin du script d'installation, on peut lancer le script contenu dans Zope à savoir zpasswd.py (lancé par la commande bin/python zpasswd.py) on pourra alors introduire un nouveau manager au serveur Zope.

Toutefois je conseille d'installer après l'installation de Plone, la version de Python(2.1.3) identique à celle de Plone ainsi que Pythonwin32 ce qui permet d'avoir un ensemble homogène permettant de pouvoir tester les scripts Python facilement.

Yves. le 10/03/2003

pour le pythonwin32 il faut le fichier win32all-151.exe qui correspond à la version de Plone

sinon le driver ODBC qui fonctionne avec cette version de Plone est ZODBCDA-3.1.0b2-win32-x86.tar

Yves. le 11/03/2003

NDA : sous Windows®

### Premier lancement de Zope :

Lancer le script de démarrage 'start' comme vu ci-dessus.

Se rendre dans la barre d'adresse d'un navigateur web et entrer :

'http://localhost:8080/manage'

Une fenêtre de login apparait, il suffit d'entrer les identifiants du manager de Zope pour voir la ZMI (Zope Management Interface) apparaître.

### La ZMI :

La ZMI se compose de plusieurs parties, au premier lancement de Zope il faut se rendre dans le dossier `acl_user` et ajouter un utilisateur qui aura le rôle de manager du serveur Zope, puis supprimer l'utilisateur "admin" initial.

### II.1.2 CMF

L'archive de CMF contient 4 dossiers :

- CMFCalendar
- CMFCore (dedans il y a le `skinpath` utile pour l'export import)
- CMFDefault
- CMFTopic (pour gérer les commentaires)

### II.1.3 Plone version 1.0.X ou précédente

Consulter le site [plone.org](http://plone.org) pour télécharger l'archive.

L'archive de CMFPlone contient 5 dossiers (au moi de juillet) :

- ActivePack (à n'utiliser que si on utilise une version de Zope inférieure à 2.6.0)
- CMFPlone (Plone et ses sous-parties)
- DCWorkflow
- i18n (à savoir "internationalisation")
- Formulator

Pour l'installation, c'est très simple :

1. Arrêter le serveur Zope
2. Copier les répertoires "*CMFPlone*", "*DCWorkflow*", "*i18n*", "*Formulator*" si vous avez Zope 2.6.x, ajoutez le répertoire "*ActivePack*" si vous avez un Zope 2.5.X, le tout est à copier dans le répertoire `/cheminDeZope/lib/python/Produ`.
3. Redémarrer le serveur Zope.

Pour l'éditeur externe il manque dans la distribution Plone le fichier `ZopeEdit.ini`.

Yves. le 10/03/2003

NDA : pour Windows®

### II.1.4 Plone version 1.1.X ou 2.X

Consulter le site [plone.org](http://plone.org) pour télécharger l'archive.

L'archive de CMFPlone contient 8 dossiers (pour la version 2.0-beta2) :

- BTreeFolder
- CMFPlone
- CMFActionIcons
- CMFFormController
- i18n (à savoir "internationalisation")
- CMFQuickInstallerTool
- GroupUserFolder (GRUF ;) pour gérer les groupes d'utilisateurs.)
- Formulator

Pour l'installation, c'est très simple :

1. Arrêter le serveur Zope
2. Copier les 8 répertoires dans le répertoire /chemindeZope/lib/python/Products/ .
3. Redémarrer le serveur Zope.

## II.2 Les autres produits Zope utilisés

### II.2.1 Localizer et TranslationService

Pour une utilisation du support des langues.

Localizer :

<http://sourceforge.net/projects/leu>

TranslationService :

<http://www.zope.org/Members/efge/TranslationService>

Pour TranslationService un problème se posait dans la version 0.2 (avec utf-8 et iso-8895-1), corrigé dans la version 0.3 par Florent Guillaume. En effet dans le changelog on note que l'erreur (innommable) "UnicodeError : ASCII decoding error : ordinal not in range(128)" est réglée. .

Installation dans Zope :

1. On arrête Zope
2. tar -xvzf des 2 archives téléchargées
3. On copie les répertoires obtenus "Localizer" et "TranslationService" dans /chemindeZope/lib/python/Products/
4. On redémarre Zope

### II.2.2 CMFForum

[http://www.zope.org/Members/dreamcatcher/cmfforum\\_first\\_release](http://www.zope.org/Members/dreamcatcher/cmfforum_first_release)

Pour avoir des Forums. (CMFForum est une application de forum, il en existe d'autres)

### II.2.3 ZPhotoSlides et LocalFS

<http://www.zope.org/Members/p3b/ZPhotoSlides>

<http://sourceforge.net/projects/localfs>

Pour les albums Photos.

## II.2.4 CMFTypes

Ce projet a été renommé en Archetypes.

<http://sourceforge.net/projects/collective>

Pour uploader un document MSWord ® qui va être transformé à la volée par vware en html (tableaux...)

Pour créer des nouveaux types de contenu dans Plone.

## Chapitre III

# L'OBJET PLONE SITE

### III.1 Ajout de l'objet de type Plone Site

A la racine de la ZMI on ajoute un objet de type "Plone Site".

On obtient un formulaire pour définir cet objet :

**Id** un nom court sans espace et en lettres minuscules

**Title** le titre du portail intranet

**Membership source** Créer un nouveau "acl\_users" (= correspond à l'action "Add User Folder" répertoire contenant la liste des utilisateurs et leurs rôles (Create a new user folder in the portal) ou bien utiliser un "acl\_users" préexistant (I have an existing user folder and want to use that instead ).

**Description** la description du site intranet

**Site type** Soit "Default Plone", ce sera un site où les anonymes auront accès en lecture aux informations publiées du site, soit "Private Plone site" où seuls les utilisateurs identifiés pourront accéder au contenu publié du site.

En cliquant sur le bouton "Add Plone Site" on se retrouve avec la ZMI partagée en deux, la liste des objets à gauche et l'interface graphique de notre site Plone à droite.

Pour revenir à la vue du contenu de notre site, on clique dans la liste des objets sur "Refresh" puis sur le nom (id) de notre site Plone qui vient d'apparaître dans cette liste.

### III.2 Fonctionnement de Plone - Contenu de l'objet Plone Site

#### III.2.1 MailHost

Est un formulaire permettant de configurer le smtp qui va être utilisé pour envoyer les emails lors des oublis de mots de passe par exemple.

#### III.2.2 Members (Members)

Dossier où chaque utilisateur aura son espace personnel (si autorisé voir dans la suite du document pour plus d'informations sur la configuration de Plone)

#### III.2.3 acl\_users (User Folder)

Dossier contenant la liste des utilisateurs du portail. On ajoute un utilisateur en cliquant sur "Add" on doit alors entrer son nom d'utilisateur (sans espace) son mot de passe et la confirmation de ce dernier.

Pour utiliser GRUF, il faut supprimer ce répertoire et le remplacer par un GroupUserFolder (via la ZMI).

### III.2.4 **content\_type\_registry**

Existe si on a utilisé le produit CMFTypes.

### III.2.5 **cookie\_authentication**

Là où on va spécifier la forme des cookies.

### III.2.6 **index\_html(Welcome to Plone)**

Page d'accueil du site, à personnaliser (voir aussi le chapitre VI sur l'organisation de la navigation).

### III.2.7 **plone\_utils (Various Plone Utility methods)**

### III.2.8 **portal\_actions (Contains custom tabs and buttons)**

Les actions de que l'on trouve ici sont celles qui sont appelées par le header des pages :

- portal\_tabs : les onglets (par défaut : rechercher, les news, les membres)
- folder\_buttons : les boutons d'actions possibles que l'on peut réaliser sur les objets contenus dans un dossier ou bien dans l'objet Plone Site
- object\_tabs : les onglets de propriétés de chaque objet pour les utilisateurs (état, appartenance de l'objet)
- object : action spécifique à un objet. Un objet Plone Site est vu comme un dossier et que donc chaque objet même à la racine du site Plone est contenu dans un dossier. Donc une action par défaut d'un objet (mis à part voir ) serait de lister le dossier auquel il appartient
- folder : action spécifique à l'objet dossier "voir le / accéder au contenu du dossier"

### III.2.9 **portal\_calendar (Controls how Events are shown)**

L'onglet intéressant dans cette partie est "Configure". On va pouvoir définir quel type d'objet sera visible dans le calendrier.

Pour qu'un type soit vu dans le calendrier (en plus de configurer cet onglet) il faudra remplir les champs début (date de publication) et fin (date d'expiration) de chaque objet que l'on veut voir dans le calendrier.

On va aussi choisir si on veut utiliser les sessions pour se rappeler des états du calendrier.

### III.2.10 **portal\_catalog (Indexes all content in the site)**

Cette partie d'un Plone Site est très utile pour connaître le contenu du site.

#### **L'onglet Contents**

Ce dossier contient un objet appelé "Vocabulary" qui est en fait le recensement de tous les mots, numéros... écrits dans les pages du site.

#### **L'onglet Catalog**

Cet onglet permet de gérer les items du catalogue du site.

En cliquant sur un des items de type /id\_du\_portail/x/... on obtient une page de renseignements très complets sur cet objet du portail.

Ces informations renseignent entre autres sur : le metatype, les dates de création, de parution, d'expiration, le créateur, l'id de l'objet, l'état, la description, le type, le titre, les allowedRolesAndUsers ...

### L'onglet Properties

Donne la valeur de titre pour l'objet portal\_catalog.

### L'onglet Indexes

Il présente le résultat de l'indexation, suivant des types par exemple :

- les FieldIndex donc les champs (pour le type, les dates d'expiration, de création...),
- les mots clés (KeywordIndex),
- les textes (TextIndex) pour les descriptions ...

### L'onglet Metadata

Contient les types de données qui vont être cataloguées (vus dans l'onglet Indexes).

### L'onglet Find Objects

Permet de trouver des objets et / ou bien de cataloguer des types d'objets suivant des champs correspondant aux types, ids, contenus, droits, permissions...

### L'onglet Advanced

Permet la gestion du catalogue en le mettant à jour (réindexation) le nettoyer pour le recréer ensuite.

Une partie de cet onglet permet aussi de gérer les "subtransactions". Cela entraîne une réduction de la demande en mémoire du catalogue, au détriment de la rapidité de recherche.

### L'onglet Undo

Annuler des transactions (les défaire)

### L'onglet Security

La sécurité sur l'objet portal\_catalog

### L'onglet Ownership

L'appartenance de l'objet portal\_catalog.

### L'onglet Actions

Si l'on veut ajouter des actions possibles correspondant à cet objet portal\_catalog.

### L'onglet Overview

#### III.2.11 portal\_discussion(Control how discussions are stored by default on content)

Un onglet action, par défaut sur "Reply"

#### III.2.12 portal\_factory (Responsible for the creation of content objects)

Aucune configuration possible.

**III.2.13 portal\_form (Used together with templates to do validation and navigation)**

Aucune configuration possible.

**III.2.14 portal\_form\_validation (Deprecated, not in use)**

Obsolète.

**III.2.15 portal\_memberdata (Handles the available properties on Members)**

Permet de gérer des données sur chaque utilisateur, rajouter des actions de gestion de membres ...

**III.2.16 portal\_membership****L'onglet Configuration**

La partie haute de cet onglet concerne les rôles des membres du portail (manager, reviewer...).

La partie basse de cet onglet va autoriser ou non la création lors de l'ajout d'un utilisateur d'un espace réservé à cet utilisateur.

**L'onglet Actions**

La catégorie de ces actions est "user", c'est à dire qu'elles ne concernent qu'un utilisateur authentifié.

Chaque action se présente sous la forme suivante :

Name	<i>My Folder</i>
Id	<i>mystuff</i>
Action	<i>string : \$portal/portal_membership/getHomeUrl/folder_contents</i>
Condition	<i>python : member and portal.portal_membership.getHomeFolder()</i>
Permission	<i>View</i>
Category	<i>use</i>
Visible ?	<i>x</i>

Remarque : si l'on a mis dans l'onglet "Configuration" que l'on ne voulait pas de dossier personnels pour les utilisateurs, cocher la case "Visible ?" restera sans effet.

**III.2.17 portal\_metadata****L'onglet Actions****L'onglet Overview****L'onglet Properties**

Cet onglet propose trois champs :

**1. Update Metadata Tool Properties**

Publisher :

**2. Add Metadata Element**

- Element :
- Multi-valued ?

### 3. Remove Metadata Element

#### L'onglet Elements = Update Element Metadata Policies

Les éléments par défaut sont Subject, Rights, Format, Language, Description, Title.

Pour chaque Content type (Evénement, Document, News...), on demande pour chaque élément :

1. s'il est requis
2. d'entrer une réponse par défaut
3. de proposer la réponse par défaut
4. d'entrer des termes appelés "vocabulaire" (correspond à la liste des "subjects")
5. de forcer l'utilisation des termes du vocabulaire

### III.2.18 portal\_migration

Cet outil sert lorsqu'une nouvelle version de Plone sort. Le reste du temps on a le message que notre objet Plone site est *up to date*.

La procédure en cas de nouvelle version de Plone, est d'extraire de l'archive la nouvelle version. D'arrêter Zope, puis de copier le contenu de CMFPlone-1.x.x, sauf Active Pack si on a un Zope de type 2.6.x, dans chemin\_de\_Zope/lib/python/Produ (afin de remplacer les anciens scripts) et ensuite d'aller dans portail fait sous l'ancienne version de Plone, de se rendre dans *portal\_migration* qui dit cette fois que le portail est *out of date* et il faudrait cliquer sur upgrader

### III.2.19 portal\_navigation

Aucune configuration possible.

### III.2.20 portal\_properties

Il contient quatre parties (donner des valeurs aux objets Zope) et des actions :

#### a. form\_properties

sert en particulier à faire valider le remplissage des champs de chaque Métatype.

#### b. navigation\_properties

donne des actions et des chemins de scripts à suivre suivant le succès ou l'échec d'une validation d'objet.

#### c. navtree\_properties

Les paramètres configurables sont :  
(Type : l = lines , b = boolean, i = int)

Name	Value(exemple)	Type	Explications
showMyFolderUserOnly	<i>x</i>	b	ne voir que son dossier dans Membres(Members)
includeTop	<i>x</i>	b	Voir dans le slot de navigation l'objet Plone Site
showMyFolderishSiblingsOnly	<i>x</i>	b	
showMyFolderishChildrenOnly	<i>x</i>	b	
showNonFolderishObject	<i>x</i>	b	ne pas voir que les objets de types "dossier"
topLevel	<i>0</i>	i	
batchSize	<i>30</i>	i	Longueur des champs de type <input type="text">
showTopicResults	<i>x</i>	b	Voir les commentaires
rolesSeeUnpublishedContent	<i>Manager – Owner..</i>	lines	Les rôles autorisés à voir le contenu non publié
sortCriteria	<i>isPrinipiaFolderish,desc</i>	l	Tri de contenu d'abord puis alphabétique
metatypesNotToList	<i>CMF Collector</i>	l	Ce qui ne peut être ajouté par l'interface "items"
parentMetatypesNotToQuery		l	
croppingLength	<i>18</i>	i	
forceParentsInBatch		b	
skipIndex_html	<i>x</i>	b	passer s'il existe un fichier index_html
rolesSeeContentsView	<i>Manager-Owner</i>	l	Les rôles pouvant voir la vue "contents"
rolesSeeHiddenContent	<i>Manager</i>	l	Les rôles pouvant voir le contenu caché du site

#### d. site\_properties (Site wide properties)

Name	Value (exemple)	Type	Explications
allowAnonymousViewAbout	<i>x</i>	boolean	
localTimeFormat	<i>%Y%m%d</i>	string	année mois jour
localLongTimeFormat	<i>%Y%m%d %I%M%p</i>	string	année mois jour heures minutes période(AM-PM)
default_language	<i>fr</i>	string	
default_charset	<i>iso-8859-1</i>	string	
use_folder_tabs	<i>Folder</i>	lines	sur les objets de type dossier
use_folder_contents	<i>Folder</i>	lines	sur les objets de type dossier
ext_editor		boolean	avoir un éditeur externe
available_editors	<i>None</i>	lines	
allowRolesToAddKeywords	<i>Manager – Reviewer</i>	lines	
auth_cookie_length	<i>0</i>	int	
allow_sendto		boolean	
enable_navigation_logging		boolean	

#### III.2.21 portal\_registration (Handles registration of new users)

Dans cette section on a une seule action par défaut : autoriser ou non l'inscription de nouveaux utilisateurs au portail. Cocher ou décocher la case à gauche de l'interrogation "Visible ?" permet à cette action d'être réalisée dans le portail.

#### III.2.22 portal\_skins (Control skin behaviour (search order etc))

Les dossiers de portal\_skins sont :

##### Images

Ce dossier contient les images de Zope (Le logo de Zope que l'on voit dans la ZMI par exemple).

### calendar

Ce dossier contient les images correspondant au calendrier d'un objet de type CMF site ainsi que les dtml-method de son affichage.

Pour faire commencer le calendrier le lundi et pas le dimanche comme par défaut il faut modifier `calendartool.py` qui est dans `plone/data/products/cmfcalendar` et modifier la valeur de `calendar.SetFirstWeekday`.  
Yves. 10/03/2003

### cmft

Apparait si on a ajouté le produit CMFTypes.

### content

Dans ce dossier on trouve les scripts d'édition dont les noms sont du type "*métatype\_édit*" et des dtml-method pour l'affichage, l'édition des métatypes dans un objet de type CMF Site.

### control

Ce dossier contient les scripts et quelques dtml-method correspondant aux actions (`folder_cut` par exemple) réalisables sur les objets d'un site de type CMF Site.

### custom

Ce dossier est spécial. C'est le seul dossier de `portal_skins` (en dehors de la racine de `portal_skins`) où l'on possède le droit d'écriture.

C'est donc le seul endroit où l'on va pouvoir créer des objets images, des dossiers, des `ZopePageTemplates...`

Ce dossier est spécial aussi car pour modifier (`customiser / personnaliser`) un objet (`ZopePageTemplates`, `dtml-method...`) existant, on va ouvrir l'objet à modifier et l'interface de Plone va proposer un bouton appelé *customize*. A la gauche de ce bouton se trouve une liste déroulant qui permet de choisir l'endroit où va être copié l'objet à modifier. Par défaut cette destination est le répertoire *custom*. Si l'on crée d'autres répertoires dans *custom* alors on pourra les choisir comme destination de la copie de l'objet.

Ainsi on comprend que pour que la modification d'un objet prenne effet, le dossier *custom* sera lu avant les autres répertoires de *portal\_skins*.

### generic

On va trouver dans ce dossier les formulaires (d'édition, d'ajout, des métadonnées) ainsi que les RSS d'un objet de type CMF Site.

### no\_css

Ce dossier ne contient qu'un objet *stylesheet\_properties* qui doit correspondre à la version imprimable des pages.

### nouvelle

Ce dossier contient deux objets, l'un intitulé *nouvelle\_stylesheet* et l'autre *stylesheet\_properties*. Ils correspondent à des feuilles de styles.

Les feuilles de styles seront détaillées dans le chapitre X "Design du site" dans la section X.2 intitulée "les css".

### plone\_3rdParty

Contient des objets de type *Filesystem Directory View* correspondant à des produits Zope intégrables dans un Plone site (CMFCalendar, CMFCollector, CMFTopic, CMFWeblog, CMFWiki, ratings).

### plone\_content

Dans ce dossier on trouve les ZopePageTemplates correspondant à l'affichage des objets des différents métatypes ainsi que les *métatype\_edit\_form*

### plone\_ecmascript

Les scripts javascripts : *plone\_formtooltips.jp* et *plone\_javascripts.js*.

### plone\_forms

Dans ce dossier on trouve les ZopePageTemplates d'affichage des objets des différents métatypes ainsi que les actions\_form. Par exemple, *personalize\_form* ou bien *member\_search\_form* ou encore *login\_form* ... On va aussi trouver un script appelé *logged\_in* (permet le login d'un utilisateur avec *getAuthenticatedMember*)

### plone\_images

On va y trouver les objets images : le logo de plone, les *metatype\_icons* ... Toujours la même chose, pour les personnaliser : les ouvrir, cliquer sur *Customize* (l'objet est alors copié dans le dossier *custom*, on nous propose d'entrer un titre, de parcourir le contenu d'un disque et d'uploader l'image que l'on souhaite avoir en remplacement).

### plone\_scripts

Comme son nom l'indique ce dossier contient des scripts python tels que la réalisation d'actions concernant les objets du Plone Site (*folder\_delete*) ainsi que des scripts pour le classement des objets (*filter et sort*), la création des objets (*createObject*), la fonction *Undo* pour l'annulation des modifications effectuées...

Ce dossier contient aussi le dossier *form\_scripts* qui lui contient les scripts de validation d'un formulaire d'édition/création d'un objet tel que le *validate\_id* et les scripts servant à l'édition d'un métatype en particulier.

### plone\_styles

Contient un *stylesheet\_properties*, *plonePrint.css*, *plonePresentation.css*, *plone.css*, *ploneNS4.css* et des dossiers contenant les mêmes feuilles de styles ainsi que des images de navigation (user, bullet...) sur un autre thème que celui par défaut.

### plone\_templates

On retrouve entre autres le *footer* et le *header* de toutes les pages dans ce répertoire.

(les détails de contenu des header et footer sont développés dans le chapitre VI intitulé "L'organisation la navigation" dans les sections correspondantes (header et footer).

Dans le *header* on trouve :

- appel de la skin du site
- la searchBox
- le logo du site

- les `portal_tabs`
  - les actions de la `personalBar`
  - le chemin de la page en cours (par exemple `home` -> les `news` -> la sortie de Plone 1.0)
- Dans le footer on trouve des références à Plone, et un logo `plone_powered`.

Dans ce répertoire, on va aussi avoir les `ZopePageTemplates` (ZPT) correspondant à `"main_template"`, `"folder_contents"`...

Ce dossier contient aussi le dossier `"ui_slots"` qui les `ZopePageTemplates` des slots (carrés situés à gauche et à droite du centre de l'affichage du site Plone) tels que `news_slot` qui va afficher les 5 derniers titres des 5 dernières `news` comme des liens cliquables.

On y trouve aussi un script python `"navigation_tree_builder"` qui va construire le slot `navigation_tree` en fonction de ce qui a été configuré dans le dossier `portal_properties` dans l'espace `navtree_properties`.

### `plone_wysiwyg`

Il contient `wysiwyg_popup` et `wysiwyg_support`.

### `topic`

Ce dossier contient les scripts, `ZopePageTemplates` et les `dtml-method` correspondant à l'ajout, affichage des commentaires (affichés si autorisés).

### `zpt_calendar`

La version du dossier `calendar` mais il est réalisé avec des `ZopePageTemplates` et des scripts python.

### `zpt_content`

La version du dossier `content` mais il est réalisé avec des `ZopePageTemplates` et des scripts python.

### `zpt_control`

La version du dossier `control` mais il est réalisé avec des `ZopePageTemplates` et des scripts python.

### `zpt_forum`

Apparaît si on a ajouté le produit `CMFForum`, on y trouve les `ZopePageTemplates` et scripts python correspondants.

### `zpt_generic`

La version du dossier `generic` mais il est réalisé avec des `ZopePageTemplates` et des scripts python.

## III.2.23 `portal_syndication` (Generate RSS folders)

Permet la syndication (envoi d'un flux d'informations vers d'autres sites web) et l'implémentation des politiques de syndication dans le site.

### III.2.24 `portal_types` (Controls the available Content Types in your portal)

Pour chaque type on peut définir les propriétés

Name	Value (exemple)	Type	Explications
Title		string	non obligatoire si i18n
Description	<i>Document can contain text...</i>	text	traduit via i18n
Icon	<i>document_icon.gif</i>	string	icône du métatype
Product meta type	<i>Document</i>	string	
Product name	<i>CMFDefault</i>	string	
Product factory method	<i>addDocument</i>	string	
Initial view name	<i>document_edit_form</i>	string	vue 1 = l'édition
Implicitly addable ?	<i>x</i>	boolean	
Filter content types	<i>x</i>	boolean	
Allowed content types	<i>une liste</i>	multiple selection	
Allow Discussion	<i>Manager – Reviewer</i>	boolean	

Pour chaque type on peut définir les actions

Chaque action (dans un onglet) à ajouter ou bien existante se présente ainsi :

Nom	champ à remplir	commentaires
Name	<i>View</i>	ne pas traduire si i18n utilisé
Id	<i>view</i>	<b>lire le chap.VI, section VI.1.1</b>
Action	<i>document_view</i>	objet appelé
Permission	<i>View</i>	permission nécessaire
Category	<i>object</i>	
Visible ?	<i>x</i>	oui ou non

### III.2.25 `portal_undo` (Defines actions and functionality related to undo)

Cette partie du Plone Site est utilisée pour défaire les actions réalisées.

Dans le premier onglet "Actions" deux actions sont proposées : liste des changements ou bien une annulation rapide de la dernière modification.

L'onglet Undo liste toutes les modifications depuis la création du portail.

### III.2.26 `portal_url` (Methods to anchor you to the root of your Plone site)

### III.2.27 `portal_workflow` (Contains workflow definitions for your portal)

Cette partie du Plone Site contient les actions correspondant à l'action de relecture puis validation des contenus...

La plus grande partie du workflow intégrée à Plone est déjà paramétrée.

## Chapitre IV

# INSTALLATION ET INTÉGRATION DES PRODUITS ZOPE DANS PLONE

### IV.1 [CMFForum](#)

Pour installer CMF Forum, dans la ZMI on crée une "External Method"

Id : install\_forum

Title : Install Forum (optional)

Module : CMFForum.Install

Function : install

### IV.2 [CMTypes](#)

**En premier :**

Pour avoir les deux types de documents de démonstration, on doit éditer le fichier `__init__` et passer la valeur de la clé `INCLUDE_DEMO_TYPES` de 0 à 1

**En second** dans le site Plone on crée un objet de type External Method :

Id : CMFTinstall

Title : Demo type installer

Module : CMFTypes.Install

Function : install

### IV.3 [ZPhotoSlides](#)

Les modifications possible à faire pour une intégration à Plone (css ...) :

### IV.3.1 La css de présentation des albums

La css de chaque album se trouve dans le dossier correspondant à chaque album. On peut la modifier aussi par l'interface graphique de management intégrée dans le produit ZPhotoSlides.

La feuille de style présentée ici correspond plutôt au style chromatique de Plone Default.

```

/* standard color */
body {
    font-family : verdana, arial, helvetica, sans-serif;
    font-size : 15px;
    color : #003333;
    background : #ffffff;
}
/* set margins on body content and focus tables with a div */
/* (classing the table does nix in Nav4) */
.margins {
    margin-left: 2%;
    margin-right: 2%;
}
.boxborder {
BORDER-RIGHT: #000000 1px solid;
BORDER-TOP: #000000 1px solid;
BORDER-LEFT: #000000 1px solid;
BORDER-BOTTOM: #000000 1px solid
}
.boxborderR{
BORDER-TOP: #000000 1px solid;
BORDER-LEFT: #000000 1px solid;
BORDER-BOTTOM: #000000 1px solid
}
.boxborderNOTOP{
BORDER-RIGHT: #000000 1px solid;
BORDER-LEFT: #000000 1px solid;
BORDER-BOTTOM: #000000 1px solid
}
.boxborderL{
BORDER-BOTTOM: #000000 1px solid;
BORDER-RIGHT: #000000 1px solid;
}
.boxborderRIGHT{
BORDER-RIGHT: #000000 1px solid;
}
p {
    font-size : 15px;
    font-family : verdana, arial, helvetica, sans-serif;
    color : #000000;
}
h1 {
    font-size : 24px;
    font-weight : bold;
    font-family : verdana, arial, helvetica, sans-serif;
    color : #8CACBB;
    background : transparent;
}
h2 {
    font-size : 22px;
    font-weight : bold;
    font-family : verdana, arial, helvetica, sans-serif;
    color : #000000;
    background : transparent;
}
h2.right {
    font-size : 20px;
    font-weight : bold;
    font-family : verdana, arial, helvetica, sans-serif;
    color : #000000;
}

```

```
    background : transparent;
    text-align : right;
}
h3 {
    font-size : 18px;
    font-weight : bold;
    font-family : verdana, arial, helvetica, sans-serif;
    color : #000000;
    background : transparent;
}
h4 {
    font-size: 11px;
    font-family : verdana, arial, helvetica, sans-serif;
    color: #000000;
    background : transparent;
}
td {
    font-size : 12px;
    font-family : arial, helvetica, sans-serif;
    color : #000000;
}
ul {
    font-size : 11px;
    font-family : verdana, arial, helvetica, sans-serif;
    color : #000000;
    background : transparent;
}
form {
    margin : 0px;
    padding : 0px;
    border : 0px;
}
input {
    font-family : clean, fixed, courier, monospace;
    font-size : 12px;
    color : #000000;
    background : #DEE7EC;
    border-left : 1px solid #333333;
    border-top : 1px solid #333333;
    border-right : 1px solid #333333;
    border-bottom : 1px solid #333333;
}
input[type="submit"]:hover, input[type="button"]:hover {
    color : #000000;
    background : #ffffff;
}

input[type="submit"]:focus, input[type="button"]:focus {
    color : #ffffff;
    background : #DEE7EC;
}
a:link {
    color : #436976;
    text-decoration : none;
}
a:visited {
    color : #436976;
    text-decoration : none;
}
a:active {
    color : #436976
}
a:hover {
    color : #436976;
    text-decoration : underline;
}
```

## IV.4 CMFCollector

**Auteur :** <http://dev.zope.org/Members/ajung> - Andreas Jung

**Licence :** ZPL (2.0)

**Version testée :** 0.9a (cvs)

**Téléchargeable :** <http://cvs.zope.org/CMF/CMFCollector/CMFCollector.tar.gz>

**Page du produit :** [http://www.zope.org/Members/ajung/CMFCollectorNG/CMFCollector\\_0\\_20beta1\\_released](http://www.zope.org/Members/ajung/CMFCollectorNG/CMFCollector_0_20beta1_released)

**Utilité :** Avoir un collecteur de problèmes et d'erreurs sur le site Plone. Le collecteur se présente sous la forme d'un dossier dont le premier onglet sera un formulaire de recherche des erreurs publiées, le second un formulaire lors de la rencontre d'un problème (on peut ajouter un fichier et/ou une image), les erreurs (issues) collectées peuvent être suivies, il existe aussi un onglet de configuration pour ce collecteur.

### IV.4.1 Installation du Produit :

1. Télécharger le produit CMFCollector
2. Dans "Chemin de Zope/lib/python/Products/", détacher l'archive tar -xzf CMFCollector.tar.gz
3. Redémarrer le serveur Zope.

### IV.4.2 Installation à la racine du Site Plone considéré :

1. Ajouter une "external method" (cliquer dans la liste d'objet ajoutable sur "external method") :
  - id : `install_cmfcollector`
  - title : `installation de CMFCollector`
  - module name : `CMFCollector.Install`
  - function name : `install`
2. Cliquer sur le bouton "add".
3. Pour réaliser l'installation (install), choisir l'onglet "test" parmi les onglets pour la configuration de l'external method "installation de CMFCollector".

### IV.4.3 Installation du Workflow pour CMFCollector

1. Dans le dossier du produit CMFcollector se trouve "collector\_issue\_workflow.zexp", le copier dans le répertoire de Zope appelé "import"
  2. Se rendre dans "portal\_workflow" du site Plone considéré et choisir l'onglet "contents"
  3. Cliquer sur le bouton "import/export"
  4. Comme nom, entrer "collector\_issue\_workflow.zexp"
  5. Se rendre dans l'onglet "workflows"
  6. Remplir les champs : Collector Issue -> collector\_issue\_workflow et cliquer sur le bouton "change"
- Il existe IssueManager dans la même catégorie de produits.

## IV.5 External Editor

### IV.5.1 Installer External Editor

Ce produit pour Zope qui va permettre d'éditer les scripts, pages templates et autres joyeusetés de la ZMI dans son éditeur préféré. Ci-dessous tout est expliqué pour les utilisateurs d'une distribution GNU/Linux, les utilisateurs de Windows® pourront se reporter à la documentation se trouvant sur <http://zope.org/Members/Caseman/ExternalEditor/install-win32>

Il est nécessaire d'avoir Python (2.1 ou supérieur) et Tk

– Télécharger :

1. Le produit Zope : <http://www.zope.org/Members/Caseman/ExternalEditor/>
2. La "Helper Application" : <http://www.zope.org/Members/Caseman/ExternalEditor/>

– Etapes d'installation :

1. Arrêter le serveur Zope
2. Détarrer l'archive téléchargée par exemple : `tar -xzvf ExternalEditor-0.7.tgz` dans `/cheminZope/libe/python/Product/`
3. Redémarrer le serveur Zope (si tout a bien fonctionné on obtient l'icone d'un petit crayon à droite de chaque fichier)

### IV.5.2 Configurer External Editor

Ci-dessous tout est expliqué pour les utilisateurs d'une distribution GNU/Linux, les utilisateurs de Windows® pourront se reporter à la documentation se trouvant sur <http://zope.org/Members/Caseman/ExternalEditor/install-win32>

– Pour paramétrer l'éditeur :

1. Dans un répertoire (position indifférente) détarrer l'archive Helper Application par exemple : `tar -xzvf zopeedit-0.7-src.tgz`
2. Se rendre dans le répertoire `zopeedit-0.7-src` créé, et exécuter avec les droits de root (sudo par exemple) : `python2.1 setup.py install`
3. Cela va créer dans `/usr/bin/` un exécutable appelé `zopeedit.py`

– Paramétrer Mozilla :

1. Edit (ou édition) Preferences (préférences)
2. Menu Navigator/Helper-Application (Navigateur, Applications) et créer un nouveau type (via le bouton New Type) et remplir SEULEMENT les champs suivants :
3. MIME Type : `application/x-zope-edit`
4. Description : Pour ExternalEditor Zope
5. Open it with : `/usr/bin/zopeedit.py`

– Configurer l'éditeur par défaut :

1. Aller dans la ZMI (avec le Mozilla paramétré bien sûr)
2. Cliquer sur une des petites icône "crayon"
3. Une fenêtre va s'ouvrir en haut à gauche si vous souhaitez Nedit comme éditeur, entrez Nedit (comme pour tous les éditeurs graphiques de X), si vous souhaitez vi il faudra ouvrir le fichier `/usr/bin/zopeedit.py` et à la ligne 664 (dans ma version) il faudra décommenter la ligne et entrer : `editor = xterm -e vi` , pour vim on pourra entrer `editor = gvim -f`

## IV.6 Localizer et TranslationService

### IV.6.1 prérequis

Il est nécessaire de récupérer aussi le fichier de langue (langage.po) qui nous intéresse dans i18n-Plone cvs sur <http://cvs.sourceforge.net/cgi-bin/viewcvs.cgi/plone-i18n/i18n/>

### IV.6.2 Via la ZMI, à la racine de notre site Plone

1. On ajoute un objet "Localizer" (appelé "français", langue : fr .
2. Puis on ajoute un objet "TranslationService" qui doit s'appeler "translation\_service" .

### IV.6.3 Pour que le langage utilisé soit celui du fichier que l'on a téléchargé

A savoir : plone-fr-FR.po

On copie simplement notre fichier langage.po téléchargé dans le répertoire i18n de Zope (il se trouve dans /cheminde-Zope/lib/python/Products/).

### IV.6.4 Ajouter le support des langues dans le MessageCatalog

On se trouve dans notre site Plone.

1. Dans l'objet Localizer, on ajoute un objet de type "MessageCatalog", qu'il faut appeler *Plone*.
2. Dans cet objet intitulé "*Plone*" de Localizer, aller sur l'onglet "Languages" et ajouter les langues dont on possède les fichiers.po et que l'on veut activer .
3. Une fois cela fait, cliquer sur le bouton *Import/Export*.
  - Dans la section "*Import*" on sélectionne le langage pour lequel on va aller chercher le fichier langage.po
  - On clique sur le bouton "*Browse*" et on va aller dans le répertoire /chemindeZope/lib/python/Products/i18n et choisir d'importer le fichier langage.po que l'on a téléchargé il n'y a pas si longtemps...
  - Cliquer sur l'import de ce fichier
4. Répéter ces étapes pour chaque langue que l'on veut que notre site Plone supporte.

### IV.6.5 Activer l'utilisation du Catalogue

Cela revient à faire un lien entre le catalogue et la traduction

1. On va sur l'objet du site Plone "*translation\_service*"
2. Dans "*Default Domain*" on écrit "*Localizer/Plone*" et on clique sur le bouton intitulé "*Change*"

### IV.6.6 Supporter plusieurs langues

C'est l'utilisateur qui va choisir :

1. Par défaut on choisit dans le Plone Setup quelle langue par défaut on va utiliser pour le site
2. Si le site supporte d'autres langues, la langue utilisée sera aussi fonction des préférences de l'utilisateur lors du choix de la langue de son navigateur

#### **IV.6.7 Caractères accentués**

D'autre part pour avoir dans les nouveaux Folder les caractères accentués il faut aller dans `/plone/portal_properties/site_properties` et fixer `default_charset` à `iso-8859-1` au lieu de `utf-8`.

Yves. le 10/03/2003

## Chapitre V

# EXPORTER PUIS IMPORTER SON SITE

### V.1 Le bouton import/export

Le format d'export/import sera .zexp

#### V.1.1 Exporter son site

1. On se place dans la ZMI de façon à voir notre objet de type Plone Site et de pouvoir cocher la case qui lui correspond
2. On sélectionne notre site Plone
3. On clique sur le bouton "Import/Export"
4. On arrive sur une page dont la partie haute concerne l'Export
5. Dans le champ "*Export object id*" est entré le nom (id) de l'objet Plone Site (on peut le changer sans majuscule ni espace)
6. Un choix reste à faire, à savoir pour "*Export to*" soit on choisit "*Download to local machine*" qui va être équivalent d'un "save as" sur la machine cliente, soit on peut choisir "*Save to file on server*" qui va créer dans le répertoire `/le_chemin_vers_Zope/var/` un fichier nommé "idduportail.zexp". (il y a une case à cocher pour obtenir un format XML)
7. Puis on clique sur le bouton intitulé "*Export*"

#### V.1.2 Importer son site

1. On se place dans la ZMI à l'endroit où l'on veut retrouver son site Plone
2. On clique sur le bouton "*import/export*"
3. On arrive sur une page dont la partie basse concerne l'Import
4. Dans une console on copie le idduportail.zexp que l'on a exporté précédemment du serveur local ou d'un serveur distant dans le répertoire intitulé `/le_chemin_vers_Zope/import/`
5. Dans le champ "*Import file name*" on entre le nom de l'objet à importer du type idduportail.zexp
6. Quand à l'appartenance, on peut faire le choix dans "*Ownership*" de s'approprier les objets importés (Take ownership of imported objects), soit de conserver les informations originales d'appartenance (Retain existing ownership information).
7. Puis on clique sur le bouton intitulé "*Import*"

## V.2 Après l'import, on va avoir l'erreur de type `__getitem__`

### V.2.1 Pourquoi ?

Car le chemin vers les dossiers de skins sont codés en dur dans la vue de chaque répertoire.

Si on déplace le site vers un autre serveur alors il faut corriger le chemin recherché pour chaque objet utilisant les skins.

### V.2.2 Corriger

Grâce à un script trouvé sur <http://www.zopelabs.com/> soumis par Robert le 14/02/2003 (merci beaucoup)  
( au 23/02/2003 sur <http://www.zopelabs.com/cookbook/1021216110> )

```
## Script (Python) "setDirpath"
##bind container=container
##bind context=context
##bind namespace=
##bind script=script
##bind subpath=traverse_subpath
##parameters=
##title=
##
import string

#get possible skins
newDirs=container.manage_addProduct['CMFCore'].manage_listAvailableDirectories()

#we can not use os.sep so we must look ourselves what
# the path separator is
if newDirs[0].find('/') > -1:
    newSep = '/'
else:
    newSep = '\\'

objs = context.objectItems('Filesystem Directory View')
for n,o in objs:
    found = 0
    path = o.getDirPath()
    if path.find('/') > -1:
        sep = '/'
    else:
        sep = '\\'
    #look for new path by comparing the last 2 elements
    dArr = string.split(path,sep)
    for i in range(len(newDirs)):
        # if we found the directory skip the rest
        if found:
            continue
        ndArr = string.split(newDirs[i],newSep)
        if string.lower(ndArr[-1]) != string.lower(dArr[-1]):
            continue
        if string.lower(ndArr[-2]) != string.lower(dArr[-2]):
            continue
        # if we land here, we found the real one
        found = 1
        o.manage_properties(dirpath = newDirs[i])
    if(found == 0):
        print path, ' not found'

print 'finished'
return printed
```

## Chapitre VI

# L'ORGANISATION DE LA NAVIGATION

### VI.1 Les dossiers

Dans le site Plone, chaque objet est un lien vers le contenu de l'objet. Ainsi, pour créer des rubriques (en fonction des services de l'entreprise par exemple) le plus simple est de créer un dossier correspondant à chaque service, à le publier et à rajouter à l'intérieur ce qui correspond au service.

#### VI.1.1 Créer une rubrique (onglet par exemple)

C'est plutôt simple à réaliser :

1. Dans la ZMI, ouvrir le site plone, puis ouvrir portal\_actions. Chaque partie composée de 5 à 6 champs, correspond à un bouton (onglet ou lien de la personnal bar). La catégorie des onglets est "portal\_tabs".
2. Aller en bas de cette page et ajouter un onglet en remplissant les champs d'un nouveau bouton pour qu'ils correspondent à un lien vers le dossier qui propose le service que l'on veut voir apparaître en onglet (pour réaliser un onglet il y a des champs comme id (forum), action (lien vers le dossier), condition (member...), permission (view...), category (portal\_tabs), visible)

#### Remarque sur le nom donné à l'id :

Si dans portal\_actions on ne donne pas la même chaîne de caractères que ce qu'il va chercher, pour définir "id", on obtient un problème.

Le problème sera que l'onglet créé n'aura pas de coloration de fond lorsqu'il sera sélectionné comme c'est le cas pour news, accueil... Pourquoi ? car dans le *header* le paramètre passé pour savoir si la classe de css va être class="plain" ou class="selected" est **id** !

exemple qui applique la bonne classe du style (plain et selected) :

```
name=s'identifier
id=login_form
string :$portal_url/login_form
permission=view
category=portal_tabs visible (coché)
```

#### VI.1.2 Droits sur les dossiers :

Pour ce qui est des droits sur ces dossiers il existe lors de la création de l'objet (et pour le propriétaire de l'objet) un onglet sur l'objet intitulés "Local Roles". Dans cet onglet on trouve une interface de gestion de membres : on peut alors ajouter au dossier un ou des managers, des relecteurs, des membres.

## VI.2 Les slots

Par défaut l'organisation du site Plone ressemble à :

Logo	
Onglets de navigation	
Barre personnalisée (actions pour utilisateur authentifié)	
Slots :: Contenu	:: Slots
Gauches :: de la page	:: Droits
:: d'accueil	::
Bas de page	

Il faut penser que la navigation pour les droitiers est plus aisée si il faut cliquer à droite et pour les gauchers s'il faut cliquer à gauche.

Comme on ne peut dupliquer cette information de navigation, on va créer une double entrée : une navigation écrite en ligne en haut à gauche, une au centre plus détaillées, une en haut brève et imagée, une à droite (listing). Le tableau ci dessous pourrait correspondre à ce que l'on vient de décrire.

Logo	Navigation concise imagée
Onglets de navigation et navigation personnalisée (actions pour membre)	
Index	:: Slots
image	:: Droits
	détaillé ::
Bas de page (lien vers haut de la page)	

### VI.2.1 Créer un slot

On crée un fichier qui va afficher "à la slot" les informations que l'on veut.

#### Remarque importante

Les slots sont comme vu précédemment dans /monplone/portal\_skins/templates/ui\_slots. Pour en créer un, il est intéressant de s'inspirer des templates de slots déjà présents.

Pour celui de recherche on prend le code contenu dans /monplone/portal\_skins/plone\_templates/header et on rajoute un entête de fichier et une fin de fichier.

Pour créer ce fichier on doit se placer dans /monplone/portal\_skins/custom (le seul répertoire de portal\_skins où on a le droit d'écriture) et on crée un objet de type "page template" qu'on appelle, pour garder la nomenclature, "search\_slot". (noter la présence de <div metal :define-macro="searchBox">, pour pouvoir l'appeler par here/search\_slot/macros/searchBox et respecter ainsi la nomenclature du site Plone.).

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//FR"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

```
<html xmlns:tal="http://xml.zope.org/namespaces/tal"
      xmlns:metal="http://xml.zope.org/namespaces/metal"
      i18n:domain="plone">

<body>
<!-- The search box -->

<div metal:define-macro="searchBox">
  <div class="box">
    <h5>Rechercher</h5>
    <div class="body">
      <br>
      <form name="searchform"
            action="search"
            tal:attributes="action string:${portal_url}/search" >
&nbsp; <input id="searchGadget"
           name="SearchableText"
           type="text"
           size="16"
           value=""
           tal:attributes="value request/SearchableText|nothing;" />
&nbsp; <input class="context searchButton"
           type="submit"
           value="Search"
           i18n:attributes="value" />
      <br><br>
    </form>
  </div>
</div>
</body>
</html>
```

### VI.2.2 Organiser les slots

A la racine de notre site Plone, on se trouve par défaut dans l'onglet "Contents", il suffit d'aller sur l'onglet "Prop-erties".

Il se présente ainsi :

Name	textitValue (exemple)	Type	Explications
Title	<i>Intranet</i>	string	le nom de du site Intranet
Description	<i>Le site intranet</i>	text	
email_from_address	<i>postmaster@localhost</i>	string	
email_from_name	<i>Administrateur du Portail</i>	string	
validate_email		boolean	oui ou non
left_slots		lines	colonne à gauche
right_slots	<i>/here/navigation_tree_slot/macros/navigationBox</i> <i>here/login_slot/macros/loginBox</i> <i>here/search_slot/macros/searchBox</i> <i>here/news_slot/macros/newsBox</i>	lines	navigation le login la recherche les 5 dernières news

On valide en utilisant le bouton "SaveChanges"

Voilà, en cliquant sur l'onglet "View" on s'aperçoit que les slots n'apparaissent plus qu'à droite des pages.

## VI.3 Header et Footer

### VI.3.1 Le header

Le fichier header original se situe dans l'objet PloneSite puis `portal_skins/plone_templates`

Pour le comprendre le voici commenté (l'indentation et le mode verbatim de  $\LaTeX$  ne permettent pas un rendu fidèle du code) :

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<head metal:define-macro="html_header"
tal:define="global portal_url here/portal_url;
          ztu modules/ZTUtils;
          current_skin python:ztu.make_query(skin=request.get(here.portal_skins.getRequestVarname(), ''))">
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1"
      tal:define="charset here/portal_properties/site_properties/default_charset|string:iso-8859-1"
      tal:attributes="content string:text/html;;charset=${charset}" />
```

Le titre de la page

```
<title tal:define="portal_title here/portal_properties/title;
                here_title python:test( portal_title != here.title_or_id()
                , here.title_or_id()
                , template.title_or_id());"
      tal:content="string:$portal_title - $here_title">Title of page</title>
```

Là où interviennent les "keyvalues" des pages (comparables aux métatags)

```
<metal:block metal:define-slot="base" />

<meta tal:condition="here/listMetaTags|nothing"
      tal:repeat="keyval here/listMetaTags"
      tal:attributes="name python:keyval[0];
                    content python:keyval[1];" />
```

Appel des css.

Les css sont des feuilles de styles (cascading style sheets)

```
<!-- Basic crude style for Netscape4.x -
This can be removed if you don't want the special NS4 look
- it will still work, just be plain text instead. -->
```

```

<link rel="Stylesheet" type="text/css" href=""
tal:attributes="href string:$portal_url/ploneNS4.css" />

<!-- Main style sheets for CSS2 capable browsers -->
<style type="text/css" media="all"
tal:content="string: @import url({here/portal_url}/plone.css?${current_skin});">
</style>

<!-- Custom style sheet if available -->
<style type="text/css" media="all"
tal:condition="exists: here/ploneCustom.css"
tal:content="string:@import url({here/portal_url}/ploneCustom.css);"></style>

<!-- Style sheet used for printing -->
<link rel="stylesheet" type="text/css" media="print"
href="" tal:attributes="href string:$portal_url/plonePrint.css" />

<!-- Style sheet used for presentations
(Opera is the only browser supporting this at the moment) -->
<link rel="stylesheet" type="text/css" media="projection"
href="" tal:attributes="href string:$portal_url/plonePresentation.css" />

<link rel="shortcut icon" href="" type="image/x-icon"
tal:attributes="href string:$portal_url/favicon.ico" />

<link rel="home" href="" title="Front Page"
tal:attributes="href string:$portal_url" />

<link rel="search" href=""
title="Search this site"
tal:attributes="href string:$portal_url/search_form" />

<!-- Disable IE6 image toolbar -->
<meta http-equiv="imagetoolbar" content="no" />

<metal:block metal:define-slot="css_slot" tal:replace="nothing">
Inserts CSS specified from a page.
</metal:block>

<metal:block metal:define-slot="head_slot" tal:replace="nothing">
Flexibility for published templates to inject stuff into the head tag.
</metal:block>

<metal:block metal:define-slot="javascript_head_slot" tal:replace="nothing">
Inserts javascript specified from a page.
</metal:block>

<!-- Common Plone ECMAScripts -->
<script type="text/javascript"
tal:attributes="src here/plone_javascripts.js/absolute_url">
</script>

<tal:block
tal:define="defaulttips python:here.portal_memberdata.getProperty('formtooltips');
showdefault python:test(defaulttips, 'true', 0);
formtooltips python:getattr(here.portal_membership.getAuthenticatedMember(),
'formtooltips', showdefault)"
tal:omit-tag="">

```

```
<script type="text/javascript"
    tal:condition="formtooltips"
    tal:attributes="src here/plone_formtooltip.js/absolute_url"></script>
<metal:block tal:condition="not:formtooltips">
  <tal:block tal:replace="structure string:&lt;/script
  language='javascript' type='text/javascript' &gt;" />

<metal:block >function formtooltip(el,flag){}</metal:block>
  <tal:block tal:replace="structure python:'&lt;/script&gt;' " />
</metal:block>

</tal:block>
</head>
```

Ici prennent fin les balises html <head></head> qui comme on a pu le voir contiennent les métat-tags, les balises <title></title> ainsi que les appels des css.

```
<div metal:define-macro="portal_header"
  il8n:domain="plone"
  tal:define="portal_url here/portal_url;
    ztu modules/ZTUtils;
    current_skin
    python:ztu.make_query(skin=request.get(here.portal_skins.getRequestVarname()),
  '))">

<tal:block tal:condition="not: actions|nothing">
  <span tal:define="global actions here/portal_actions/listFilteredActionsFor"
    tal:replace="nothing" >
    This is here just in case actions is not defined.
    This would happen if you were calling from a DTML Document
  </span>
</tal:block>
```

Ci dessous le code de l'espace de recherche (la search box) :

```
<div class="top">

  <div class="searchBox">
    <form name="searchform"
      action="search"
      tal:attributes="action string:${portal_url}/search" >

      <input id="searchGadget"
        name="SearchableText"
        type="text"
        size="20"
        value=""
        tal:attributes="value request/SearchableText|nothing;" />

      <input class="context searchButton"
        type="submit"
        value="Search"
        il8n:attributes="value" />
    </form>
  </div>
```

Le logo du site (en haut à gauche) :

```
<a href="/"
  tal:attributes="href portal_url"
  tal:define="logo here/logo.jpg;
             uri logo/absolute_url" >

  </a>
</div>
```

Les onglets du portail sont appelé par le code suivant :

```
<hr size="" class="netscape4" />
<div class="tabs"
  metal:define-macro="portal_tabs"
  tal:define="portal_tabs actions/portal_tabs|nothing;
             global selected_tabs python:here.selectedTabs(template.getId(), here);
             selected_portal_tab selected_tabs/portal;" >
  <metal:block tal:repeat="tab portal_tabs">
<a href="" class=""
  tal:attributes="href tab/url;
                 class python:test(selected_portal_tab==tab['id'], 'selected', 'plain')">
  <span tal:omit-tag="" il8n:translate="" >
  <span tal:replace="tab/name">dummy</span></span>
</a>
  </metal:block>
</div>
```

La barre comportant le nom de l'utilisateur loggué - le chemin vers son dossier - l'accès au personalize\_form (ses préférences) - l'action de se déconnecter s'appelle "personalBar"

Les actions définies des utilisateurs (membres ou non) sont donc appelées par le code qui suit :

```
<div class="personalBar"
  metal:define-macro="personalBar"
  tal:define="portal_url here/portal_url;
             user_actions actions/user;
             global_actions actions/global;
             display_actions python:user_actions[:-1]+global_actions+user_actions[-1:];">&nbsp;
  <a href=""
    tal:omit-tag="not: here/portal_membership/getHomeFolder"
    tal:condition="not:here/portal_membership/isAnonymousUser"
    tal:attributes="href string:${here/portal_membership/getHomeUrl}">
```

```

    <img tal:replace="structure here/user.gif" />

    <span tal:replace="user/getUserName">john</span>
</a>

    <span tal:condition="here/portal_membership/isAnonymousUser"
i18n:translate="you_are_not_logged_in">
You are not logged in</span>

    <metal:block tal:repeat="action python:display_actions">
    <a href=""
        tal:attributes="href action/url">
    <img tal:replace="structure here/linkOpaque.gif" />

        <span tal:omit-tag="" i18n:translate="">
<span tal:replace="action/name">dummy</span></span>
        </a>
    </metal:block>
</div>

```

Début de la barre de chemin (barre commençant par accueil et se poursuivant par le chemin parcouru dans le Plone-site, chaque lieu passé est un lien) :

```

<div class="pathBar"
    metal:define-macro="pathBar"
    tal:define="breadcrumbs python:here.breadcrumbs(here);
                portal_url here/portal_url;">

    <a href="#"
        tal:condition="here/portal_membership/getHomeFolder"
        tal:attributes="href string:${here/absolute_url}/addtoFavorites"
        tal:define="addFavorite here/addFavorite.gif">

<img
    tal:replace="structure python:addFavorite.tag(css_class='addFavorite',
                                                title='Add this location to My Favorites')"
    />

</a>

    <span tal:omit-tag="" i18n:translate="you_are_here">You are here:</span>
<span tal:condition="breadcrumbs" tal:omit-tag="">
    <a i18n:translate="tabs_home" tal:attributes="href portal_url">Home</a>
    <strong>&raquo;</strong>
</span>

    <span tal:condition="not: breadcrumbs"
i18n:translate="tabs_home" tal:omit-tag="">
    Home
</span>

    <span tal:repeat="crumb breadcrumbs">

    <metal:block tal:define="global is_last repeat/crumb/end" />

        <a href="#"
            tal:condition="python:not is_last"
            tal:attributes="href python:crumb[1]"
            tal:content="structure python:crumb[0]">crumb</a>
        <strong tal:condition="python:not is_last"> &raquo;</strong>
        <span tal:condition="python: is_last"
            tal:content="structure python:crumb[0]">crumb</span>

```

```

        </span>

    </div>

    <hr size="" class="netscape4" />

</div>

```

Fin de la barre de chemin.

```

<div class="message"
    i18n:domain="plone"
    metal:define-macro="portal_message"
    tal:define="portal_message request/portal_status_message | nothing"
    tal:condition="portal_message">

    <img tal:replace="structure here/info.gif" />

    <strong tal:content="portal_message" i18n:translate=""></strong>
</div>

```

### Quelques modifications possibles :

Pour l'interface de navigation :

#### VI.3.2 Le footer

Le fichier "footer" original se situe aussi dans l'objet PloneSite puis portal\_skins/plone\_templates

### Le bas de page se présente ainsi :

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<div class="footer" metal:define-macro="portal_footer" i18n:domain="plone"
    tal:define="portal_url here/portal_url">

<a href="http://plone.org">
<img src=""
    height="30"
    width="96"
    border="0"
    alt="Powered by Plone"
    title="This site was built using the Plone
        Content Management System.Click for more information."
    tal:attributes="src string:$portal_url/plone_powered.gif"
    i18n:attributes="alt title" />
</a>

```

Ci dessus le logo de Plone, que les auteurs demandent de laisser ou de remplacer par un lien ou de créer une page évoquant la présence d'un Plone Site.

```
<!--
```

```
    Please keep the Plone Powered button (or a textual link to us) if you use
    Plone on your site. It's a nice token of gratitude, and we appreciate your
    help in promoting the Plone name.
```

```
    Plone is powered by the combined forces of Zope and CMF, two absolutely
    great systems made by Zope Corporation (http://zope.com) and they in turn
    are based on the best programming language in the world - Python
    (http://www.python.org). We owe these guys a lot, thanks for making Plone
    possible!
```

```
-->
```

Les contributeurs :

```
<br />
```

```
<span il8n:translate="description_copyright" tal:omit-tag="">
Plone is Copyright &copy; 2000-
<span il8n:name="current_year"
tal:define="now modules/DateTime/DateTime" tal:content="now/year" />
by
<span il8n:name="limi"><a href="http://limi.net">Alexander Limi</a></span>,
<span il8n:name="runyaga"><a href="http://www.runyaga.com">Alan Runyan</a></span>,
<span il8n:name="blacktar"><a href="http://blacktar.com">Vidar Andersen</a></span>.
</span>
```

Mot de la fin ;)

```
<hr size="" class="netscape4" />
```

```
<strong class="netscape4">
If you can read this text, it means you are not experiencing the Plone design at its best.
Plone makes heavy use of CSS, which means it is accessible to any internet browser,
but the design needs a
<a href="http://www.webstandards.org/upgrade/">
standards-compliant browser to look like we intended it</a>.
Just so you know ;)
</strong>
```

```
</div>
```

### Quelques modifications possibles :

Le rajout d'un lien vers le haut de la page :

```
<span class="description">Site Intranet de .... — <a href="javascript :window.scrollTo(0,0);"> Retour au Haut de la
Page</a></span>
```

## VI.4 Avoir un slot de navigation (à la racine) en images

Pour ne pas aller dans la ZMI pour créer et remplir ce slot de navigation, que l'on va imaginer vertical, montrant seulement les répertoires de type 'Plone Folder', publiés et présents à la racine du site. On va publier dans les répertoires

concernés, un contenu de type image appelé 'image.gif' qui est en réalité un bouton.

### VI.4.1 le code

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns:tal="http://xml.zope.org/namespaces/tal"
      xmlns:metal="http://xml.zope.org/namespaces/metal"
      i18n:domain="plone">

<body>
<div metal:define-macro="navdeuxBox"
      i18n:domain="plone"
      tal:define="isContainer here/isPrincipiaFolderish|nothing;
                  portal here/portal_url/getPortalObject;
                  global portal_url portal;"
      tal:condition="python:test(not site_properties.allowAnonymousViewAbout and
                                site_properties.portal_membership.isAnonymousUser(), 0, 1)"
      tal:omit-tag="">

<table>

<metal:block tal:repeat="obj python:here.portal_url.getFolderContents(suppressHiddenFiles=1)"
      tal:condition="python:test(not site_properties.allowAnonymousViewAbout
                                and site_properties.portal_membership.isAnonymousUser(), 0, 1)" >
<tr>
    <td>
<a href="#" tal:condition="python:obj.meta_type=='Plone Folder'"
      tal:attributes="href string:${obj/absolute_url}/view">
    </a>
    </td>
</tr>
</metal:block>
</table>
</div>
</body>
</html>
```

### VI.4.2 les remarques

Pour que la page ne soit pas trop lourde il faut ajouter un RAMcache pour ces images.

## Chapitre VII

# LES MÉTADONNÉES

### VII.1 Utilité et Présentation

Une des méthodes pertinentes pour améliorer la recherche de documents est l'ajout de métadonnées à ces documents.

Les métadonnées sont des informations techniques et descriptives ajoutées à un document pour mieux le qualifier (titre, auteur, date de publication, brève description...).

Le schéma des métadonnées utilisé dans Plone par défaut est basé sur celui proposé par l'organisation appelée Dublin Core Metadata Initiative (DCMI) apparaissant dans Plone comme "Dublin Core".

Il existe d'autres types de schémas correspondant à des documents au contenu plus pédagogique sont proposés en particulier le schéma Learning Object Mode (LOM), issu de l'organisme de standardisation IEEE.

### VII.2 Les métadonnées dans Plone

Elles se présentent donc d'après le schéma Dublin Core.

On peut remplir les champs suivants :

Nom dans Plone pour le manager	Pour l'utilisateur	Explications
Identifier		chemin absolu du document ou dossier et nom (id)
Title	Titre	le titre du document
Description	Description	brève description
Subject	Mots clés	peuvent être nouveaux, forcés...
Contributors	Contributeurs	
Creation Date		non modifiable
Last Modified Date		non modifiable
Effective Date	Date de publication	parution
Expiration Date	Date d'expiration	obsolète
Format	Format	par exemple text/html
Language	Langue	
Rights	Droits ou Copyright	

C'est donc dans l'objet `portal_metadata` à la racine du site Plone que l'on peut paramétrer les métadonnées et leurs

entrées.

#### **Rappels sur l'onglet Elements = Update Element Metadata Policies**

Les éléments par défaut sont Subject, Rights, Format, Language, Description, Title.

Pour chaque Content type (Evénement, Document, News...), on demande pour chaque élément :

1. s'il est requis
2. d'entrer une réponse par défaut
3. de proposer la réponse par défaut
4. d'entrer des termes appelés "vocabulaire" (correspond à la liste des "subjects")
5. de forcer l'utilisation des termes du vocabulaire

### **VII.3 Validator ou comment obliger l'entrée des métadonnées**

Pour comprendre cette méthode, un exemple est démonstratif :

#### **VII.3.1 Script validant la création d'un dossier**

Ce script se trouve dans `portal_skins/plone_scripts/form_scripts` et se nomme `validate_folder_edit` :

```
## Script (Python) "validate_folder_edit"
##bind container=container
##bind context=context
##bind namespace=
##bind script=script
##bind subpath=traverse_subpath
##parameters=
##title=Validates a folder edit_form contents
##
validator = context.portal_form.createForm()
validator.addField('title', 'String', required=1)
errors=validator.validate(context.REQUEST)
if errors:
    return ('failure', errors,
           {'portal_status_message':'Please correct the indicated errors.'})
return ('success', errors,
       {'portal_status_message':'Your folder properties have been saved.'})
```

Dans ce script on note que l'on peut rajouter pour chaque métadonnées que l'on veut une ligne du type de :  
`validator.addField('title', 'String', required=1)`

En changeant la vérification du champ 'title' et le type de remplissage par les champs pour lesquels on souhaite forcer le remplissage.

On peut donc rajouter :

```
validator.addField('description', 'Text', required=1)
```

### VII.3.2 L'aspect du formulaire d'édition doit montrer que le champ est obligatoire

Cela ne suffit pas car l'utilisateur ne saura pas que le champ doit être rempli, on va le lui indiquer en changeant dans le formulaire le style du label du champ description.

On va donc ouvrir `/portal_skins/plone_forms/folder_edit_form`, seul le code concernant le champ description du formulaire est répété ci-dessous :

```
<div class="row">
  <div class="label">
    <span i18n:translate="label_description">Description</span>

    <div id="description_help"
          i18n:translate="help_description"
          class="help"
          style="visibility:hidden">
      Enter a brief description of the folder.
    </div>
  </div>
  <div class="field">
    tal:define="description request/description | here/Description ">
    <textarea cols="40"
              rows="5"
              tabindex=""
              name="description"
              onfocus="formtooltip('description_help',1)"
              onblur="formtooltip('description_help',0)"
              tal:content="description"
              tal:attributes="tabindex tabindex/next"
            >
      Description
    </textarea>
  </div>
</div>
```

La seule chose à changer pour ajouter un point rouge à côté du label du champ texte à remplir est :  
`<div class="label">` que l'on change en `<div class="label required">`

## Chapitre VIII

# RÉFLEXION SUR LES MÉTADONNÉES

### VIII.1 Le concept

Chaque utilisateur peut avoir (si on le décide voir l'onglet configuration de l'objet site Plone) un dossier personnel. C'est cette possibilité qui va rendre la réflexion sur les métadonnées possible.

#### VIII.1.1 Le contexte

Imaginons que vous désirez :

- Un site intranet / extranet où des utilisateurs vont pouvoir publier des contenus
- Un besoin de modération avant publication
- Une idée de dossiers où, par thème (donné plus loin de niveau 0) puis sous-thème (donné plus loin comme niveau 1) un utilisateur pourra publier ses contenus

#### VIII.1.2 Design de ce contexte

1. Installation de Zope, installation des produits CMF et Plone
2. Création d'un manager pour le serveur Zope (dans le acl\_users de la ZMI)
3. Création d'un objet Plone Site (nommage...)
4. Création de quelques utilisateurs dans le acl\_users du site Plone créé (un manager, un relecteur, deux membres)
5. Passage à l'interface view du site Plone, se loguer en manager
6. Passage dans la boîte de navigation sur la droite en "switch to content view"
7. Création des dossiers se rapportant aux thèmes de niveaux 0 par exemple comme titre "Biologie"
8. Paramétrage des métadonnées, pour pouvoir classer les contenus par thèmes de niveau 0, puis par sous-thèmes de niveau 1 qui pourront être par exemple "Biologie moléculaire", "Embryologie" et "Bactériologie"
9. Création des pages d'affichages (templates) qui correspondront à l'intérieur du dossier "Biologie" aux deux sous-thèmes de niveau 1.
10. Mise en place de ces templates dans le dossier "Biologie" pour qu'un membre (voire un anonyme par défaut) puisse se rendre dans la rubrique "Biologie" et consulter la liste de tous les contenus correspondant à la "Biologie Moléculaire"

## VIII.2 Les détails de la mise en oeuvre

### VIII.2.1 Installations

- On installe Zope, puis CMF et Plone comme décrit dans le chapitre sur les installations.
- Dans la ZMI : on réalise :
  - la création d'un manager de Zope dans le conteneur acl\_users
  - la création d'un objet Site Plone
  - à l'intérieur de ce site Plone on ajoute des utilisateurs dans le conteneur acl\_users
- Notez qu'on peut Localiser le site Plone lors de cette étape, comme toujours ajout d'un objet Localizer avec un message catalog Plone, ajout d'un objet Translation\_service avec comme domaine par défaut Localizer/Plone.

### VIII.2.2 Adaptation de l'architecture du site Plone

- On s'identifie en manager dans notre site Plone (via `http://zopeserver:8080/monsieplone/` par exemple).
- On va se situer dans la zone de la boîte de navigation et passer en mode "Switch to content view"
- La liste de certains des dossiers contenus à la racine du site Plone s'affiche. Ajouter un dossier appelé "biologie" comme id, et "Biologie" comme titre, remplir la description du dossier "Tous les contenus relatifs à la Biologie"

### VIII.2.3 Paramétrage des métadonnées

- Se rendre dans la ZMI du site Plone (via `http://zopeserver:8080/monsieplone/manage` par exemple)
- Ouvrir `portal_metadata`, se rendre sur l'onglet Elements (Update Element Metadata Policies)
- Si pour chaque type de contenu, on veut forcer l'édition des métadonnées, et surtout l'édition du Sujet "SUBJECT" (selon l'appellation Dublin Core), les mots clés dans l'appellation courante, il faut considerer les champs correspondants à "Content Type <default>"
- Le paramétrage serait alors :

Nom du Champ	Action à réaliser	Explications
Supply default	Cocher ou non	Cocher si on veut proposer quelque chose par défaut
Enforce Vocabulary	Cocher ou non	Obliger l'utilisation du vocabulaire décrit dans Vocabulary
Required	Cocher ou non	Obliger la sélection d'au moins un mot clé (subject)
Default	Ecrire une chaîne de caractère ou non	Sera le mot clé proposé par défaut
Vocabulary	Ecrire une liste de mots clés	Seront les mots clés proposés (aller à la ligne pour changer de mot clé) pour notre exemple, entrer : Biologie Moléculaire Embryologie Bactériologie

- Ne pas oublier de cliquer sur le bouton "Update" pour valider le paramétrage

### VIII.2.4 Ecrire les pages templates pour le classement

#### Premier objectif :

Le premier objectif est de faire une requête sur le sujet "Biologie moléculaire", pour ce faire :

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en-US"
    lang="en-US"
    metal:use-macro="here/main_template/macros/master"
    il8n:domain="plone">

    <metal:block fill-slot="base">
        <tal:block tal:define="dummy python:request.set('disable_border', 1)"
            tal:replace="nothing">
            Disable the tabs
        </tal:block>
    </metal:block>

<!-- Change the following line -->

<div metal:fill-slot="main"
    tal:define="results python:request.get('subject',
    here.portal_catalog.searchResults( Subject='Biologie moléculaire'
                                     , sort_on='Date'
                                     , sort_order='reverse'
                                     , review_state='published'));"
    tal:condition="python:test(results, 1, 0)">

    <div>
    <h1 tal:content="here/Title"> Le titre </h1>
    <br><br>
        <div>
            <table align="center" width="90%">
                <tr>
                    <td>Type</td>
                    <td>Titre</td>
                </tr>

                <tr tal:repeat="item results">

                    <span tal:define="object item/getObject">

                        <td>
                            <img tal:attributes="src string:${object/getIcon}">
                            <span tal:content="object/portal_type"> Type </span>
                        </td>
                        <td>
                            <a href=""
                                tal:attributes="href object/absolute_url">

                                <span tal:content="object/Title"> Titre </span>
                            </a>
                        </td>
                    </span>
                </tr>
            </table>
        </div>
    </div>
</div>
</html>

```

## Deuxième objectif :

Le second objectif sera de créer une page template, permettant l'affichage des contenus se rapportant dans notre exemple à l'ensemble Biologie, à savoir Embryologie, Bactériologie et Biologie moléculaire.

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

```

```

<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en-US"
  lang="en-US"
  metal:use-macro="here/main_template/macros/master"
  i18n:domain="plone">

  <metal:block fill-slot="base">
    <tal:block tal:define="dummy python:request.set('disable_border', 1)"
      tal:replace="nothing">
      Disable the tabs
    </tal:block>
  </metal:block>

  <!-- Change the following line -->

  <div metal:fill-slot="main"
    tal:define="results python:request.get('subject',
      here.portal_catalog.searchResults( Subject=['Biologie moléculaire', 'Embryologie', 'Bactériologie']
        , sort_on='Date'
        , sort_order='reverse'
        , review_state='published'));"
    tal:condition="python:test(results, 1, 0)">

    <div>
      <h1 tal:content="here/Title"> Le titre </h1>
      <br><br>
      <div>
        <table align="center" width="90%">
          <tr>
            <td>Type</td>
            <td>Titre</td>
          </tr>

          <tr tal:repeat="item results">

            <span tal:define="object item/getObject">

              <td>
                <img tal:attributes="src string:${object/getIcon}">
                <span tal:content="object/portal_type"> Type </span>
              </td>

              <td>
                <a href=" "
                  tal:attributes="href object/absolute_url">

                  <span tal:content="object/Title"> Titre </span>
                </a>
              </td>
            </span>
          </tr>
        </table>
      </div>
    </div>
  </div>
</html>

```

### Troisième objectif :

Le dernier objectif sera de créer une page d'accueil à l'intérieur de notre répertoire "Biologie", permettant au visiteur du site de choisir les contenus de sous-thème ou tous les contenus du thème qu'il veut consulter.

#### *Première façon de procéder :*

- Créer une page d'accueil à l'objet "Biologie" par le bouton situé sur le site web en étant loggué en manager (via <http://zopeserver:8080/monsieplone/>)



## Chapitre IX

# LE WORKFLOW DANS PLONE

### IX.1 Ce qu'est un workflow

Il faut imaginer un projet. Il y a alors des tâches à accomplir avec un certain enchaînement, des restrictions d'entrées (gardes)... Ces tâches seront réalisées par des personnes ayant différents rôles. Tout cela se schématise dans ce que l'on appelle un workflow.

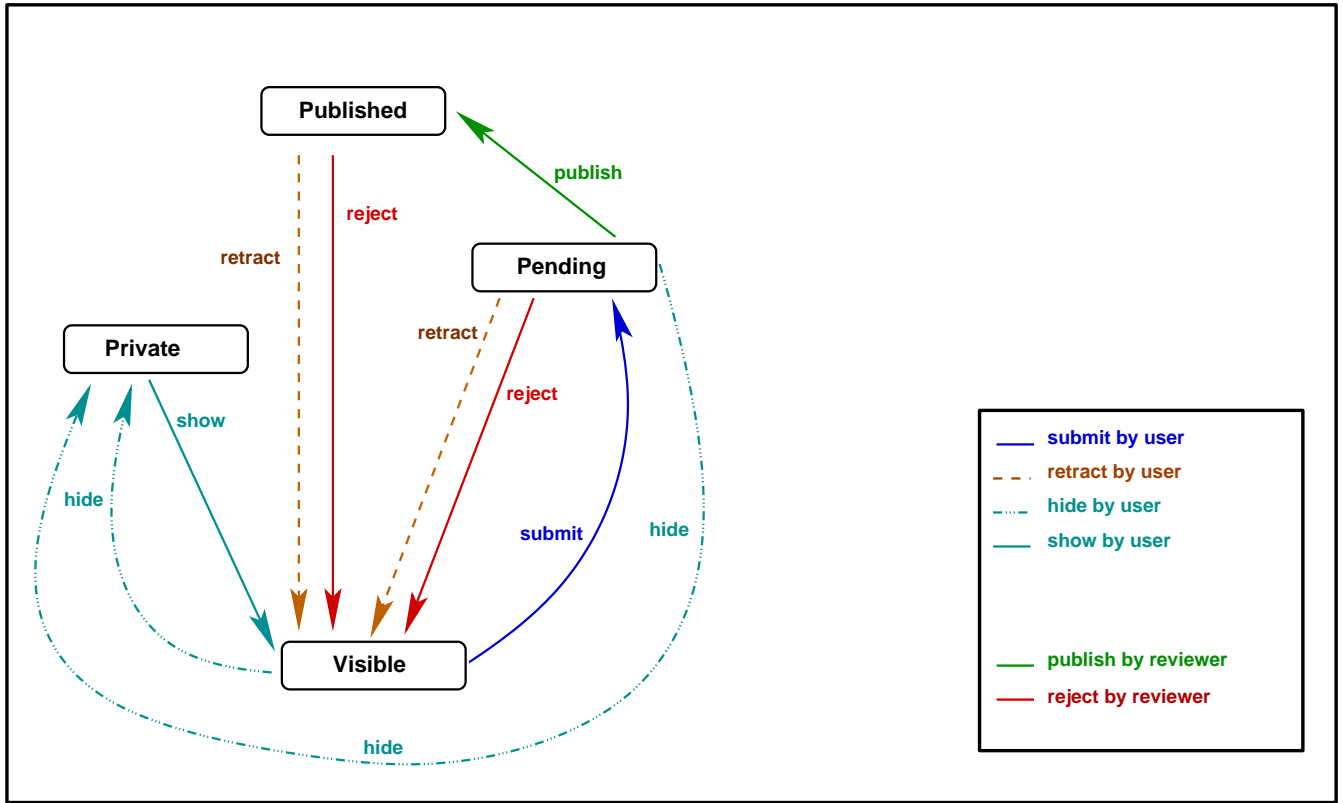
Le moteur de Workflow comme DCWorkflow va permettre d'aider à l'organisation en mettant en place le schéma du workflow (suivant un ordre chronologique, en fonction de tel rôle, telles tâches devront être réalisées). Le moteur de Workflow reçoit des informations de fin de réalisation de tâches des utilisateurs et prévient les autres utilisateurs qu'ils peuvent commencer une tâche découlant de la réalisation d'une ou de tâches précédemment achevées.

### IX.2 DCWorkflow, le workflow dans Plone

Pour trouver une très bonne documentation sur DCWorkflow, je vous invite à consulter celle réalisée par un membre de zope.org : hathawsh :

[http://www.zope.org/Members/hathawsh/DCWorkflow\\_docs](http://www.zope.org/Members/hathawsh/DCWorkflow_docs)

### IX.2.1 DCWorkflow, description



### IX.2.2 D'autres Workflow pour Plone

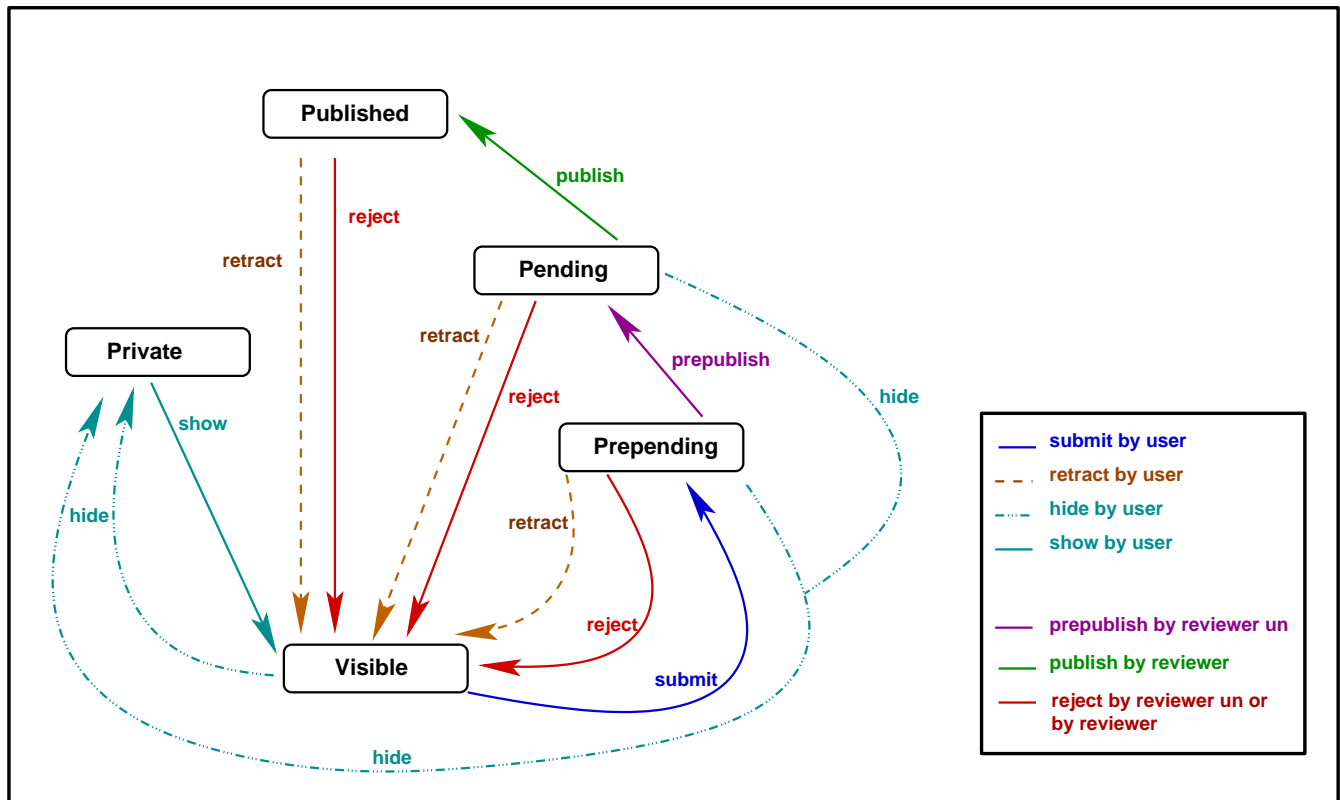
Il existe d'autres Workflow intégrables dans Zope, par exemple Reflow appelé aussi CMFOpenFlow.

## IX.3 Un exemple de personnalisation du Workflow dans Plone

### IX.3.1 Objectif visé

Dans cet exemple, on veut atteindre une politique de publication plus échelonnée qu'avec le Workflow par défaut de Plone. Il faudrait avoir une pré-lecture des contenus soumis à la publication, afin de respecter une hiérarchie peut-être pré-existante dans une entreprise.

### IX.3.2 Schéma du nouveau workflow



On note donc qu'il est nécessaire de :

1. Créer un nouveau rôle dans le site Plone
2. Créer un nouveau workflow
3. Créer un nouvel état appelé "prepending" dans ce nouveau workflow
4. Créer une transition appelée "prepublish" dans ce nouveau workflow
5. Créer un moyen d'indiquer aux utilisateurs ayant le nouveau rôle qu'ils ont des contenus à relire
6. Affecter ce nouveau workflow à des types de contenus

### IX.3.3 Un nouveau rôle dans le site Plone

Le nouveau rôle sera celui du pré-relecteur (reviewerun). Cet utilisateur fera donc un premier tri avant de laisser passer le document au relecteur final (reviewer).

Pour ce faire :

- Se rendre à la racine du site Plone considéré,
- puis sur l'onglet "Security",
- en bas de la liste des permissions se trouve le bouton "Add Role", entrer "Reviewerun" dans le champ et cliquer sur le bouton "Add Role"
- et donner à ce rôle les mêmes permissions que le rôle "Reviewer"

### IX.3.4 Créer et configurer ce nouveau workflow

- Se rendre à la racine du site Plone considéré

- Puis sur l'objet "portal\_workflow"
- Puis sur l'onglet "Contents"
- Cliquer sur le bouton "Add workflow"
- Entrer un nom à ce nouveau workflow "review\_workflow" par exemple
- Comme type de workflow, choisir "default\_workflow (Web-configurable workflow [Revision 2])" et cliquer sur Add.
- Entrer dans ce workflow pour le configurer.

### IX.3.5 Configurer ce nouveau workflow

#### Ajouter une permission pour les relecteurs

- Entrer dans ce workflow pour le configurer,
- Se rendre sur l'onglet "Permission"
- Ajouter une nouvelle permission "Review portal content" en cliquant sur le bouton "Add"

#### Ajouter l'état : "prepending"

- Se rendre à la racine du workflow intitulé "review\_workflow"
- Se rendre sur l'onglet "States"
- Ajouter un nouvel état appelé "prepending" en cliquant sur le bouton "Add a state"
- Cliquer sur ce nouvel état pour le configurer
- Ajouter comme transitions possibles : hide, prepublish, reject, retract
- Sauvegarder les changements

#### Ajouter la transition : "prepublish"

- Se rendre à la racine du workflow intitulé "review\_workflow"
- Se rendre sur l'onglet "Transition"
- Ajouter une nouvelle transition appelée "Prepublish" en cliquant sur le bouton "Add a transition"
- Cliquer sur cette nouvelle transition pour la configurer
- Entrer comme Titre : "Reviewerun prepublishes content"
- Dans le champs "Destination state" choisir "pending"
- Conserver le "Trigger type" : "initiated by user action"
- Pour les "Guard", entrer seulement une restriction par rôle et non par permission : Role(s) : Reviewerun
- Comme nom pour l'action (name formatted) entrer "Prepublish"
- Comme page après validation (url formatted) entrer la page de visualisation de l'objet "%(content\_url)s"
- La catégorie reste "workflow"

#### Modifier les autres transitions

La transition "submit" :

- Se rendre à la racine du workflow intitulé "review\_workflow"
- Se rendre sur l'onglet "Transition"
- Se rendre sur la transition "submit",
- Comme vu dans notre schéma, submit est une transition qui passe de l'état "Visible" à l'état "Pending". Ici, il faut modifier le champ "Destination state" pour avoir "prepending" comme état d'arrivée, puis sauvegarder les changements.

La transition "publish" :

- Se rendre à la racine du workflow intitulé "review\_workflow"
- Se rendre sur l'onglet "Transition"
- Se rendre sur la transition "publish"
- Conserver tous les paramètres exceptés ceux pour les "Guard", entrer seulement une restriction par rôle et non par permission : "Role(s) : Reviewer". (donc effacer dans le champ permission "Review portal content")

### IX.3.6 Prévenir les gens ayant le rôle de Reviewerun de contenus en attente de validation

Pour prévenir les utilisateurs ayant comme rôle "Reviewerun" de contenus en attente de validation de leur part. Il faut donc ajouter une liste de tâches à accomplir, visible seulement par les utilisateurs ayant le rôle de "Reviewerun", qui s'incrémentera automatiquement en fonction du nombre de contenus à revoir (= dans l'état "prepending").

- Se rendre à la racine du workflow intitulé "review\_workflow"
- Se rendre sur l'onglet "Worklists"
- Ajouter une liste de tâches à accomplir (add a worklist), la nommer "reviewerun\_queue"
- Remplir le champ "review\_state" par "prepending" (le nom de l'état que l'on veut voir listé)
- Pour le champ "Name (formatted)", entrer "Prepending (%(count)d)"
- Pour le champ "URL (formatted)", entrer "%(portal\_url)s/search ?review\_state=prepending"
- L'indication de catégorie (category) reste "global"
- Dans l'espace "Guard", le champ "Rôle(s)" est "Reviewerun"
- Sauvegarder les changements

Par la même occasion, changer dans l'espace Guard de la worklist intitulée "reviewer\_queue" le champ "Rôle(s)" en "Reviewer" (au cas où ce soit le champ "Permission" qui soit rempli avec "Review portal content").

### IX.3.7 Affecter ce nouveau workflow à des types de contenus

- Se rendre à la racine du site Plone
- Se rendre sur l'objet "portal\_workflow"
- Dans l'onglet "Workflows" considérer les types de contenus dont dispose l'objet "Plone site"

#### Changer le workflow pour un type de contenu (excepté folder qui a un workflow particulier)

- Modifier le champ qui porte le nom du type de contenu "Document" par exemple qui contient "(Default)" en "review\_workflow"
- Appliquer les changements via le bouton "Change"

#### Changer le workflow pour tous les types de contenus (excepté folder qui a un workflow particulier)

- Modifier le champ intitulé "(Default)" qui contient "plone\_workflow" en "review\_workflow"
- Appliquer les changements via le bouton "Change"

### IX.3.8 Vérification

Pour tester ce nouveau schéma de workflow on va supposer qu'il a été affecté au type de contenu "Document".

- Se rendre à la racine du site Plone
- Se rendre sur l'objet "acl\_users"
- Ajouter 3 utilisateurs : user1 qui va être membre, user2 qui va être reviewerun, user3 qui va être reviewer
- Etape 1 : user1 s'identifie et propose un document à la publication puis se déconnecte du portail Plone

- Etape 2 : user2 s'identifie et voit dans sa "personnal\_bar" (appelée dans le header) qu'un lien appelé "Prepending (1)" est apparu. Il clique sur ce lien et arrive sur une page donnant la liste des contenus dans l'état "prepending". Il voit le document de user1 et va pouvoir le rejeter (reject) ou le prépublier (prepublish). Il choisit de prépublier le document.
- Etape 3 : user3 s'identifie et dans sa "personnal\_bar" se trouve sa liste de tâches de relecture finale (contenus dans l'état "pending") il trouve le document de user1 prépublié par user2.

# Chapitre X

## GRAPHISME DU SITE

*Ayons tous une pensée pour les développeurs qui ont réussi la performance qu'est la réalisation de l'interface du site, même avec zoom, pour tous les navigateurs l'ensemble se tient, est homogène : bref un très bon travail.*

### X.1 Les images

Pour améliorer la navigation, pour une internationalisation, pour une harmonie visuelle on peut ajouter des images.

#### X.1.1 Modifier une image

Se rendre dans plone\_skins/plone\_images et choisir une image en cliquant dessus. On se retrouve devant une interface montrant :

Nom	<i>valeur (exemple)</i>	Explications
Id	folder.gif	nom de l'image
Content Type	image/gif	format
Dimensions	32 x 32	
Size	1,097 bytes	la taille
Last modified	2003/02/25 19 :16 :55 GMT+1	dernière modification
Source	file /lib/python/CMFPlone/skins/plone_images/folder.gif	

#### X.1.2 Insérer une nouvelle image

Comme on l'a dit précédemment, le dossier "Custom" est le premier à être lu dans portal\_skins.

C'est donc là où il sera simple d'ajouter une image et encore plus d'écrire son chemin pour son insertion dans une des pages du portail.

#### Pour ajouter une image :

1. Se rendre dans le répertoire portal\_skins/custom
2. Trouver Image dans la liste correspondant aux objets ajoutables
3. Il y a trois champs à remplir : Id (image.png), Title (titre de l'image, par exemple : image de Tux), File assorti du bouton parcourir (chemin de l'image source /home/tux/images/pour\_portail/pingouin.png)
4. Cliquer sur Add

### Inclure l'image dans une page :

Il existe de nombreuses manières d'inclure une image mais une de ces manières serait quelque soit la page du Plone Site (index\_html....) de l'inclure par le code suivant :

```

```

### X.1.3 Une organisation stratégique des images

Pour modifier fréquemment l'aspect du portail (saisonnier par exemple), on peut créer plusieurs css ou utiliser celles déjà existante, mais surtout on peut modifier les images de la navigation par exemple.

#### Restrictions

Le plus simple serait encore de savoir que l'image du logo, celles des icônes représentant tel ou tel type de document, et celles indiquant la navigation (petites flèches...) doivent rester à la racine du répertoire "custom".

*Explications* : on a vu que lorsque l'on clique sur le bouton "customize" pour une image ou un document on copie par défaut l'objet en question dans le répertoire "custom".

Ce chemin est important puisque c'est en le suivant que Plone va trouver l'objet à afficher. Si on laisse la destination "custom" pour le "logo.png" du Plone Site puis que dans ce répertoire "custom" on décide de créer un dossier "mesimages" et que l'on y déplace l'objet "logo.png", le logo personnalisé ne s'affichera pas dans le rendu du site, ce sera celui d'origine (Plone) qui s'affichera.

Il faut noter que le fait de changer la destination de ce type d'objet "logo" ou icon\_edit (etc...) ne fonctionne pas non plus.

#### Organisations des nouvelles images

Par contre l'ajout de nouvelles images (celles utilisées pour la navigation par exemple) peut très bien se faire dans un dossier créé dans portal\_skins/custom.

Dans l'exemple ce dossier est intitulé "imgsna".

On verra donc dans "
<HEAD>
<style>
```

```

p {font :13px Verdana, Helvetica} /* sera utilisé par toutes les balises de type<p> */
a {color :blue} /* sera utilisé par toutes les balises de type <a> */
.diff {color :purple} /* est une classe et doit être précisé */
</style>

</HEAD>

<BODY>

<p>Découvrez les fonctionnalités de PStricks suivant votre niveau : (rappelez-vous la nomenclature : bleu si vous débutez, violet si vous avez déjà une première expérience) :</p>
<a href="niveau1.html">Lien standard en bleu</a><br>

<a href="niveau2.html" class="diff">Lien violet</a>
</BODY>

</HTML>

```

Interne signifie que la feuille de style est écrite dans la page et doit être répétée dans chaque page.

### Externes à la page

Ce sont évidemment les plus intéressantes, les internes servaient ici à montrer la notion de "class" très utilisée dans Plone.

Dans une page html normale elles sont appelées ainsi entre les balises <head></head> de la page :

```

<style type="text/css" media="screen,projection"><!--
@import url("presentation.css");
--></style>

```

Le fichier presentation.css contient le code suivant :

```

TEXTAREA {
color:#333333;
background-color:#ccffcc;
font-size:12px;
font-family:arial;
}
A:LINK {
color:#990033;
font-size:12px;
text-decoration:underline;
font-weight:bold;
}
A:HOVER {
color:#993366;
font-size:12px;
text-decoration:underline;
font-weight:bold;
}

```

```

}
.diff {
color:#990000;
text-decoration:none;
font-weight:normal;
}

```

Cette partie d'exemple de feuille de style montre que l'on peut faire beaucoup (positionnement des menus, des div...) avec une feuille de style (voir le site de Eric A. Meyer pour cela

<http://www.meyerweb.com/eric/>

Dans Plone, les css sont appelées ainsi que le montrent ces exemples (dans le *header*) :

```

<!-- Main style sheets for CSS2 capable browsers -->
<style type="text/css" media="all"
tal:content="string: @import url(${here/portal_url}/plone.css?${current_skin});">
</style>

<!-- Custom style sheet if available -->
<style type="text/css" media="all" tal:condition="exists: here/ploneCustom.css"
tal:content="string:@import url(${here/portal_url}/ploneCustom.css);">
</style>

```

## X.2.2 Modifier les css de Plone

Dans Plone comme dans Zope, le fond est séparé de la forme. Cela s'applique aussi à la section des css dans Plone. On peut choisir entre plusieurs styles de présentation (donc plusieurs css) déjà réalisés.

Ce choix entraîne l'utilisation de la "*current\_skin*", c'est à dire la présentation choisie. (dans Plone setup / Configuration).

Ainsi il existe un modèle de css appelant les valeurs pour chaque style. Ce modèle est dans *portal\_skins/plone\_styles/plone.css*.

Petit morceau choisi de ce fichier :

```

h1, h2, h3, h4, h5, h6 {
color: &dtml-headingFontColor;;
clear: left;
font: &dtml-headingFont;;
margin: 0;
padding-top: 0.5em;
border-bottom: 1px solid &dtml-mainTabBorderColor;;
}

```

On voit donc un clair appel à des variables, variables remplies et personnalisables (par la méthode "customize" toujours) dans le fichier :

*portal\_skins/plone\_styles/stylesheet\_properties* .

On peut remplir les champs correspondant à `&dtml-headingFont` , `&dtml-headingFontColor`...

Il faut noter que ce fichier correspond au style Plone Default, il existe un tel fichier "styleshett\_properties dans chaque répertoire de skin (dans `portal_skins/plone_styles`) par exemple dans le répertoire de la skin "Mozilla", dans celui la skin "Kitty", dans celui de l'apparence "Autumn" ...

## Annexe A

# GNU Free Documentation License

GNU Free Documentation License  
Version 1.2, November 2002

Copyright (C) 2000,2001,2002 Free Software Foundation, Inc.  
59 Temple Place, Suite 330, Boston, MA 02111-1307 USA  
Everyone is permitted to copy and distribute verbatim copies  
of this license document, but changing it is not allowed.

### 0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondly, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

## 1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input

to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

A section "Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "Acknowledgements", "Dedications", "Endorsements", or "History".) To "Preserve the Title" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

## 2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use

technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

### 3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

## 4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

- A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.\\
- B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.\\
- C. State on the Title page the name of the publisher of the Modified Version, as the publisher.\\
- D. Preserve all the copyright notices of the Document.\\
- E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.\\
- F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.\\
- G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.
- H. Include an unaltered copy of this License.\\
- I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.\\
- J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.\\
- K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements

- and/or dedications given therein.\\
- L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.\\
  - M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.\\
  - N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.\\
  - O. Preserve any Warranty Disclaimers.\\

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties--for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

## 5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements".

## 6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

## 7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole

aggregate.

## 8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

## 9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

## 10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See <http://www.gnu.org/copyleft/>.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or

of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

```
Copyright (c) YEAR YOUR NAME.  
Permission is granted to copy, distribute and/or modify this document  
under the terms of the GNU Free Documentation License, Version 1.2  
or any later version published by the Free Software Foundation;  
with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts.  
A copy of the license is included in the section entitled "GNU  
Free Documentation License".
```

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the "with...Texts." line with this:

```
with the Invariant Sections being LIST THEIR TITLES, with the  
Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.
```

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.