



freedom-toolbox 3.0

Manuel de paramétrage et de développement



Table des matières

1	Introduction.....	9
1.1	Présentation.....	9
1.2	Informations complémentaires.....	9
1.2.1	Comité de rédaction.....	9
1.2.2	Anakeen.....	9
1.2.3	La communauté freedom.....	9
2	Paramétrage.....	10
2.1	Famille.....	10
2.1.1	Création d'une famille.....	10
2.1.1.1	Créer une nouvelle famille depuis l'interface Web.....	10
2.1.1.2	Présentation des champs de base nécessaires à la création d'une famille.....	10
2.1.1.3	Exemple d'une famille simple.....	11
2.1.1.4	Fonctionnement de l'héritage entre les familles.....	11
2.1.1.5	Créer une famille sous OpenOffice.org et l'importer dans Freedom.....	11
2.1.2	Importation de documents famille.....	11
2.1.2.1	But de ce chapitre.....	11
2.1.2.2	Importer des familles en utilisant l'interface Web.....	11
2.1.2.3	Importer des familles en utilisant la ligne de commandes.....	12
2.1.2.4	Supprimer une famille.....	12
2.1.2.5	Description du fichier d'importation.....	12
2.1.3	Attributs.....	14
2.1.3.1	Les attributs d'une famille.....	14
2.1.3.2	Options des attributs.....	18
2.1.3.3	Types Spéciaux.....	21
2.1.3.4	Valeurs par défaut.....	21
2.1.3.5	Les attributs énumérés.....	22
2.1.3.6	Créer un lien hypertexte.....	24
2.1.3.7	Création d'une relation entre documents.....	25
2.1.3.8	Ajouter une aide à la saisie.....	26
2.1.4	Paramètres.....	31
2.1.5	Modification d'une famille.....	31
2.1.5.1	Modifier un attribut père.....	31
2.1.5.2	Supprimer un attribut.....	32
2.1.5.3	Supprimer un attribut père.....	32
2.1.6	Modifier une famille depuis l'interface Web.....	32
2.1.6.1	Ajouter un attribut.....	32
2.1.6.2	Paramétrage d'une famille depuis l'interface.....	32
2.1.6.3	Exemples de familles à importer dans Freedom.....	32
2.2	Gestion des utilisateurs.....	33
2.2.1	Présentation.....	33
2.2.2	Gestion des groupes.....	34
2.2.2.1	Création de groupes.....	34
2.2.2.2	Modification d'un groupe.....	36
2.2.2.3	2.3. Consultation du groupe.....	37
2.2.2.4	2.4. Suppression de groupe.....	38
2.2.3	3. Gestion des utilisateurs.....	38
2.2.3.1	3.1. Création d'utilisateurs.....	38
2.2.3.2	3.2. Modification d'utilisateurs.....	39
2.2.3.3	3.3. Consultation d'utilisateurs.....	40
2.2.3.4	3.4. Suppression d'utilisateur.....	40
2.2.4	4. Administration.....	40
2.2.4.1	4.1. Mise à jour.....	41
2.2.4.2	Importation.....	41
2.2.4.3	Catégories.....	45
2.2.4.4	Configuration LDAP.....	46

2.2.5 Connexions anonymes.....	51
2.3 Sécurité.....	51
2.3.1 Les droits applicatifs.....	51
2.3.1.1 FDL - Bibliothèque FREEDOM.....	51
2.3.1.2 GENERIC - Manipulation d'une famille.....	52
2.3.1.3 ONEFAM - Manipulation groupement de familles.....	52
2.3.1.4 FREEDOM - Gestion documentaire.....	52
2.3.2 Profils de documents.....	52
2.3.2.1 Définition des droits.....	52
2.3.2.2 Modification.....	54
2.3.2.3 Affecter manuellement un profil à un document.....	55
2.3.2.4 Profil par défaut des nouveaux documents.....	55
2.3.2.5 Profil par défaut lors de l'enregistrement d'un document dans un dossier.....	55
2.3.2.6 Profil dynamique.....	55
2.3.2.7 Profil de transitions de cycle de vie.....	56
2.3.2.8 Profils d'état de cycle de vie.....	56
2.3.2.9 Administration des profils.....	57
2.4 Importation et exportation de documents.....	58
2.4.1 Exportation.....	58
2.4.1.1 Exportation CSV.....	58
2.4.1.2 Exportation XML.....	59
2.4.2 Importation.....	61
2.4.2.1 Importation CSV/ODS.....	61
2.4.2.2 Importation XML.....	69
2.4.2.3 Importation en ligne de commande.....	70
2.5 Archives.....	71
2.5.1 Introduction.....	71
2.5.2 Constitution de l'archive.....	71
2.5.3 Archivage des documents.....	73
2.5.4 Destruction des documents archivés.....	75
2.6 Fonctions Wsh.....	76
2.6.1 Comment exécuter une fonction WSH.....	76
2.6.2 listapi.....	77
2.6.3 fdl_adoc.....	77
2.6.4 freedom_clean.....	77
2.6.5 freedom_convert.....	77
2.6.6 freedom_groups.....	77
2.6.7 freedom_import.....	78
2.6.8 freedom_refresh.....	78
2.6.9 freedom_refresh avec une méthode en argument.....	78
2.6.10 ods2csv.....	79
3 Développement.....	80
3.1 Utilisation de l'API.....	80
3.1.1 Présentation de l'utilisation de l'API.....	80
3.1.2 Manipulation de documents (GetParam, new Doc et new_Doc).....	80
3.1.3 Consultation de documents (getValue, getTValue et getTDoc).....	82
3.1.3.1 Les propriétés remarquables.....	83
3.1.3.2 Accès aux propriétés (getValue).....	84
3.1.3.3 Récupérer les différentes valeurs d'un attribut multiple (getTValue).....	84
3.1.3.4 Récupérer un array complet ou une ligne de l'array (getAValues).....	85
3.1.3.5 Accès à l'ensemble des attributs (getValues).....	85
3.1.3.6 Accès aux valeurs des documents (getTdoc).....	86
3.1.4 Modification de documents (setValue, getValues et deleteValue).....	86
3.1.5 Création de documents (createDoc).....	87
3.1.6 Placement de documents dans un dossier (addFile et delFile).....	88

3.2 Recherche de documents.....	90
3.2.1 Retour de valeurs d'attributs.....	90
3.2.2 Retour d'objets documentaires.....	90
3.2.3 Debug.....	91
3.2.4 Recherche dans un dossier et recherche récursive.....	92
3.2.5 Recherche par critère sur les valeurs d'attributs.....	92
3.2.5.1 Recherche sur le titre des documents (title).....	92
3.2.5.2 Recherche sur n'importe quel attribut (values).....	93
3.2.5.3 Recherche dans une famille.....	93
3.2.5.4 Recherche avec l'opérateur 'or'.....	94
3.2.5.5 Recherche sur des attributs de type énumérés (getKindDoc).....	94
3.2.5.6 Recherche dans un array.....	94
3.2.6 Recherche spécialisée.....	95
3.3 Aides à la saisie.....	96
3.3.1 Principe et spécification.....	96
3.3.2 Retour unique.....	96
3.3.3 Retour multiple.....	98
3.3.4 Cas des attributs relation (docid).....	99
3.4 Attributs calculés.....	99
3.4.1 Principe et spécification.....	99
3.4.2 Calcul simple.....	99
3.4.3 Calcul multiple.....	100
3.4.4 Afficher un message d'avertissement en cas d'erreur (addWarningMsg).....	101
3.5 Contraintes.....	101
3.5.1 La fonction de contrôle.....	101
3.5.2 Contrainte système.....	102
3.6 Surcharges des méthodes du document.....	102
3.6.1 Contrôle à la création.....	102
3.6.2 Contrôle à la modification.....	103
3.6.3 Contrôle à la suppression.....	104
3.6.4 Autres méthodes.....	104
3.6.5 Ordre d'appel des méthodes.....	105
3.6.5.1 Consultation de document.....	105
3.6.5.2 Édition de document.....	105
3.6.5.3 Création de document.....	105
3.6.5.4 Sauvegarde de document.....	105
3.6.5.5 Duplication de document.....	105
3.6.5.6 Suppression de document.....	105
3.6.5.7 Import de document.....	105
3.7 Principales méthodes de la classe Doc.....	106
3.7.1 lock.....	106
3.7.2 unlock.....	106
3.7.3 addRevision.....	106
3.7.4 attachTimer.....	106
3.8 Vues et éditions particulières.....	107
3.8.1 Vues par défaut.....	107
3.8.2 Syntaxe d'un fichier Layout.....	108
3.8.2.1 Données atomiques.....	108
3.8.2.2 Données listes.....	109
3.8.2.3 Données tableau.....	110
3.8.2.4 Conditions.....	111
3.8.2.5 Internationalisation.....	112
3.8.2.6 Paramètres d'environnement.....	113
3.8.2.7 Les zones.....	113

3.8.3	Vue de consultation par défaut.....	114
3.8.3.1	Consultation avec la méthode par défaut.....	115
3.8.3.2	Consultation avec méthode spécifique.....	116
3.8.4	Vues de consultation spécifiques.....	117
3.8.5	Vues de consultation OpenDocument Text.....	118
3.8.6	Vues de consultation avec transformation.....	121
3.8.7	Vues d'édition.....	123
3.8.8	Interface spécifique de contrôle d'un formulaire document.....	127
3.8.8.1	Manipulation de tableau.....	131
3.8.8.2	Affectation d'attribut.....	132
3.8.8.3	Récupération de valeur d'attribut.....	132
3.8.8.4	Utilisation des attributs de retour définis dans l'aide à la saisie.....	132
3.8.9	Mise en forme de rangée de tableau.....	133
3.8.9.1	Exemple de consultation.....	133
3.8.9.2	Exemple d'édition.....	135
3.8.9.3	Pour éditer et consulter avec le même template.....	137
3.8.10	Vues d'attributs.....	137
3.8.10.1	Vue d'attributs de consultation.....	137
3.8.10.2	Vue d'attributs d'édition.....	139
3.9	Cycle de vie.....	141
3.9.1	Développement.....	141
3.9.1.1	Définition d'un cycle de vie.....	141
3.9.1.2	Construction d'un cycle de vie.....	142
3.9.1.3	Paramètres de transition.....	148
3.9.1.4	Options de transitions.....	151
3.9.2	Paramétrage.....	153
3.9.2.1	Créer un Cycle de vie (Workflow).....	153
3.9.2.2	Créer un cycle de vie localisé.....	165
3.9.2.3	Modèle de courriel pour les cycles.....	169
3.9.2.4	Ajout de courriel pour les cycles.....	172
3.9.2.5	Minuteurs.....	172
3.9.2.6	Les accords.....	177
3.9.2.7	Cycle de vie de publication de documents.....	179
3.10	Créer une nouvelle application Freedom.....	182
3.10.1	But de ce document.....	182
3.10.2	Arborescence des fichiers d'une application.....	182
3.10.3	MONAPPLI.app.....	183
3.10.4	MONAPPLI_init.php.....	184
3.10.5	MONAPPLI_post.....	184
3.10.6	Initialisation ou mise à jour de l'application.....	185
3.10.7	Liens pour avoir d'autres informations.....	186
3.10.8	Actions particulières.....	186
3.10.8.1	Principe et spécification des actions particulières.....	186
3.10.8.2	Introduction.....	186
3.10.8.3	Fichier 'MONAPP.app'.....	187
3.10.8.4	Fichier 'MONAPP_init.php'.....	188
3.10.8.5	Fichier 'monaction1.php'.....	188
3.10.8.6	Fichier monaction1.xml.....	189
3.10.8.7	Gestion des droits sur les actions.....	190
3.10.8.8	Initialisation de l'application et des actions.....	190
3.10.8.9	Action dans les menus contextuels.....	190
3.11	Localisation (Traduction de freedom dans d'autres langues).....	193
3.11.1	But de ce chapitre.....	193
3.11.2	Quel mécanisme utilise freedom pour la localisation ?.....	193
3.11.3	Localiser son application.....	193
3.11.3.1	Localisation des familles.....	193
3.11.3.2	Localisation des fichiers .php (Methode, Class, Action, Api,..)	193
3.11.3.3	Localisation des fichiers .xml (Layout et Action).....	193
3.11.3.4	Localisation des fichiers .app (Application).....	194
3.11.3.5	Localisation des cycles de vie.....	194
3.11.3.6	Localisation des attributs des familles.....	194

3.11.3.7	Recherche des chaînes à traduire (Génération fichier .po).....	195
3.11.3.8	Comment ça marche.....	195
3.11.4	Traduire un module Freedom dans une autre langue.....	196
3.11.5	Outils pour lire les fichiers .po et traduire les chaînes.....	198
3.11.6	Génération des fichiers .mo (binaires).....	199
3.11.7	Initialisation de freedom et d'Apache pour prendre en compte les localisations.....	199
3.11.8	Script pour automatiser les tâches.....	199
3.12 Bases de données PostgreSQL.....	200	
3.12.1	Tables de la base anakeen.....	200
3.12.1.1	acl.....	200
3.12.1.2	action.....	200
3.12.1.3	application.....	201
3.12.1.4	docfrom.....	201
3.12.1.5	docname.....	201
3.12.1.6	domain.....	201
3.12.1.7	groups.....	201
3.12.1.8	mailaccount.....	201
3.12.1.9	paramdef.....	201
3.12.1.10	paramv.....	202
3.12.1.11	permission.....	202
3.12.1.12	session_conf.....	202
3.12.1.13	sessions.....	202
3.12.1.14	style.....	202
3.12.1.15	users.....	202
3.12.2	Tables de la base freedom.....	203
3.12.2.1	Héritage entre les tables.....	203
3.12.2.2	doc.....	204
3.12.2.3	doc<idfam>.....	204
3.12.2.4	docattr.....	204
3.12.2.5	docattrldap.....	204
3.12.2.6	docfam.....	204
3.12.2.7	docfrom.....	204
3.12.2.8	dochisto.....	204
3.12.2.9	docname.....	204
3.12.2.10	docperm.....	205
3.12.2.11	docread.....	205
3.12.2.12	docrel.....	205
3.12.2.13	docutag.....	205
3.12.2.14	docvaultindex.....	206
3.12.2.15	fld.....	206
3.12.2.16	groups.....	206
3.12.2.17	pg_ts_cfg, pg_ts_cfgmap, pg_ts_dict et pg_ts_parser.....	206
3.12.2.18	vgroup.....	206
3.12.3	Tables du vault (base freedom).....	206
3.12.3.1	vaultdiskdirstorage.....	206
3.12.3.2	vaultdiskfsstorage.....	206
3.12.3.3	vaultdiskstorage.....	207
4 Exploitation.....	208	
4.1 dbaccess.php.....	208	
4.1.1	Fichier "dbaccess.php".....	208
4.1.1.1	freedom_authtype.....	208
4.1.1.2	freedom_authprovider.....	209
4.2 Développer un provider.....	211	
4.2.1	validateCredential().....	211
4.2.2	validateAuthorization().....	211
4.2.3	__construct().....	212
4.2.4	initializeUser().....	212
4.2.5	Exemple.....	212
4.2.5.1	Fichier `context/default/dbaccess.php'.....	212
4.2.5.2	Fichier `WHAT/providers/Class.pamProvider.php'.....	213
4.2.5.3	Fichier `/etc/pam.d/freedom'.....	214

4.3 Debuggage.....	214
4.3.1 Affichage des avertissements PHP.....	214
4.3.2 Affichage des requêtes PostgreSQL.....	215
4.3.3 freedom en mode debug.....	215
4.3.4 Tracer le temps d'exécution des actions dans Freedom.....	215
4.3.5 Firebug.....	216
4.4 Diagnostic des bases de données.....	217
4.4.1 main connection db.....	217
4.4.2 unreference user in group.....	217
4.4.3 user as group.....	217
4.4.4 unreference actions.....	218
4.4.5 unreference parameters.....	218
4.4.6 unreference acl.....	218
4.4.7 unreference permission.....	218
4.4.8 connection db freedom.....	218
4.4.9 double doc id.....	218
4.4.10 double doc name.....	219
4.4.11 family inheritance.....	219
4.4.12 user group synchro.....	220
4.4.13 connection db webdav.....	220
4.4.14 connection USER LDAP.....	220
4.5 Exécution périodique.....	220
4.6 Paramétrage de Freedom.....	221
4.6.1 Les modules de Freedom.....	221
4.6.2 Accès au paramétrage des modules de Freedom.....	222
4.6.3 ACCESS - Accessibilités.....	222
4.6.4 APPMNG - Gestion des applications.....	222
4.6.5 AUTHENT - Authent.....	223
4.6.6 CORE - Noyau.....	223
4.6.7 FDL - Bibliothèque FREEDOM.....	224
4.6.8 GENERIC - Manipulation d'une famille.....	225
4.6.9 FREEDOM - Gestion documentaire.....	225
4.7 Quelques commande PostgreSQL pratiques.....	225
4.7.1 Voir la taille des tables/index.....	225
4.7.2 Autres ressources.....	226
4.8 Sauvegarde et restauration des bases de données.....	226
4.9 VAULT - Coffre de stockage des fichiers.....	226
4.9.1 Interface.....	226
4.9.2 Création d'un coffre.....	226
4.9.3 Modification du volume d'un coffre.....	227
4.9.4 Déplacement d'un coffre.....	227
4.9.5 Commande shell freedom.....	227
4.10 Type MIME.....	227
4.10.1 Fichier d'association local `\$CONTEXT_ROOT/admin/mime-user.conf`.....	227
4.10.2 Fichier d'association global `\$CONTEXT_ROOT/admin/mime.conf`.....	228

1 Introduction

1.1 Présentation

freedom-toolbox permet de construire des applications métiers pour des structures confrontés aux problématiques de [gestion de contenu \(l'ECM\)](#) et du Case Management.

Ce document décrit les mécanismes proposés et leur mise en oeuvre. Il est articulé en deux parties. La première présente le paramétrage qui permet de préciser et modifier les comportements du produit sans avoir recours au scripting ou codage. La seconde apporte des mécanismes plus complets nécessitant eux des bases dans le langage PHP. En préambule à l'utilisation de ce document, nous précisons qu'il est important de maîtriser les concepts freedom.

1.2 Informations complémentaires

Le wiki accessible à l'adresse <http://www.freedom-eclm.org> offre l'ensemble de la documentation freedom-toolbox et des modules associées.

1.2.1 Comité de rédaction

Ce manuel est le fruit du travail de l'équipe de développement Anakeen amenée par EBR, assisté de MCL et dirigée par YLB.

1.2.2 Anakeen

Anakeen est éditeur du produit freedom-toolbox. Société couvrant l'ensemble des prestations habituelles d'un éditeur, Anakeen a décidé depuis sa création de proposer des produits sous licence libre.

1.2.3 La communauté freedom

Elle existe au travers des outils d'échange accessibles via [le site communautaire freedom](#). Cette communauté permet à chaque jour de corriger, préciser ce manuel par ses remarques pertinentes.

2 Paramétrage

Cette première partie décrit le paramétrage que l'on peut effectuer sur freedom sans utilisation de codage PHP.

2.1 Famille

La famille sert à décrire en ensemble de document ayant les mêmes caractéristiques que ce soit en terme de contenu ou en terme de comportement. Les familles peuvent être diverses comme par exemple les compte-rendus, les notes techniques, les factures, les sociétés, les recettes de cuisines.

2.1.1 Crédit d'une famille

2.1.1.1 Créer une nouvelle famille depuis l'interface Web

Pour créer une nouvelle famille, il faut :

- Aller dans le module “Gestion documentaire”
- Sélectionner le menu “Création / Famille”
- Indiquer de quel famille la nouvelle famille doit hériter
- Indiquer le nom de la famille
- Cliquer sur “Créer une nouvelle famille”
- Faire un clic droit sur la nouvelle famille et sélectionner le menu “Éditer les attributs”

La fenêtre d'édition présentent les attributs de la famille et les attributs hérités qui sont non modifiables.

En bas du tableau, trois nouveaux attributs peuvent aussi être ajoutés. Si vous voulez en ajouter plus de trois il faut le faire trois par trois. Une fois les trois premiers saisis, il faut valider et faire :

- Clic droit sur la famille
- Menu “Éditer les attributs”

2.1.1.2 Présentation des champs de base nécessaires à la création d'une famille

Champ	Description
Identificateur ou ID	Nom interne de l'attribut. Il sert de référence lors des définitions des fonctions et des liens. Cet item n'est plus modifiable une fois créée. Cet identificateur doit être unique dans l'ensemble des familles.
Ordre	Défini l'ordre de présentation des attributs dans le document. Cet ordre n'est pas utile pour les attributs de type <i>frame</i> .
Nom	Texte court définissant l'attribut. Ce texte est utilisé pour désigner l'attribut lors de la visualisation ou de l'édition.
Type	Défini le type syntaxique de l'attribut (text, htmltext, frame, date, time,...)
Cadre ou fenêtre	Défini le cadre dans lequel l'attribut doit être présenté. Il faut auparavant avoir défini les cadres.
Résumé ou R?	Si vous voulez que l'attribut soit affiché dans le mode colonne ou dans les aperçus de documents dans l'application GENERIC, il faut cocher la colonne 'R?'.
Titre ou T?	Le titre d'un document peut être composé par les valeurs d'un ou de plusieurs attributs. Il faut au moins mettre un des attributs en titre en cochant la colonne "T?".
Obligatoire ou O?	Indique si l'attribut est obligatoire lors de la saisie.
Visibilité ou Vi	Indique dans quel cas l'attribut est affiché.

2.1.1.3 Exemple d'une famille simple

Cet exemple est celui de la famille “de base” :

// famille de base	titre	id	class	name						
BEGIN	0 de base			BASE						
TYPE	C									
ATTR	idattr	idframe	label	T	A	type	ord	vis	need	link
ATTR	FR_BASIC		basique	N	N	frame		0 W		
ATTR	BA_TITLE	FR_BASIC	titre	Y	N	text		1 O	Y	
END										

Commentaires :

- La première ligne permet de définir le cadre (bordure qui entoure les champs)
- La deuxième ligne permet de définir un champ de type “text” qui sera affiché dans la vue résumé et utilisé dans le titre du document. Ce champ est obligatoire

Remarque : Pour avoir d'autres exemples, le plus simple est de regarder le paramétrage des familles disponibles en standard

2.1.1.4 Fonctionnement de l'héritage entre les familles

Les familles sont soumises à une hiérarchie. On peut ainsi définir des familles spécialisées à partir de définition plus générale, on parle alors d'héritage.

Par exemple, Nous avons la famille “société” dans laquelle figure toutes les coordonnées d'une entreprise et on souhaite définir une famille “site” qui permettra de visualiser les différents sites de cette entreprise. Nous allons donc créer la famille “site” à partir de la famille “société”. On dit alors que la famille “site” hérite de la famille “société” et de tous ses attributs. La famille héritée pourra alors être modifiée pour caractériser précisément les sites de l'entreprise.

2.1.1.5 Créer une famille sous OpenOffice.org et l'importer dans Freedom

L'interface Web sert juste à ajuster les attributs d'une famille. Pour accéder à toutes les possibilités d'une famille, il faut la créer dans OpenOffice.org et l'importer dans Freedom. Tout est expliqué dans le chapitre suivant.

2.1.2 Importation de documents famille

2.1.2.1 But de ce chapitre

- Expliquer comment importer des familles dans Freedom en utilisant l'interface Web ou la ligne de commandes.
- Expliquer comment construire un fichier OpenOffice.org ou .csv permettant de définir les attributs d'une famille à importer.

2.1.2.2 Importer des familles en utilisant l'interface Web

Avant d'importer une famille, il est conseillé de suivre le chapitre concernant l'importation de documents en général.

Mais en résumé, voici ce qu'il faut faire :

- Construire un fichier OpenOffice.org ou .cvs en suivant les instructions des chapitres suivants
- Se placer dans le module “Gestion documentaire” de Freedom
- Utiliser le menu “Outils / Importer des documents”

- Cliquer sur le bouton “Parcourir” et sélectionner le document .csv ou OpenOffice.org
- Cliquer sur le bouton “Analyse” pour détecter les anomalies
- Cliquer sur le bouton “Importer les document”

Le bouton "Importer les documents" est grisé (inactif) lors de l'ouverture de la page d'import. Il se dégrise si l'analyse est correcte et si le nombre de document est inférieur au paramètre applicatif " FDL_MAX_FGEXPORTDOC" (par défaut 20 - bibliothèque freedom). Si le nombre de famille est supérieur seul l'import en tâche de fond sera activable.

2.1.2.3 Importer des familles en utilisant la ligne de commandes

Voici la commande qu'il faut utiliser pour importer un document dans Freedom :

```
$ sudo /var/www/wiff/wiff context Développement exec /bin/bash --login
[sudo] password for eric
www-data@tarfful:~$ ./wsh.php --api=freedom_import --file=/chemin/VotreFichier.ods
```

Remarque : Vous pouvez importer directement un fichier OpenOffice.org .ods ou un fichier .csv

Si vous êtes en phase de mise au point vous pouvez utiliser l'option reinitattr pour supprimer les attributs des familles déclarées dans le fichier avant l'importation

```
./wsh.php --api=freedom_import --reinitattr=yes --file=VotreFichier.ods
```

2.1.2.4 Supprimer une famille

Une fois l'importation d'une famille effectuée, il n'est plus possible de supprimer des attributs ni de les renommer. Vous serez donc certainement amené pendant votre phase de mise au point à supprimer des familles.

ATTENTION : La suppression d'une famille entraîne la suppression de tous les documents de cette famille.

Pour supprimer une famille, Il faut utiliser une commande wsh

```
$ sudo /var/www/wiff/wiff context Développement exec /bin/bash --login
[sudo] password for eric
www-data@tarfful:~$ ./wsh.php --api=fdl_deletefamily --famid=ZOO_TEST
Destroying [Test :(ZOO_TEST)]
begin;
delete from doc1182;
drop view family."zoo_test";
delete from docname where name='ZOO_TEST'
delete from docfrom where fromid=1182
drop table doc1182;
delete from docattr where docid=1182;
delete from docfam where id=1182;
commit;
Family ZOO_TEST (id : 1182) is destroyed.
```

2.1.2.5 Description du fichier d'importation

Dans les chapitres suivants, vous trouverez la description complète d'un fichier d'importation. Ce fichier peut-être au format OpenOffice.org ou au format .csv.

Remarque : Une ligne vide ou toute ligne qui commence par un mot-clé non reconnu sera ignorée.

Lors de l'utilisation de openOffice.org pour éditer les structures de familles, il est conseillé d'inhiber toutes les options pour l'autocorrection et les guillemets typographiques.

Entête

L'entête définit le titre, le nom et l'héritage de la famille.

//	fromid	titre	id	class	name
BEGIN	SOCIETY	site			SITE

Définition des colonnes :

1. BEGIN
2. identifiant numérique ou logique de la famille père. Mettre vide s'il n'y a pas d'héritage.
3. titre de la famille.
4. identifiant numérique de la famille. Le laisser à vide pour utiliser un identifiant logique. S'il est valué il faut que cet identifiant ne soit pas déjà pris par un autre document. Les valeurs entre 900 et 999 peuvent être utilisées pour vos besoins spécifiques.
5. le nom d'un fichier de définition de classe PHP qui doit être présent sur le serveur (<installdir>/FDL). Ne sert que pour des documents systèmes. Permet un héritage autre que celui prévu par défaut par les classes documentaire.
6. nom logique de la famille. Ne doit contenir que des caractères alphanumériques les caractères _ et : sont autorisés. (pas d'espace, ni de ponctuation).

Caractéristiques générales

Les caractéristiques permettent d'affecter les propriétés du documents. La syntaxe est toujours la même : 1^{ère} colonne : nom de la propriété, 2^{ème} colonne valeur de la propriété.

propriété	description
TYPE	toujours mettre C. Défini la propriété doctype du document. 'C' signifie famille de document.
ICON	nom du fichier image définissant l'icône de la famille. Cette icône doit être une image de taille 48x48 pixels. Elle doit être présente sur le serveur. Cette icône peut être modifiée à postérieur par l'interface. Si une icône est déjà présente pour cette famille, elle ne sera pas modifiée.
PROFID	identifiant du document profil pour cette famille. Ce document doit être de la famille 'profil de famille'.
CPROFID	identifiant de profil qui sera affecté à tout nouveau document créé par cette famille. Ce document doit être de la famille profil de document. Si la famille hérite de dossier, le document doit être de la famille profil de dossier. Si la famille hérite de recherche, le document doit être de la famille profil de recherche.
DFLDID	identifiant du dossier par défaut dans lequel tout nouveau document de la famille sera créé. Peut être égal au 'auto', ce qui a pour effet de créer un dossier par défaut automatiquement. Si cette propriété est déjà renseignée, la nouvelle valeur ne sera pas prise en compte.
METHOD	Indique le nom du fichier contenant les méthodes supplémentaires de la famille. Cette propriété peut être utilisée sur plusieurs lignes. Dans ce cas les valeurs de lignes suivantes devront être précédées du caractère '+'. Si on précède le valeur par '*', cela indique que ce fichier n'est pas intégré directement dans la famille; mais qu'il est utilisé comme méthodes héritées. Cela implique que les méthodes définies dans ce fichier peuvent être redéfinies dans un autre fichier méthodes.
WID	identifiant du cycle de vie associé à la famille
CVID	identifiant du contrôle de vue associé à la famille

SCHAR	caractéristique spéciale pour la révision : 'R' : signifie auto révision à chaque modification 'S' : signifie document non révisable
USEFOR	caractère désignant une utilisation spéciale. Seulement pour les documents systèmes : 'W' : pour les cycles de vie 'G' : pour les intercalaires de chemise 'P' : pour les profils
TAG	caractère désignant une marque applicative. Cette marque peut être utilisé sur plusieurs lignes ce qui a effet d'ajouter une marque. Les documents créés auront aussi cette marque.

2.1.3 Attributs

exemple famille image :

BEGIN	BASE0	image	IMAGE
TYPE	C		
SCHAR	R		
ICON	photo.gif		
//	idattr	idframe	label
		T	A
		type	ord
		vis	need linkphpf:phpfunc elink constraint options
ATTR	IMG_FRFILE	image	N N frame 0 W
ATTR	IMG_TITLE	IMG_FRFILE	titre Y N text 10 R
ATTR	IMG_FILE	IMG_FRFILE	image N Y image 20 W Y iwidth=80
ATTR	IMG_DESCRIPTION	IMG_FRFILE	description N Y longtext 30 W
ATTR	IMG_FR_CHAR	caractéristiques	N N frame 40 W
ATTR	IMG_CATG	IMG_FR_CHAR	catégorie N N enum 50 W 1 photo,2 dessin,1.1 paysage,1.2 portr
END	-		

Définition des colonnes :

1. ATTR
2. identifiant de l'attribut
3. identifiant du cadre ou du tableau englobant
4. désignation de l'attribut
5. 'Y' signifie qu'il fait partie du titre, 'N' il ne fait pas partie du titre
6. 'Y' signifie qu'il fait partie du résumé, 'N' il ne fait pas partie du résumé
7. type de l'attribut
8. ordre de l'attribut
9. visibilité de l'attribut
10. 'Y' signifie qu'il est obligatoire, 'N' il ne fait pas obligatoire
11. définition de l'hyperlien
12. nom du fichier php pour l'aide à la saisie
13. nom et attributs de la fonction pour l'aide à la saisie ou nom et attributs de la méthode de calcul (précédé de ::) s'il s'agit d'un attribut calculé (visibilité W)
14. extra lien
15. nom et attribut de la méthode de contrainte
16. options de présentation séparées par des '|'.

2.1.3.1 Les attributs d'une famille

Ce chapitre donne la description des colonnes permettant de créer ou modifier des familles.

Dans les chapitres suivants vous trouverez la description des autres colonnes pour réaliser des aides à la saisie, des menus,..

Rappel : Pour accéder à la modification d'une famille, il faut faire un clic droit sur la famille et

sélectionner le menu "Éditer les attributs"

Colonne "Identificateur" ou "ID"

Nom interne de l'attribut.

Il sert de référence lors des définitions des fonctions et des liens. Cet item n'est plus modifiable une fois créé car il peut être utilisé dans les parties spécifiques de la famille.

Cet identificateur doit être unique dans l'ensemble des familles. Il est conseillé de préfixer chacun des attributs par un trigramme identifiant la famille afin de bien percevoir les attributs par famille.

Colonne "Ordre"

Défini l'ordre de présentation des attributs dans le document. Cet ordre n'est pas utile pour les attributs de type *frame*.

Colonne "Nom"

Texte court définissant l'attribut. Ce texte est utilisé pour désigner l'attribut lors de la visualisation ou de l'édition.

Colonne "Type"

Défini le type syntaxique de l'attribut. Les types de base sont :

Type	Description
text	texte court sur une ligne
longtext	texte long sur plusieurs lignes
htmltext	texte formatable
frame	cadre dans lequel les attributs sont présentés
date	défini un attribut date de la forme JJ/MM/AAAA
time	heures et minutes
timestamp	date, heures et minutes
password	texte caché. Il est non visualisable. Des étoiles remplacent les caractères saisis.
file	fichier à télécharger.
image	image à télécharger. Formats image supportés : GIF, JPEG, PNG.
integer	nombre entier
double	nombre réel
money	nombre réel avec 2 chiffres après la virgule
enum	liste énumérée
docid("[ID_DE_FAMILLE]")	Lien vers un document de la famille ID_DE_FAMILLE ¹ Attention à bien mettre des double-quotes et non des guillemets typographique.
docid	identifiant de document ²
color	code RGB (rouge/vert/bleu) en hexadécimal (ex: #FF0000 pour rouge)
ifile	fichier intégré. Utilisable pour les petits fichiers afin d'être visualisé directement dans le document.
idoc	document intégré. ³
menu	lien vers une autre page.

Formater les attributs

Pour les attributs "text" un formatage du type (style langage C) peut être ajouté. Par exemple :

1) si ID_DE_FAMILLE n'existe pas lien vers n'importe quel document

2) Déprécié

3) plus maintenu

- text("%s environ") "pour postfixer environ après la valeur,
- text("%s") "pour afficher l'attribut en gras.

Ceci est aussi valable pour les type "int" ou "double"

- double("%.02f") : pour afficher un nombre avec 2 décimales
- integer("%d m²") : pour afficher (100 m²).
- double("%.02f %%") : pour afficher le signe pourcentage après le nombre

Pour les attributs "date", "time" et "timestamp" un formatage (style langage C "[strftime](#)") peut être ajouté. Par exemple :

- time("%H:%M:%S") pour afficher aussi les secondes,
- timestamp("%A %d %B %Y %X") donne « samedi 08 janvier 2005 10:13:00 ».

Colonne "Cadre" ou "fenêtre"

Défini le cadre dans lequel l'attribut doit être présenté. Il faut auparavant avoir défini les cadres.

Colonne "Résumé" ou "R?"

Si vous voulez que l'attribut soit affiché dans le mode colonne ou dans les aperçus de documents dans l'application GENERIC, il faut cocher la colonne 'R?'.

Colonne "Titre" ou "T?"

Le titre d'un document peut être composé par les valeurs d'un ou de plusieurs attributs. Il faut au moins mettre un des attributs en titre en cochant la colonne "T?". Le titre est composé en concaténant les valeurs des attributs titre suivant l'ordre donné.

Si vous modifiez la définition du titre, il est nécessaire d'utiliser la fonction Bash ("freedom_refresh") pour que les documents déjà créés aient leur titre modifié.

Colonne "Obligatoire" ou "O?"

Indique si l'attribut est obligatoire lors de la saisie.

Colonne "Visibilité" ou "Vi"

Cet attribut détermine dans quel cas l'attribut est affiché. La visibilité est :

Code	Description
W	attribut visible en lecture et modifiable en édition.
R	attribut visible en lecture seulement. Généralement ce sont des attributs calculés.
H	attribut caché. Généralement ce sont des attributs servant soit au calcul, soit à la génération des liens. Leur valeur n'est généralement pas exploitable directement.
O	attribut modifiable en édition mais non visible en lecture. Généralement utilisé en corrélation avec les attributs de type menu.
S	attribut visible en lecture et en édition, mais non modifiable en édition
U	uniquement pour le type array. Interdit l'ajout et la suppression de rangées dans le tableau (version >= 2.7.4). Correspond à "tableau statique".
I	invisible : l'attribut n'est pas présent en consultation et en édition dans le document. Un attribut de visibilité I n'est pas modifiable (méthode Doc::setValue inactive). Pour en modifier la valeur, il est nécessaire d'appliquer un masque de saisie modifiant sa visibilité

Les types d'attributs de structure

Pour positionner les champs dans une famille, il existe 3 types d'attributs particulier :

- frame : Un attribut de type "frame" permet d'ajouter une bordure et un titre pour regrouper des champs similaires
- array : Un attribut de type "array" permet de créer des tableaux
- tab : Un attribut de type "tab" permet de créer des onglets

Attribut de type "frame" - Regroupement de champs

L'attribut de type "frame" permet d'ajouter une bordure et un titre autour des champs.

Dans cet exemple, les deux champs de type "text" seront entourés par une bordure dont le titre est "Exemple de Frame" :

ID	fenêtre	Nom	T?	R?	type	Ordre	Vi	O?
demo_frame	-	Exemple de Frame			frame	1	W	
demo_text1	demo_frame	Texte 1			text	2	W	
demo_text2	demo_frame	Texte 2			text	3	W	

Pour paramétrer une famille, il faut donc commencer par définir les différentes frames et ensuite il faut placer les champs dans les frames créées en utilisant la colonne "fenêtre"

Remarques :

- Si vous souhaitez placer des frames l'une à côté de l'autre et non pas l'une en dessous de l'autre, il faudra créer une vue particulière comme expliqué dans le chapitre "vues de documents".

Attribut de type "array" - Crédation de tableaux

L'attribut de type "array" permet de construire un attribut qui contient une liste de tuples composés de un ou plusieurs attributs.

	ID	parent	label	T?	R?	type
ATTR	demo_frame		Cadre exemple			frame
ATTR	demo_fichiers	demo_frame	Fichier(s)			array
ATTR	demo_fichier	demo_fichiers	Fichier	N	N	file
ATTR	demo_comment	demo_fichiers	Commentaire	N	N	text

Dans cette exemple, on a un attribut "demo_fichiers" de type array qui contiendra une liste de couples { fichier, champ commentaire }. On a donc une liste de fichiers avec leur commentaire associé :

Fichier(s)	Fichier	Commentaire
+-	fichier1	commentaire 1
+-	fichier2	commentaire 2
+-	etc.	etc.

Chaque ligne sera préfixé de boutons pour ajouter/supprimer une ligne.

Attribut de type "tab" - Création d'onglets

Le type "tab" est le type structurant de plus haut niveau. Il ne peut avoir de parent. Il ne peut contenir que des attributs de type "frame".

	ID	parent	label	T?	R?	type
ATTR	demo_fr_ident		Identification	N	N	frame
ATTR	demo_title	demo_fr_ident	Titre	Y	N	text
ATTR	demo_tab_un		Onglet n°1	N	N	tab
ATTR	demo_fr_exun	demo_tab_un	Cadre exemple	N	N	frame
ATTR	demo_fichiers	demo_frame	Références	N	N	array
ATTR	demo_fichier	demo_fichiers	Fichier	N	N	file
ATTR	demo_comment	demo_fr_exun	Commentaire	N	N	text
ATTR	demo_tab_deux		Onglet n°2	N	N	tab
ATTR	demo_t_annexes	demo_tab_deux	Cadre exemple	N	N	frame
ATTR	demo_t_annexes	demo_fr_exdeux	Annexes	N	N	array
ATTR	demo_annexe	demo_t_annexes	Annexe	N	N	file
ATTR	demo_remarques	demo_t_annexes	Remarques	N	N	htmltext

Les attributs qui ne sont inclus dans des onglets sont toujours visibles quelques soit l'onglet. Dans l'exemple l'attribut 'demo_titre' est affiché en permanence.

2.1.3.2 Options des attributs

Option	Description	Restriction
esize	taille de champs de saisie en caractère	pour les attributs de type text
elabel	texte du tooltip apparaissant lorsque la souris est sur la zone de saisie	pour les attributs de type text
ititle	texte du tooltip du bouton '...' de l'aide à la saisie. Par défaut : « choisissez une valeur »	
submenu	nom du sous menu	pour les attributs de type menu et action
barmenu	mettre à yes si l'affichage nécessite la barre de menu du navigateur	pour les attributs de type menu et action
bmenu	mettre à no si l'attribut ne doit pas apparaître dans les menus des application GENERIC.	pour les attributs de type enum
system	mettre à yes si les éléments de l\$-1'énuméré doivent être écrasé par la nouvelle définition	pour les attributs de type enum
ltitle	Texte affichable en popup sur l'hyperlien lorsque la souris passe dessus	hyperlien valide
ltarget	Nom de la fenêtre destinataire de l'hyperlien. Par défaut _self.	hyperlien valide
global	Mettre à yes si le lien n'a pas de relation directe avec un document particulier. Dans ce cas, le menu apparaîtra aussi dans le menu 'outils' des applications issues de GENERIC. Ce menu apparaîtra aussi dans le menu contextuel du document famille.	pour les attributs de type menu
onlyglobal	Mettre à yes si option est déjà global et si vous ne voulez pas que le menu n'apparaisse pas dans le menu contextuel du document mais uniquement dans le menu 'outils' des applications issues de GENERIC.	pour les attributs de type menu
lconfirm	Mettre à yes pour afficher un message de confirmation avant	pour les attributs de

	l'activation du lien	type menu
tconfirm	Question fermée apparaissant pour la confirmation. Par défaut : "êtes vous sûr ?"	pour les attributs de type menu
mwidth	Largeur de la fenêtre destinataire de l'hyperlien. Par défaut 400px.	pour les attributs de type menu ou action (version >= 2.9.6)
mheight	Hauteur de la fenêtre destinataire de l'hyperlien. Par défaut 300px.	pour les attributs de type menu ou action (version >= 2.9.6)
elttitle	Texte affichable surgissant sur le bouton généré par l'extra-lien.	extra-lien valide
elsymbol	Caractère symbolique affiché sur le bouton généré par l'extra-lien.	extra-lien valide
eltarget	Nom de la fenêtre destinataire sur le bouton généré par l'extra-lien.	extra-lien valide (version >= 2.9.6)
batchfolder	Mettre à 'yes' si l'action défini doit être appliqué sur tous les éléments du dossier	Pour les attributs de type action. Uniquement dans le cadre de famille hérités de dossier
version	Mettre à 'yes' pour indiquer que la valeur de l'attribut fait partie de la composition de la version	
cwidth	largeur de la colonne pour des attributs présents dans un tableau peut être exprimer en pixel (100px) ou en pourcentage (30%)	pour les attributs de tableaux
editheight	hauteur de la zone d'édition : (400px par exemple) . Les valeurs exprimées en pourcentage ne sont pas possibles.	pour les attributs de type htmltext
toolbar	référence de la barre de menu.	pour les attributs de type htmltext. Soit Simple (le défaut), Basic (juste gras et souligné), Default (complet).
toolbarexpand	indique si la barre de menu doit être repliée ou non. Mettre à 'yes' (par défaut) pour la voir. Mettre à 'no' pour la replier	pour les attributs de type htmltext.
align	alignement horizontal pour les cellules de tableau. Valeur possible left, right, center, justify	uniquement pour les attributs insérés dans un tableau.
color	couleur du texte pour les cellules de tableau. Soit #RRGGBB soit nom de couleur (red,yellow,...)	uniquement pour les attributs insérés dans un tableau.
bgcolor	couleur du fond des cellules de tableau. Soit #RRGGBB soit nom de couleur (red,yellow,...)	uniquement pour les attributs insérés dans un tableau.
height	hauteur du corps du tableau. En pixel uniquement (pas de pourcentage). Si le corps du tableau dépasse la hauteur spécifiée, un ascenseur vertical apparaitra :!:Ne fonctionne qu'avec firefox	uniquement pour les attributs de type tableau. (version >= 2.8.1)
iwidth	largeur des images affichées pour la vue par défaut (défaut 80px). Les valeurs doivent être exprimées en pixel uniquement (ex: '100px'). Si la valeur est 'auto', l'image sera affichée à sa taille originale	uniquement pour les attributs de type image.
vlabel	visualisation du libellé en consultation et en édition [left;up;none]. left par défaut affiche le libellé à gauche de la valeur. up l'affiche dessus en souligné. none ne l'affiche pas. Dans le cas de up et none, la valeur de l'attribut prend la largeur de la fenêtre affichée.	version ≥ 2.7.0.
showempty	En consultation, voir le libellé de l'attribut même si sa valeur	version ≥ 2.7.4.

	est vide. Par exemple showempty=RIEN pour afficher RIEN s'il n'y a pas de valeur. Pour afficher juste le libellé mettre simplement un espace comme valeur showempty=	
boolcolor	En consultation, au lieu d'afficher le libelle, un carré de couleur est affiché. Exemple : boolcolor=red, green va afficher un carré rouge si pas coché et un carré vert si coché	version ≥ 2.8.1
autosuggest	En édition, sur une aide à la saisie, indique que la recherche est lancée à chaque modification du texte saisi. (par défaut yes). Mettre à no pour désactiver l'auto-suggestion	version ≥ 2.9.1
search	Indique si l'attribut est searchable. Mettre à 'no' (par défaut yes) indique que l'attribut ne fera pas l'objet d'une indexation plein texte.	version ≥ 2.9.3 - Limité au type file
inline	Indique si l'image doit être affichée dans la navigateur (mettre inline=yes). Par défaut inline=no.	version ≥ 2.9.4 - Limité au type image
rn	Méthode renommage d'un fichier. Le fichier est renommé lors du transfert du fichier sur le serveur. La syntaxe est par exemple rn=: :myNewName(). myNewName est une méthode de l'objet documentaire qui retourne une chaîne de caractères. Le premier argument de la méthode est le nom du fichier téléchargé. Il est recommandé que la méthode fournit une extension compatible avec le type mime pour l'utilisation ultérieure sur le poste client et les transformations. (pour récupérer l'extension d'un nom de fichier vous pouvez utiliser la fonction getFileExtension de la librairie Lib.FileMime.php)	version ≥ 2.11.1 - Limité aux types file et image
isuser	Sert à indiquer à l'interface d'accessibilité des profils quels sont les identificateurs qui peuvent être pris en compte. Ceci sert à filtrer les liens vers des documents qui ne sont pas des utilisateurs ou des groupes d'utilisateurs Mettre isuser=yes pour indiquer l'usage dans les profils dynamiques	version ≥ 2.11.3 - Limité au type docid
htmlclean	Nettoie les balises de fonts, de style etc. qui sont généralement issu d'un copier/coller (syntaxe : htmlclean=yes)	version ≥ 2.11.3 - Limité au type htmltext
firstopen	Indique que cet onglet doit être sélectionné à l'ouverture du document (consultation et rédaction).	version ≥ 2.11.6 - Limité au type tab
empty	Indique que le tableau s'il est vide ne doit pas afficher la première rangée. empty=yes	version ≥ 2.11.8 - Limité au type array
sortable	Indique que le tableau est triable L'utilisateur peut cliquer sur les entêtes de tableau pour trier sortable=yes	version ≥ 2.11.8 - Limité au type array
mselectsize	Indique le nombre d'items présentés pour les attributs énumérés multiples. Par défaut 3	version ≥ 2.11.11 - Limité au type enum multiple sans option eformat
multiple	Indique si un énuméré est à choix multiple (anciennement enumlist). Ceci est applicable aussi aux types docid typé afin d'indiquer la possibilité de lier plusieurs documents d'une même famille Mettre multiple=yes	version ≥ 2.11.11 - Limité aux types enum et docid
bgcolor	Indique si la couleur de fond d'un cadre. cela peut être une couleur simple (exemple : yellow) ou une couleur exprimée en RGB hexadécimal (exemple #FF335A)	version ≥ 2.11.13 - Limité au type frame
doctitle	Pour les attributs relation, indique si un attribut contenant le titre du document pointé par le lien doit être créé automatiquement. Si doctitle=auto le nom de l'attribut titre sera le nom de	version ≥ 2.11.13 - Limité au type docid

	l'attribut relation suivi de '_title'. Sinon le nom sera celui précisé dans la valeur de l'option doctitle	
preventfilechange	Pour les attributs fichiers, cela ajoute une contrainte pour que le fichier à remplacer provienne de la dernière version du serveur. Cela ne bloque pas un changement de fichier mais cela averti l'utilisateur dans le cas où le fichier ne correspond pas à cette dernière version Cela implique aussi que lors du téléchargement du fichier un code identifiant la version est ajouté dans le nom du fichier (exemple todo{i47307-56}.ods pour le fichier todo.ods) Mettre preventfilechange=yes pour activer cette option	version ≥ 2.12.9 - Limité aux types file et image
offline	Pour les attributs fichiers, cela permet d'inhiber une synchronisation avec l'outil freedom-offline. Par défaut tous les fichiers présents dans une famille sont synchronisés. Mettre offline=no pour ne pas synchroniser le fichier	version ≥ 2.14 - Limité aux types file et image
docrev	Pour les attributs docid, si docrev=latest, cela indique que la relation pointe vers la dernière révision document (ceci est la valeur par défaut). Dans ce cas ce sera l'initid du document pointé qui sera stocké en base Si docrev=fixed, cela sera l'id de la dernière révision au moment de l'appel qui sera affecté. Si docrev=state(keystate) alors le lien portera vers le document à l'état keystate. L'aide à la saisie filtrera aussi sur les documents à cet état	version ≥ 3.0 - Limité aux types docid en visibilité W

exemple : esize=3|elabel=saisissez votre prénom

Pour modifier la définition d'un attribut d'une famille mère, on utilisera **MODATTR** au lieu de ATTR. Cela indique que toute propriété non valuée aura la valeur de l'attribut défini dans la famille mère.

2.1.3.3 Types Spéciaux

IATTR

Copie d'attributs d'une frame d'une autre familles.

famille portail		id	class	name
BEGIN	DIR	dossier événements		SCALENDAR
TYPE	C			
	idattr	idframe	id famille	
IATTR	DCAL_FR_PRESENT		CALENDAR	
END				

Cette exemple copie les attributs et le frame DCAL_FR_PRESENT de la famille CALENDAR dans la famille SCALENDAR que l'on défini. Cela est utile dans le cas de ressemblance de famille mais sans héritage souhaité.

2.1.3.4 Valeurs par défaut

Les valeurs par défaut sont déclarées à l'aide du mot-clef DEFAULT dans la première colonne.

1	2	3
DEFAULT	SGATE_ACTION	GATE_WEATHER
DEFAULT	SGATE_IDRED	::userDocId()
DEFAULT	SGATE_RED	::getTitle(SGATE_IDRED)

DEFAULT	SGATE_DATE	::getDate()
DEFAULT	SGATE_ARRAY	ligne1\nligne2\nligne3

Définition des colonnes :

1. DEFAULT
2. identifiant de l'attribut
3. valeur par défaut. Soit du texte statique. Soit l'appel à une méthode connue de la famille. Pour les attributs à l'intérieur d'un tableau (array), il est possible d'initialiser plusieurs lignes du tableau en séparant chaque ligne par un '\n'

Fin de définition

Toutes définitions de familles doit ce terminer par le mot-clef END .

END	
------------	--

Remarque : Ensuite, après cette ligne, il est possible de définir une autre famille.

2.1.3.5 Les attributs énumérés**But**

Les attributs énumérés peuvent être utilisés pour définir un ensemble fini de choix de valeurs pour un attribut. On distinguera les attributs énumérés **simples** (type "enum") pour le choix d'une seule valeur parmi l'ensemble des choix, des attributs énumérés **multiples** (type "enumlist") pour le choix de plusieurs valeurs possibles.

L'ensemble des valeurs est une suite de couples <clef>|<label> séparés par des virgules. Le label est le texte qui est présenté sur l'interface; la clef la valeur qui est stockée en base de données. On peut ainsi modifier les labels sans pour autant changer la clef. La suppression d'une clef implique que c'est directement la clef qui sera affichée (les valeurs en base de données ne sont pas supprimées).

Par défaut un menu de filtrage sur les valeurs de cet attribut est ajouté dans la barre du haut pour toutes applications dérivées de GENERIC.

Mise en place

Deux options sont disponibles pour les énumérés bmenu et system. Si bmenu égal no, le menu supplémentaire ne sera pas affiché. Si system égal yes, les valeurs définies seront écrasées par les nouvelles valeurs définies. Par défaut, les valeurs des énumérés une fois initialisées ne sont pas modifiables lors de la redéfinition d'une famille.

Par défaut les attributs énumérés sont présentés à l'édition par une liste déroulante pour les énumérés simples et une liste à sélection multiple pour les énumérés multiples.

Plusieurs formats de sélection sont possibles :

- pour les énumérés simples :
 - vcheck : boutons radio (exclusifs) alignés verticalement
 - hcheck : boutons radio alignés horizontalement
 - auto : liste déroulante filtrante
 - bool : un seul bouton boîte à cocher (checkbox). Utilisable seulement pour **les énumérés à 2 valeurs**. Si la boîte n'est pas cochée, c'est la première valeur qui est affectée à l'attribut, et inversement.
- pour les énumérés multiples :
 - vcheck : boîtes à cocher alignées verticalement
 - hcheck : boîtes à cocher alignées horizontalement

- auto : liste déroulante filtrante

Le format sera décrit par l'option 'eformat'. Pour indiquer qu'un énuméré est multivalué on indiquera 'multiple=yes' dans les options.

Restriction: les énumérés multiples ne peuvent pas être utilisés dans les tableaux.

Pour les énumérés simples, si la valeur par défaut n'est pas spécifiée, c'est le premier choix qui est présenté. Si on veut donner la possibilité de ne rien choisir, il faut ajouter un élément qui a pour valeur espace (red|rouge, |sans avis). La sélection de 'sans avis' aura pour conséquence la suppression de la valeur de l'attribut.

idattr (ou ID)	label (ou Nom)	type	phpfunc (ou ft)	options
TST_COLOR1	ma couleur préférée	enum	red rouge,lightgreen vert,darkblue bleu foncé,yellow jaune	bmenu=no
TST_COLOR2	la couleur du vase	enum	red rouge,lightgreen vert,darkblue bleu foncé,yellow jaune	bmenu=no eformat=hcheck
TST_COLOR3	la couleur de la table	enum	red rouge,lightgreen vert,darkblue bleu foncé,yellow jaune	bmenu=no eformat=vcheck
TST_COLOR4	aimez-vous le vert ?	enum	red rouge,green vert	bmenu=no eformat=bool
TST_COLOR5	les couleurs de la fleur	enum	red rouge,lightgreen vert,darkblue bleu foncé,yellow jaune	bmenu=no m ultiple=yes
TST_COLOR6	les couleurs de l'arbre	enum	red rouge,lightgreen vert,darkblue bleu foncé,yellow jaune	bmenu=no eformat=hcheck multiple=yes
TST_COLOR7	les couleurs du ciel	enum	red rouge,lightgreen vert,darkblue bleu foncé,yellow jaune	bmenu=no eformat=vcheck multiple=yes

The screenshot shows a user interface with several input fields:

- "Ma couleur préférée : " followed by a dropdown menu containing "rouge" (selected), "vert", "bleu foncé", and "jaune".
- "La couleur du vase : " followed by a group of radio buttons: rouge (selected), vert, bleu foncé, jaune.
- "La couleur de la table : " followed by a group of radio buttons: rouge, vert (selected), bleu foncé, jaune.
- "Aimez vous le vert ? : " followed by a checkbox which is checked.
- "Les couleurs de la fleur : " followed by a group of checkboxes: rouge (selected), vert, bleu foncé (selected), jaune.
- "Les couleurs de l'arbre : " followed by a dropdown menu containing "vert" (selected), "bleu foncé", and "jaune".
- "Les couleurs du ciel : " followed by a group of checkboxes: rouge, vert, bleu foncé (selected), jaune.

La liste des choix peut aussi être fournie par une fonction PHP ce qui permet une réutilisabilité et une plus grande dynamique dans les choix proposés. Cette fonctionnalité est décrite dans le manuel de programmation.

Type ouvert

etype=open

Les énumérés de type 'ouvert' permettent aux utilisateurs de rajouter des entrées dans la liste des choix possibles. Si l'utilisateur a le droit d'éditer le document il a le droit de rajouter une entrée dans l'énuméré. Par contre, il ne pourra pas l'enlever directement. Seul l'administrateur pourra supprimer/corriger les entrées de cet énuméré.

Type libre

etype=free

Les énumérés de type 'libre' permettent aux utilisateurs de choisir un autre choix que ceux proposés. Ce choix libre n'est pas ajouté à la liste des entrées.

Créer ou modifier des énumérés avec l'interface Web

Pour créer ou modifier les énumérés avec une une interface Web, il faut :

- Faire un clic droit sur la famille et sélectionner le menu "Éditer les énumérés"
- Cliquer sur le bouton correspondant à l'énuméré à modifier
- Utiliser l'interface, pour ajouter, modifier ou supprimer des énumérés

Il est aussi possible de créer des sous-listes dans l'énumération grâce à l'interface web. Il suffit pour cela:

- De créer un nouvel énuméré
- Le déplacer grâce au flèche en dessous de l'énuméré titre
- Le déplacer vers la droite ou la gauche pour changer son niveau

Remarque : si l'énuméré à l'option "system=yes", l'interface de modification de l'énuméré ne sera pas disponible.

Fichier de test

FIXME internalmedia: freedom_2.14:admin:family:famille_test_enumere.ods contient un exemple de famille utilisant des attributs énumérés.

Vous pouvez suivre les instructions de [cette page](#) pour importer cette famille dans Freedom.

Aide à la création de champs enum

Si votre champ enum comporte de nombreuses entrées, FIXME internalmedia: freedom_2.14:admin:family:aide_creation_champ_enum_0.1.ods vous aidera à créer votre champ.

Localisation

Le texte peut être traduit. S'il contient le caractère "," ou ".", alors il faut précéder ce caractère de "\".

2.1.3.6 [Créer un lien hypertexte](#)

But

L'hyperlien sert à afficher une URL dans la fenêtre courante à partir de l'attribut. L'hyperlien peut être une simple URL statique ("<http://www.meteo.fr>"). L'URL peut être décrite avec des paramètres issus du document.

Création d'un hyperlien

Les références aux attributs du document sont écrites entre les caractères % en indiquant l'identifiant de l'attribut.

Soit l'attribut US_MAIL définissant le mail d'une personne. Pour déclencher l'édition d'un mail vers une personne il suffit de mettre l'hyperlien suivant :

```
mailto:%US_MAIL%
```

Autre exemple, soit l'attribut SI_TOWN indiquant la ville de la famille société. Pour avoir la météo de la ville il suffit de mettre l'hyperlien suivant :

```
http://www.viamichelin.com/viamichelin/fra/dyn/controller/weatherAmbiguous?
strLocation=%SI_TOWN%&strCountry=EUR
```

Le mot-clef %SI_TOWN% sera remplacé par la valeur de cet attribut. S'il n'y a aucune valeur pour un des attributs de l'URL, l'hyperlien ne sera pas affiché (on ne pourra pas cliquer sur l'attribut).

Deux mots-clefs spéciaux peuvent être utilisés pour la composition de l'URL :

- %S% : est remplacé par l'URL vers FreeDom avec le numéro de session. Il doit être utilisé en début de lien.
- %ID% : est remplacé par le numéro de référence du document.

Les [propriétés](#) du document peuvent aussi être utilisées comme %FROMID% par exemple afin d'avoir l'identifiant de la famille du document.

2.1.3.7 [Création d'une relation entre documents](#)

Relation simple

Pour créer une relation entre deux documents nous utiliserons le type 'docid'. Pour créer une relation entre le document et un document de la famille 'X' nous déclarerons l'attribut suivant

id	type	visibility
MA_RELATION	docid("X")	W

Cela donnera en édition une aide à la saisie classique vers un document de la famille X. En consultation l'utilisateur verra le titre du document lié avec un hyperlien vers celui-ci. Si la famille n'est pas précisée dans le type l'aide à la saisie ne sera pas proposée.

Par contre, seul le n° du document sera stocké et donc l'utilisateur ne pourra pas rechercher par le titre. Si vous voulez que cette relation soit recherchable par le titre il faut rajouter l'option doctitle.

Relation détaillé

L'aide à la saisie peut être personnalisée. Les documents ci-dessous donne plus d'explications sur la création de liens entre documents :

- [freedom_2.14:cookbook:attributes:inputhelp](#)
- [Comment créer des liens entre les documents](#)

Soit US_IDSOC l'attribut identifiant le document société de la famille personne. Pour créer un hyperlien vers le document lié il faut saisir l'URL suivante :

```
%S%app=FDL&action=FDL_CARD&id=%US_IDSOC%
```

Cette URL indique que l'application FREEDOM va effectuer l'action FDL_CARD (affichage d'un document) dont l'identificateur est celui défini dans l'attribut US_IDSOC.

Il est possible d'indiquer un texte lorsque le curseur est sur le lien en utilisant l'option "ltitle".

Link	...	Options
%S %app=FDL&action=FDL_CARD&id=%US_IDSOC%		ltitle=détail de la société

2.1.3.8 Ajouter une aide à la saisie

 : LES DOCUMENTS CI-DESSOUS SONT PLUS COMPLETS ET DEVRAIENT ETRE INTEGRES DANS CE CHAPITRE

Pour les attributs relations

Il est possible de modifier

id	type	vis	phpfile	phpfunc
MA_RELATION	docid("TST_FAM")	W	fdl.php	lfamily(D,FAM,CT):MA_RELATION,CT

L'attribut relation est composé de l'identificateur et du titre du document. Le titre du document n'est pas forcément un attribut défini comme cela est décrit ici. Pour le référencer dans un aide à la saisie on utilise le mot-clef "CT". Ce mot-clef permet de référencer le titre sur l'interface en édition.

Cas général

- [freedom_2.14:cookbook:attributes:inputhelp](#)
- [freedom_2.14:devel:aides_saisie](#)

But



Mise en place du fichier PHP et modification du paramétrage de la famille

Défini le nom du fichier PHP qui contient la fonction décrite ci-après. Ce fichier se trouve sur la machine serveur dans le répertoire

```
/usr/share/what/EXTERNALS
```

Les fonctions de base sont décrites dans le fichier "**fdl.php**".

Mise en place

Les fonctions d'aide à la saisie doivent être des fonctions PHP décrites dans le fichier défini. Leur spécification est décrite dans le manuel de programmation.

La spécification de l'appel suit la syntaxe suivante :

```
NomFonction(P1,P2,...):R1,R2,R3,...
```

- NomFonction : Nom de la fonction PHP à appeler
- P1,P2,...: Paramètres de la fonction. Pour ces attributs, il est possible de mettre:
 - D : coordonnée de la base de donnée Freedom
 - I : numéro de référence du document
 - {FAMNAME} : FAMNAME étant le nom interne de la famille. Est remplacé par le numéro de référence de la famille
 - {PARAM} : PARAM étant un identificateur de paramètre. Est remplacé par la valeur du paramètre
 - un chiffre : chiffre mis en argument tel quel
 - chaîne de caractère : mise en argument telle quelle
- R1,R2,R3,...: Identificateur du ou des attributs à modifier

La fonction d'aide à la saisie calcule une liste de choix en fonction des paramètres "P". Lorsque l'utilisateur fait son choix les attributs "R" sont modifiés.

Une fonction générale est décrite dans le fichier fdl.php. Elle peut être utilisée pour la recherche et pour lier des documents.

Cette fonction est la suivante

```
lfamily($dbaccess, $famid, $name, $catgid=0)
\(plus d'info sur lfamily\)
```

Elle possède quatre paramètres dont un optionnel :

- \$dbaccess : coordonnées de la base de données
- \$famid : identificateur de la famille recherchée
- \$name : restriction de la recherche au document dont le titre comprend \$name
- \$catgid : identificateur de la catégorie, soit le numéro de référence du dossier dans lequel on cherche le document. Zéro indique que la recherche est faite dans toute la base.

La fonction retourne deux paramètres : identificateur du document et titre du document.

Exemple

soit la famille test dont les attributs sont :

- TST_IDSOC : identificateur de la famille société
- TST_SOC : nom de la société

on met un lien sur TST_SOC pour pointer sur le document société qui est lié à la famille test:

```
%$app=FDL&action=FDL_CARD&id=%TST_IDSOC%
```

Le fichier de TST_SOC est fdl.php et la fonction d'aide à la saisie est :
[lfamily\(D,SOCIETY,TST_SOC\):TST_IDSOC,TST_SOC](#)

ceci permet de choisir parmi les documents de la famille société dont le titre contient ce qui est déjà saisi dans l'attribut TST_SOC. Le choix déclenche la mise à jour de l'identificateur et du titre. Une fois le document validé (ou créé), l'lien sera visible sur l'attribut TST_SOC.

Ajouter un attribut calculé

 A mon avis, le document ci-dessous devrait être fusionné avec ce chapitre :

- [freedom_2.14:devel:attributs_calcules](#)

But

Un attribut calculé est un attribut dont la valeur est définie par une méthode. Cette valeur ne peut pas être modifiée directement par l'utilisateur. Par conséquent, ce type d'attribut ne peut pas avoir d'aide à la saisie. Sa visibilité est soit H (caché) soit R (lecture).

Mise en place du fichier PHP et modification du paramétrage de la famille



Exemple de fonctions

La définition du calcul se fait dans la même colonne que celle de l'aide à la saisie. Le calcul est effectué en appelant la méthode décrite. La valeur renvoyée par la méthode est affectée à l'attribut. La définition de l'appel de la méthode commence par '::'. Toutes les méthodes de la classe du document peuvent être appelées (méthodes générales (Classe Doc) et méthodes spécifiques).

Par exemple si on veut qu'un attribut de type 'date' contienne la date du jour on indiquera la méthode suivante.

```
::getDate()
```

L'attribut sera remis à jour à chaque rafraîchissement du document (à chaque visualisation complète).

Des paramètres peuvent être ajoutés si la méthode le requiert. Ces paramètres peuvent être des valeurs d'attributs ou des chaînes de caractères quelconques.

```
::getFutureDate(3)
::getFutureDate(ZOO_DELAY)
::getStrDate(%d/%m/%Y,ZOO_CDATE)
```

S'il y a plusieurs paramètres, ils doivent être séparés par des virgules.

Le résultat du calcul peut être appliqué à un autre attribut que celui où on a défini le calcul. Cela est utile dans le cas où on utilise un attribut avec aide à la saisie qui sera recalculé avec les paramètres issus de l'aide à la saisie.

Exemple classique : Mettre un lien calculé sur un document:

```
::getTitle(ZOO_IDDOC1):ZOO_DOC1
```

idattr	label	vis	link	phpfile	phpfunc
CMC_IDPROPO	id propo	H			::getTitle(CMC_IDPROPO):CMC_PROPO
CMC_PROPO	proposition	W	'%S %app=FDL&action=FDL_CARD&id=%CMC_IDPROPO%	Ext.php	Ipropo(D,CMC_PROPO):CMC_IDPROPO,CMC_PROPO

Dans cet exemple, l'utilisateur choisit une proposition avec la fonction d'aide à la saisie. Ceci à pour but de récupérer l'identificateur. Ensuite, cet identificateur est utilisé pour remettre à jour le titre du lien en cas de changement du titre du document lié.

Contraintes d'attributs

But

Certains attributs peuvent faire l'objet de contraintes spécifiques. C'est à dire que la valeur de l'attribut peut être soumis à certaines conditions que ce soit de syntaxe ou plus générales. La contrainte fait appel à une méthode de la classe d'objet documentaire.

Mise en place

idattr	label	vis	link	constraint
SI_MAIL	courriel	W	mailto:%SI_MAIL%	::parseMail(SI_MAIL)

On peut comme pour le champ "phpfunc" mettre comme paramètre à la méthode des attributs du document.

Si la contrainte de l'attribut n'est pas respectée, le document ne peut être enregistré. Par contre, si vous êtes connecté en temps que utilisateur 'admin', un bouton 'Valider!' apparaîtra afin d'outrepasser les contraintes. Seul cet utilisateur peut utiliser et voir ce bouton.

CARNET DE SANTE SANS TITRE 1064

Identification

Nom : Zoé

Liste Intervention

	date	description	mail	veterinaire
	18/12/2003	albert	hurlaaeAau@zo.net	
	17/12/2003	robert	aa@zee.net	

Valider Valider! Annuler

outrepasser les contraintes

Créer une entrée de menu contextuel



: [Ce document](#) donne d'autres informations concernant la modification des menus.

But

Ces attributs permettent d'ajouter des actions particulières dans le menu contextuel du document. Le lien définit l'url qui sera activée lors de la sélection du menu.

Mise en place



: Chapitre à expliciter

idattr (ou ID)	label (ou Nom)	type	link (ou Lien)	phpfunc (ou ft)	option
CMC_M_M1	Proposition n°1	menu	%S %app=FDL&action=FDL_CARD&id=%CMC_IDPROPO1%		submenu=proposition
CMC_M_M2	Proposition n°2	menu	%S %app=FDL&action=FDL_CARD&id=%CMC_IDPROPO2%		submenu=proposition
CMC_M_PROPO	Voir la proposition	menu	%S %app=FDL&action=FDL_CARD&id=%CMC_IDPROPO%	::verifyProposition() lconfirm=yes	

La colonne "phpfunc" permet d'indiquer un pré-condition à l'affichage du menu. Cette méthode de l'objet documentaire retourne l'activité possible du lien (actif, inactif, invisible).

Modifier la fenêtre de destination

Par défaut, le résultat de l'activation d'un item de menu s'ouvre dans une fenêtre autonome. Chaque item de menu à sa propre fenêtre de résultat. La fenêtre cliente de destination peut être modifiée en spécifiant le nom de la fenêtre dans l'option "ltarget".

Exemple, pour avoir le résultat dans la fenêtre 'test' :

link (ou Lien)	...	options
%S %app=FDL&action=FDL_CARD&id =%US_IDSOC%		ltarget=test

Pour avoir le résultat dans la fenêtre courante on utilisera le nom '_self'

link (ou Lien)	...	options
%S %app=FDL&action=FDL_CARD&id =%US_IDSOC%		ltarget=_self

Pour avoir le résultat dans une fenêtre différente à chaque activation, on utilisera le nom '_blank'

link (ou Lien)	...	options
%S %app=FDL&action=FDL_CARD&id =%US_IDSOC%		ltarget=_blank

Demander une confirmation pour une action critique

Si le résultat d'une action menu est critique, une confirmation peut-être demandée avant l'exécution de l'action. Cette confirmation est indiquée en mettant l'option "**lconfirm=yes**".

Le texte de la confirmation peut être personnalisée avec l'option "**tconfirm=<mon texte>**"

link (ou Lien)	...	options
%S %app=FDL&action=FDL_CARD&id =%US_IDSOC%		ltarget=test lconfirm=yes tconfirm=Action très risquée\nVous voulez continuer ?

Mettre le menu accessible uniquement avec la touche CTRL

Si l'item du menu ne doit être utilisé que dans de rare occasion, il peut être visible que lors de l'appui conjugué avec la touche 'Ctrl'. On indique cette possibilité en mettant l'option "**lcontrol=yes**". Cela indique aussi que le menu sera disponible dans le menu "autres" de la barre de menu du document.

link (ou Lien)	...	options
%S %app=FDL&action=FDL_CARD&id =%US_IDSOC%		lcontrol=yes
%S %app=FDL&action=FDL_CARD&id =%US_IDSOC%		ltarget=test lconfirm=yes lcontrol=yes

Fichier de test

Ces fichiers contiennent des exemples de menu :

-

```
FIXME internalmedia: freedom_2.14:admin:family:famille_test_menu.ods
```

-

```
FIXME internalmedia:  
freedom_2.14:admin:family:menu Rafraîchir et précédent_0.1.ods
```

Vous pouvez suivre les instructions de [cette page](#) pour importer ces familles dans Freedom.

Créer un extra lien

L'extra lien permet d'afficher un bouton supplémentaire lors de l'édition d'un document. Ce bouton est spécifié par un lien comme pour les hyperliens définis ci-dessus. Cet extra-lien est utilisé le plus souvent pour que l'utilisateur crée ou modifie un document secondaire lors de l'édition d'un document principal.

Il est possible de préfixer le lien pour changer le caractère sur la bouton (par défaut '+') et le texte surgissant affiché.

id	type	vis	link	...	eLink	...	Options
MA_RELATIO N	docid("FACT ")	W			%S %app=GENERIC&action =GENERIC_EDIT&classid =FACT&id= %MA_RELATION%		elttitle=éditer une facture elsymbol=\$

L'exemple ci-dessus affichera le bouton '\$' avec le texte surgissant « éditer une facture ». Le lien GENERIC_EDIT est l'action par défaut d'édition de document.

2.1.4 Paramètres

La définition d'un paramètre est identique à celle d'un attribut normal. Il suffit de mettre PARAM dans la première colonne au lieu de ATTR.

//	idattr	idframe	label	T	A	type	ord
PARAM	WSGAT_FR_ PAR		météo	N	N	frame	10
PARAM	WSGATE_UR L	WSGAT_FR_ PAR	url site météo	N	N	text	20

2.1.5 Modification d'une famille

2.1.5.1 Modifier un attribut père

Pour modifier un attribut hérité donc normalement non modifiable, il faut créer un nouvel attribut (nouvelle ligne) ayant le même identifiant que l'attribut à modifier.

Il est nécessaire de redéfinir toutes les propriétés de l'attribut.

L'attribut modifié perd toute relation avec l'attribut défini dans ses ancêtres. Si les propriétés de l'attribut de l'ancêtre sont modifiés, l'attribut fils modifié n'est pas impacté.

Voici un exemple avec une nouvelle famille basée sur la famille "de base"

$\wedge \text{ID}^{\wedge} \text{ordre}^{\wedge} \text{Nom}^{\wedge} \text{type}^{\wedge} \text{fenêtre}^{\wedge} \text{R?}^{\wedge} \text{T?}^{\wedge} \text{O?}^{\wedge} \text{Vi}^{\wedge}$

fr_basic	0	basique	frame	-					W
ba_title	1	titre	text	basic		x	x		O
ba_title	1	Mon nouveau titre	longtext	basic		x	x		W

Dans ce cas, le champ "titre" ne sera plus visible car il sera remplacé par le champ "Mon nouveau titre"

2.1.5.2 Supprimer un attribut

On ne peut pas supprimer les attributs créés. Par contre on peut les rendre invisibles en affectant leur visibilité à H (invisible).

2.1.5.3 Supprimer un attribut père

Si on ne veut pas hériter d'un des attributs des ancêtres, il faut :

- Créer un nouvel attribut ayant le même identifiant (comme pour modifier un attribut)
- Rendre ce nouvel attribut invisible (comme on le fait pour la suppression de ses propres attributs)

2.1.6 Modifier une famille depuis l'interface Web

Pour modifier une famille, il faut :

- Aller dans le module "Gestion documentaire"
- Sélectionner la recherche "les familles"
- Sélectionner la famille à modifier
- Faire un clic droit sur la famille et sélectionner le menu "Éditer les attributs"

2.1.6.1 Ajouter un attribut

L'ajout d'attribut se fait sans condition en utilisant les trois dernières lignes de l'interface "éditer les attributs".

2.1.6.2 Paramétrage d'une famille depuis l'interface

Dossier racine d'une famille

Ce dossier sert de dossier racine dans l'application ONEFAM (Gestion par famille). Si une famille a un dossier par défaut, elle peut être gérée par l'application ONEFAM.

Icône de la famille

Pour changer l'icône d'une famille, il faut :

- Faire un clic droit sur la famille et sélectionner le menu "Changer d'icône"
- Cliquer sur "Parcourir" et sélectionner une image sur le disque dur
- Cliquer sur le bouton "Importer"

2.1.6.3 Exemples de familles à importer dans Freedom

Vous trouverez dans ce chapitre des familles à importer dans Freedom. Le but est de vous

familiariser dans la création de familles et vous donnant des exemples.

Pour importer ces familles, il faut suivre [ce chapitre](#).

Mais en résumé, cette commande permet d'importer un document dans Freedom :

```
/usr/share/what/wsh.php --api=freedom_import --file=VotreFichier.ods
```

```
* FIXME internalmedia: freedom_2.14:admin:family:famille_test_frame.ods
```

•

```
FIXME internalmedia: freedom_2.14:admin:family:famille_test_attributs.ods
```

•

```
FIXME internalmedia: freedom_2.14:admin:family:famille_test_array.ods
```

•

```
FIXME internalmedia: freedom_2.14:admin:family:famille_test_tab.ods
```

•

```
FIXME internalmedia: freedom_2.14:admin:family:famille_test_heritage.ods
```

•

```
FIXME internalmedia: freedom_2.14:admin:family:famille_test_enumere.ods
```

•

```
FIXME internalmedia: freedom_2.14:admin:family:famille_test_menu.ods
```

2.2 Gestion des utilisateurs



Une mise à jour est nécessaire, ces informations étaient valables pour la version 2.3.0 nous sommes aujourd'hui en 2.10.0

2.2.1 Présentation

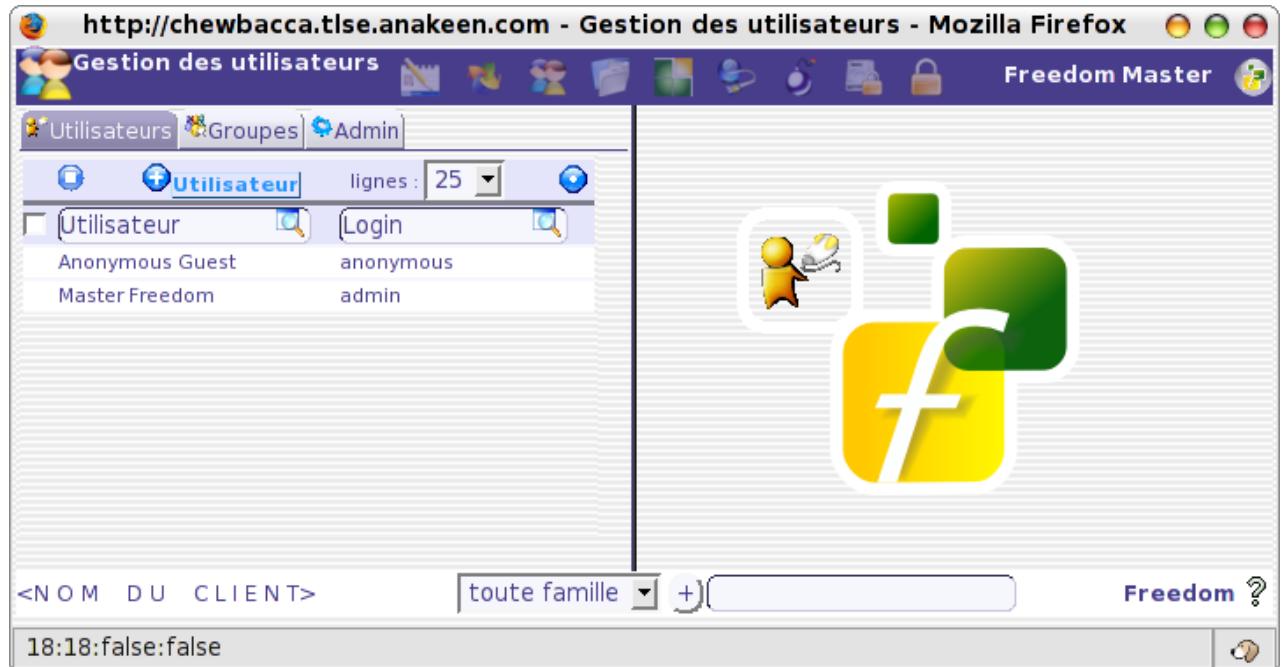
Ce manuel s'adresse aux personnes qui ont en charge de gérer les comptes d'accès à l'applicatif **freedom**.

L'accès aux applications hébergées par **freedom** est contrôlé. L'utilisateur doit s'authentifier pour

accéder à une application (ou à un service fourni par le serveur).

En fonction de son profil, il est autorisé ou non à utiliser ces applications ou certaines de leurs fonctions.

Afin de simplifier la gestion des utilisateurs et en particulier, la gestion de leur droits d'accès aux différentes applications, fonctions et services, l'administrateur définit des groupes. Un groupe rassemble un ensemble d'utilisateurs ayant des droits identiques pour la majorité d'entre eux au moins. La possibilité de modifier les droits d'un utilisateur (suppression ou ajout) par rapport au groupe reste possible.



L'interface principale se compose de deux parties.

Celle de gauche permet de rechercher des utilisateurs ou des groupes. La partie de droite sert à consulter les informations sur les utilisateurs ou des groupes de manière détaillée et sert aux interfaces d'éditions des groupes et des utilisateurs.

2.2.2 Gestion des groupes

2.2.2.1 Crédation de groupes

À l'initialisation de FREEDOM, deux groupes ont été créés : le groupe « utilisateurs » et le groupe « administrateurs ». Le premier de ces groupes est le groupe dans lequel sont affectés les nouveaux utilisateurs par défaut. Le deuxième groupe contient les utilisateurs qui ont les permissions d'administration informatique. Ces administrateurs ont le droit (par défaut) de créer de nouveaux groupes et de nouveaux utilisateurs. Ils peuvent aussi modifier n'importe quelles informations sur les utilisateurs (mot de passe, nom, prénom, ...).

Par défaut, pour qu'un utilisateur accède à l'application « gestion des utilisateurs », il doit appartenir au groupe administrateur. À l'initialisation de FREEDOM, un super administrateur (login: *admin*) a la responsabilité de créer un administrateur. Dans le cas contraire seul ce super administrateur pourra utiliser cette application.

Pour créer un nouveau groupe il faut cliquer sur l'onglet « Groupes » puis sur

freedom-toolbox 3.0

le bouton « **Créer un nouveau groupe** ».

Cela a pour effet d'afficher l'interface d'édition de groupes. Cette interface est composée de deux parties : une partie « **système** » et une partie « **groupe** ». Les informations qui doivent être obligatoirement renseignées sont toujours en gras. Ceci est valable quel que soit l'interface d'édition proposée par FREEDOM. Pour créer un groupe il faut saisir au moins les deux informations suivantes :

- **Nom** : dénomination du groupe. Souvent c'est sa fonction. Exemple : commerciaux, recherche & développement.
- **Login** : nom système. Il sert à l'identification du groupe. Ce nom ne doit contenir que des caractères minuscules alphabétique ou des chiffres ainsi que les caractères (tiret), _ (blanc souligné), . point. Il ne doit pas contenir d'espaces ni de caractères accentués. Ce nom doit être unique parmi l'ensemble des autres noms de groupes mais aussi des utilisateurs.

Les autres renseignements sont facultatifs.

- **Domaine** : est par défaut à local. Cela veut dire que le groupes n'est pas connecté à un serveur de messagerie géré par FREEDOM. Si le module de messagerie est installé alors les domaines gérés par FREEDOM apparaîtront. Si le domaine est « local », l'adresse email du groupe sera la concaténation des adresses emails des membres du groupes. Dans le cas contraire, l'adresse email sera composée de login@domaine. Dans ce cas c'est le serveur de messagerie géré par FREEDOM qui relayera à tous les membres les courriels adressés aux membres du groupe.
- **Rôle** : texte définissant le rôle, l'activité du groupe.
- **Fonction** : par défaut les groupes constitués sont des groupes fonctionnels. Ils servent à rassembler des utilisateurs ayant des fonctions similaires dans l'entreprise. Cet attribut sert à préciser le lien entre les membres. L'option « service » indique que cela est un groupe qui rassemble les membres d'un même service. Dans ce cas le nom du groupe est généralement le nom du service. L'option « bureau » indique que les membres sont localisés dans le même « bureau ». D'autres options peuvent être ajoutées.

Par défaut le nouveau groupe appartient au groupe « utilisateurs ». Si vous voulez le faire appartenir à un autre groupe (ou à aucun groupe), cliquer sur l'onglet « **Groupe** ». Cela affiche l'arborescence des groupes. Pour déplier entièrement l'arbre des groupes cliquez sur le premier bouton **+**.

Pour faire appartenir votre groupe à un autre groupe cliquez sur le nom du groupe choisi. Celui-ci s'affiche alors en vert. Les groupes affichés en vert indiquent les groupes dont votre groupe fait parti **directement**. Les groupes affichés en bleu indiquent les groupes dont votre groupe fait parti **indirectement**, c'est à dire qu'un des ses groupes parents fait parti des groupes en bleu. Pour désélectionner un groupe il suffit de recliquer dessus. La hiérarchie des groupes suit une logique arborescente. Il est par conséquent interdit de faire appartenir le groupe à un de ses fils ou à lui-même.

Dans ce cas si vous sélectionnez un des fils ou lui-même, les groupes non autorisés s'afficheront en rouge. Il faut, dans ce cas, recliquer dessus pour enlever l'alerte.





Pour des raisons de maintenance et de performances, il est conseillé de ne pas sélectionner un groupe qui est déjà bleu. Cela n'a pour effet que de répéter le fait qu'il appartient à ce groupe. Au niveau des droits d'accès il n'est fait aucune différences entre un groupe indirect (bleu) et un groupe direct (vert).

Pour voir le descriptif détaillé pour un groupe, vous pouvez cliquer sur l'icone affichée à gauche du nom. Cela ouvre un petit cadre avec la description du groupe.

Lorsque les informations sont remplis, il faut valider votre ajout ou modification en appuyant sur le bouton **Créer** ou **Sauver** suivant que vous êtes en création ou en modification.

Vous obtenez alors le document du nouveau groupe.

2.2.2.2 Modification d'un groupe

Pour modifier un groupe il faut se placer dans l'onglet « **Groupes** » du cadre gauche. Ensuite vous cliquez sur l'icone placée à gauche du nom du groupe. Cela affiche la description complète du groupe.



Ensuite si vous cliquez sur la document descriptif du groupe avec le bouton droit de la souris un menu contextuel s'affiche. De manière générale, si le curseur de la souris devient une croix cela indique que vous avez accès à un menu contextuel. Ceci est valable pour toute consultation de document sous FREEDOM. Sélectionnez alors **éditer** pour afficher l'interface d'édition. Vous pouvez aussi double-cliquer sur un onglet pour passer en mode édition.



L'interface d'édition est identique à celle de création et fonctionne de la même façon. Cependant le login n'est pas modifiable une fois la création effectuée. De même si un domaine de messagerie est associé au groupe il ne peut plus être modifié.

2.2.2.3 2.3. Consultation du groupe

Vous avez trois manières de voir les membres du groupes.

Dans l'onglet Groupes cliquez sur le nom du groupe. Cela fait apparaître l'ensemble des utilisateurs qui sont directement rattachés au groupe

Sur le document descriptif du groupe, cliquer sur l'onglet **membres**. Cela fait apparaître deux tableaux. Le premiers « Personnes du groupes » indique les utilisateurs qui sont directement dans ce groupes. Le deuxième tableau montre en plus les utilisateurs qui sont rattachés indirectement à ce groupe. Dans l'onglet « **Groupes** », vous pouvez voir deux tableaux : celui des « sous-groupes » et celui des « groupes parents ». Le tableau des sous-groupes contient la liste des groupes directement attachés à ce groupe. Le tableaux « groupes parents » contient la liste des groupes dans lesquels le groupe fait partie directement. Si ces tableaux sont vides ils n'apparaissent pas dans la description. Afin d'accéder à la consultation de membre vous pouvez cliquer sur le nom afin d'afficher l'information détaillée. Si vous cliquez sur le nom de l'utilisateur avec le bouton droit de la souris une mini vue s'ouvre avec le détail du groupe ou de l'utilisateur correspondant. Ceci vous permet de voir un détail sans quitter la page.

La troisième manière de voir les membres du groupe est d'utiliser l'item « ouvrir » du menu contextuel. Il est accessible à partir du clic droit du document de description du groupe. Cela vous ouvre une fenêtre qui contient la liste des groupes et des utilisateurs appartenant directement au groupe.



2.2.2.4 2.4. Suppression de groupe

Pour supprimer un groupe, il faut utiliser le menu contextuel du document descriptif du groupe. Dans ce menu contextuel, il faut sélectionner l'item « **supprimer** » afin de supprimer le groupe. Attention, cette opération est irréversible. Une confirmation de suppression vous est demandée afin de valider votre demande.

Cette opération a pour effet de détacher tous les membres du groupe de ce groupe. Cela ne supprime pas les membres du groupes (utilisateurs et sous-groupes).

Une fois l'opération achevée le document descriptif est affiché avec la mention « Ce document a été supprimé ».



2.2.3 3. Gestion des utilisateurs

2.2.3.1 3.1. Crédit d'utilisateurs

La création d'utilisateur se fait en cliquant sur le bouton « **utilisateur** » présent dans l'onglet « **utilisateurs** ». Ceci vous ouvre l'interface de création d'utilisateur. Les informations obligatoires pour la création d'utilisateur sont indiquées en gras et sont toutes sur l'onglet « système ». Le

login d'un utilisateur ne doit contenir que des caractères alphabétiques ou numériques. Il ne doit pas contenir ni d'espaces ni de caractères accentués. De plus ce login doit être unique. Si vous saisissez un login incorrect un bouton rouge apparaîtra à côté du champ de saisie pour vous indiquer que le login saisi n'est pas valide. Le login ne peut être saisi qu'à la création. Vous ne pourrez pas modifier ce login après la création.

Le mot de passe doit être saisi deux fois afin d'éviter des erreurs de saisie (vérifiez bien que le clavier n'a pas le CAPS LOCK enclenché).

L'administrateur peut préciser une période de renouvellement du mot de passe. Ceci est utile dans le cas où vous voulez forcer l'utilisateur à changer son mot de passe tous les mois par exemple.

Pour indiquer un changement tous les mois vous indiquer 30 dans le champ « délai d'expiration en jours ». Si vous ne remplissez pas la date d'expiration, elle sera calculée à la date du jour + 30 jours. Si vous indiquez une date c'est à cette date que l'utilisateur devra changer son mot de passe. À la date d'expiration, lorsque l'utilisateur se connectera, une demande de changement de mot de passe lui sera proposée. Une fois son mot de passe changé la nouvelle date d'expiration sera la date de changement + le délai d'expiration. Pour annuler cette notion d'expiration vous devez saisir « -1 » dans le champ « délai d'expiration ».

Vous pouvez interdire l'accès à un utilisateur temporairement en mettant l'attribut « état » à désactivé. Ceci à pour effet d'interdire l'accès à l'applicatif dans sa globalité.

L'attribut « domaine » indique le domaine de messagerie de l'utilisateur. Si vous n'avez pas le module de messagerie installé, seul « local » et « externe » sont proposés. « local » indique que l'utilisateur n'a pas de domaine et donc pas d'adresse email. « externe » indique que l'adresse email de l'utilisateur n'est pas géré par FREEDOM. Dans ce cas, on peut saisir une adresse email quelconque valide. Dans le cas d'un domaine de messagerie géré par FREEDOM, l'adresse email est constituée par le [login@domain](#).

Le deuxième onglet concerne l'appartenance à des groupes. Par défaut l'utilisateur appartient au groupe « utilisateurs ». L'utilisation est la même que celle décrite page 3 §2.1.

Le troisième onglet concernent des informations générales sur l'utilisateur.

2.2.3.2 3.2. Modification d'utilisateurs

L'interface d'édition peut être affichée de 3 façons différentes:

7. Soit en cliquant sur l'icone de l'utilisateur lorsque vous êtes dans l'onglet « utilisateurs ».
8. Soit par le menu contextuel (item « éditer ») lorsque vous êtes sur le document descriptif de l'utilisateur.
9. Soit en double-cliquant sur un des onglets du document descriptif.

Les champs affichés sont les mêmes que ceux de la création. La seule contrainte est que vous ne pouvez pas modifier le login.

Affecter un utilisateur dans un groupe : Pour affecter un utilisateur dans un groupe, il faut cliquer sur l'icône précédant le nom du groupe, afin d'accéder à l'interface d'édition du groupe. Cette interface propose le menu "Gérer les membres", qui permet donc de rajouter cet utilisateur dans les membres de ce groupe.

Permettre à un utilisateur de gérer les autres utilisateurs : le compte admin permet de gérer les utilisateurs. Malgré cela, si on veut donner le droit de gérer les utilisateurs et les groupes à un utilisateur particulier, il faut alors le mettre dans le groupe "Administrateur" puis de lui affecter l'ACL "FUSERS" de l'application "FUSERS".

2.2.3.3 3.3. Consultation d'utilisateurs

La consultation des utilisateurs peut être fait soit par groupes (onglet « groupes ») soit de manière générale onglet « utilisateurs ». Vous pouvez rechercher un utilisateur soit par son nom et prénom soit par son login en saisissant dans les champs marqués le début ou une partie du critère recherché. Vous envoyez votre demande de recherche en appuyant sur la touche « entrée » du clavier. Ensuite il suffit de cliquer sur la rangée pour afficher le document complet décrivant l'utilisateur.

Dans l'onglet groupe, la hiérarchie des groupes vous est présentée. Si vous cliquez sur le nom d'un groupe vous retrouvez l'interface de consultation des utilisateurs mais limité au groupe. Si vous faites des recherches dans cette interface, elle portera uniquement sur les membres directs du groupe.

2.2.3.4 3.4. Suppression d'utilisateur

Pour supprimer un utilisateur, il faut utiliser le menu contextuel du document descriptif de l'utilisateur. Dans ce menu contextuel, il faut sélectionner l'item « supprimer » afin de supprimer l'utilisateur. Attention, cette opération est irréversible. Une confirmation de suppression vous est demandée afin de valider votre demande.

Cette opération a pour effet de ne plus autoriser la connexion à cet utilisateur. Tous les documents (contacts, sociétés, etc.) que cet utilisateur a créé sont conservés.

Une fois l'opération achevée le document descriptif est affiché avec la mention « Ce document a été supprimé ».



2.2.4 4. Administration



Les fonctions d'administration concernent la mise à jour des utilisateurs et groupes ainsi que l'importation et la gestion des attributs « catégorie » des utilisateurs et des groupes.

2.2.4.1 4.1. Mise à jours

Le bouton « **Actualiser les utilisateurs** » effectue une ré-importation des groupes et utilisateurs entre la base de données et FREEDOM. Vous pouvez utiliser ce bouton dans le cas où des mises à jours ont eu lieu directement en base de données. Si vous utilisez uniquement l'interface FREEDOM pour gérer les utilisateurs, il n'est pas nécessaire d'effectuer cette opération.

“Actualiser” les utilisateurs veut dire de mettre à jours les informations systèmes des utilisateurs depuis la base de données anakeen vers la base freedom. Ceci, dans un fonctionnement normal, n'est pas nécessaire. Cela est nécessaire quand :

- la base “anakeen” / table user / a été mise à jour directement à la main (par le requeteur postgresql)
- lorsque l'on a détruit accidentellement des données dans la base “freedom” / table doc128 (utilisateur) et doc127 (groupes) /

La commande, lorsqu'il y a de nombreux utilisateurs est consommatrice en temps. En effet, l'appartenances aux groupes, les adresses email de groupes, les données LDAP sont recalculés et mises à jour.

La commande shell d'actualisation est : (à effectuer en tant que root sur la machine serveur d'applications)

```
[root@www ~]# . /etc/freedom.conf
[root@www ~]# wsh --api=usercard_iuser
```

Pour raffraîchir un seul utilisateur :

```
[root@www ~]# wsh --api=usercard_iuser --whatid=10
```

10 est l'identifiant système de l'utilisateur (indiqué sur la fiche FREEDOM) - c'est l'attribut id de la table user de la base anakeen.

Le bouton « **Actualiser LDAP** » ré-importe tous les utilisateurs dans l'annuaire LDAP. Ceci n'est valable que si le paramètre « Activer LDAP » est activé³.

En ce qui concerne les groupes, une partie de leur attributs n'est pas actualisée directement à cause du temps de calcul qui peut être important lorsque les membres du groupes ou des groupes parents sont nombreux. Cela concerne les attributs (membres et mail du groupes°). Lorsque vous créez un nouvel utilisateur ou un nouveau groupe, le bouton « **Tous les groupes sont actualisés** » est remplacé par le bouton « **x groupe(s) à actualiser** ». Cela indique que certains groupes n'ont pas leur liste de membre à jours. Cela n'a aucune incidence sur les droits des utilisateurs. Vous pouvez alors déclencher le rafraîchissement du groupes en appuyant sur ce bouton. Ceci n'est pas obligatoire, FREEDOM vérifie s'il y a des groupes à actualiser toutes les heures et les mets à jours si nécessaire.

2.2.4.2 Importation

L'importation de groupes et d'utilisateurs est effectuée à l'aide d'un fichier d'importation CSV (texte séparé par des point-virgules).

Les quatre premières colonnes” sont des colonnes « système ». Elles permettent de savoir quel type d'importation doit être fait.”

Importation d'utilisateurs

Pour importer des utilisateurs, les paramètres suivants doivent être renseignés.

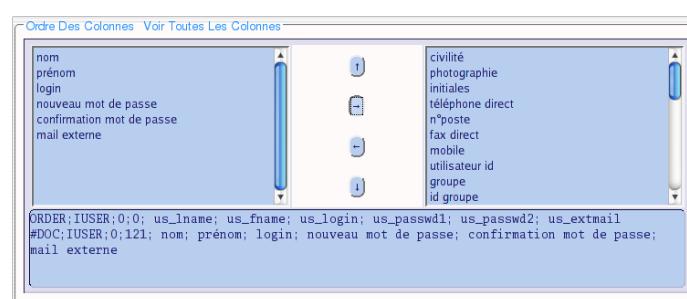
DOC	IUSER	0	0
-----	-------	---	---

N°	Signification	Valeurs
1	Indique l'action à faire. Mettre toujours "DOC" pour indiquer l'importation.	DOC
2	Identifiant de la famille à importer. Pour des utilisateurs mettre "IUSER".	IUSER
3	Identifiant spécifique pour le document descriptif correspondant à l'importation. Laisser à zéro si c'est un nouvel utilisateur. Si le numéro est celui d'un document descriptif déjà existant, celui-ci sera modifié.	
4	Identifiant du groupe d'appartenance de l'utilisateur. 0 indique qu'il appartient par défaut au groupe "utilisateurs".	

Les colonnes suivantes définissent les caractéristiques de l'utilisateur. Par défaut les colonnes suivantes concernent tous les attributs modifiables. Par contre certaines colonnes servent à stocker des attributs « système », ils ne doivent être remplis qu'en connaissance de cause.

Ci-dessous, voici les attributs par défaut qui peuvent être renseigné. En vert, ce sont les attributs obligatoirement présent. En bleu, ce sont les attributs optionnels et en rouge ce sont des attributs « système » qui ne doivent pas être renseignés (ils sont calculés par le système). Les attributs oranges sont des attributs qui ne doivent être modifiés que par des administrateurs expérimentés dans FREEDOM. Leurs définitions et leurs impacts ne sont pas décrite dans ce manuel.

Pour choisir les colonnes à importer, vous pouvez utiliser l'interface suivante. La sélection de gauche concerne les attributs que vous voulez renseigner. La partie de droite concerne les attributs que vous ne voulez pas renseigner.



Vous pouvez soit supprimer des colonnes, soit modifier leur ordre. Pour supprimer il faut sélectionner la (les) colonne(s) et les déplacer dans le cadre de droite (bouton →). Pour changer l'ordre des colonnes, il faut utiliser les boutons ↑ et ↓. L'ordre et la définition des colonnes utilisées seront celui du cadre de gauche. Lorsque vous modifiez cet ordre, l'écriture des colonnes en bas est aussitôt modifiée.

Vous pouvez utiliser ces deux lignes inscrites dans le cadre pour votre fichier CSV afin d'indiquer l'ordre de vos colonnes. La première ligne 'ORDER' indique au logiciel le nouvel ordre des attributs d'importation établi pour cette famille. S'il n'y a pas de ligne ORDER dans le fichier, l'ordre sera

us_civilty	civilité
us_lname	nom
us_fname	prénom
us_photo	photographie
us_initials	initiales
us_pphone	téléphone direct
us_intphone	n°poste
us_pfax	fax direct
us_mobile	mobile
us_maid	utilisateur id
us_login	login
us_passwd1	nouveau mot de passe
us_passwd2	confirmation mot de passe
us_idgroup	id groupe
us_groupe	groupe
us_status	état
us_daydelay	délai d'expiration en jours
us_expiresd	date d'expiration
us_expirest	heure d'expiration
us_iddomain	id domain
us_domain	domaine
us_extmail	mail externe
us_idsociety	id société
us_socaddr	même adresse ?
us_society	site société
us_type	fonction
us_idservice	service id
us_service	service
us_job	métier
us_role	rôle
us_workaddr	adresse
us_worktown	code postal
us_workcedex	ville
us_country	cedex
us_web	pays
us_scatg	web
us_wgcal_gid	catégorie
	id

celui défini dans le cadre gauche. Dans le cas contraire ce sera l'ordre défini dans le fichier qui sera pris en compte. Les quatre premières colonnes d'une ligne ORDER sont composées du mot-clé 'ORDER', de l'identificateur de la famille puis de deux colonnes non utilisées (servant seulement pour l'alignement avec les lignes DOC). Les colonnes suivantes contiennent les identifiants des attributs de la famille. La deuxième ligne inscrite dans le cadre du bas ('#DOC') sert juste à aider l'utilisateur sur la signification des colonnes.

ORDER	IUSER	0	0	us_lname	us_fname	us_login	us_passwd1	us_passwd2	us_extmail
#DOC	IUSER	0	0	nom	prénom	login	nouveau mot de passe	confirmation mot de passe	mail externe
DOC	IUSER	0	0	Deschamps	Arthur	arthur.deschamps	m.dparthur	m.dparthur	arthur.deschamps@zoo.net
DOC	IUSER	0	0	Trompette	Gilberte	gtrompette	m.dpgilberte	m.dpgilberte	gtrompette@zoo.net

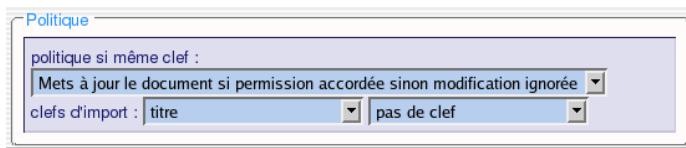
On peut avoir un fichier minimalist comme celui-ci:

Le fichier CSV produit par le tableur doit être comme ci-dessous. Il ne doit pas avoir de retour à la ligne entre 2 lignes d'importation.

```
ORDER;IUSER;0;0;us_lname;us_fname;us_login;us_passwd1; us_passwd2; us_extmail
#DOC;IUSER;0;121; nom; prénom; login; nouveau mot de passe; confirmation mot
de passe; mail externe
DOC;IUSER;0;0;Deschamps;Arthur;arthur.deschamps;Anakeen;Anakeen;arthur.descham
ps@zoo.net
DOC;IUSER;0;0;Trompette;Gilberte;gtrompette;Anakeen;Anakeen;gtrompette@zoo.net
```



Le cadre **dossiers** indique tous les groupes d'utilisateurs dans lesquels les utilisateurs peuvent appartenir. Si vous cochez un ou plusieurs groupes les utilisateurs importés seront rattachés aux groupes sélectionnés. Pour des raisons de simplifications, il est conseillé d'importer les utilisateurs par groupes. C'est à dire, qu'il faut produire un fichier CSV par groupe.



Le cadre 'Politique' permet de choisir la politique d'importation lorsqu'un utilisateur à importer possède des similitudes avec un utilisateur existant (par défaut même nom et prénom). La similitude est détectée par défaut si le titre du document décrivant l'utilisateur possède le même titre qu'un autre document de la même famille.

Dans le cadre d'importation d'utilisateurs, il n'est pas souhaitable de modifier la politique par défaut. Si vous avez des homonymes, vous pouvez utiliser le login comme clef d'import car il est une clef unique et on ne peut avoir deux utilisateurs avec le même login.

La cadre '**valeurs par défaut**' permet d'affecter des valeurs à toutes valeurs nulles (vides) définies dans le fichier d'importation. Ces valeurs par défaut sont uniquement utilisées pour des nouveaux documents, pas pour les mise à jours.

Dans le cas d'une modification si une valeur est vide cela signifiera que la précédente valeur sera inchangée. Pour supprimer une valeur il faut mettre le caractère 'espace'. Ceci indique explicitement la suppression de la valeur.

La case **Analyse seulement** est cochée par défaut. Si vous appuyez sur '**Importation CSV**' alors l'analyse du fichier sera effectuée et le résultat sera affiché dans le cadre du bas.

Résultat de l'analyse de l'import						
2 documents à prendre en compte.						
ligne	titre	dossier	id famille	action	message	modifications
1					nouvel ordre des colonnes us_lname - us_fname - us_login - us_passwd1 - us_passwd2 - us_extmail	
3	Deschamps Arthur	icamel d'adresses			ajouté Deschamps Arthur et ajout dans dossier icamel d'adresses	<ul style="list-style-type: none"> • [new-Deschamps] • [exists-Arthur] • [login-arthur-deschamps] • [nouveau mot de passe Anakeen] • [confirmation mot de passe Anakeen] • [mail externe arthur.deschamps@zoo.net]
4	Trompette Gilberte	icamel d'adresses			ajouté Trompette Gilberte et ajout dans dossier icamel d'adresses	<ul style="list-style-type: none"> • [new-Trompette] • [exists-Gilberte] • [login-gtrompette] • [nouveau mot de passe Anakeen] • [confirmation mot de passe Anakeen] • [mail externe gtrompette@zoo.net]

Les ajouts sont notifiés en vert, les modifications en jaune et les document ignorés en rouge.

Le résultat de l'analyse indique le nombre de lignes à prendre en compte (ajouté ou modifié). Il indique pour

chaque ligne du fichier, l'interprétation effectuée et l'action qu'il entreprendra. Si l'analyse est conforme aux attendus vous pouvez décocher 'analyse seulement' est lancer réellement l'importation. Le résultat apparaîtra dans le cadre du bas à la place de l'analyse.

Importation de groupes

L'interface d'importation des groupes est identiques à celle des utilisateurs. Pour indiquer qu'il s'agit d'une importation de groupe il faut indiquer IGROUP dans la deuxième colonne au lieu de IUSER (réservé pour les utilisateurs).

DOC	IGROUP	0	0
-----	--------	---	---

La liste des attributs nécessaires pour la création du groupe est la suivante.

grp_name	nom
grp_role	rôle
grp_type	fonction
us_login	login
us_meid	groupe id
us_iddomain	id domain
us_domain	domaine
grp_iduser	id membre
grp_idgroup	id groupe
grp_idruser	id membre
grp_idpgroup	id groupe
fld_fam_ids	id familles
fld_pdocid	id profil par défaut de document
fld_pdirid	id profil par défaut de dossier

La signalétique des couleurs est celle décrite au paragraphe précédent.

L'exemple suivant montre le fichier de création minimalistes pour la création de groupes

ORDER	IGROUP	0	0	grp_name	us_login
#DOC	IGROUP	0	0	nom	login
DOC	IGROUP	0	0	Recherche & Développement	gdevelop
DOC	IGROUP	0	0	Communication	gcomm

Le fichier CSV produit par le tableur aura la forme suivante:

```
ORDER;IGROUP;0;0; grp_name; us_login;;;;
#DOC;IGROUP;0;0; nom; prénom;;;;
DOC;IGROUP;0;0;Recherche & Développement;gdevelop;;;;
DOC;IGROUP;0;0;Communication;gcomm;;;
```

Comme pour les utilisateurs vous pouvez renseigner les valeurs par défaut et les groupes d'appartenances (onglet **dossiers**). Le résultat de l'importation donnera le résultat suivant.

2 documents à prendre en compte.							
ligne	titre	dossier	id famille	action	message	modifications	erreur
1					nouvel ordre des colonnes grp_name - us_login -----		
3	Recherche & Développement	/carnet d'adresses		ajouté	Recherche & Développement et ajout dans dossier carnet d'adresses	<ul style="list-style-type: none"> [nom:Recherche & Développement] [login:gdevelop] 	
4	Communication	/carnet d'adresses		ajouté	Communication et ajout dans dossier carnet d'adresses	<ul style="list-style-type: none"> [nom:Communication] [login:gcomm] 	

L'accès d'un groupe d'utilisateurs à une ACL peut être indiqué lors de l'importation des groupes utilisateurs. Pour cela, il faut ajouter une ligne dans le tableur d'import selon :

mot clé	nom_du_groupe	nom de l'appli	nom des acl	...
ACCESS	Group_A	APPLI LAMBDA	ACL1_APPLI_LAMBDA	ACL2_APPLI_LAMBDA
ACCESS	Group_B	APPLI ALPHA	ACL_APPLI_ALPHA	

Affectation de groupes

L'affectation des utilisateurs (ou des sous-groupes) à des groupes peut aussi être effectuée à l'aide du fichier CSV d'importation. Il faut pour cela nommer les groupes dans le fichier CSV puis ensuite référencer les utilisateurs (ou sous-groupes) au groupe voulus.

Pour affecter un nom à un groupe, il faut renseigner la colonne n°3 avec une clef. Cette clef est une chaîne de caractères sans espace et ne contenant pas de caractère accentué. Cette clef doit être unique pour tous les utilisateurs, groupes et autres documents nécessaires à FREEDOM. Pour cela il est fortement conseillé de préfixer la clef par "GRP_" pour les groupes et par "USER_" pour les utilisateurs.

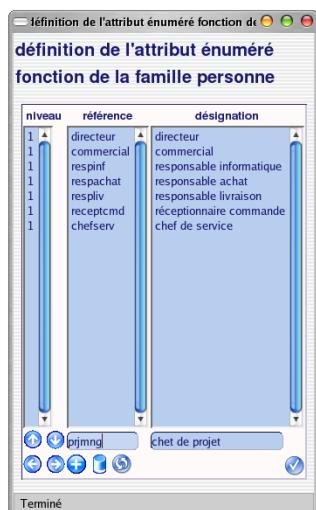
Si vous n'avez pas nommé le groupe à la création, vous pouvez le faire a posteriori en reprenant les informations composant le titre du document décrivant le groupe. Dans le cas du groupe, il faut au moins re-renseigner le nom ("grp_name") pour que FREEDOM puisse faire l'association avec un groupe existant (ceci est valable pour la politique d'importation par défaut puisque la clef est le titre). Pour affecter un utilisateur (ou un sous-groupe) à un groupe il faut mettre la clef du groupe dans la colonne n°4.

ORDER	IGROUP	0	0	grp_name	us_login
#DOC	IGROUP	id	fldid	nom	prénom
DOC	IGROUP	GRP_DEVEL		0 Recherche & Développement	gdevel
DOC	IGROUP	GRP_COMM	GRP_DEVEL	Communication	gcomm
ORDER	IUSER	0	0	us_lname	us_fnname
#DOC	IUSER	id	fldid	nom	prénom
DOC	IUSER		0 GRP_COMM	Deschamps	Arthur
DOC	IUSER		0 GRP_COMM	Trompette	Gilberte
DOC	IUSER		0 GRP_DEVEL	Deschamps	Arthur

Dans l'exemple ci-dessus, on affecte la clef GRP_DEVEL et GRP_COMM aux groupes "gdevel" et "gcomm". Le groupe "gcomm" devient un sous groupe de "gdevel". L'utilisateur « arthur deschamps » appartient aux groupes "gcomm" et "gdevel" et l'utilisatrice « gilberte trompette » appartient au groupe "gcomm".

Il n'est pas possible par l'importation CSV d'enlever des utilisateurs (ou des sous-groupes) à un groupe.

2.2.4.3 Catégories



Les éléments des attributs à choix multiples peuvent être modifiés en cliquant sur le bouton « **Modifier** » de l'attribut choisi. Cela ouvre l'interface de modification des catégories.

Pour ajouter un élément, il faut renseigner deux champs : la référence et la désignation. La référence est la clef qui sera stockée dans la base de données. La désignation est la signification de la référence. La référence ne doit contenir que des caractères alphabétiques ou numérique (pas d'espaces ni de caractères accentués ni de ponctuations). Une fois la référence et la désignation saisies , cliquer sur le bouton  pour ajouter le nouveau choix. Pour supprimer un choix, il faut d'abord sélectionner un élément en cliquant sur la référence puis sur le bouton  pour supprimer le choix. Vous pouvez modifier une désignation mais vous ne devez pas modifier une clef. En effet, si vous modifiez une désignation cela aura pour incidence de changer le nom du choix dans les différents documents. Le fait de supprimer une clef implique que les documents qui ont cette clef auront un choix qui n'est plus disponible. Si un document à un choix qui n'est plus disponible, ce sera le nom de la clef qui sera affiché. Pour modifier une désignation il faut d'abord sélectionner la référence. Cela affiche la sélection dans les champs du bas. Ensuite modifiez la désignation puis appuyez sur  pour valider le changement.

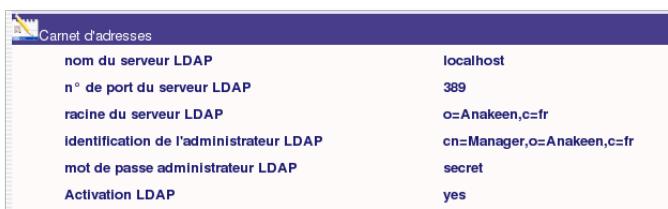
Ces choix peuvent suivre une hiérarchie. Pour changer un choix de niveau, il suffit de le sélectionner et de cliquer sur les boutons  et  pour augmenter ou diminuer le niveau.

Une fois vos modifications effectuées, appuyez sur le bouton  pour enregistrer les modifications. Tant que vous n'avait pas enregistré les modifications vous pouvez quitter la page sans sauver.

2.2.4.4 Configuration LDAP

Les documents décrivant les personnes et les utilisateurs peuvent être copiés dans un serveur LDAP. Ce serveur LDAP permet notamment aux clients de messagerie d'être reliés au carnet d'adresses des utilisateurs et des personnes.

Installation et paramétrage LDAP



Le serveur LDAP est accessible en lecture seulement. Seul l'administrateur du LDAP peut écrire dans cette base. Pour ce connecter au serveur LDAP il faut définir les 5 paramètres suivant:

- nom du serveur LDAP : nom ou adresse IP de la machine où le serveur LDAP est installé.
- n° du port : normalement 389 (n° d'écoute pour le serveur)
- racine du serveur : indique l'arborescence où les données seront placées
- administrateur LDAP : login (LDAP DN) de l'administrateur LDAP
- mot de passe LDAP : mot de passe en clair de l'administrateur

La modification de ces paramètres est accessible en cliquant sur votre nom (en haut à droite) et en choisissant l'onglet « paramètres applicatifs ». Ces 5 paramètres doivent se retrouver dans la configuration système du serveur LDAP.

Pour activer la copie avec le serveur LDAP il faut ensuite mettre le paramètre « Activation LDAP » à "yes" dans le paramétrage applicatif du carnet d'adresses. Dans le cas contraire la copie n'est pas effectuée.

On utilisera comme serveur LDAP le logiciel [openldap](#).

Dans le répertoire "/etc/openldap" se trouve un fichier de configuration fourni par FREEDOM :

"/etc/openldap/slapd_anakeen.conf". Ce fichier reprend la paramétrage par défaut que l'on a sur l'interface web de FREEDOM.

```
#view only its own organizationalUnit
access to dn.regex="^ou=([^\,]*),dc=people,(.*)$"
by dn.regex="uid=$1,dc=users,$2" read
by group.expand="cn=$1,dc=people,$2" read
by * none
#view only its private card
access to dn.regex="^uid.*,ou=([^\,]*),dc=people,(.*)$"
by dn.regex="uid=$1,dc=users,$2" read
by group.expand="cn=$1,dc=people,$2" read
by * none
access to attr=userPassword
by anonymous auth
by self read
by * none
#others
access to *
by dn="cn=Manager,o=Anakeen,c=fr" write
by anonymous read
by * read

database bdb

suffix "o=Anakeen,c=fr"

rootdn "cn=Manager,o=Anakeen,c=fr"

rootpw secret

...
```

Le fichier de configuration de openldap est "/etc/openldap/slapd.conf". Ce fichier doit inclure le fichier "slapd_anakeen.conf" soit par copie soit par "include". Le fichier de configuration principal doit contenir au moins les quatre schémas suivants ainsi que le chargement du module bdb (berkeley database).

```
include /etc/openldap/schema/core.schema
include /etc/openldap/schema/cosine.schema
include /etc/openldap/schema/inetorgperson.schema
include /etc/openldap/schema/nis.schema
```

Lorsque le fichier de configuration est correct il suffit de démarrer le serveur LDAP:

```
[root@chewbacca ~]# /etc/rc.d/init.d/ldap start
Checking configuration files for slapd: config file testing succeeded **[ OK ]
**
Démarrage de slapd :**[ OK ]**
```

Ensuite il faut initialiser la base LDAP. Pour cela on activera se script "usercard_ldapinit" comme le montre l'exemple suivant.

```
[root@chewbacca ~]# . /etc/freedom.conf
[root@chewbacca ~]# wsh --api=usercard_ldapinit
4)anonymous guest: **updated**
```

```
3)Master Freedom: **updated**  
2)Administrateurs: **updated**  
1)Utilisateurs: **updated**
```

Cette dernière opération peut être faite aussi via l'interface (cf §4.1).

Ensuite les documents personnes ou utilisateurs seront copiés sur le serveur LDAP lorsqu'ils seront modifiés ou créés par l'interface FREEDOM.

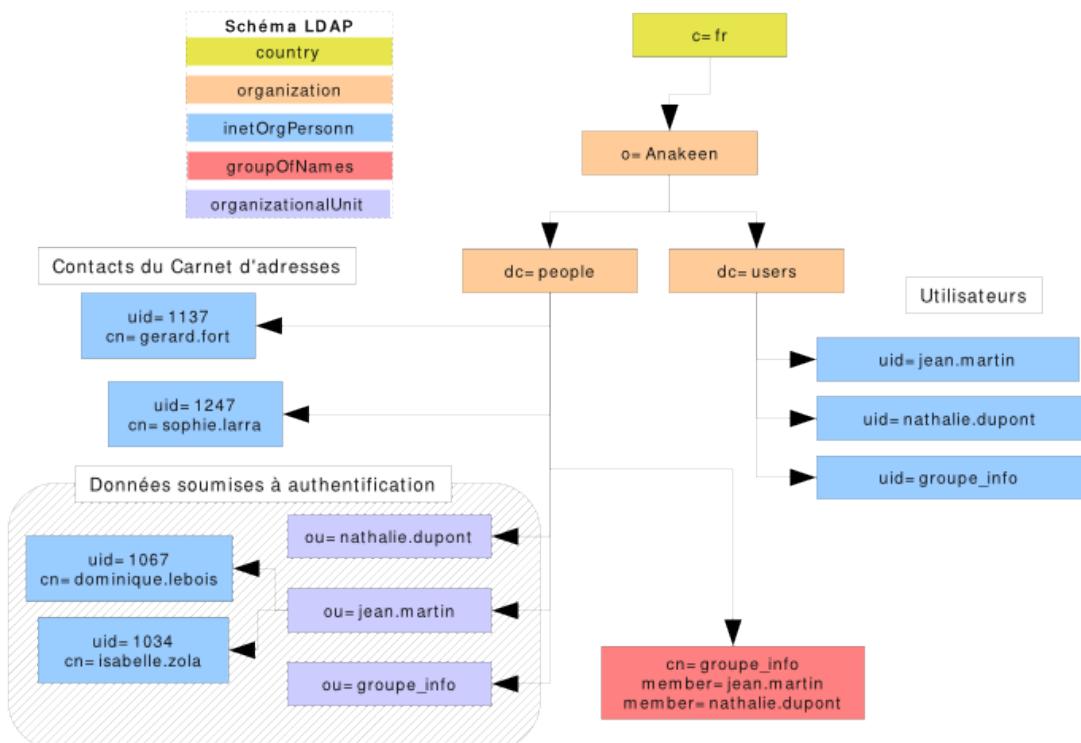
Note

Dans le cas où la racine de la base LDAP ne serait pas sous la forme "o=xxx,c=xxx", il faudra manuellement créer au préalable la base correspondante en important la définition LDIF de la racine organisation. Exemple pour une racine dc=anakeen,dc=fr :

```
# vi organisation-anakeen.ldif  
dn: dc=anakeen,dc=fr  
objectClass: dcobject  
objectClass: organization  
o: Anakeen  
dc: Anakeen  
# ldapadd -x -D 'cn=Manager,dc=anakeen,dc=fr' -W -f ./organisation-  
anakeen.ldif
```

Organisation du LDAP

L'annuaire LDAP est composé de deux branches principales : les utilisateurs ("dc=users") et les personnes du carnet d'adresses ("dc=people"). Les coordonnées des utilisateurs et des personnes à accès publics sont disponibles sans authentification. Les personnes déclarés avec un accès restreint ne sont accessibles qu'avec une authentification appropriée.



Configuration des clients LDAP

Ce paragraphe décrit comment paramétrier l'accès au serveur LDAP pour un client de messagerie.

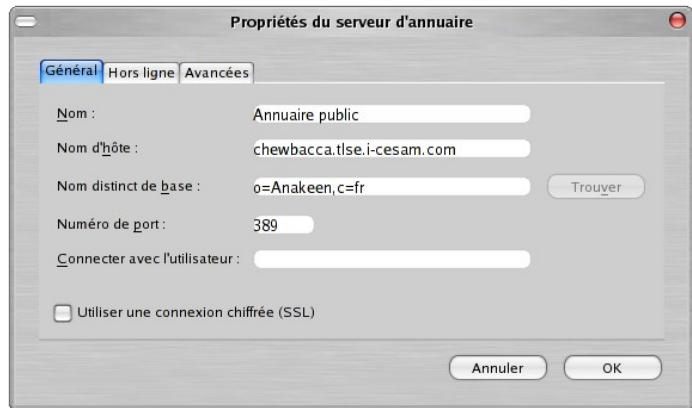
Le client de messagerie choisi pour l'exemple est le logiciel [thunderbird](#), mais le principe est le même pour Microsoft Outlook.

Sur thunderbird, la création d'un accès LDAP se fait via le carnet d'adresses. Une fois le carnet d'adresse ouvert, aller dans le menu Fichier/Nouveau/Annuaire LDAP. Ceci ouvre une fenêtre de dialogue afin de rentrer les coordonnées du serveur LDAP.

Pour ce connecter au serveur il faut au moins renseigner le nom, le nom d'hôte, et le chemin d'accès à la base LDAP.

Paramétrage annuaire public

Ce type de paramétrage ne demande pas d'authentification. Cela permet à toute personne d'accès aux informations des personnes et des utilisateurs.



Il suffit d'indiquer le nom de votre machine qui héberge le serveur LDAP et la racine de l'annuaire : "o=Anakeen,c=fr" est la valeur par défaut.

Paramétrage Annuaire public et privé



Pour accéder en plus à vos contacts privés il est nécessaire d'indiquer un login d'authentification LDAP. Ce login est constitué de la manière suivante :

"uid=<login>,dc=users,o=Anakeen,c=fr."

où "<login>" désigne le nom de connexion de l'utilisateur. Ce login LDAP est indiqué dans tous les documents décrivant les utilisateurs.



Paramétrage Annuaire privé seul

Le paramétrage est le même que le précédent au détail près que vous devez restreindre votre recherche à votre branche privée :

"ou=<login>,dc=people,o=Anakeen,c=fr"



1Pour les souris avec un seul bouton, vous pouvez utiliser la combinaison shift-click à la place.

2La touche « menu » de certains claviers fonctionne aussi si le focus est sur la fenêtre. Dans ce cas il faut ensuite avec les touches de tabulation se positionner sur l'item puis appuyer sur « entrée ».

3Le paramétrage LDAP client est expliqué dans le manuel du carnet d'adresses.

2.2.5 Connexions anonymes

Freedom supporte la connexion et la consultation en anonyme, ne nécessitant donc pas de s'authentifier et d'avoir un compte sur le système.

L'accès au mode anonyme se fait en accédant à la page "guest.php" (à la place du "index.php" par défaut) :

```
http://MON_SERVEUR/freedom/guest.php
```

La connexion à cette page s'effectue donc automatiquement sous le compte "Anonymous Guest".

La gestion de ce qu'il est possible de faire en mode anonyme, s'effectue de manière classique dans l'interface "Accessibilité" en modifiant les droits de cet utilisateur "Anonymous Guest" sur les différentes applications installées.

2.3 Sécurité

2.3.1 Les droits applicatifs

Les droits applicatifs doivent être enregistrés avec l'application ACCESS (Gestion des accessibilités). Ils permettent de définir des droits sur les actions applicatives. Ils ne s'appliquent pas aux documents mais seulement aux actions dans leur ensemble.

2.3.1.1 FDL - Bibliothèque FREEDOM

Nom	Définition
NORMAL	Autorise l'envoi des documents par mail. Permet l'aide à la saisie. À activer si GENERIC ou FREEDOM et activé.
EXPORT	Autorise l'exportation de dossiers ou de recherches. Doit être utilisé en corrélation avec FREEDOM_MASTER.
FAMILY	Autorise la modification des paramètres pour les documents famille. Doit être utilisé en corrélation

avec FREEDOM_MASTER.

2.3.1.2 GENERIC - Manipulation d'une famille

Nom	Définition
GENERIC_READ	Accès aux documents en lecture seule. Aucune fonction de modification n'est autorisée.
GENERIC	Accès à la modifications des documents.
GENERIC_MASTER	Accès à l'ajout de catégories et à l'import de document

2.3.1.3 ONEFAM - Manipulation groupement de familles

Nom	Définition
ONEFAM_READ	Accès à la liste de famille défini par l'administrateur en lecture seule
ONEFAM	Accès à une liste personnelle modifiable par l'utilisateur
GENERIC_MASTER	Accès en écriture à la liste administrative

2.3.1.4 FREEDOM - Gestion documentaire

Nom	Définition
FREEDOM_READ	Accès aux documents en lecture seule. Aucune fonction de modification n'est autorisée.
FREEDOM	Accès à la modifications des documents et des plans de classements
FREEDOM_MASTER	Accès aux fonctions d'import/export. Accès à la définition des attributs de familles
FREEDOM_GED	Accès à l'interface principale de Gestion documentaire

2.3.2 Profils de documents

L'administrateur a en charge de définir les profils de création associés aux familles.

Afin de spécifier de manière globale l'ensemble des profils, vous pourrez vous aider du fichier suivant :
FIXME internalmedia: freedom_2.14:admin:definition_droits_familles.ods

2.3.2.1 Définition des droits

Quatre familles de profils sont définies :

- profil de famille (droit de créer des documents ou de voir la famille)
- profil de document (droit de voir, modifier ou supprimer des documents)
- profil de dossier (droit de voir le contenu du dossier)

- profil de recherche (droit d'exécuter la recherche)

Les droits définis sont :

Nom interne	Description	Description longue	Pour
view	voir	Voir les caractéristiques du document, du dossier ou de la recherche. Le fait de ne pas voir un dossier n'implique pas de ne pas voir les documents contenus dans le dossier.	DFR
edit	éditer	Modifier les caractéristiques du document, du dossier. Les recherches ne sont pas modifiables.	DF
delete	supprimer	Supprimer le document, le dossier, la recherche; c'est à dire le mettre à la poubelle.	DFR
send	envoyer	Envoyer par email le document.	D
unlock	déverrouiller	déverrouiller le document.	DFR
viewacl	voir les droits	Voir les droits du document.	DFR
modifyacl	modifier les droits	Modifier les droits du document.	DFR
open	ouvrir	Ouvrir le dossier. Permet de voir le contenu du dossier	F
modify	modifier	Modifier le contenu du dossier. Permet d'ajouter ou de supprimer des documents dans le dossier.	F
execute	executer	Permet d'exécuter la recherche.	R
create	créer	Autorise la création de document de cette famille.	C
icreate	créer manuellement	Autorise la création de document de cette famille à partir de l'interface. Si ce droit n'est pas mis est que <i>create</i> est mis, l'utilisateur ne pourra créer le document que de manière indirecte (soit sur une transition, soit sur toutes autres actions particulière mis en place par l'administrateur). Sans ce droit les menus de création de cette famille sont inaccessibles. Si ce droit est mis il faut que le droit	C

		'create' soit aussi mis.	
confidential	voir document confidentiel	Permet d'utiliser normalement un document qui est confidentiel. (Confidentiel est une propriété de document).	DFR
forum	poster sur un forum	Autorise le post sur un forum de document (utilisateur n'ayant pas le droit d'édition).	DF

D pour document, F pour dossier (folder), R pour recherche et C pour Famille (classe)

2.3.2.2 Modification

Le moyen de création et de modification des profils est décrit dans le manuel utilisateur. Nous décrirons dans ce manuel la méthode de création de profil.

Le but du profil est de décrire un ensemble de droits pour les groupes de travail de l'entreprise. Les groupes de travail doivent représenter des groupes **fonctionnels** (groupe des achats, groupe commercial, groupe invité, groupe administration) et peuvent être déclarés de manière hiérarchique : le groupe direction commerciale est un sous groupe du groupe commercial.

Pour que les profils soient utilisées de manière optimum, il est conseillé de ne travailler les profils que sur les groupes et non sur les personnes. Les droits des personnes doivent être déduits de leurs groupes d'appartenances. Si les droits sont bien placés pour chaque groupe, il suffit de placer les personnes dans les groupes pour changer leur droits. Il n'est alors plus nécessaire de modifier les profils.

Pour placer les droits nous vous conseillons de remplir un tableau comme celui-ci pour chaque droit :

Droit Voir (view)				
Groupes/Familles	Fichier	Contrat	Facture	Personne
défaut				+
- administrateur		+	+	o
- commercial		+		o
-- direction commerciale		o	+	o
- achat			+	o
invité				+

Les cellules vertes (+) représentent les droits ajoutés, les cellules grises (o) représentent les droits déduits suivant la hiérarchie des groupes. Ici, on a décidé que la famille *fichier* n'avait pas de droit par défaut donc chaque nouveau document de la famille *fichier* est accessible de tout le monde. Le groupe *achat* a le droit de voir les factures et les personnes (du fait de l'appartenance au groupe *défaut*). Par contre il n'a pas le droit de voir les contrats. On peut aussi utiliser des cellules rouges (-) pour indiquer les restrictions de droits.

Une fois les tableaux remplis il suffit de créer un profil par famille avec les droits indiqués dans les cellules vertes (et rouges). Si des familles ont la même définition de profil, on peut associer le même profil à ces familles. Le but est d'avoir le moins de profil possible pour simplifier les modifications ultérieures.

Normalement pour changer les droits d'une personne, il suffit de changer ses groupes d'appartenances².

Bien sûr, il est toujours possible de modifier un profil pour une personne particulière si cette

personne ne correspond à aucun groupe.

2.3.2.3 Affecter manuellement un profil à un document

Pour affecter manuellement un profil à un document existant, il faut utiliser le menu :

- Autre / Sécurité / Changer de profil

2.3.2.4 Profil par défaut des nouveaux documents

Pour chaque famille, il est possible d'indiquer le profil à appliquer aux nouveaux documents de cette famille :

- Menu "Sécurité / Changer le profil pour les nouveaux documents"

2.3.2.5 Profil par défaut lors de l'enregistrement d'un document dans un dossier

Pour chaque document de type dossier, il est possible d'indiquer le profil à appliquer aux nouveaux documents ajoutés au dossier :

- Menu "Édition restriction"

2.3.2.6 Profil dynamique

Les profils dynamiques permettent de mettre des droits par rapport au contenu du document. Si un attribut est du type docid, et qu'il référence un document de la famille *Utilisateur* ou *Groupe d'utilisateurs* sa valeur peut être utilisée pour définir des droits sur le profil dynamique. Pour indiquer qu'un profil est dynamique, il suffit d'indiquer une famille dans le cadre dynamique lors de l'édition.

Attribut	Visible
Intégration	Visible
Infos Sauvegardes	Visible
id émetteur	Visible
service id	Visible
id client	Visible
id affaire	Visible
id correcteur	Visible

Dans l'exemple montré ci-dessus, on crée un profil pour la famille compte-rendu. Dans cette famille plusieurs attributs sont de type docid. Dans l'interface d'accessibilité ces attributs sont présentés en bas avec un bouton jaune pour les attributs qui ne sont pas dans un tableau; en orange pour ceux qui sont dans un tableau.

Intégration	Visible
Infos Sauvegardes	Visible
id émetteur	Visible
service id	Visible
id client	Visible
id affaire	Visible
id correcteur	Visible

Tous les attributs de type docid sont présentés. Cependant seuls ceux qui font référence à un utilisateur ou à un groupe d'utilisateurs auront une pertinence dans le calcul des droits. Dans l'exemple ci dessus, l'attribut « id affaire » ne doit pas être pris en compte pour l'élaboration du profil.

Si l'attribut de type docid est dans un tableau, il ne peut pas être pris en compte dans le profil dynamique. (version < 2.11.3) Depuis la version 2.11.3, le docid peut être dans un tableau. Les attributs visibles sont de type 'docid' et depuis la version 2.11.3 pensez à utiliser l'[option d'attribut isuser](#) pour ne pas prendre en compte l'attribut "id affaire" par exemple.

Si un profil n'a pas besoins de dynamique, il ne faut pas renseigner la famille dans le cadre dynamique car cela implique des calculs supplémentaires qui ne sont pas nécessaires.

2.3.2.7 Profil de transitions de cycle de vie

Les familles *cycles de vie* définissent une famille de profil spécifique. Les droits des documents *cycles de vie* permettent de savoir qui a le droit d'effectuer les transitions. Ces droits sont applicables à tous les documents associés à ce document de cycle de vie.

workflow incident	voir	éditer	supprimer	vers enregistré	vers qualifié	vers analysé	vers traité	vers rejeté	vers suspendu	vers clôt	voir les droits	modifier les droits
default group	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Groupe Achats	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Groupe Administrateur	<input checked="" type="checkbox"/>											
Développement	<input type="checkbox"/>											
Anakeen	<input type="checkbox"/>											
Groupe Commercial	<input type="checkbox"/>											
Groupe Direction	<input type="checkbox"/>											
Groupe Encadrement	<input type="checkbox"/>											
Groupe Gestion&financier	<input type="checkbox"/>											
Groupe Incident	<input type="checkbox"/>											

Les droits d'un cycle de vie comportent cinq droits classiques (voir, éditer, supprimer, voir les droits et modifier les droits), et N droits portant sur les transitions. Ces droits sur les transitions sont définis lors de la programmation du cycle de vie³.

Lorsque l'utilisateur effectue un changement d'état, il ne peut choisir l'état suivant que parmi ceux dont il a le droit. Dans l'exemple ci-dessus, le droit *vers qualifié* doit être activé si l'utilisateur veut changer d'état vers *qualifié*. Si ce droit n'est pas accordé, le changement d'état n'est pas proposé. Pour placer les droits sur une famille *cycles de vie*, il faut se rapporter à sa fiche de description. Chaque cycle de vie décrit ces droits de transition particuliers en fonction de son cycle de vie.

2.3.2.8 Profils d'état de cycle de vie

Les profils par défaut ne sont pas applicables aux familles *cycles de vie*. Dans les documents *cycles de vie* un profil par état peut être défini. Le profil par défaut peut être considéré comme le profil associé à l'état initial du document. On modifie les profils du document en éditant le document *cycle*.

À chaque changement d'état, le profil associé est affecté au document. Si un état n'a pas de profil associé, le document conserve son profil courant.

The screenshot shows the configuration interface for a 'CYCLE PAR DÉFAUT INCIDENT'. It includes sections for 'Basique' (Basic) and various state profiles:

- Basique:**
 - Titre:** cycle par défaut
 - Description:** (empty)
 - Famille:** incident
- Paramètres Pour L'état Enregistré:**
 - Profil enregistré:** profil recorder
 - Masque enregistré:** enregistrement incident
- Paramètres Pour L'état Qualifié:**
 - Profil qualifié:** profil qualified
 - Masque qualifié:** (empty)
- Paramètres Pour L'état Rejeté:**
 - Profil rejeté:** profil rejected
 - Masque rejeté:** (empty)

Si le cycle contient des spécificités par rapport à une famille, c'est à dire qu'il contient des actions particulières utilisables seulement par une famille, il est nécessaire de spécifier cette famille. Ainsi le cycle ne pourra être associé qu'avec cette famille ou ses descendants.

2.3.2.9 Administration des profils

! Version 2.14.3 Afin de réaliser la mise en place des droits les plus courants, le document famille dispose d'une interface permettant d'accéder directement à la définition du paramétrage des droits.

Famille sans cycle de vie

Pour une famille sans cycle de vie par défaut vous avez accès aux droits pour le document famille et pour les documents de la famille.

Si vous n'avez encore effectué aucun profilage, les liens "initialisé les droits" apparaîtront. Si vous cliquez sur le lien concernant la famille cela va créer un profil de famille avec aucun droits positionnés hormis les droits pour celui qui vient de la faire. Le lien est alors remplacé par "modifier les droits". Si vous cliquez dessus, l'interface de modification des droits apparaît et vous pouvez affecter les différents droits aux groupes et utilisateurs.

De même pour les droits sur les documents. L'initialisation va créer un document profil qui sera activé avec aucun droit sauf pour l'utilisateur qui a créé le profil. Vous cliquez ensuite sur "modifier le profil" pour placer les droits des groupes et utilisateurs.

Famille avec cycle de vie

Si la famille a un cycle de vie par défaut, les différents profils associés aux états apparaîtront. Le profil par défaut des documents n'apparaît pas car c'est le rôle du cycle d'imposer sa sécurité. Si vous voulez quand même mettre un profil par défaut, le menu "sécurité/changer le profil pour les nouveaux documents" est toujours accessible. Pour chacun des états vous pouvez "initialiser les droits" si aucun profil n'est déjà associé. Si le profil est déjà associé, vous pouvez alors le modifier ou le désactiver. La liste des états associés au profil est indiqué dans la liste (exemple "rédigé, publié"). Si vous voulez lier un profil à plusieurs état, il faudra passer par le cycle de vie pour le faire. Lorsqu'un profil est indiqué en rouge, cela signifie qu'aucun droits n'est positionné (tout le monde peut tout faire).

Cette interface ne donne pas accès à toutes les possibilités de mise en place de sécurités liées à une famille. Son rôle est de permettre à un administrateur d'accéder aux modifications ponctuelles de droits.

2.4 Importation et exportation de documents

2.4.1 Exportation

2.4.1.1 Exportation CSV

Le contenu d'un dossier ou le résultat d'une recherche peut être exporté dans un fichier au format CSV avec comme délimiteur le **point-virgule**.



L'export n'est possible que si l'administrateur a le droit EXPORT sur l'application FDL et l'acl FREEDOM_MASTER de FREEDOM.

Pour exporter, il faut cliquer sur le menu *outils/exportation du dossier* depuis la fenêtre d'ouverture de dossier (ou de recherche). L'accès à l'exportation est aussi disponible sur le menu contextuel affichable en cliquant sur les icônes de dossier ou de recherche présentes dans l'arborescence.

L'export produit un fichier CSV (séparé par des point-virgules) ou un fichier archive (Zip) si vous avez sélectionnez avec les fichiers. Explications des options :

option	signification
Encodage	Deux encodages sont proposés. Cela est nécessaire pour les valeurs textuelles qui comportent des accents. Si votre poste client est sur le système Windows et que vous voulez utiliser MicroSoft Excel, il est préférable de choisir Latin1. Sinon pour openOffice.org il est conseillé d'utiliser UTF-8. Lors d'une importation FREEDOM détecte automatiquement l'encodage et supporte correctement ces 2 encodages.
Profil	Cela est utile pour sauvegarder les profils associé à un document. Seuls les profils des documents avec profils dédiés (généralement les documents <i>profils</i>) sont exportés. Cette option est surtout utile lorsqu'un concepteur de famille veut récupérer son paramétrage profil.

Fichiers	L'option <i>sans les fichiers</i> indique que seul le fichier csv contenant les valeurs des attributs sera exporté. Les fichiers liés aux attributs de type <i>file</i> ou <i>image</i> ne sont pas exportés. L'option avec <i>les fichiers</i> indique que l'exportation générera un fichier archive de type <i>Zip</i> . Cette archive contiendra le fichier <i>fdl.csv</i> qui contient les valeurs des attributs et les fichiers attachées aux documents exportés. Cette archive peut être réimporté à l'importation d'archive de <i>FREEDOM</i> (Voir ci après). Pour des raisons de compatibilité avec les différents systèmes d'exploitations les noms des fichiers exportés ne comportent pas d'accents
Identificateur	Permet de sauvegarder les identificateurs numériques des documents. Cela implique que votre fichier ne sera ré-importable que dans la même base. Cela identifie de manière non ambiguë le document dans l'objectif d'une restauration. Dans tous les cas, si les documents comportent des noms logiques (propriété <i>name</i>), ces noms sont exportés.

Chaque ligne commençant par **DOC** présente les attributs visibles et invisibles (type *hidden*) d'un document. Les documents sont rangés par famille. À chaque changement de famille, une rangée indique les définitions des attributs de la famille (1ère colonne : **/FAM**).

L'ordre de présentation des attributs suit l'ordre défini par les attributs de la famille. Les quatre premières colonnes sont communes pour tous les documents.

- 1ère colonne : toujours égale à **DOC**
- 2ème colonne : identificateur de la famille
- 3ème colonne : identificateur du document
- 4ème colonne : identificateur du dossier (ou de la recherche) où a été extrait le document

Le document est exploitable sur tout tableur tel que Microsoft Excel, OpenOffice.org ou Gnumeric en spécifiant que le caractère séparateur est le **point-virgule**.

	A	B	C	D	E	F	G	H	I	J	K
1	//FAM	fichiers(FILE)	<specid>	<fldid>	titre	titre	sujet	mots-clés	résumé	autres	de
2	ORDER	FILE			fi_title	fi_titlew	fi_subject	fi_keyword	fi_description	ft_t_oformat	fi
3	DOC	FILE	-		Divers.tar.gz		71		les collines du Gers		
4	DOC	FILE	-		Collines.zip						
5	DOC	FILE	-		definition_droits_familles.sxc						
6	DOC	FILE	-		archivage.odg						
7	//FAM	fichier(SIMPLEFILE)	<specid>	<fldid>	nom du fichier	titre	sujet	mots-clés	résumé	type court	ty
8	ORDER	SIMPLEFILE			sfi_title	sfi_titlew	sfi_subject	sfi_keyword	sfi_description	sfi_mimetxtshort	sfi
9	DOC	SIMPLEFILE	-		C5 (copie).html	C					
10											
11											
12											
13											
14											

2.4.1.2 Exportation XML

Les documents, hormis les familles, peuvent être exportés en XML. Les fichiers XML produits suivent un schéma de famille imposé par *freedom*. Chaque famille de document a son propre schéma XML.

L'exportation XML d'un contenu de dossier ou de recherche peut se faire de deux façons : en générant un fichier par document ou avec un seul fichier xml.

Si vous voulez utiliser le fichier XML pour un exportation vers un autre contexte *freedom*, il faut indiquer l'option "sans les identificateurs", sinon les documents importés seront considérés comme des mise à jours.

Un fichier xml par document

Dans ce cas une archive (zip) est constituée avec un fichier par document. Chaque fichier est nommé à l'aide du titre et de l'identifiant du document (ex :alligator {4567}.xml). Chaque fichier fait référence à un schéma xml issus de la famille. Ce schéma est aussi présent dans l'archive

 Totor Alligator{1148}.xml	1,1 Kio	document XML
 Viel Ali Alligator{1289}.xml	1,3 Kio	document XML
 zoo_animal.xsd	21,3 Kio	W3C XML schema

Un fichier xml seul

dans ce cas, la sortie est un seul fichier xml contenant l'ensemble des documents. La balise racine est dans ce cas "documents". La balise documents porte la date de génération et l'auteur.

```
<documents date="2010-06-07T12:46:45"
    author="Default Master"
    name="recherche_animal">
<zoo_animal .> </zoo_animal>
<zoo_animal .> </zoo_animal>
</documents>
```

Avec cette option les Xml schema des familles ne sont pas importés.

Exemple d'un document exporté:

```
<zoo_animal xsi:noNamespaceSchemaLocation="zoo_animal.xsd" id="1147" name="" title="Alli Alligator" revision="0" modification-date="2010-06-02T13:53:33" version="" state="">
<an_tab1>
  <an_identification>
    <an_nom>Alli</an_nom>
    <an_descr xsi:nil="true"/>
    <an_tatouage xsi:nil="true"/>
    <an_espece_title>Alligator</an_espece_title>
    <an_espece id="1127">Alligator</an_espece>
    <an_ordre>Crocodiliens</an_ordre>
    <an_classe id="1123" name="Reptilia">Reptilia</an_classe>
    <an_sexe>M</an_sexe>
    <an_photo xsi:nil="true"/>
    <an_photo vid="27" mime="image/png" href="http://localhost:80/dev/?app=FDL&action=EXPORTFILE&cache=no&inline=no&vid=27&docid=1146&attrid=an_photo&index=-1" title="alligator.png"/>

    <an_naissance>2010-05-12</an_naissance>
    <an_entree>2010-05-18</an_entree>
    <an_pere id="1289">Viel Ali Alligator</an_pere>
  </an_identification>
</an_tab1>
</zoo_animal>
```

Exportation xml avec fichier

Si vous cochez l'option avec les fichiers ceux-ci seront importés "inline" dans les fichiers xml. La balise de l'attribut contiendra dans ce cas le fichier encodé en base 64.

```
<an_photo vid="27" mime="image/png" title="alligator.png">
```

```
iVBORw0KGgoAAAANSUhEUgAAAQAAAAEACAYAAABccqhmAAAACXBWIWXMAAAsTAAALEwEAmpwYAAAABG
dBTUEALG0fPtRkwAACBjSFJNAAB6JQAAgIMAAPn/AACA6QAAdTAAA0pgAAA6mAAF2+SX8VGAAEo
A01EQVR42mL8//8/wwgE==</an_photo>
```

Récupération des Xml schema seuls

Pour récupérer un fichier Xml schema de manière autonome, vous devez utiliser l'interface d'importation de document via l'application "gestion documentaire". Sur cette interface un bouton "Xml Schema" permet de récupérer le schéma compatible avec les documents XML exportés.

Exportation Xml par ligne de commande

Vous pouvez lancer l'export avec wsh. La commande est :

```
./wsh.php --app=FDL --action=EXPORTXMLFLD --id=<folderid> --eformat=[X|Y]
--wfile=[N|X]
```

export Xml sans fichier

```
./wsh.php --app=FDL --action=EXPORTXMLFLD --id=1180 --eformat=Y > t.xml
```

export Xml avec fichier

```
./wsh.php --app=FDL --action=EXPORTXMLFLD --id=1180 --eformat=Y --wfile=Y>
t.xml
```

export Zip sans fichier

```
./wsh.php --app=FDL --action=EXPORTXMLFLD --id=1180 --eformat=X > t.zip
```

export Zip avec fichier

```
./wsh.php --app=FDL --action=EXPORTXMLFLD --id=1180 --eformat=X --wfile=Y>
t.zip
```

2.4.2 Importation

2.4.2.1 Importation CSV/ODS

Constitution du fichier d'importation

Le fichier d'import peut être au format :

- CSV (**séparateur point-virgule** / pas virgule) avec un encodage latin9 ou utf-8
- OpenDocument Spreadsheet (tableur OpenOffice.org .ods)

Généralités

Comme pour l'export les quatre premières colonnes définissent les paramètres du document :

- 1ère colonne : toujours DOC
- 2ème colonne : identifiant de la famille
- 3ème colonne : identifiant du document. Utiliser un identifiant existant indique une modification du document existant. Un identifiant vide ou égal à zéro indique une création de document.
- 4ème colonne : identifiant du dossier où insérer le document. Si un identificateur est précisé le document est référencé dans le dossier (s'il n'y est pas déjà). Si l'identificateur n'est pas précisé (vide ou zéro), le document est référencé dans le dossier courant (celui qui est sélectionné dans l'arborescence). Si l'identifiant est '-' alors il ne sera référencé dans aucun dossier.

Les colonnes suivantes décrivent les valeurs des attributs du document.

Ordre des attributs



ORDER

Clés d'imports



KEYS

Importation générale de documents sans fichier



L'importation de document utilise un fichier de même format que pour l'exportation. **Par contre, certains attributs présents dans le fichiers d'exportation peuvent ne pas être présents dans le fichier d'importation.** Pour réutiliser un fichier issu de l'exportation, il faudra bien s'assurer que les colonnes soient les mêmes (ordre et définition) que celles demandées.



On ne peut pas importer des attributs qui sont calculés en fonction des données de la base : ils seraient systématiquement écrasé par le nouveau calcul.

Importer des documents

Description Du Format D'import

pour **de base** **Import spécifique**

DOC;BASE;<special id>;<special dirid> ; titre

DOC	BASE	<special id>	<special dirid>	titre
-----	------	--------------	-----------------	-------

Fichier à importer : **Parcourir...** **Analyse**

politique si même titre : **Mets à jour le document si permission accordée sinon modification ignorée**

Import Direct

Importer la description d'un document

Import En Tache De Fond

rapport à (email) **albert.durant@somewhere.org**

Lancer l'import en tache de fond

Pour importer un fichier décrivant des documents, il faut sélectionner *import de documents* dans la barre de menu Outils. Afin de voir les formats d'importation des familles il suffit de sélectionner la famille souhaitée.

L'import de fichiers associés aux attributs de type *file* et *image* n'est pas possible dans ce cas.

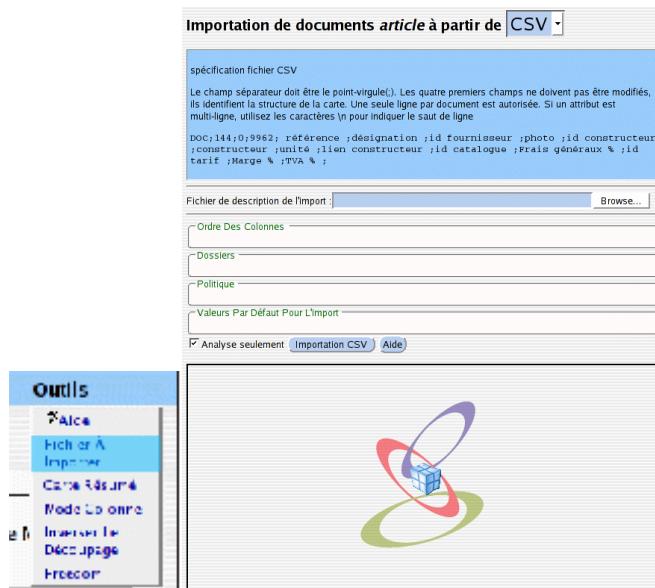
Avant l'import, il est nécessaire d'analyser le fichier d'import pour être sûr de ce qu'on importe. L'analyse indique le titre du document importé si le document est un nouveau ou une mise à jour et dans quel dossier il va être placé.

S'il y a au moins un document, les boutons d'importations sont activés. Si le nombre de documents est supérieur au paramètre FDL_MAX_FGEXPORTDOC, l'importation directe reste désactivée seul l'importation en tâche de fond est possible.

Les nouveaux documents (pas d'identificateur spécifié) qui ont le même titre qu'un document existant de la même famille sont ignorés par défaut. Ils peuvent être ajouté si on change la politique d'import. Dans le cas '*Ajout nouveau document*', un nouveau document avec le même titre qu'un document existant sera créé.

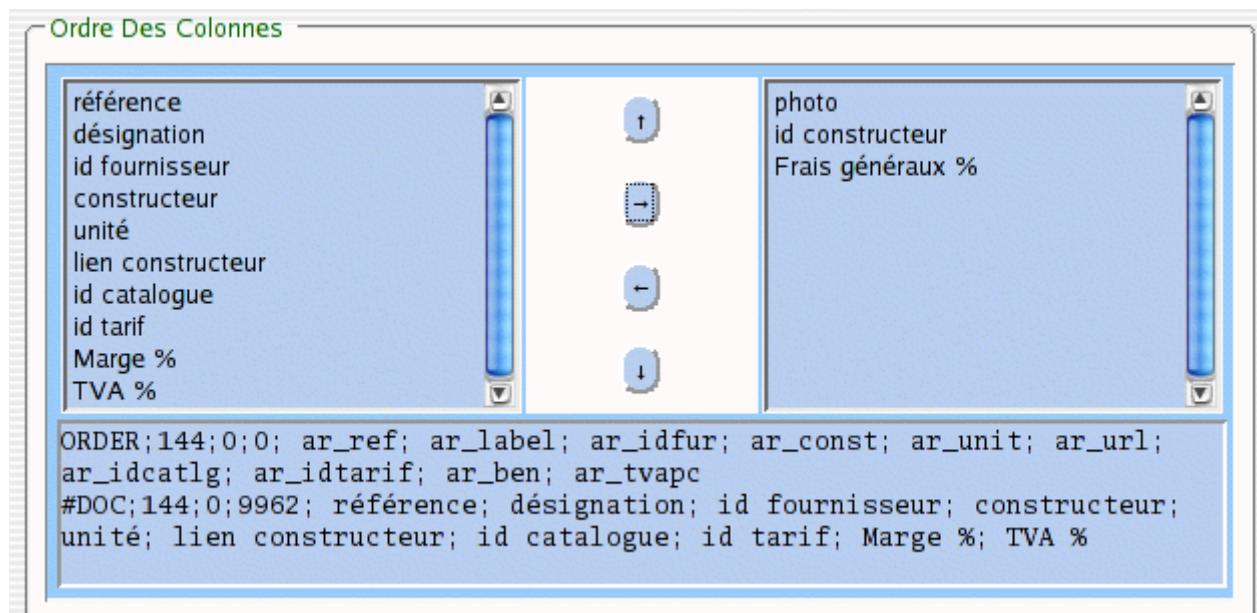
Dans le cas de l'importation en tâche de fond, le rapport d'importation sera envoyé au mail spécifié (par défaut celui de l'utilisateur) une fois l'import effectué.

Importation spécifique de documents sans fichier

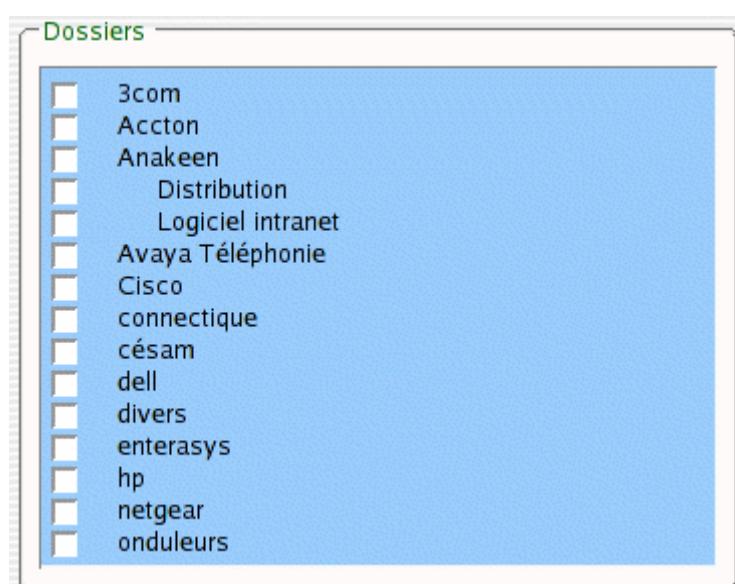


Plus d'options d'importation sont disponibles avec ce type d'importation. Par contre la construction du fichier CSV reste la même. Cette importation est disponible via l'application '*Gestion par famille*', dans le menu '*outils*' choisir '*fichier à importer*' ou par le bouton '*import spécifique*' disponible sur l'interface d'importation générale.

Cette importation peut être utilisée pour l'importation d'un ensemble de documents de la même famille. Il faut auparavant sélectionner la famille que l'on veut importer avant de sélectionner '*fichier à importer*'.



Avec ce type d'importation, l'utilisateur peut choisir l'ordre des colonnes composant le CSV en cliquant sur '*ordre des colonnes*'. Les quatre premières colonnes (DOC;<id famille>;<id document>; <id dossier> restent immuables. Par défaut l'ordre des colonnes est celui défini par la famille. Vous pouvez soit supprimer des colonnes, soit modifier leur ordre. Pour supprimer il faut sélectionner la (les) colonne(s) et les déplacer dans le cadre de droite (bouton \$-3\$). Pour changer l'ordre des colonnes, il faut utiliser les boutons \$1\$ et \$2\$. L'ordre et la définition des colonnes utilisés sera celui du cadre de gauche. Lorsque vous modifiez cet ordre, l'écriture des colonnes en bas est aussitôt modifié. Vous pouvez utiliser ces deux lignes inscrites dans le cadre pour votre fichier CSV afin d'indiquer l'ordre de vos colonnes. La première ligne 'ORDER' indique au logiciel le nouvel ordre des attributs d'importation établi pour cette famille. S'il n'y a pas de ligne ORDER dans le fichier, l'ordre sera celui défini dans le cadre gauche. Dans le cas contraire ce sera l'ordre défini dans le fichier qui sera pris en compte. Les quatre premières colonnes d'une ligne ORDER sont composées du mot-clef 'ORDER', de l'identificateur de la famille puis de deux colonnes non utilisées (servant seulement pour l'alignement avec les lignes DOC). Les colonnes suivantes contiennent les identifiant des attributs de la famille. La deuxième ligne inscrite dans le cadre du bas ('#DOC') sert juste à aider l'utilisateur sur la signification des colonnes.



Le cadre dossier indique tous les dossiers fils et petit-fils du dossier par défaut défini par la famille. Lors d'un importation, vous pouvez choisir l'emplacement des documents importé dans un ou plusieurs des dossiers proposés.

Le cadre 'Politique' permet de choisir la politique d'importation lorsqu'un document à importer possède des similitudes avec un document existant. La similitude est détectée par défaut si un

document d'une même famille possède le même titre. L'utilisateur peut modifier les clefs de recherche de similitude en utilisant un ou deux attributs de la famille (exemple : pour un article manufacturé on peut définir que l'on a affaire à un même document si la référence constructeur et le constructeur sont identiques).

Politique

politique si même clef :	Mets à jour le document si permission accordée sinon modification ignorée
clefs d'import :	titre
	pas de clef

La cadre 'valeurs par défaut' permet d'affecter des valeurs à toutes valeurs nulles (vides) définies dans le fichier d'importation. Ces valeurs par défaut sont uniquement utilisées pour des nouveaux documents, pas pour les mise à jours.

Valeurs Par Défaut Pour L'import

Identification
Référence :
Désignation :
Photo :
Constructeur :
Unité :
Lien constructeur :
Coûts
Frais généraux % :
Marge % :
TVA % :

La case 'Analyse seulement' est cochée par défaut. Si vous appuyez sur 'Importation CSV' alors l'analyse du fichier sera effectuée et le résultat sera affiché dans le cadre du bas.

Résultat de l'analyse de l'import

10 documents à prendre en compte.

ligne	titre	dossier	id	famille	action	message	modifications	erreur
2	CMBPA-10K Powerware	/onduleurs	22113		mis à jour	à mettre à jour CMBPA-10K Powerware [22113]	<ul style="list-style-type: none"> [référence:CMBPA-10K] [désignation:coffret métal by-pass automatique 10KVA pour commutation automatique de la charge sur le réseau en cas de panne onduleur et sans coupure] [id fournisseur:15894] [id constructeur:19631] [constructeur:Powerware] 	
3	BRDA02 Powerware	/onduleurs			ajouté	BRDA02 Powerware à ajouter	<ul style="list-style-type: none"> [référence:BRDA02] [désignation:Boîtier report d'alarme onduleur - pose à l'extérieur du local technique pour visualiser les alarmes] [id fournisseur:15894] [id constructeur:19631] 	

Les ajouts sont notifiés en vert, les modifications en jaune et les document ignorés en rouge.

Le résultat de l'analyse indique le nombre de documents à prendre en compte (ajouté ou modifié). Il indique pour chaque ligne du fichier, l'interprétation effectuée et l'action qu'il entreprendra. Si l'analyse est conforme aux attendus vous pouvez décocher 'analyse seulement' et lancer réellement l'importation. Le résultat apparaîtra dans le cadre du bas à la place de l'analyse.

Importation d'arborescence



L'importation d'une arborescence de fichiers peut être effectuée avec l'importation d'archive. L'archive contient une arborescence de répertoires contenant des fichiers. L'arborescence de répertoires sera rattachée au plan de classement en créant une arborescence de dossiers. Ensuite les fichiers ou les descriptions de documents seront insérés dans ces dossiers.

Cette archive permet :

- de créer un plan de classement
- d'ajouter des documents contenant des fichiers avec des caractéristiques
- d'insérer des fichiers sans caractéristique

Constitution de l'archive

Vous devez créer sur votre machine locale votre arborescence de répertoires. Dans chacun de vos répertoires vous placez les fichiers à importer. Ces fichiers seront par défaut ajoutés dans le dossier correspondant à votre plan de classement.

Dans chaque répertoire, un fichier de description nommé *fdl.csv* doit être créé pour indiquer les caractéristiques des fichiers. Si ce fichier n'est pas présent la conversion par défaut sera effectuée et aucune caractéristique ne sera présente dans les documents contenant les fichiers du répertoire. Ce fichier de description est tel qu'il est décrit dans le paragraphe précédent. Pour les attributs de type *fichiers*, il suffit d'indiquer le chemin relatif vers le fichier.

FAM	image(IMAGE)	<specid>	<fldid>	titre	image	description
ORDER	IMAGE			img_title	img_file	img_descripti on
DOC	IMAGE		-	place principale	photos/Carna val 005.jpg	vue du balcon\avec les bandas
DOC	IMAGE		-	animations	photos/Carna val 006.jpg	

Dans l'exemple ci-dessus, le fichier *fdl.csv* indique la création de trois documents en attachant trois fichiers de l'archives. Le répertoire courant contient ici un sous répertoire *photos* contenant les fichiers jpeg.

Pour chaque fichier de description, le mot-clé DFAMID indique la famille de conversion par défaut pour les fichiers non référencés. Ces fichiers sont ceux qui n'ont pas été spécifiés dans un des fichiers de description de l'archive. S'il n'y a pas de redéfinition de DFAMID dans les sous répertoires, la famille par défaut définie par le répertoire père sera réutilisé. Le fichier de description décrit le document du dossier courant. Il ne peut pas décrire des documents d'autres dossiers. Les colonnes *<special id>* et *<special dirid>* doivent être vides dans ce cas pour l'ajout de document.

De même que pour la famille des fichiers par défaut, le mot-clé DFLDID indique la famille du container par défaut pour les répertoires. Par défaut DFLDID est l'identifiant de la famille dossier. Il faut obligatoirement que cet identificateur soit une famille dérivée de dossier.

Une fois l'arborescence, les fichiers et les fichiers de description remplis, il suffit de créer une archive sur votre poste local. Le format de fichier de l'archive peut être un tar compressé (tar.gz, tgz, tar.bz2). La compression du tar peut être effectuée soit par gzip (par défaut), soit par bzip2. Ce format peut aussi être un Zip (répertoire compressé par défaut sous Windows).

Voici un exemple très simple d'une archive contenant 2 images :
:FIXME internalmedia: freedom_2.14:admin:mes_images.zip

Téléchargement de l'archive

Importer une archive

Import D'archives

Formats supportés:

- Tar compressé (.tgz .tar.gz .tar.bz2)
- Fichier Zip (.zip)

Importer un fichier (max 50M bytes)

Voir Les Archives

Une fois l'archive créé, il faut la transférer sur le serveur documentaire. Si l'archive est assez petite en taille, un transfert classique peut être effectué. La taille maximum d'importation dépend de la configuration de PHP ([Upload configuration](#)) au niveau du serveur. Une modification de ce paramètre implique un redémarrage de Apache.

Si l'archive est trop volumineuse, un transfert plus classique (scp, ftp) peut être effectué par l'administrateur. Il faut alors la placer sous le répertoire suivant du serveur : <FREEDOM_UPLOADDIR>/<login>/tars

Où FREEDOM_UPLOADDIR est le paramètre applicatif indiquant le répertoire de stockage (par défaut /tmp/upload), login est le nom de connexion de l'utilisateur. Par exemple pour jean.martin le répertoire de stockage des archives sera /tmp/upload/jean.martin/tars/.

Une fois l'archive téléchargée, il suffit d'aller aux archives pour voir le résultat de l'extraction.

Les fichiers commençant par un point sont ignorés.

Analyse de l'archive

Importer une archive

Choisir L'archive

pour

- carnaval
 - carnaval.html
 - fdl.csv
 - fdl.gnumeric
- photos
 - 04-02-21 - Carnaval Saint Sébastien 001.jpg
 - 04-02-21 - Carnaval Saint Sébastien 002.jpg
 - 04-02-21 - Carnaval Saint Sébastien 003.jpg
 - 04-02-21 - Carnaval Saint Sébastien 004.jpg
 - 04-02-21 - Carnaval Saint Sébastien 005.jpg
 - 04-02-21 - Carnaval Saint Sébastien 006.jpg

Le résultat de l'extraction est visible dans la première partie de la fenêtre. S'il y a plusieurs archives on peut sélectionner celle que l'on souhaite. Seules les archives téléchargées par l'utilisateur sont présentées ici. Ces archives sont dans un espace temporaire et elles peuvent être supprimées.

La seconde partie montre ce qui sera effectué lors de l'importation dans la base documentaire.

Trois options d'importation sont proposées:

Dans tous les cas les dossiers sont créés conformément à l'arborescence de fichiers.

- Seul fichier CSV : si coché ne traite que des documents qui sont décrits dans les fichiers *fdl.csv*. Sinon traite d'abord les documents issus des fichiers de description puis des fichiers non référencés.

- Famille fichier par défaut : famille utilisée pour les fichiers non référencés dans le cas où DFAMID n'est pas renseigné
- Famille dossier par défaut : famille utilisée pour les répertoires dans le cas où DFLDID n'est pas renseigné
- Dossier racine : ajoute un dossier pour contenir l'ensemble de l'importation de l'archive. Le nom du dossier est celui de l'archive. Il peut être modifié après l'importation.

À chaque modification d'options, il faut relancer l'analyse pour effectuer l'importation.

Analyse de l'archive

Options D'importation

famille fichier par défaut **fichier**

famille répertoire par défaut **dossier**

Seul fichier CSV

Ajouter un dossier racine

Analyse

Lancer L'import

rapport à (email)

Effectuer l'import en tâche de fond de 30 documents

Résultat De L'analyse					
dossier		fichier	famille	action	err
Eric Brison/		carnaval			
Eric Brison/carnaval		photos/04-02-21 - Carnaval Saint Sebastien 004.jpg	image	ajouté	
Eric Brison/carnaval		photos/04-02-21 - Carnaval Saint Sebastien 006.jpg	image	ajouté	
Eric Brison/carnaval		carnaval.html	fichier	ajouté	
Eric Brison/carnaval		carnaval.html	fichier	à ajouter	
Eric Brison/carnaval		fdl.csv	fichier	à ajouter	
Eric Brison/carnaval		fdl.gnumeric	fichier	à ajouter	
Eric Brison/carnaval		photos			
Eric Brison/carnaval/photos		04-02-21 - Carnaval Saint Sebastien 005.jpg	dossier	à ajouter	
			image	ajouté	

Importation de l'archive

Une fois que l'analyse est conforme à vos attentes il suffit de lancer l'importation en tâche de fond. Un email vous sera envoyé avec le résultat de l'analyse dès que l'importation sera finie.

Après importation, vous pouvez alors supprimer l'archive afin d'éviter une deuxième importation non souhaitée.

Importation de profils

Création de profil



PROFIL

Affectation de profil



PROFIL

Définitions des accessibilités

 ACCESS

Divers

Affectation d'icone

 DOCICON

Balise applicative

 ATAG

Importation de documents famille

Voir chapitre [Importation famille](#)

Importation de table d'échanges FREEDOM -> LDAP

 LDAPMAP

Importation de masse

Pour les importations importantes de document sans fichier, vous pouvez accélérer votre importation en utilisant le script wsh csv2sql. Ce script transforme un fichier CSV (avec ';' comme séparateur) (ou ODS) contenant des DOC en commande sql prêtes à l'insertion en base de données.

Ce script permet une importation brute. C'est à dire pas d'appel aux méthodes Doc::PostModify ou Doc::postCreate, pas de vérification d'unicité.

```
# . /etc/freedom.conf
# wsh --api=csv2sql --file=/var/tmp/test.csv | psql freedom anakeen
```

Il est bien sûr fortement conseillé de bien vérifier la sortie du script avant l'injection.

Dans les fichiers CSV, il faut obligatoirement que les DOC soient précédés d'une ligne ORDER indiquant la correspondance entre les colonnes et les attributs.

2.4.2.2 [Importation XML](#)

L'importation de document peut être faite à l'aide de fichier XML tels que ceux produits par l'exportation XML. L'interface graphique pour importer est la même que pour les fichiers CSV et ODS. On y accède depuis la gestion documentaire (menu "outils/importer des documents").

Le type de fichier accepté est soit un fichiers XML avec balise root "<documents>", soit un fichier zip contenant un ensemble de fichier XML importés. Ces fichiers XML doivent être conformes au schéma XML associé à la famille de document.

Si vous importez un XML contenant des attributs relations et que dans votre XML vous n'indiquez pas explicitement un id ou un name alors une recherche par titre sera faite afin de trouver automatiquement le document qui a le même titre en cohérence avec la famille déclarée dans la relation au niveau de la famille. Si aucun id n'est trouvé la valeur de l'attribut sera vide.

```
<an_espece>Alligator</an_espece>
```

S'il y a un attribut 'id' ou 'name' c'est celui ci qui sera utilisé pour effectuer la relation ; dans ce

cas le nom (contenu de la balise) est ignoré.

```
<an_espece id="4563">Alligator</an_espece>
<an_classe name="Reptilia">Reptile</an_classe>
```

Les fichiers encodés contenus dans le XML sont insérés dans le document et remplace le fichier original.

Test de validité des fichiers XML

Les fichiers XML peuvent être validés à l'aide du schéma avant importation.

Cette vérification est à faire manuellement à l'aide du schéma correspondant. Lorsque vous importer une recherche sur une famille spécifique et que vous choisissez l'option "import XML avec Zip" alors cette archive contient le schéma XML. Vous pouvez utiliser le logiciel xmllint pour vérifier vos fichiers XML

```
eric@tarfful:~/Bureau$ ls -l Recherche_totor.zip
-rw-r--r-- 1 dev dev 189675 2010-06-16 13:39 Recherche_totor.zip
dev@tarfful:~/Bureau$ mkdir test
dev@tarfful:~/Bureau$ cd test/
dev@tarfful:~/Bureau/test$ unzip ../Recherche_totor.zip
Archive: ../Recherche_totor.zip
  inflating: Totor Alligator{1148}.xml
  inflating: Totor junior 1 Alligator{3307}.xml
  inflating: Totor junior 2 Alligator{3308}.xml
  inflating: Totor junior 3 Alligator{3309}.xml
  inflating: Totor junior 4 Alligator{3310}.xml
  inflating: zoo_animal.xsd
dev@tarfful:~/Bureau/test$ xmllint --noout --schema zoo_animal.xsd *xml
Totor Alligator{1148}.xml validates
Totor junior 1 Alligator{3307}.xml validates
Totor junior 2 Alligator{3308}.xml validates
Totor junior 3 Alligator{3309}.xml validates
Totor junior 4 Alligator{3310}.xml validates
```

2.4.2.3 Importation en ligne de commande

La commande wsh freedom_import permet d'importer des documents en ligne de commande. Les types CVS, ODS, XML sont supportés par le scripts :

```
./wsh.php --api=freedom_import --file=/home/dev/Bureau/recherche_animal.xml
```

Si vous voulez un compte rendu et une pré-analyse :

```
./wsh.php --api=freedom_import --analyze=Y --htmlmode=Y
--file=/home/eric/Bureau/recherche_animal.xml > report.html
```

Vous pouvez aussi envoyer le rapport par mail

```
./wsh.php --api=freedom_import --to=somone@somewhere.org
--file=/home/eric/Bureau/recherche_animal.xml
```

2.5 Archives

2.5.1 Introduction

freedom permet d'archiver des documents afin qu'ils ne soient plus vus comme des documents "vivants".

Un document archivé n'est plus accessible par les recherches classiques.

Un document archivé a un profil donné par l'administrateur de l'archive

Les documents archivés ne sont plus vus dans les recherches par défaut comme pour les documents mis à la poubelle.

2.5.2 Constitution de l'archive

Pour construire une archive, il est nécessaire de créer un document "archive". Le droit de créer des archives est donné par défaut au groupe "administrateurs". Les utilisateurs par défaut ne peuvent ni créer ni voir les documents archives.

Les seules données à remplir pour créer une archive sont de renseigner son nom et sa description.



Une fois créée, l'archive a le statut "Ouvert". Cela signifie que l'administrateur peut ajouter ou enlever des documents dans l'archive. L'archive étant un dossier, tout les moyens d'insertions classiques de document dans un dossier peuvent être employés pour constituer l'archive. Cependant pour faciliter la constitution de l'archive des menus supplémentaires ont été ajoutés aux documents et aux contenus de collections.

Si vous avez les droits d'insertions (droit "modifier" et "voir") sur les archives ouvertes, alors les boutons "insérer dans l'archive <xxx>" apparaissent sur les documents. Si le document est une collection le menu "tout insérer dans l'archive <xxx>" apparaît en plus. Dans la gestion documentaire, sur les résultats de recherche un menu "tout insérer dans l'archive <xxx>" apparaît dans le menu "outils".

recherche Compte rendu - 37 documents

	Affichage	Outils
Objet 1/CR1 cr_init		Propriétés Tout insérer dans le porte-documents Ouvrir dans une chemise Créer un traitement Exportation du dossier Tout insérer dans l'archive Compte rendu de 2009
Objet 10/CR10 cr_init		31/05/10 0
Objet 11/CR11 cr_init		31/05/10 0
Objet 12/CR12 cr_init		31/05/10 0
Objet 13/CR13 cr_init		31/05/10 0
Objet 14/CR14 cr_init		31/05/10 0
Objet 15/CR15 cr_init		31/05/10 0
Objet 16/CR16 cr_init		31/05/10 0
Objet 17/CR17 cr_init		31/05/10 0
Objet 18/CR18 cr_init		31/05/10 0
Objet 19/CR19 cr_init		31/05/10 0
Objet 2/CR2 cr_init		31/05/10 0
Objet 20/CR20 cr_init		31/05/10 0

recherche détaillée

CRÉATION RECHERCHE DÉTAILLÉE

[Créer](#) | [Annuler](#)

famille :

Conditions

		révision : <input type="text" value="courante"/>		
		Document : <input type="checkbox"/> Modifiable <input type="checkbox"/> Supprimable		
		Sans sous famille : <input type="checkbox"/> Avec les sous familles		
		<input checked="" type="radio"/> satisfait toutes les conditions <input type="radio"/> satisfait au moins une condition		
Opérateur	Parenthèse gauche	attributs	fonctions	mot-clefs
global	<input type="button" value=""/>	<input type="text" value="date de rédaction"/>	>	<input type="text" value="01/01/2009"/>
global	<input type="button" value=""/>	<input type="text" value="date de rédaction"/>	<	<input type="text" value="01/01/2010"/>
attributs	Etat			

Lancer la recherche

Les documents archivés ne peuvent pas être archivés par une autre archive.

Pour voir les documents archivés, il faut aller sur l'archive et cliquer sur le menu "ouvrir". Si le nombre de document archivé est supérieur à 1000 ou si vous recherchez un document précis dans une archive en cours de constitution vous pouvez cliquer sur "voir les documents". Cela permet d'accéder au filtre sur les valeurs des documents.

2.5.3 Archivage des documents

Une fois que l'archive est constituée vous pouvez archiver son contenu. Cela va consister à marquer les documents comme "archivé" et à leur mettre le profil applicable de l'archive. Chaque archive à un profil applicable différent. Vous pouvez à tout moment modifier ce profil : pendant la constitution ou après. Si vous ne voulez pas que les utilisateurs puissent modifier les archives il faut mettre ce profil en lecture seule. Si vous ne voulez pas qu'ils y accèdent vous ne leur donnez pas le droit de voir tout simplement. Le profil applicable par défaut ne donne aucun droit à personne à part au créateur de l'archive. freedom n'interdit pas que l'on puisse modifier des documents archivés si vous donnez le droit "éditer" sur le profil.

Une fois l'archive close, vous ne pouvez plus ajouter ou enlever des documents de cette archive.

Si vous affichez un document archivé l'icône indiquera cet état.

Document	Date	Archivage
Objet 1/CR1 cr_init	31/05/10	
Objet 10/CR10 cr_init	31/05/10	
Objet 11/CR11 cr_init	31/05/10	
Objet 12/CR12 cr_init	31/05/10	
Objet 13/CR13 cr_init	31/05/10	
Objet 14/CR14 cr_init	31/05/10	
Objet 15/CR15 cr_init	31/05/10	

Si vous êtes précipité lors de l'archivage, il est possible de revenir en arrière en désarchivant l'archive. Dans ce cas l'ensemble des documents archivés par cette archive seront restaurés comme auparavant. Leurs profils initiaux seront restaurés. L'archivage comme le désarchivage sont des procédures qui peuvent être longues car elles dépendent du nombre de documents à archiver. Il

est donc conseillé de réaliser ces manipulations lorsque peu de personnes sont connectées au risque de pénaliser les performances. Chaque archivage est notifié dans chacun des historiques des documents impactés.

Vous pouvez rechercher dans les documents archivés si vous avez le droit de voir le document archive et si vous avez le droit de voir les documents archivés. Si vous avez ces droits, l'option "Dans l'archive" de la recherche simple vous permettra de trouver des documents archivés. Vous pouvez aussi à partir de menu "voir les documents archivés" de l'archive accéder à la recherche sur son contenu.

recherche

CRÉATION RECHERCHE

[Créer](#) | [Annuler](#)

mot-clé : <input type="text"/>	Révision : <input type="text" value="courante"/>
famille : <input type="text" value="toute famille"/>	À partir du dossier courant : <input type="checkbox"/>
	Mode : <input type="text" value="expression sans casse"/>
	Sans sous famille : <input type="text" value="Avec les sous familles"/>
	Document : <input type="checkbox"/> Modifiable <input type="checkbox"/> Supprimable
	Dans la poubelle : <input type="text" value="non"/>
	Dans l'archive : <input type="text" value="Compte rendu de 2009"/> Les animaux
basique	Titre : <input type="text"/>
	Couleur intercalaire : <input type="color"/>

2.5.4 Destruction des documents archivés

Une fois l'archive close vous pouvez détruire son contenu. Cette étape permet de supprimer physiquement les documents.

Cette destruction est irréversible.

Si vous activez cette ultime étape, chacun des documents sera supprimé de la base de données. L'archive gardera les titres et références de l'ensemble des documents détruits.

Archive

COMPTE RENDU DE 2009

[Éditer](#) | [Édition Restriction](#) | [Supprimer](#) | [Historique](#) | [Ajouter un post-it](#)
[Ouvrir](#) | [Archivage↓](#) | [Autres↓](#)

Basique

Description : Tous les comptes-rendus rédigés en 2009
Statut : Purgé
Date de clôture : 31/05/2010
Profil applicable : [Profil pour l'archive Compte rendu de 2009](#)

Purge

Date de destruction : 31/05/2010
Documents détruits :

- 1. [Objet 10/CR10](#)
- 2. [Objet 11/CR11](#)
- 3. [Objet 12/CR12](#)
- 4. [Objet 13/CR13](#)
- 5. [Objet 14/CR14](#)
- 6. [Objet 15/CR15](#)

Si vous cliquez sur un lien référençant un document détruit vous pourrez accéder à ses dernières traces conservées dans les log et l'historiques.

Dernières traces du document 1266 : Objet 17/CR17

Derniers logs

Date	Utilisateur	Message	Code
31/05/2010 12:54:51.317445	Default Master		delete
31/05/2010 12:54:36.242389	Default Master	Archiving into Compte rendu de 2009	archive
31/05/2010 10:35:23.532465	Default Master	Unarchiving from Compte rendu de 2009	unarchive
31/05/2010 10:25:54.140537	Default Master	Archiving into Compte rendu de 2009	archive
31/05/2010 10:17:17.661795	Default Master		create

Derniers historiques

Date	Utilisateur	Message
31/05/2010 12:54:51	Default Master	Détruit par la purge de l'archive Compte rendu de 2009
31/05/2010 12:54:36	Default Master	Archiving into Compte rendu de 2009
31/05/2010 10:35:23	Default Master	Unarchiving from Compte rendu de 2009
31/05/2010 10:25:54	Default Master	Archiving into Compte rendu de 2009
31/05/2010 10:17:17	Default Master	Création du document
31/05/2010 10:17:17	Default Master	Document inséré dans le dossier Default Master
31/05/2010 10:17:17	Default Master	mise à jour par import

2.6 Fonctions Wsh

Les fonctions bash sont activables via le programme "wsh.php". Elles doivent être exécutées dans une fenêtre shell sur la machine serveur.

2.6.1 Comment exécuter une fonction WSH

Pour exécuter une fonction WSH, il y a deux méthodes :

1 - Indiquer le chemin complet de la fonction. Exemple :

```
<wiff_context_root>/wsh.php --listapi
```

Le <wiff_context_root> est le répertoire d'installation de freedom.

2 - Utiliser un alias wsh. Exemple :

```
alias wsh=<wiff_context_root>/wsh.php
wsh --listapi
```

Remarque : Il est possible de charger ce fichier de configuration automatiquement au démarrage en ajoutant ces lignes au fichier "~/.bashrc":

```
alias wsh=<wiff_context_root>/wsh.php"
```

ATTENTION : Pour faciliter la lecture et la saisie des commandes, nous allons utiliser uniquement l'alias "wsh" dans ce document.

2.6.2 listapi

Cette commande, permet de connaître les fonctions de l'API disponible :

```
wsh --listapi
```

Remarque : Le résultat de cette commande, donne en fait la liste des fichiers .php du répertoire "/usr/share/what/API"

2.6.3 fdl_adoc

Permet de reconstruire les classes PHP conformément à la définition décrite dans la table docattr de la base Postgresql Freedom.

Paramètres :

- docid : référence de la famille à mettre à jour. Si non précisé toutes les familles sont mises à jour.

```
wsh --api=fdl_adoc --docid=100
```

À utiliser si des modifications ont été faites directement dans la base de données.

2.6.4 freedom_clean

Supprime les documents temporaires et nettoie les contenus des dossiers.

Paramètres : aucun.

```
wsh --api=freedom_clean
```

Ce nettoyage est effectué tous les jours de manière automatique.

2.6.5 freedom_convert

Convertit un document dans une autre famille. Le document perd tous les attributs qui n'existent pas dans la nouvelle famille.

Paramètres :

- tofamid : référence de la nouvelle famille
- docid : référence du document

```
wsh --api=freedom_convert --tofamid=124 --docid=4568
```

2.6.6 freedom_groups

Mets à jour la table des groupes FREEDOM par rapport à base de données WHAT. Copie la table *groups* de la base *anakeen* vers la base *freedom*.

Paramètres : aucun

```
wsh --api=freedom_groups
```

Cette fonction est appelée par l'application USER de WHAT à chaque changement de groupes.

2.6.7 freedom_import

Permet d'importer des documents au format CSV ou OpenOffice.org dans la base Freedom.

Paramètres :

- file : chemin du fichier
- analyze : (optionnel) si Y alors analyse le fichier seulement. Si N (défaut), l'importation est effectuée.
- dirid : (optionnel) référence du dossier par défaut pour classer les documents. Le document est classé dans ce dossier si sa référence de dossier (3ème colonne) est vide ou égale à zéro. Si cet attribut vaut zéro et qu'il n'a pas de référence dossier, alors le document n'est pas classé. Si cet attribut n'est pas précisé et qu'il n'a pas de référence dossier, il est classé dans le dossier par défaut de la famille s'il existe sinon il est classé dans le dossier « Import » (référence 10).
- double : (optionnel) : si Y alors autorise l'import d'un nouveau document de même titre et de même famille qu'un document existant. Si N (défaut), les nouveaux documents de même titre sont ignorés.
- reinitattr : (optionnel) : si yes, dans le cas d'un import de description de famille, alors les précédentes descriptions des attributs (dans la table docattr) sont supprimées. Le fichier de Class (Method.*.php) est re-généré. Dans le cas de l'import des familles de type cycle de vie, il faut "initialiser" tous les documents de ces familles pour reconstituer les attributs. Dans tous les cas, les valeurs en base de données ne sont pas affectées. Remarque : cette option n'a pas à être utilisée autrement que pour importer des familles (par exemple, à ne pas utiliser pour l'import de paramétrages de documents "système" (masque, contrôle de vue, profils, ...))

```
wsh --api=freedom_import --file=/home/eric/Banques.csv --analyze=Y
```

Le rapport des documents importés est écrit sur la sortie standard.

2.6.8 freedom_refresh

Permet de recalculer les valeurs des attributs dynamiques d'un document ainsi que de leur titre. Le rafraîchissement se fait pour tous les documents d'une même famille.

Paramètres :

- famid : référence de la famille des documents à rafraîchir.

```
wsh --api=freedom_refresh --famid=100
```

2.6.9 freedom_refresh avec une méthode en argument

La méthode freedom_refresh permet de passer une méthode en argument. Cet exemple montre comment changer le profil d'un document et de tous les documents d'une famille.

Paramètres :

- famid = id (logique ou nombre) de la famille
- method = méthode de la famille à activer
- docid (OPTIONNEL) = id du doc (si pas précisé = tous les documents)
- arg = chaîne de caractère étant les arguments passé à la méthode

Affecter le profil 1234 au document 8776 de la famille MA_FAMILY :

```
wsh --api=freedom_refresh --famid=MA_FAMILLE --docid=8776 --method=setProfil  
--arg=1234
```

Affecter le contrôle de vue 2345 à tous les documents de la famille MA_FAMILLE :

```
wsh --api=freedom_refresh --famid=MA_FAMILLE --method=setcvid --arg=2345
```

Affecter le contrôle de vue 2345 uniquement au document 8776 de la famille MA_FAMILLE :

```
wsh --api=freedom_refresh --famid=MA_FAMILLE --docid=8776 --method=setcvid  
--arg=2345
```

2.6.10 ods2csv

Cette fonction permet de convertir un fichier OpenOffice.org Calc (.ods) en fichier CSV. Le fichier CSV peut ensuite être utilisé pour réaliser des importations avec la fonction “freedom_import” documentée ci-dessus.

Paramètres :

- odsfile : Permet d'indiquer le nom du fichier .ods à convertir
- csvfile (optionnel) : Permet d'indiquer le nom du fichier .csv de destination. Si cette option n'est pas utilisée, le résultat est donné sur la sortie standard.

```
wsh --api=ods2csv --odsfile=/MonChemin/MonFichier.ods  
wsh --api=ods2csv --odsfile=/MonChemin/MonFichier.ods  
--csvfile=/MonChemin/MonFichier.csv
```

Remarque : Il n'est pas nécessaire de convertir un fichier OpenOffice.org en fichier CSV avant de l'importer dans Freedom, car Freedom est capable d'importer directement le fichier OpenOffice.org.

3 Développement

3.1 Utilisation de l'API

Ce chapitre a pour but de montrer par l'exemple l'utilisation des méthodes et fonctions les plus courantes de l'API freedom :

3.1.1 Présentation de l'utilisation de l'API

Ce chapitre a pour but de montrer par l'exemple l'utilisation des méthodes et fonctions les plus courantes de l'API freedom.

Pour tester ces exemples, il faut copier/coller le code PHP dans un fichier (ex: test.php) et l'enregistrer dans le dossier "/usr/share/what/API".

Cette commande donne la liste des fonctions de l'API disponible :

```
/usr/share/what/wsh.php --listapi
```

Cette commande permet d'exécuter une fonction de l'API comme votre fichier de test :

```
/usr/share/what/wsh.php --api=test
```

Pour plus d'informations sur l'utilisation des fonctions WSH, vous pouvez lire [ce chapitre](#).

3.1.2 Manipulation de documents (GetParam, new_Doc et new_Doc)

Chaque document est identifié par un numéro unique : son identificateur. Cet identificateur est différent pour chaque révision d'un même document. Cet identificateur permet d'accéder à l'objet document correspondant. Le développeur n'a pas besoin de connaître la famille de document pour extraire les caractéristiques propres de ce document. La fonction new_Doc() retourne l'objet document dans la classe décrite pour la famille du document précisé.

```
include_once("FDL/Class.Doc.php")
$docid = 38922;
$dbaccess = GetParam("FREEDOM_DB");
$doc = new_Doc($dbaccess, $docid);
print_r($doc);
```

Le résultat sera le suivant :

```
doc128 Object
(
    [dbtable] => doc128
    [dbseq] => seq_doc128
    [fromid] => 128
    [serveur] =>
    [port] =>
    [racine] =>
    [rootdn] =>
    [rootpw] =>
    [dbaccess] => user=anakeen port=5432 dbname=freedom
    [orginit] =>
    [action] =>
```

```
[defaultabstract] => USERCARD:VIEWABSTRACTCARD
[defaultedit] => USERCARD:EDITUSERCARD
[defDoctype] => F
[defClassname] => DocFile
[fields] => Array
(
    [0] => id
    [1] => owner
    [2] => title
    [3] => revision
    [4] => initid
    [5] => fromid
    [6] => doctype
    [7] => locked
    [8] => icon
    [9] => lmodify
    [10] => profid
    [11] => usefor
    [12] => revdate
    [13] => comment
    [14] => classname
    [15] => state
    [16] => wid
    [17] => values
    [18] => attrids
    [19] => postid
    [20] => us_matricule
    [21] => usm_hobby
    [22] => usm_ecm
    [23] => usm_comment
    [24] => usm_birthdate
    [25] => us_intphone
    [26] => us_lname
    [27] => us_fname
    [28] => us_photo
    [29] => us_initials
    [30] => us_mail
    [31] => us_phone
    [32] => us_fax
    [33] => us_mobile
    [34] => us_idsociety
    [35] => us_socaddr
    [36] => us_society
    [37] => us_type
    [38] => us_job
    [39] => us_role
    [40] => us_workaddr
    [41] => us_workpostalcode
    [42] => us_worktown
    [43] => us_country
    [44] => us_workcedex
    [45] => us_workweb
    [46] => us_scatg
    [47] => us_privcard
    [48] => us_login
    [49] => us_passwd
    [50] => us_whatid
    [51] => us_group
    [52] => us_idgroup
    [53] => us_service
    [54] => us_idservice
```

```

    )
}
```

L'objet retourné est de la classe doc128 (un utilisateur). Il contient tous les attributs et méthodes de cette classe.

Cela aurait donné le même résultat que si on avait écrit le code suivant :

```

include_once("FDLGEN/Class.**Doc128**.php")

$docid = 38922;
$dbaccess = GetParam("FREEDOM_DB");
$doc = new Doc128($dbaccess, $docid);
print_r($doc);
```

L'utilisation directe des classes spécialisée est à proscrire car elle présuppose de l'appartenance à la classe. De plus l'appel direct court-circuite les optimisations d'accès placées dans la fonction new_Doc(). Le constructeur new Doc() ne doit pas être utilisé pour instancier des objets documentaire car ces objets n'aurait pas les attributs et méthodes spécifiques qui leur sont dus.

Avant de manipuler un document, il faut s'assurer qu'il existe et qu'il vit. Pour cela, on utilise les méthodes Doc::isAlive() ou DbObj::isAffected().

```

include_once("FDL/Class.Doc.php")
$docid = getHttpVars("docid");
$dbaccess = GetParam("FREEDOM_DB");
$doc = new Doc($dbaccess, $docid);
if ($doc->isAlive()) {
    // le document est valide//
} else if ($doc->isAffected()) {
    // le document existe mais n'est pas vivant => c'est un zombie//
} else {
    // le document 38922 n'existe pas//
}
```

Note sur getParam :

La méthode getParam permet de récupérer la valeur d'un paramètre d'une application. La déclaration de ce paramètre se fait dans le fichier MONAPPLI_init.php.ini. Exemple :

```
$N = getParam("Name_of_the_parameter", "0.5");
```

\$N prendra pour valeur :

- celle du paramètre nommé "Name_of_the_parameter"
- ou 0.5 si l'accès à la valeur de ce paramètre échoue.

3.1.3 Consultation de documents (getValue, getTValue et getTDoc)

Une fois assuré de la validité du document, la consultation des données du document peut avoir lieu. Les informations portées par le document appartiennent à 2 catégories :

10. les propriétés : présentent pour tout document ⁴
11. les attributs : définis pour une classe de document et hérités par les classes dérivées.

4) ce sont les attributs de la classe Doc

3.1.3.1 Les propriétés remarquables

id	Identifiant unique du document
owner	Identifiant système de l'utilisateur
title	Titre du document
revision	numéro de révision : 0 est la première révision
version	libellé de la version : vide par défaut
initid	Identifiant initial = identifiant de la révision 0
fromid	Identifiant de la famille d'appartenance
doctype	Type de document 'F' (Normal), 'D' Dossier, 'S' Recherche, 'P' Profil, 'T' Temporaire, 'Z' Zombie, 'W' Cycle de vie, 'C' Famille (C pour Class)
locked	Chiffre négatif = Identifiant de l'utilisateur ayant verrouillé le document (Le verrou est supprimé automatiquement à la fermeture de Firefox et par un script lancé toute les nuits). Chiffre positif = Document verrouillé manuellement par l'utilisateur. 0 = Pas de verrou. -1 = Document révisé (figé).
allocated	Identifiant de l'utilisateur alloué au document
icon	référence au fichier icon : référence locale ou référence VAULT
lmodify	Y = Le document a été modifié depuis la dernière révision. L = Révision N-1. N = Document révisé mais non modifié
profid	Identifiant du profil de document. Chiffre négatif = Profil non activé. Si identique à l'id du document = contrôle dédié. Vide = Pas de profil. Chiffre positif = profil actif
attrids	Concaténation des noms attributs valués
values	Concaténation des valeurs des attributs valués (différent de vide)
usefor	type d'utilisation (obsolète)
cdate	date de création de la révision. La date de création du document est celle de la révision 0
adate	date de dernier accès
revdate	date de dernière modification ou date de révision dans le cas d'un document révisé
comment	commentaire de révision (obsolète) ⇒ voir Objet DocHisto
classname	Classe du document - Utilisé seulement pour les cycles de vie
state	état du document ou référence à un identifiant de document 'état libre'
wid	identifiant du document cycle de vie
postitid	identifiant du document 'note'
cvid	identifiant du document contrôle de vue
name	identifiant <i>logique</i> du document
dprofid	Identifiant utilisé pour les profils dynamiques. Si le document à un profid alors le profid est égale à l'id.
atags	balises applicatives (Tags positionnés sur le document)
prelid	identifiant du document (dossier) de relation primaire (généralement le dossier parent)
confidential	=1 si document confidentiel (restriction supplémentaire)
ldapdn	chemin LDAP dans le cas d'une copie sur un

	serveur LDAP
Depuis 2.7.1	
fulltext	vectorisation des attributs
svalues	concaténation des valeurs des attributs recherchable (hors image/password/idoc)

3.1.3.2 Accès aux propriétés (getValue)

L'accès aux propriétés se fait par simple accès à l'objet. L'accès aux valeurs des attributs se fait principalement par la méthode Doc::getValue(\$attrid, \$default=""). L'identificateur de l'attribut peut être indiqué en majuscule ou en en minuscule.

```
include_once("FDL/Class.Doc.php");

$docid = 38922;
$dbaccess = GetParam("FREEDOM_DB");
$doc = new_Doc($dbaccess, $docid);
if ($doc->isAlive()) {
    // le document est valide//
    $id = $doc->id;                                // c'est 38922 ($docid)//
    $rev = $doc->getProperty('revision');           // le numéro
de révision du document//
    $society = $doc->getValue("US_SOCIETY");         // la société du document
personne intranet//
    $usermail = $doc->getValue("US_MAIL","nomail"); // l'email du document
personne intranet//
    if ($usermail == "nomail") {
        // cet utilisateur n'a pas d'adresse email//
    }
}
```

3.1.3.3 Récupérer les différentes valeurs d'un attribut multiple (getTValue)

La méthode Doc::getValue() retourne toujours une chaîne de caractères même pour les attributs multiples (ceux qui sont dans des tableaux) ou ceux qui ont comme type *list tels que integerlist ou textlist. Pour récupérer les différentes valeurs d'un attribut multiple, on utilise la méthode Doc::getTValue().

```
include_once("FDL/Class.Doc.php");

$dbaccess=GetParam("FREEDOM_DB");
$doc = new_Doc($dbaccess,38922);
if ($doc->isAlive()) {
    print "[.".($doc->getValue("US_IDGROUP")."]\n");
    print_r ($doc->getTValue("US_IDGROUP")); // affichage du tableau
}
```

Le résultat de ce programme donne :

```
[38981
 38995
 38985]
Array(
    [0] => 38981
    [1] => 38995
    [2] => 38985
)
```

Le premier affichage (getValue) donne la valeur sous forme de chaîne de caractère. Les différents éléments sont séparés par un retour chariot. Le deuxième affichage montre la décomposition de la valeur multiple en 3 valeurs distinctes. Ceci permet de contrôler chaque valeur plus facilement.

Dans l'exemple ci-dessous, on affiche les noms des groupes d'appartenance de la personne référencée par le numéro 38922.

```
include_once("FDL/Class.Doc.php");

$dbaccess=GetParam("FREEDOM_DB");

$doc = new_Doc($dbaccess,38922);
if ($doc->isAlive()) {
    $tidgroup = $doc->getTValue("US_IDGROUP");
    foreach($tidgroup as $k=>$v) {
        $gdoc = new_Doc($dbaccess, $v); // constitution du document
        groupe de personnes
        if ($gdoc->isAlive()) print $gdoc->getTitle()."\\n"; // affichage des
        titres des groupes
    }
}
```

3.1.3.4 Récupérer un array complet ou une ligne de l'array (getAValues)

getAValues permet de retourner un tableau complet de valeurs ou une rangée si l'argument index est indiqué :

```
$t=$this->getAValues("EN_T_ANIMAUX");
```

Pour plus d'infos : [methodGetAValues](#)

3.1.3.5 Accès à l'ensemble des attributs (getValues)

Pour avoir l'ensemble des valeurs des attributs d'un document, on utilise la méthode Doc::getValues(). Cette méthode retourne les valeurs (sous forme de chaîne de caractères) de tous les attributs du documents. Les index du tableau retourné sont les identificateurs des attributs.

```
include_once("FDL/Class.Doc.php");

$dbaccess=GetParam("FREEDOM_DB");
$doc = new_Doc($dbaccess,38922);
if ($doc->isAlive()) {
    $tvalues = $doc->getValues();
    foreach($tvalues as $k=>$v) {
        if ($v != "") print "$k : $v\\n"; // affiche les valeurs non nulles
    }
}
```

Cet exemple retourne toutes les valeurs non nulles des attributs. Le résultat produit est le suivant :

```
us_lname : martin
us_fname : jean
us_mail : jean.martin@i-cesam.com
us_login : jean.martin
us_passwd : Vto9bHWMZGSPw
us_whatid : 54
us_group : default group
```

```
Groupe Incident
Groupe Technique
us_idgroup : 38981
38995
38985
```

3.1.3.6 Accès aux valeurs des documents (getTdoc)

Afin d'optimiser l'accès aux documents, on peut utiliser la fonction getTdoc() pour récupérer l'ensemble des valeurs d'un document. Cette fonction ne retourne pas un objet documentaire mais simplement un tableau de valeur des propriétés et attributs.

```
include_once("FDL/Class.Doc.php");

$dbaccess=GetParam("FREEDOM_DB");
$tdoc = getTDoc($dbaccess,38922); // tableau de valeurs//
if ($tdoc) {
    foreach($tdoc as $k=>$v) {
        if ($v != "") print "$k : $v\n"; // affiche les valeurs non nulles
    }
}
```

3.1.4 Modification de documents (setValue, getValues et deleteValue)

La méthode Doc::setValue(\$attrid, \$val) sert à modifier les valeurs d'un document. Cette méthode de mets pas à jour directement en base mais uniquement dans l'objet document. La modification effective en base de données est faite par l'appel à la méthode DbObj::modify().

```
include_once("FDL/Class.Doc.php");

$dbaccess=GetParam("FREEDOM_DB");
$doc = new_Doc($dbaccess,38922);
if ($doc->isAlive()) {
    $doc->setValue("US_ROLE","responsable développement logiciel");
    $err = $doc->Modify(); // affectation des valeurs dans la base de données//
    if ($err != "") {
        print $err; // l'utilisateur n'a probablement pas le droit de modifier ce
document//
    } else {
        $tvalues = $doc->getValues();
        foreach($tvalues as $k=>$v) {
            if ($v != "") print "$k : $v\n";
        }
    }
}
```

Le résultat indique que l'attribut US_ROLE a maintenant une nouvelle valeur :

```
us_lname : martin
us_fname : jean
us_mail : jean.martin@i-cesam.com
us_login : jean.martin
us_passwd : Vto9bHWMZGSPw
us_whatid : 54
```

```

us_group : default group
Groupe Incident
Groupe Technique
us_idgroup : 38981
38995
38985
us_role : responsable développement logiciel

```

Si la valeur passée en paramètre de setValue est une chaîne vide, la modification est ignorée. Pour supprimer une valeur, il faut utiliser la méthode deleteValue.

```

include_once("FDL/Class.Doc.php");
$dbaccess=GetParam("FREEDOM_DB");
$doc = new_Doc($dbaccess,38922);
if ($doc->isAlive()) {
    $doc->deleteValue("US_ROLE"); // suppression du rôle//
    $err = $doc->Modify(); // affectation des valeurs dans la base de
données//
    if ($err != "") {
        print $err; // l'utilisateur n'a probablement pas le droit de modifier
ce document//
    }
}

```

3.1.5 Crédation de documents (createDoc)

La création d'un document vide (sans valeur) se fait par l'appel à la fonction createDoc(). Cette fonction permet d'initialiser un objet document en précisant sa famille. Le document retourné contient les valeurs par défaut de la famille ainsi que le profil d'accès par défaut.

```

include_once("FDL/Class.Doc.php");

$dbaccess=GetParam("FREEDOM_DB");
$doc = createDoc($dbaccess,"USER");
if ($doc === false) {
    // l'utilisateur n'est pas habilité à créer ce type de document//
} else {
    // l'objet doc120 (USER) est créé//
    $err=$doc->setValue("us_fname","Paul");
    if (! $err) $err = $doc->setValue("us_lname","Dujardin");
    if (! $err) $err = $doc->Add();
    if ($err != "") {
        // erreur à la création en base//
        print $err;
    } else {
        // nouveau document en base//
        print "nouvel identifiant :".$doc->id. "\nobjet :".get_class($doc).
        "\ntitre :".$doc->getTitle();
    }
}

```

Ce programme crée un nouveau document vide de la famille USER (personne). Le résultat de cet exemple est le suivant :

```
nouvel identifiant :39692
objet :doc120
titre :personne sans titre 39692
```

La création de document peut aussi se faire pour duplication d'un autre document. Pour cela on utilise la méthode Doc::copy().

```
include_once("FDL/Class.Doc.php");
$dbaccess=GetParam("FREEDOM_DB");

$doc = new_Doc($dbaccess,38922);

if ($doc->isAlive()) {
    $copy = $doc->copy();
    if (! is_object($copy)) {
        // erreur pendant la copie//
        print $copy; //// message d'erreur//
    } else {
        // nouveau document en base//
        print "nouvel identifiant :".$copy->id. "\nobjet :".get_class($copy).
            "\ntitre :".$copy->getTitle();
    }
}
```

Le résultat est que l'on obtient une réplique du document 38922. Il est identique à part son identifiant qui est 39693 dans l'exemple.

```
nouvel identifiant :39693
objet :doc128
titre :martin jean
```

La création d'un document **temporaire** se fait par la fonction createTmpDoc(). Cela crée un document temporaire et il n'est pas soumis au contrôle de l'acl create au contraire de la fonction createDoc(). Le document temporaire s'il est mis en base (appel de Doc::Add()) sera supprimé la nuit par le robot (cron) de nettoyage.

3.1.6 Placement de documents dans un dossier (addFile et delFile)

Les documents que l'on créé par createDoc() ou Doc::copy() ne sont rattachés à aucun dossier⁵. Si on veut les référencer⁶ dans des dossiers précis il faut le demander. Ces ajouts ou suppressions de référence à des documents dans les dossiers se font à l'aide des méthodes Dir::AddFile() et Dir::DelFile().

```
include_once("FDL/Class.Doc.php");

$dbaccess=GetParam("FREEDOM_DB");
$doc = new_Doc($dbaccess,38922);
if ($doc->isAlive()) {
    $fld = new_Doc($dbaccess,38618);
```

5) ce n'est pas gênant en soi... FreeDom tout est dans tout et réciproquement

6) un document peut être référencé plusieurs fois, n'oubliez pas

```

if ($fld->isAlive()) {
    if ($fld->doctype != "D") {
        // ce n'est pas un dossier//
        print $fld->id." n'est pas un dossier";
    } else {
        $err = $fld->addFile($doc->id);
        if ($err != "") {
            // le document n'a pas pu être inséré//
            print $err;
        } else {
            //// le document est maintenant inséré//
        }
    }
}

```

Pour l'exemple le document n°38618 est un dossier. L'exécution du programme produit le résultat suivant :

LOG::(I)::Maître Chewie [1] - Ajout martin jean dans dossier What Master

La méthode Dir::AddFile() mets un message de log si l'insertion réussie. Si on ré-exécute le programme nous obtenons :

déjà existant : pas ajouté

Dans le deuxième cas, la méthode Dir::AddFile() retourne un message d'erreur car on ne peut référencer deux fois le même document dans un dossier.

Pour supprimer la référence à un document d'un dossier, il suffit d'appeler la méthode Dir::DelFile().

```

include_once("FDL/Class.Doc.php");

$dbaccess=GetParam("FREEDOM_DB");
$doc = new_Doc($dbaccess,38922);
if ($doc->isAlive()) {
    $fld = new_Doc($dbaccess,38618);
    if ($fld->isAlive()) {
        if ($fld->doctype != "D") {
            // ce n'est pas un dossier//
            print $fld->id." n'est pas un dossier";
        } else {
            $err = $fld->delFile($doc->id);
            if ($err != "") {
                // la référence au document n'a pas pu être supprimée//
                print $err;
            } else {
                // la référence au document est maintenant supprimée//
            }
        }
    }
}

```

L'exécution du programme produit le résultat suivant :

LOG::(I)::Maître Chewie [1] - Suppression n°38922 du dossier What Master

Si on ré-exécute le programme nous obtenons :

échec de la suppression de la référence (document 38922, dossier 38618)

La méthode Dir::DelFile() indique qu'on a essayé de supprimer la référence à un document qui n'est pas dans le dossier.

3.2 Recherche de documents

Utilisation de la classe [SearchDoc](#)⁷. Cette classe est à utiliser en remplacement de la fonction getChildDoc qui est plus délicate à appeler à cause du nombre croissant de paramètres.

3.2.1 Retour de valeurs d'attributs

Ce programme permet d'écrire tous les titres des documents visible⁸ de la famille personne intranet. Ici, on n'a utilisé que le critère d'appartenance à une famille. Le retour de méthode DocSearch::Search() est un **tableau de valeurs** (tableau de tableaux).

Avec une boucle foreach.

```
include_once("FDL/Class.SearchDoc.php");

$db=GetParam("FREEDOM_DB");
$s=new SearchDoc($db,"IUSER");
$t=$s->search();

foreach ($t as $k=>$v) {
    print "$k) ".$v["title"].".".getv($v,"us_mail","nomail")."\n";
}
```

Avec une boucle while:

```
include_once("FDL/Class.SearchDoc.php");

$db=GetParam("FREEDOM_DB");
$s=new SearchDoc($db,"IUSER");
$s->search();

$k=0;
while ($v=$s->nextDoc()) {
    print "$k) ".$v["title"].".".getv($v,"us_mail","nomail")."\n";
    $k++;
}
```

3.2.2 Retour d'objets documentaires

Ce programme permet d'écrire tous les titres des documents visible 90 de la famille personne intranet. Ici, on n'a utilisé que le critère d'appartenance à une famille.

L'appel à la méthode DocSearch::setObjectReturn() indique que le retour de méthode DocSearch::Search() est un **tableau d'objets documentaires**. Dans ce cas, il est possible d'appliquer les méthodes des objets sur les retours (exemple Doc::getValue).

```
include_once("FDL/Class.SearchDoc.php");

$db=GetParam("FREEDOM_DB");

$s=new SearchDoc($db,"IUSER");
$s->setObjectReturn(); // retour d'objets documentaires
$s->addFilter('us_extmail is not null'); // retour d'objets documentaires
$s->search(); // déclenchement de la recherche
```

7) L'accès à l'api complète se fait [ici](#)

8) La recherche retourne seulement les documents que l'utilisateur à le droit de voir -accès VIEW-

```
$c=$s->count();
print "count $c\n";
$k=0;
while ($doc=$s->nextDoc()) {
    print "$k) ".$doc->getTitle().".". $doc->getValue("US_MAIL","nomail")."\n";
    $k++;
}
```

On peut limiter le nombre de résultats avec l'attribut `slice`.

```
$s->slice=10; // limit to first 10 results
```

On peut utiliser la pagination en combinant avec `start` et `slice` : Les résultats de 15 à 25

```
$s->start=15; // commence au 15ème
$s->slice=10; // limit to first 10 results
```

3.2.3 Debug

Si la requête échoue suite à des erreurs de filtres, vous pouvez voir les erreurs en utilisant les méthodes `::searchError()` et `::getError()`.

```
<?php
include_once("FDL/Class.SearchDoc.php");
$s=new SearchDoc($dbaccess,$famId);
$s->setObjectReturn();
if ($docid > 0) $s->addFilter("id = %s",$docid);
if ($fldid > 0) $s->dirid=$fldid;
if ($allrev) $s->latest=false;
$s->addFilter("pas bon"); // ici une erreur de filtre
$s->search();

if ($s->searchError()) {
    $action->exitError(sprintf("search error : %s",$s->getError()));
}
?>
Erreur : search error : ERREUR: erreur de syntaxe sur ou près de « bon »
LIGNE 1 : ... and (doctype != 'T') and (locked != -1) and (pas bon) ORDER...
```

La classe `SearchDoc` propose un mode debug qui permet d'avoir le détail de la requête effectué. Le mode debug est activable avec la méthode `DocSearch::setDebugMode()`, et le résultat est consultable avec la méthode `DocSearch::getDebugInfo()` :

```
include_once("FDL/Class.SearchDoc.php");

$db=GetParam("FREEDOM_DB");
$s=new SearchDoc($db,"IUSER");
$s->addFilter('us_extmail is not null');
```

```
$s->setDebugMode(); // Active le mode debug avant de lancer la recherche
$s->search();

// Affiche les messages de debug avec leur type
foreach($s->getDebugInfo() as $type => $msg) {
    error_log("DEBUG [$type] = $msg");
}
```

3.2.4 Recherche dans un dossier et recherche récursive

La classe SearchDoc permet de faire des recherches dans un dossier spécifique au moyen de la propriété DocSearch::dirid. **Cette recherche n'est pas récursive par défaut**, c'est à dire qu'elle ne recherchera que dans le dossier indiqué.

Il est possible de faire des recherches récursives à l'aide de la méthode DocSearch::setRecursiveSearch(). Le niveau de profondeur de la recherche est ensuite défini au moyen de la propriété DocSearch::folderRecursiveLevel, positionné à 2 par défaut⁹

 Ajouter un exemple de code

3.2.5 Recherche par critère sur les valeurs d'attributs

Ce type de recherche est souvent combiné avec le critère d'appartenance à la famille. S'il n'y a pas de critère d'appartenance à une famille seules les propriétés peuvent être utilisés comme conditions de sélection.

L'ajout de critère se fait avec la méthode ::addFilter(). Celle-ci ajoute une condition supplémentaire sur la requête. Si votre requête comporte des parties variables, il faut les déclarer dans les arguments suivant le format de la requête comme pour la fonction sprintf. Cela permet d'éviter une injection SQL et garanti que la chaîne de caractère sera correctement traduite en SQL.

Exemple :

```
$s->addFilter("title ~= '%s'", $mytitle);
$s->addFilter("id = %d and cdate > '%s'", $myid, $s->getDate());
```

3.2.5.1 Recherche sur le titre des documents (title)

Cet exemple montre la recherche de document par le titre. Ici, tout type de document est retourné si son titre contient 'jean' en majuscule ou minuscule. Les opérateurs utilisables sont les opérateurs SQL de PostgreSQL¹⁰.

```
<?php
include_once("FDL/Class.Doc.php");
include_once("FDL/Class.SearchDoc.php");

$dbaccess=GetParam("FREEDOM_DB");
$s=new SearchDoc($dbaccess); // famid : toute famille
$nom="jean";
$s->addFilter("title ~* '%s'", $nom); // titre contient jean
$s->start=0;
```

9) On notera que c'est le niveau de **récursivité**. Par exemple, folderRecursiveLevel=0 vaut dire que l'on recherche dans le dossier, alors que folderRecursiveLevel=1 indique de rechercher dans le dossier et ses sous-dossiers.

10) Se reporter au manuel de référence PostgreSQL [Chapter 6. Fonctions and Operators](#)

```

$s->slice=10;// les 10 premiers//
$s->setObjectReturn();
$s->search();

while ($doc=$s->nextDoc()) {
    print "$k".$doc->getTitle()."\n";
}
?>

```

3.2.5.2 Recherche sur n'importe quel attribut (values)

Dans le cas où on recherche une valeurs dans n'importe quel attribut du document, on utilisera l'attribut **svalues**. Cet attribut contient la concaténation des valeurs des autres attributs.

```

<?php
include_once("FDL/Class.Doc.php");
include_once("FDL/Class.SearchDoc.php");

$dbaccess=GetParam("FREEDOM_DB");
$s=new SearchDoc($dbaccess);// famid : toute famille
$s->addFilter("svalues ~* 'jean'");// une des valeurs contient jean
$s->start=0;
$s->slice=10;// les 10 premiers//
$s->setObjectReturn();
$s->search();

while ($doc=$s->nextDoc()) {
    print "$k".$doc->getTitle()."\n";
}
?>

```

3.2.5.3 Recherche dans une famille

Dans le cas où une famille d'appartenance est précisée, il est possible d'utiliser les attributs de cette famille comme critères.

```

<?php
include_once("FDL/Class.Doc.php");
include_once("FDL/Class.SearchDoc.php");

$dbaccess=GetParam("FREEDOM_DB");
$s=new SearchDoc($dbaccess,"USER");// famid : toute famille
$s->addFilter("us_fname ~* 'jean'");// prénom contient jean
$s->start=0;
$s->slice=10;// les 10 premiers//
$s->setObjectReturn();
$s->search();

while ($doc=$s->nextDoc()) {
    print "$k".$doc->getTitle().".".getValue("us_mail","nomail")."\n";
}
?>

```

L'exemple ci-dessus montre la recherche de toutes les personnes dont l'adresse email est un organisme et dont le prénom contient jean. Le paramètre sqlfilters établit une conjonction¹¹ de condition.

¹¹⁾ET logique

3.2.5.4 Recherche avec l'opérateur 'or'

Si on veut établir une disjonction¹², il faut l'écrire manuellement en SQL en utilisant l'opérateur or.

```
<?php
include_once("FDL/Class.Doc.php");
include_once("FDL/Class.SearchDoc.php");

$dbaccess=GetParam("FREEDOM_DB");
$s=new SearchDoc($dbaccess,"USER");// famid : toute famille
$s->addFilter("us_fname ~* 'jean|patrick'"); // prénom contient jean ou patrick
// mail fini par .org ou cp commence par 31//
$s->addFilter("(us_mail ~ '\.org$') or (us_workpostalcode ~ '^31')");
$s->start=0;
$s->slice=10;// les 10 premiers//
$s->setObjectReturn();
$s->search();

while ($doc=$s->nextDoc()) {
    print "$k".$doc->title."(".$doc->getValue("us_mail","nomail").":".
          $doc->getValue("us_workpostalcode").")\n";
}
?>
```

3.2.5.5 Recherche sur des attributs de type énumérés (getKindDoc)

Pour rechercher des documents suivant des attributs de type énumérés, une fonction simplifiée de recherche existe. Cette fonction getKindDoc() est basée sur getChildDoc(), elle a juste pour but de construire la règle de filtrage adéquate pour tenir compte de la hiérarchie dans ce type d'attribut.

```
include_once("FDL/Class.Doc.php");
include_once("FDL/Lib.Dir.php");

$dbaccess=GetParam("FREEDOM_DB");
$tdoc=getKindDoc($dbaccess,
    "USER",           // nom de la famille//
    "us_type",        // attribut énuméré où s'applique le filtrage//
    "chefserv");      // clef à rechercher//

while (list($k, $v) = each($tdoc)) {
    print "$k".$v["title"]."(".$v["us_mail","nomail"].")\n";
}
```

Cet exemple permet de sélectionner la liste des chefs de service. Le chef de service a pour clef 'chefserv'.

3.2.5.6 Recherche dans un array

Si vous avez une famille avec un array contenant une liste d'id, vous pouvez utiliser ce filtre Postgresql pour retourner tous les documents contenant cet id :

```
$s->addFilter="AN_IDENFANT ~ '\\\\y'.$id.'\\\\y'";
```

^{12)OU logique}

3.2.6 Recherche spécialisée

Pour toutes recherches non prévus en standard par l'interface, vous pouvez programmer des recherches spécifiques. Elles pourront ensuite être utilisées comme une recherche "normale" depuis l'interface grâce à la famille "recherche spécialisée".

Lorsque vous éditez une recherche spécialisée, vous devez renseigner le fichier php où se trouve la fonction de recherche et le nom de cette fonction. Le fichier PHP devra être dans le répertoire EXTERNALS (/usr/share/what) sur votre serveur.

Les arguments de la fonction sont au minimum de 3 :

- start : start index pour la recherche
- slice : nombre max d'éléments à retourner
- userid : utilisateur courant

Exemple issu du fichier fdlsearches.php (livré en standard).

```

function mytagdoc($start,$slice,$tag,$uid) {
    include_once("FDL/Class.DocUTag.php");
    include_once("FDL/Lib.Dir.php");
    $dbaccess=getParam("FREEDOM_DB");
    $q=new QueryDb($dbaccess,"DocUTag");
    $q->AddQuery("uid=$uid");
    $q->AddQuery("tag='$tag'");
    $lq=$q->Query($start,$slice,"TABLE");
    $lid=array();
    if ($q->nb > 0) {
        foreach($lq as $k=>$v) {
            $lid[$v["initid"]]= $v["id"];
        }
    }

    $ltdoc=getDocsFromIds($dbaccess,$lid);

    return $ltdoc;
}

/**

```

```

* function use for specialised search
* return all document tagged TOVIEWDOC for current user
*
* @param int $start start cursor
* @param int $slice offset ("ALL" means no limit)
* @param int $userid user system identifier (NOT USE in this function)
*/
function mytoviewdoc($start="0", $slice="ALL", $userid=0) {
    return mytagdoc($start,$slice,"TOVIEW",$userid);
}

```

Des arguments supplémentaires peuvent être ajoutés dans l'attribut 'Argument PHP'. Il sont ajoutés dans l'appel à partir de la quatrième position. Pour rajouter plusieurs arguments, il faut les séparer par une virgule (exemple : 1234, ceci est un test, dernier argument).

Dans ces arguments, il est possible de référencer des attributs du document recherche lui-même. Il faut alors utiliser la notation suivante %TITLE% pour avoir le titre de la recherche, ou %SE_IDCFLD% pour avoir l'identifiant du dossier dans lequel s'exécute la recherche. N'importe quel attribut ou propriété de la recherche est accessible. Le mot clef %THIS%, permet d'obtenir l'objet recherche dans sa globalité.

La fonction de recherche doit retourner un tableau de document. Ces documents retournés doivent être de type array (pas object). Ce type est celui retourné par la classe SearchDoc et les fonctions getChildDoc (mode TABLE), getTdoc ou getDocsFromIds.

3.3 Aides à la saisie

3.3.1 Principe et spécification

Les fonctions d'aide à la saisie permettent de compléter ou de remplir des zones de saisie lors de l'édition des documents Freedom.

Les paramètres d'entrées peuvent être :

- des valeurs déjà saisies lors de l'édition de la fiche
- les coordonnées de la base de données FREEDOM
- des valeurs statiques

Cette fonction doit toujours retourner une matrice comportant les choix possibles. Cette matrice est un tableau de tableau de chaîne de caractères. La première colonne de chaque rangée contient la chaîne de caractères présentée à l'utilisateur. Les colonnes suivantes contiennent les valeurs qui seront insérées dans les zones de saisies spécifiées à l'appel.

Les fonctions d'aide à la saisie doivent être définies dans des fichiers PHP qui seront placé dans le répertoire /usr/share/what/EXTERNALS sur le serveur.

3.3.2 Retour unique

Le premier exemple nous montre comment retourner une liste de choix statique.

```

function getGravite() {
    return array(0=> array("mineure", "Mi"),
                1=> array("majeure", "Ma"),
                2=> array("bloquante", "Bl"));
}

```

utilisation possible :

Id	Description	Vis	Phpfile	phpfunc
-----------	--------------------	------------	----------------	----------------

TE_GRAV	Gravité	W	Test.php	getGravite():TE_GRAV
---------	---------	---	----------	----------------------

L'utilisateur, lorsqu'il demandera l'aide à la saisie, verra les trois choix: mineure, majeure et bloquante. S'il choisit *mineure* par exemple, la valeur *Mi* sera insérée dans la zone de saisie spécifiée (TE_GRAV).

Le deuxième exemple montre une utilisation plus complète avec la manipulation de documents. Cela permet de rechercher les personnes qui appartiennent à la société passée en paramètre. Un deuxième filtre sur le titre des personnes permet de restreindre cette recherche. Cette fonction a pour but de rechercher le nom et prénom d'une personne ou de les compléter.

```
function getUserOfSociety($dbaccess,$socid,$name="") {
    include_once("FDL/Class.SearchDoc.php");

    $s=new SearchDoc($dbaccess,"USER");
    $s->addFilter(sprintf("us_idsociety='%d'", $socid));//// filtre sur
    l'appartenance à la société/
    if ($name != "") $s->addFilter(sprintf("title ~*
    '%s",pg_escape_string($name)));//// filtre éventuel sur le titre//
    $s->slice=100;
    $s->setObjectReturn();
    $s->search();
    $tret=array();
    while ($doc=$s->nextDoc()) {
        $tret[]=$doc->getTitle(), $doc->getTitle());
    }
    return $tret;
}
```

utilisation possible :

Id	Description	Vis	Phpfile	phpfunc
TE_IDSOC	identification société	H		
TE_USER	une personne	W	test.php	getUserOfSociety(D, TE_IDSOC,TE_USER) :TE_USER

Pour retourner une erreur, il suffit de retourner une chaîne de caractère au lieu du tableau. Cela affichera l'erreur au lieu de la liste de choix. Exemple :

```
function searchLDAPinfo($login) {
    $err=searchLDAPFromLogin($login,false,$tinfo);
    if ($err == "") {
        $conf=getLDAPConf(getParam("NU_LDAP_KIND"));
        $tout=array();
        foreach ($tinfo as $k=>$v) {
            $login=$v[$conf["LDAP_USERLOGIN"]];
            $fn=$v["givenName"];
            $ln=$v["sn"];

            $tout[]=$login,$login,$fn,$ln); // le premier login est la valeur
            présentée
        }
    }
    if ($err) return $err;
    return $tout;
}
```

```
}
```

3.3.3 Retour multiple

Les retours multiples permettent de remplir plusieurs zone d'édition avec une seule aide à la saisie. Leur réalisation est identique à celle du retour unique sauf que le nombre de colonne dans la matrice est supérieur à deux.

Nous reprenons l'exemple précédent mais cette fois nous retournons aussi le mail et le téléphone du document personne. Si la personne n'a pas de téléphone, l'aide à la saisie retournera le numéro de téléphone de la société.

Lorsqu'une des valeurs vaut '?' cela signifie que la zone de saisie ne doit pas être modifiée. Par contre si la valeur vaut '' (chaîne vide) cela signifie que le zone de saisie sera effacée et par conséquent l'ancienne valeur en base de données sera conservée. Par conséquent, dans l'exemple, si la personne n'a pas de mail et qu'on a déjà saisi une adresse email celle-ci sera conservée.

```
function getUserOfSociety($dbaccess,$socid,$name="") {
    include_once("FDL/Class.Doc.php");
    include_once("FDL/Class.SearchDoc.php");

    $tret=array(); ///// matrice à retourner///
    $docsoc= new_Doc($dbaccess, $socid);
    if ($docsoc->isAlive()) {

        $s=new SearchDoc($dbaccess,"USER");
        $s->setObjectReturn();
        $s->addFilter(sprintf("us_idsociety='%d'", $socid));/// filtre sur
        l'appartenance à la société/
        if ($name != "") $s->addFilter(sprintf("title ~*
        '%s'", pg_escape_string($name)));/// filtre éventuel sur le titre//
        $s->slice=100;
        $s->search();

        while ($doc=$s->nextDoc()) {
            $tret[] = array(sprintf("%s (%s)", $doc->getTitle(), $docsoc-
            >getTitle()), // on présente la personne et la société
            $doc->getTitle(),
            $doc->getValue("us_mail","?"), //***// ne supprime pas le mail si déjà
            saisi et s'il n'existe pas**/
            $doc->getValue("us_phone",$docsoc->getValue("SI_PHONE"))); //***//
            retourne le téléphone société si téléphone personnel non trouvé*/
        }
    }
    return $tret;
}
```

utilisation possible :

Id	Description	Vis	Phpfile	phpfunc
TE_IDSOC	identification société	W		
TE_USER	une personne	W	test.php	getUserOfSociety(D, TE_IDSOC,TE_USER) :TE_USER,TE_MAIL,T E_PHONE
TE_MAIL	Son mail	W		

TE_PHONE	Son téléphone	W		
----------	---------------	---	--	--

Remarque : La fonction « **func_get_args()** » permet de connaître le nombre d'arguments indiqués dans le fichier OpenOffice.org et donc de retourner les arguments nécessaires dans le code.

3.3.4 Cas des attributs relation (docid)

Pour les attributs relation, une aide à la saisie est automatiquement mise si la déclaration fait référence à une famille : docid("TST_MA_FAMILY").

L'aide à la saisie automatique est la suivante :

Id	Description	Type	Vis	Phpfile	phpfunc
TE_SOC	Société	docid("SOCIETY")	W	fdl.php	Ifamily(D,SOCIETY,CT):TE_SOC,CT

Le mot-clef CT désigne le titre (Current Title).

Depuis la version 3.0.4 de freedom, il est aussi possible d'utiliser le mot-clef CT[TE_DEPT] pour désigner le titre d'une autre relation.

Id	Description	Type	Vis	Phpfile	phpfunc
TE_SOC	Société	docid("SOCIETY")	W	fdl.php	Ifamilyvalue(D,SOCIETY,CT,'title,'id,'soc_depttitle,'soc_dept):TE_SOC,CT,CT[TE_DEPT]
TE_DEPT	Département	docid("DEPT")	W		

3.3.5

3.4 Attributs calculés

3.4.1 Principe et spécification

Les valeurs des attributs calculés sont issues de méthodes de la classe de document. Comme pour les fonctions d'aide à la saisie, les paramètres d'entrées peuvent être

- les coordonnées de la base de données
- des valeurs d'attributs du document
- des valeurs statiques

Ces méthodes retournent un seul résultat sous forme de chaînes de caractères. Le résultat est mis dans l'attribut sur lequel s'applique le calcul (sauf redirection explicite faite lors de la déclaration de l'attribut). Ces fonctions sont appelées lors du rafraîchissement de document Doc::refresh(). D'un point de vue utilisateur, ces attributs sont rafraîchis avant chaque consultation de document.

3.4.2 Calcul simple

Si vous voulez un attribut qui donne l'heure courante pour la famille TEST, la construction du calcul sera la suivante:

```
// Method.Test.php
public function getTime() {
    return strftime(time());
}
```

Les fonctions de calcul sont des méthodes du document : méthodes standards ou définies dans les fichiers METHOD des familles.

Pour constituer un attribut calculé qui donne l'heure courante, il suffit de rajouter un attribut tel que celui défini ci-dessous.

id	définition	type	vis	phpfun
TE_HOUR	heure courant	time	R	::getTime()

Les fichiers METHOD doivent contenir l'ensemble des méthodes de calculs nécessaire à la famille.

Exemple :

```
// Method.Test.php

public function getTime() {
    Return strftime(time());
}

public function getMail($userdocid) {
    $udoc = new_Doc($this->dbaccess, $userdocid);
    if ($udoc->isAlive()) return $udoc->getValue("US_MAIL");
    return ""
}

public function getAddress($userdocid) {
    $udoc = new_Doc($this->dbaccess, $userdocid);
    if ($udoc->isAlive()) {
        return sprintf("%s\n%s %s",
            $udoc->getValue("US_WORKADDR"),
            $udoc->getValue("US_WORKPOSTALCODE"),
            $udoc->getValue("US_WORKTOWN"));
    }
    return "";
}
```

La valeur retourné par la méthode est directement inséré dans l'attribut calculé. Si la méthode ne retourne la chaîne de caractère vide celle-ci ne sera pas affectée à l'attribut.

Les attributs utilisables peuvent être :

Id	Description	type	vis	phpfunc
TE_IDUSER	Une personne	docid	W	
TE_MAIL	Son mail	text	R	::getMail(TE_IDUSER)
TE_CADDR	Adresse composée	longtext	R	::getAddress(TE_IDUSER)

3.4.3 Calcul multiple

Si la famille que vous construisez utilise de nombreux attributs calculés pouvant être mis à jour avec le même algorithme, on utilise la méthode Doc::specRefresh().

Cette méthode est une surcharge de la classe père¹³. Elle est appelée par la méthode Doc::refresh() comme les autres méthodes de calculs. Au contraire des méthodes de calcul

13) initialement la classe Doc

simples, cette méthode doit mettre à jour les valeurs des attributs à l'aide de la méthode Doc::setValue().

```
// Method.Test.php
public function specRefresh() {
    Parent::specRefresh(); // réutilisation éventuelle de la méthode du père//
    //spécifie les paramètres attribut d'entrée et de sortie
    $this->AddParamRefresh("TE_USERID","TE_MAIL,TE_CADDR");

    $udoc= new_Doc($this->dbaccess, $this->getValue("TE_USERID"));
    if ($udoc->isAlive()) {
        $mail=$udoc->getValue("US_MAIL");
        $caddr=sprintf ("%s\n%s %s",
        $udoc->getValue("US_WORKADDR"),
        $udoc->getValue("US_WORKPOSTALCODE"),
        $udoc->getValue("US_WORKTOWN"))
        $this->setValue("TE_MAIL",$mail);
        $this->setValue("TE_CADDR",$caddr);
    }
}
```

La méthode Doc::specRefresh() peut retourner un message d'avertissement qui est affiché lors de la consultation du document.

3.4.4 Afficher un message d'avertissement en cas d'erreur (addWarningMsg)

Il est possible d'afficher un popup à l'écran pour avertir l'utilisateur d'un problème quelconque :

```
addWarningMsg("Votre message");
```

3.5 Contraintes

L'utilisation des contraintes est décrite dans [le manuel des familles](#).

3.5.1 La fonction de contrôle

La fonction de contrôle déclenchée est une méthode du document. Son prototype standard est le suivant :

```
public function ma_contrainte([P,[P,[...]]) {
    $error_message = "";
    $proposal = array();

    // ... some code ...

    if ($this->getValue("fam_attr")=12 && $P<0) {
        $error_message = "Invalid value given ($P)";
        $proposal[] = "10";
        $proposal[] = "12";
    }

    // ... some code ...

    return array( "err" => $error_message,
                  "sug" => $proposal );
}
```

Cette méthode doit retourner un tableau avec deux entrées. Une entrée "err" qui contient le

message d'erreur de la contrainte. S'il n'y a pas d'erreur le message d'erreur doit être vide.

3.5.2 Contrainte système

La classe documentaire mets à disposition des contraintes pré-définies:

12. ::parseMail(\$mail) - permet de teste la valider d'une adresse email
13. ::isFutureDate(\$date) - vérifie que la date est dans le futur
1. ::isFloat(\$x,\$min,\$max) - teste si le nombre est compris entre min et max
2. ::isInteger(\$x,\$min,\$max)- teste si le nombre est compris entre min et max
3. ::isString(\$s,\$pattern)- teste si le texte correspond au pattern (pattern au sens perl pattern voir manuel php preg_match).

Exemple mobile ::isString(te_mobile,0[6|7][0-9]{6}) numéro à 8 chiffres commençant par 06 ou 07

3.6 Surcharges des méthodes du document

Les surcharges de méthodes sont faites via les fichiers METHOD.

! Si vous modifiez les fichiers Method sur le serveur, il faut lancer le shell fdl_adoc pour regénérer la classe PHP documentaire correspondante :

```
[root@chewbacca Freedom]# wsh --api=fdl_adoc --docid=MAILBOX
/usr/share/what/FDLGEN/Class.Doc1403.php [boîte aux lettres(MAILBOX)]
```

3.6.1 Contrôle à la création

La création de document est conditionnée par les droits de l'utilisateur courant lorsqu'on utilise la fonction createDoc(). La création en base de données est faite par la méthode Doc::Add(). Elle est aussi appelée à chaque révision du document¹⁴. Cette méthode appelle la méthode Doc::PreCreated() qui peut être définie par les différentes classes de documents créées. Cette méthode permet de rajouter des contraintes ou des calculs nécessaires avant la création en base. Si cette méthode retourne un message d'erreur l'insertion en base de données sera abandonnée.

```
function preCreated() {
    if ( $this->revision == 0 ) { //// traitement seulement pour la première
        révision// }

    //// vérifie si l'état de la commande client est "à facturer" //
    $cmcid=$this->getValue("FACT_IDCMC"); //// recherche commade client
    associée//
    if ($cmcid > 0) {

        $cmd= new_Doc($this->dbaccess,$cmcid);
        $cmdid= $cmd->latestId(); //// dernière révision//
        $cmd= new_Doc($this->dbaccess,$cmdid);

        if ($cmd->state != "forbilled")
            return sprintf(_("the state of the commande %s must be forbilled"),
            $cmd->title);

    }
}
```

14)une révision du document est un nouveau document

```
    return "";
}
```

L'exemple ci-dessus montre l'utilisation de contraintes supplémentaires : le document *facture* ne sera créé que si la commande client associée est dans l'état « à facturer ».

Des traitements supplémentaires peuvent aussi être effectués après la création du document en base de données. Ces traitements doivent être définis dans la méthode Doc::postCreated(). Généralement cette méthode est utilisée pour mettre à jour ou pour créer des documents corollaires.

```
function postCreated() {
    if ( $this->revision == 0 ) {
        //// mise à jour de numéro de séquence////
        $this->setValue("FACT_NUMBER",$this->getCurSequence());
    }
}
```

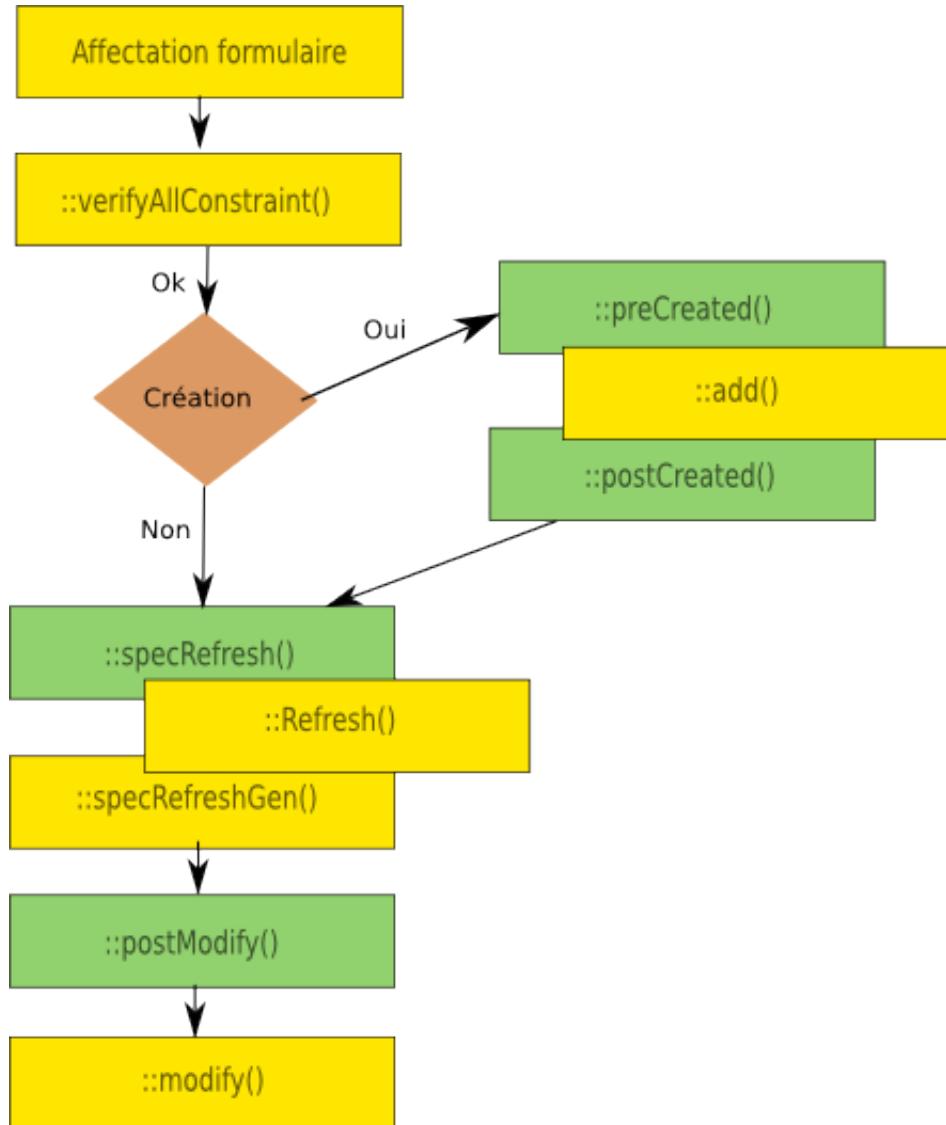
Cette exemple montre la mise à jour d'un attribut calculé qui n'est fait qu'une fois à la création. Cette méthode ne retourne rien.

3.6.2 Contrôle à la modification

Le contrôle d'accès en modification est faite par la méthode Doc::preUpdate(). Cette méthode vérifie que l'utilisateur courant a les droits en modification pour ce document. Ce contrôle peut être inhibé temporairement en utilisant la méthode Doc::disableEditControl(). Le contrôle peut être réactivé en utilisant la méthode Doc::enableEditControl().

La méthode Doc::postModify() est modifiable par les classes de documents. Cette méthode est appelée lors d'une modification par l'utilisateur (par la fonction modcard()), ou lors d'importation de documents. Elle permet d'effectuer des calculs d'attributs sur le document ou sur les documents corollaires.

Si le calcul d'un attribut est invariable en fonction du contenu du document, il est préférable d'effectuer le calcul lors du Doc::postModify() plutôt que lors du Doc::specRefresh() : le calcul ne sera pas ré-effectué à chaque consultation.



Si cette méthode retourne un message d'erreur, la modification est quand même effectuée, mais un message d'erreur est affiché lorsque l'utilisateur effectue la modification du document.

3.6.3 Contrôle à la suppression

La méthode `Doc::delete()` contrôle l'accès à la suppression du document (droit `delete`). Les méthodes `DbObj::preDelete()` et `DbObj::postDelete()` peuvent être surchargées pour effectuer des traitements ou vérifications supplémentaires. Ces deux méthodes sont appelées dans les cas de suppression réelle de la base de données ou de mise en zombie du document. Si la méthode `DbObj::preDelete()` retourne un message d'erreur le suppression n'est pas effectuée.

3.6.4 Autres méthodes

- `Doc::preCopy` et `Doc::postCopy()` : ces méthodes sont exécutées lors de la copie d'un document respectivement avant et après son stockage dans la base `FreeDom`.
- `Doc::preImport()` et `Doc::postImport()` : ces méthodes sont exécutées lors de l'importation d'un document (avant et après le stockage en base).

Pour plus d'information rendez-vous sur la [documentation de l'API FreeDom](#).

3.6.5 Ordre d'appel des méthodes

3.6.5.1 Consultation de document

Lors d'un appel à l'action FDL/FDL_CARD les méthodes "surchargeables" appelées sont :

14. ::specRefresh()
15. ::preConsultation() (version ≥ 2.12.9)

3.6.5.2 Edition de document

Lors d'un appel à l'action GENERIC/Generic_EDIT les méthodes "surchargeables" appelées sont :

1. ::specRefresh()
2. ::preEdition() (version > 2.11.11)

3.6.5.3 Création de document

Lors d'un appel à l'action GENERIC/Generic_MOD les méthodes "surchargeables" appelées sont :

1. ::preCreated()
2. ::postCreated()
3. ::specRefresh()
4. ::postModify()

FIXME internalmedia: freedom_2.14:devel:method.svg

Légende : En jaune, les méthodes non surchargeables, en vert les méthodes surchargeables.

3.6.5.4 Sauvegarde de document

Lors d'un appel à l'action GENERIC/Generic_MOD les méthodes "surchargeables" appelées sont :

1. ::specRefresh()
2. ::postModify()

3.6.5.5 Duplication de document

Lors d'un appel à l'action FREEDOM/FREEDOM_DUPLICATE les méthodes "surchargeables" appelées sont :

1. ::preCopy()
2. ::postCopy()
3. ::specRefresh()
4. ::postModify()

3.6.5.6 Suppression de document

Lors d'un appel à l'action GENERIC/Generic_DEL les méthodes "surchargeables" appelées sont :

1. ::preDelete()
2. ::postDelete()

3.6.5.7 Import de document

Lors d'un appel à l'action FREEDOM/FREEDOM_IMPORT les méthodes "surchargeables" appelées sont :

1. ::preImport()

2. ::preCreated() (*en cas de création seulement*)
3. ::postCreated() (*en cas de création seulement*)
4. ::specRefresh()
5. ::postModify()
6. ::postImport()

3.7 Principales méthodes de la classe Doc

[La classe Doc](#)

3.7.1 lock

La méthode [lock](#) permet de verrouiller un document avec l'utilisateur couramment connecté.

```
include_once("FDL/Class.Doc.php");

$dbaccess=getParam("FREEDOM_DB");
$doc=new_doc($dbaccess,4567);
if ($doc->isAlive()) {
    $err=$doc->lock();
}
```

La méthode [isLocked\(\)](#) permet de savoir s'il y a un verrou posé. La propriété [locked](#) contient l'identificateur système de l'utilisateur qui a verrouillé.

3.7.2 unlock

3.7.3 addRevision

3.7.4 attachTimer

La méthode [attachTimer](#) permet d'attacher un [minuteur](#) à un document.

```
include_once("FDL/Class.Doc.php");

$dbaccess=getParam("FREEDOM_DB");
$doc=new_doc($dbaccess,4567);
$timer=new_doc($dbaccess,47232);
if ($doc->isAlive()) {
    $err=$doc->attachTimer($timer);
    if ($err) print "$err";
}
```

Cette méthode possède un deuxième attribut optionnel pour indiquer quel est le document d'origine du minuteur. Cela est utile par exemple lors qu'un document pilote d'autres documents. Le pilote peut ainsi savoir les minuteurs qu'il a posé. Le troisième attribut optionnel est la date d'exécution de la première action. Si elle n'est pas renseignée, la date d'exécution est égale à maintenant + le premier délai. Si elle est renseignée, cela sera la date posée. Si la date est dans un passé récent (voir [minuteur dépassé](#), cela signifiera quelle sera exécutée à la prochaine vérification (toutes les 5 minutes). Autres méthodes relatives :

16. [unattachTimer](#)
17. [resetDynamicTimers](#)

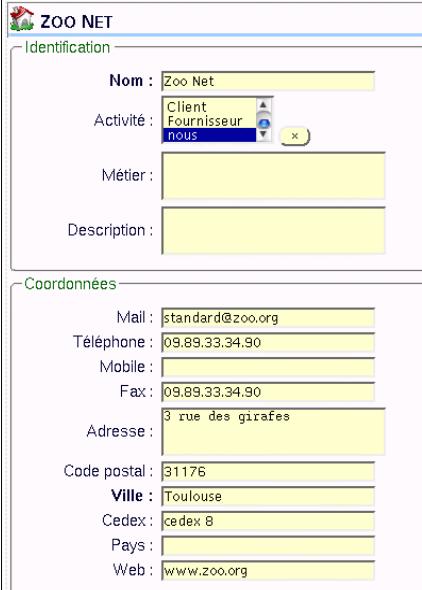
18. [unattachAllTimers](#)
19. [getAttachedTimers](#)

3.8 Vues et éditions particulières

Ce chapitre aborde les techniques liées à la modification de la présentation des documents en consultation et édition... :

3.8.1 Vues par défaut

La présentation des valeurs des documents est effectuée à l'aide de vue. La vue permet de définir la représentation graphique du document. Un même document peut avoir plusieurs vues à sa disposition. Ces différentes vues permettent de représenter le document sous différents aspects soit pour mettre le focus sur certaines parties du document ou afin d'être utilisé par différents intervenants. La vue peut être utilisée pour la consultation, pour l'envoi de mail, pour l'impression ou pour l'édition.

Consultation complète	Consultation résumé	Édition
		

Les vues sont générées à l'aide de la méthode `Doc::viewDoc()`. Cette méthode est utilisée principalement dans les fonctions `editcard()` et `viewcard()`.

Pour construire une vue particulière il est nécessaire de définir un squelette (template) XML. Ce squelette décrit la représentation du document (généralement pour produire du HTML). Ce squelette contient des parties variables qui sont complétées par une méthode spécifique du document (par défaut `Doc::viewdefaultcard()`). La mécanique de génération est effectuée par la classe Layout de WHAT.

La méthode `Doc::viewDoc()` utilise comme paramètre `$layout`. Ce paramètre permet de spécifier la vue à utiliser pour la génération de la représentation. Ce paramètre est une chaîne de caractères composées de trois parties séparées par le caractère ':' (la troisième partie étant facultative)

- la localisation du template ¹⁵
- le nom du template ¹⁶
- un paramètre optionnel -décrit plus tard-

15)nom de l'application fournissant le template

16)celui de la vue en minuscule suffixé de .xml

Par défaut pour la vue de consultation, ce paramètre vaut FDL:VIEWBODYCARD. Cela indique que la méthode viewDoc utilisera le fichier squelette <freedom-root>¹⁷/FDL/viewbodycard.xml et que la méthode appliquée pour compléter le squelette sera ::viewbodycard().

Les trois vues par défaut sont définies par les attributs de la classe Doc. Ces valeurs peuvent être redéfinies par les classes filles . Ces redéfinitions se font dans le fichier Method associé à la famille.

```
//// vue de consultation complète//
var $defaultview = "FDL:VIEWBODYCARD";

//// vue d'édition//
var $defaultedit = "FDL:EDITBODYCARD"

//// vue de consultation résumé//
var $defaultabstract = "FDL:VIEWABSTRACTCARD";
```

On différencie les vues de consultations des vues d'éditions du fait que ces dernières doivent forcément produire du HTML afin de récupérer les informations saisies.

3.8.2 Syntaxe d'un fichier Layout

Le fichier Layout décrit une présentation graphique. Cette représentation définit le contenu du fichier qui sera transmis au navigateur. Ce fichier contient des parties statiques et des parties dynamiques. Les parties dynamiques sont notées entre crochets. Tout ce qui n'est pas entre crochets est statique et ne sera pas modifié.

La génération du layout consiste à remplir les parties dynamiques. Cette instanciation est faite à l'aide de quatre sources :

- les données propres au squelette : ces données doivent être fournies par une fonction spécifique de remplissage.
- Les données de traduction : ces données sont issues de fichier gettext suivant la langue.
- Les données de paramètres : ces paramètres dépendent de l'application (paramètre applicatifs) et de l'utilisateur (préférences utilisateur).
- Les zones : permettent de faire référence à d'autre squelette. La composition d'un squelette peut alors être faite comme par assemblage de layout.

3.8.2.1 Données atomiques

Les données atomiques sont les données qui ont des valeurs non structurées. Une donnée atomique est référencée dans le layout par la syntaxe **[DATA]** DATA étant le nom de la variable. Leur valeur est donnée explicitement par la fonction d'instanciation du layout. L'association entre le nom de la variable du layout et sa valeur est faite avec la méthode Layout::Set().

```
include_once("Class.Layout.php");

$lay = new Layout(getLayoutFile("TEST","test.xml"),$action);
$x=34;
$y=78;
$lay->set("X",$x);
$lay->set("Y",$y);
$lay->set("XplusY",$x+$y);
$lay->set("XfoisY",$x*$y);
print $lay->gen();
```

¹⁷⁾/usr/share/what

test.xml :

```
La somme de [X] + [Y] = [XplusY]
Le produit de [X] + [Y] = [XfoisY]
```

Résultat de la génération :

```
La somme de 34 + 78 = 112
Le produit de 34 + 78 = 2652
```

Comme le montre l'exemple toutes les occurrences d'une même variable sont remplacées dans le fichier. Les noms des variables sont sensibles à la casse de caractères. Ainsi, [X] et [x] identifient deux variables distinctes.

La modification du fichier n'a lieu que lors de la génération (`Layout::gen()`). Les valeurs associées peuvent être modifiées tant que la génération n'a pas eu lieu.

```
include_once("Class.Layout.php");

$lay = new Layout(getLayoutFile("TEST","test.xml"),$action);
$x=34;
$y=78;
$lay->set("X",$x);
$lay->set("Y",$y);

$x1=$lay->get("X"); // récupération de X
$lay->set("X",$x1+1); // mise à jour de X
$lay->set("XplusY",$x+$y);
$lay->set("XfoisY",$x*$y);

print $lay->gen();
```

test.xml :

```
La somme de [X] + [Y] = [XplusY]
Le produit de [X] + [Y] = [XfoisY]
```

Résultat de la génération :

```
La somme de 35 + 78 = 113
Le produit de 35 + 78 = 2730
```

La méthode `Layout::get()` permet de récupérer la valeur précédemment affectée dans le layout.

3.8.2.2 Données listes

La classe `Layout` permet d'utiliser les variables de type liste. Pour indiquer une liste dans un layout, on utilise les mot-clef **BLOCK** et **ENDBLOCK**. L'association se fait à l'aide de la méthode `Layout::setBlockData()`.

```
include_once("Class.Layout.php");

$lay = new Layout(getLayoutFile("TEST","test.xml"),$action);
```

```
$x=3;
$tmul = array(); // tableau pour le bloc MUL
for ($i=1;$i<11;$i++) {
    $tmul[] = array("Y"=>$i,
    "XfoisY"=>$i*$x);
}

$lay->set("X",$x);
$lay->setBlockData("MUL",$tmul);

print $lay->gen();
```

test.xml

```
Table de multiplication de [X]
[BLOCK MUL]
- [X] * [Y] = [XfoisY][ENDBLOCK MUL]
```

Résultat :

```
Table de multiplication de 3

- 3 * 1 = 3
- 3 * 2 = 6
- 3 * 3 = 9
- 3 * 4 = 12
- 3 * 5 = 15
- 3 * 6 = 18
- 3 * 7 = 21
- 3 * 8 = 24
- 3 * 9 = 27
- 3 * 10 = 30
```

La portée des variables atomiques va aussi dans les blocs. Ainsi [X] est instancié dans chaque occurrence de la liste. Chaque ligne de la liste est un tableau qui contient les différentes valeurs du bloc. Chaque ligne peut alors contenir plusieurs variables.

Au contraire des variables atomiques, un bloc ne peut être utilisé qu'une fois dans le layout avec le même identifiant. Dans l'exemple, on ne peut pas avoir deux blocs nommés MUL.

3.8.2.3 Données tableau

Le principe d'instanciation des tableaux consiste à imbriquer deux niveaux de blocs. Un bloc pour délimiter les rangées du tableau et un bloc pour délimiter les colonnes.

Le nom des sous/blocs est dynamique et doit être unique. Dans l'exemple, on choisit de les préfixer par "rmul". Leur unicité est donnée par le numéro de rangée \$i.

```
include_once("Class.Layout.php");

$lay = new Layout(getLayoutFile("TEST","test.xml"),$action);

$x=3;
$tmul = array(); // tableau pour le bloc MUL
$thmul = array(); // tableau pour le header HMUL
for ($i=1;$i<10;$i++) {
```

```

// construction de la tête de tableau
$thmul[] = array("X" => sprintf("%3d", $i));
// construction du corps du tableau
$tmul[] = array("RMUL" => "rmul$i",
"Y" => sprintf("%3d", $i));
$trmul = array(); // nouvelle rangée RMUL
for ($j=1; $j<10; $j++) { // 9 colonnes
    $trmul[] = array("XfoisY" => sprintf("%3d", $i*$j));
}
// on mémorise la nouvelle rangée
$lay->setBlockData("rmul$i", $trmul);
}
$lay->setBlockData("MUL", $tmul);
$lay->setBlockData("HMUL", $thmul);

print $lay->gen();

```

test.xml

```

Table de multiplications
-----
x|[BLOCK HMUL][X]| [ENDBLOCK HMUL]
=====
[Y])[BLOCK [RMUL]][XfoisY]| [ENDBLOCK [RMUL]]
-----+-----+-----+-----+-----+-----+[ENDBLOCK MUL]

```

Résultat :

```

Table de multiplications
-----
x| 1| 2| 3| 4| 5| 6| 7| 8| 9|
=====
1) | 1| 2| 3| 4| 5| 6| 7| 8| 9|
-----+-----+-----+-----+-----+-----+
2) | 2| 4| 6| 8| 10| 12| 14| 16| 18|
-----+-----+-----+-----+-----+-----+
3) | 3| 6| 9| 12| 15| 18| 21| 24| 27|
-----+-----+-----+-----+-----+-----+
4) | 4| 8| 12| 16| 20| 24| 28| 32| 36|
-----+-----+-----+-----+-----+-----+
5) | 5| 10| 15| 20| 25| 30| 35| 40| 45|
-----+-----+-----+-----+-----+-----+
6) | 6| 12| 18| 24| 30| 36| 42| 48| 54|
-----+-----+-----+-----+-----+-----+
7) | 7| 14| 21| 28| 35| 42| 49| 56| 63|
-----+-----+-----+-----+-----+-----+
8) | 8| 16| 24| 32| 40| 48| 56| 64| 72|
-----+-----+-----+-----+-----+-----+
9) | 9| 18| 27| 36| 45| 54| 63| 72| 81|
-----+-----+-----+-----+-----+-----+

```

3.8.2.4 Conditions

Certaines parties du layout peuvent être soumises à une condition. La valeur de la condition se fait par la méthode `Layout::set()` comme pour les données atomiques. Par contre, la valeur du paramètre doit être un booléen. Pour indiquer les parties conditionnelles on utilise les mots-clés **IF** ou **IFNOT** suivis de **ENDIF**.

L'exemple comporte deux valeurs conditionnelles : Interdit et Pair. Le premier est affecté directement à l'aide de la méthode `layout::set()`. Pour valoriser des conditions dans un bloc, il suffit d'affecter la valeur booléenne comme on le ferait pour un mot-clef.

```
include_once("Class.Layout.php");

$lay = new Layout(getLayoutFile("TEST","test.xml"),$action);

$t = array(); // tableau pour le bloc LOOP
for ($i=1;$i<10;$i++) {
    // construction de la liste de 10 nombre
    $t[] = array("X"=>sprintf("%3d",$i),
                 "Pair"=>((($i%2)==0));
}
$lay->setBlockData("LOOP",$t);
$lay->set("Interdit",false);

print $lay->gen();
```

test.xml

```
[IF Interdit]Ne pas afficher[ENDIF Interdit]
Détection de la parité
[IFNOT Interdit]Voir absolument[ENDIF Interdit]
[BLOCK LOOP]
[X)][IF Pair][X] est pair[ENDIF Pair][IFNOT Pair][X] est impair[ENDIF Pair]
[ENDBLOCK LOOP]
```

Résultat :

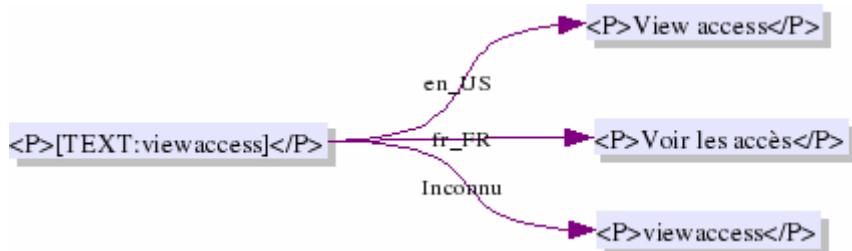
```
Détection de la parité
Voir absolument

1) 1 est impair
2) 2 est pair
3) 3 est impair
4) 4 est pair
5) 5 est impair
6) 6 est pair
7) 7 est impair
8) 8 est pair
9) 9 est impair
```

3.8.2.5 Internationalisation

Si vous voulez des représentations qui dépendent de la langue, il est nécessaire de marquer les textes avec le mot-clef **TEXT**. La langue est stocké dans le paramètre applicatif **CORE_LANG**. Il peut être égal à fr_FR (français) ou en_US (anglais américain). La traduction est faite à l'aide des fichiers au format gettext. Le fichier de traduction utilisé est : <freedom-root>107/what/locale/[en,fr]/LC_MESSAGES/what.mo. Ce fichier est la concaténation de tous les fichiers .mo disponibles dans ce répertoire.

Si vous voulez rajouter un fichier de traduction (.mo), il faut l'insérer dans le répertoire. Ensuite, il suffit de lancer la commande <freedom-root>107/what/whattext pour refaire la concaténation. Il faut enfin relancer le serveur apache pour prendre en compte les nouvelles traductions.



Si le mot est non traduit ou si la langue est inconnue, le mot original est retourné.

3.8.2.6 Paramètres d'environnement

Tous les paramètres de l'environnement applicatif sont remplacés systématiquement. Ces paramètres peuvent avoir des valeurs différentes suivant l'application WHAT utilisée ou suivant l'utilisateur connecté.



Si le paramètre est inconnu, le texte et les crochets correspondants au paramètre restent affichés.

3.8.2.7 Les zones

Les zones permettent de faire référence à d'autres layout. Cela permet une plus grande réutilisabilité des layout. La référence d'une zone est composée de deux parties séparé par ':'.

Par exemple [ZONE FDL:VIEWSCARD], fait référence au layout “FDL/Layout/viewscard.xml”. Si le fichier “FDL/viewscard.php” existe la fonction viewscard() est appelée.

Des paramètres supplémentaires peuvent être indiqués lors de l'appel à la zone. Ces paramètres suivent la même syntaxe que pour les paramètres d'URL. Ils sont transmis à la zone sous forme de variables HTTP et sont accessibles avec la fonction getHttpVars(). Exemple

[ZONE FDL:VIEWSCARD?id=52965]

Vous trouverez des exemples d'utilisation des zones au chapitre suivant :

- [Exemples de Layout](#)

3.8.3 Vue de consultation par défaut



La vue de consultation complète est une vue qui est affichée à l'aide de l'action FDL_CARD. Le layout de cette action inclus la zone FDL:VIEWCARD qui elle-même inclus le corps du document. C'est dans cette dernière zone que l'on peut personnaliser la représentation de la vue de consultation complète par défaut.

Par défaut, la vue est FDL:VIEWBODYCARD. Elle représente des attributs dans l'ordre donné dans la définition des attributs et dans des cadres arrondis.

Pour modifier la vue par défaut, il est nécessaire d'associer un fichier METHOD à la famille. Dans ce fichier, l'attribut defaultview de la classe Doc doit être redéfini. Ensuite il suffit de créer le layout et la fonction de remplissage. Pour les documents cette fonction sera une méthode de l'objet documentaire. Les méthodes de remplissage utilisées pour les vues doivent avoir trois paramètres:

- \$target : nom de la fenêtre graphique qui sera utilisé pour les hyperliens (_self par défaut)
 - \$ulink : booléen : vrai indique s'il faut générer les hyperliens (vrai par défaut)
 - \$abstract : booléen : vrai indique qu'il ne faut pas générer les attributs non résumés (faut par défaut).

La déclaration d'une méthode de remplissage doit toujours être :

```
ma_vue($target="_self",$ulink=true,$abstract=false)
```

La partie modifiable pour la consultation par défaut est la partie visible en jaune clair (FDL:VIEWBODYCARD). Le reste est imposé par l'application par soucis d'homogénéité.

Exemple : fdl_card.xml.

```
<html>
<head>
  <title>[TEXT:Properties:] [TITLE]</title>
  <LINK REL="icon" HREF="[CORE_PUBURL]/CORE/Images/logo-1-mini.ico" >
  <LINK REL="SHORTCUT ICON" HREF="[CORE_PUBURL]/[iconssrc]" >
  <LINK REL="stylesheet" type="text/css" >
```

```

    HREF="[CORE_BASEURL]app=CORE&action=CORE_CSS" >
      <LINK REL="stylesheet" type="text/css"
    HREF="[CORE_BASEURL]app=FDL&action=FDL_CSS" >
      [CSS:REF]

      <style type="text/css">
        [CSS:CODE]
      </style>
      [JS:REF]
      <script language="JavaScript">
        [JS:CODE]
      </script>
    </head>
<body onClick="closeMenu('popupcard');" >
  [ZONE FDL:POPUPCARD]
  <div width="100%" class="tableborder"
    style="cursor:crosshair;" 
    onContextMenu="openMenu(event,'popupcard',1);return false"
    onMouseDown="if (shiftKeyPushed(event)) openMenu(event,'popupcard',1)"
  >
    [ZONE FDL:VIEWCARD]
  </div></body>
</html>

```

3.8.3.1 Consultation avec la méthode par défaut

Pour illustrer les différentes vues, on utilise la famille société. Comme dit précédemment, on attache un fichier METHOD à la cette famille (Method.Society.php).

```
var $defaultview= "FDL:VIEWSOCIETY";
```

Dans ce fichier FDL/Method.Society.php, on indique seulement la référence à la vue de consultation par défaut. La méthode viewsociety() n'existant pas ce sera la méthode Doc::viewDefaultCard() qui sera utilisée pour compléter cette vue. Cette méthode permet d'insérer toutes valeurs d'attributs en indiquant leur identifiant:

- **V_<attr>** : valeur de l'attribut formatté (avec lien si existant et formatage si il est défini)
- **L_<attr>** : description de l'attribut

Cette méthode permet aussi d'afficher la valeur brute des propriétés ou attributs en indiquant leur identification en majuscule.

Afin de simplifier l'affichage, la zone FDL:VIEWFRAME permet d'afficher tous les attributs présents dans un cadre. Cette zone nécessite un paramètre (frameid) pour indiquer l'identificateur du cadre. Cette zone retourne une rangée de tableau HTML (commence par <tr> et fini par </tr>). Cette zone doit donc être incluse dans un tableau (balise père est <table>, <thead>, <tbody> ou <tfoot>).

viewsociety.xml	Représentation
<H1>TEXTE STATIQUE</H1>	 ZOO NET TEXTE STATIQUE
<P>le numéro de téléphone de [V_SI_SOCIETY] est [V_SI_PHONE] Autres renseignements : [L_SI_FAX] : [V_SI_FAX] [L_SI_TOWN] : [V_SI_TOWN] [L_SI_WEB] : [V_SI_WEB] 	 ZOO NET le numéro de téléphone de Zoo Net est 09.89.33.34.90 Autres renseignements : fax : 09.89.33.34.90 ville : Toulouse web : www.zoo.org

<pre><P>les propriétés sont titre : [TITLE] date de modification : [REVDATE] n° de révision : [REVISION] commentaire : <pre>[COMMENT]</pre> </pre>	<p>ZOO NET</p> <p>les propriétés sont</p> <p>titre : Zoo Net date de modification : 1063641531 n° de révision : 0 commentaire :</p> <p>15/09/2003 17:58 [Maitre Chewie] modification 15/09/2003 17:57 [Maitre Chewie] modification 15/09/2003 17:56 [Maitre Chewie] modification 08/09/2003 17:58 [Maitre Chewie] modification 08/09/2003 17:58 [Maitre Chewie] modification 08/09/2003 17:50 [Maitre Chewie] création</p>
<pre><iframe width="100%" src="http://[SI_WEB]"></iframe> <table width="100%"><tr> <td style="background-color:moccasin"> <table width="100%"> [ZONE FDL:VIEWFRAME? frameid=SI_FR_IDENT] </table></td> <td style="background-color:burlywood"> <table width="100%"> [ZONE FDL:VIEWFRAME?frameid=SI_FR_COORD] </table></td> </tr></table></pre>	<p>ZOO NET</p> <p>Woodland Park Zoo Seattle, WA General Info Zoo Events Conservation Zoo Travel Animal Facts Education Memberships</p> <p>Woodland Park Zoo Seattle, WA</p> <p>Identification</p> <p>Nom : Zoo Net Activité : nous</p> <p>Coordonnées</p> <p>Mail : standard@zoo.org Téléphone : 09.89.33.34.90 Fax : 09.89.33.34.90 Adresse : 3 rue des girafes Code postal : 31176 Ville : Toulouse Cedex : cedex 8 Web : www.zoo.org</p>

Le principal inconvénient des vues personnalisées est que la vue est très liées aux attributs de la famille. Ainsi chaque changement d'attribut (ajout/suppression) risque d'impliquer une modification de la représentation. Si la vue indique explicitement des attributs ou des cadres, il faut rajouter ces nouveaux attributs ou cadres à la représentation à chaque changement.

Pour les attributs de type tableaux, il sont visualisable par une zone comme pour les cadres. La syntaxe de la zone est la suivante : [ZONE FDL:VIEWARRAY?arrayid=SI_T_SITES] par exemple pour le tableau SI_T_SITES.

3.8.3.2 Consultation avec méthode spécifique

L'utilisation de la méthode spécifique offre plus de possibilités dans la représentation. Cette méthode va permettre de calculer la représentation en fonction du contenu.

L'objet layout du document est l'attribut Doc::lay de l'objet document. On utilise cet attribut pour insérer les valeurs dans le squelette. Vue circulaire :

Représentation
<pre>viewsociety.xml <div style="height:[dy]px" border=1> [BLOCK ATTR] <P style="position:absolute;left:[x]px;top:[y]px" title="[frame]/[label]"> [value] </P> [ENDBLOCK ATTR] </div> Method.society.php public \$defaultview= "FDL:VIEWSOCIETY"; public function viewsociety(\$target=_self,\$ulink=true,\$abstract=false) { \$rx=300; // rayon X \$ry=200; // rayon Y if (\$abstract){ \$listattr = \$this->GetAbstractAttributes(); } else { \$listattr = \$this->GetNormalAttributes(); } reset(\$listattr); \$tattr=array(); while (list(\$i,\$attr) = each(\$listattr)) {</pre>

```

$value = chop($this->GetValue($i));
if ($value != "") {
    $tattr[] = array("value" => $this->GetHtmlValue($attr, $value, $target, $ulink),
                    "label" => $this->getLabel($i),
                    "frame" => $this->getLabel($attr->fieldSet->id) );
}
}
reset($tattr);
$delta = 2*pi()/count($tattr);
while (list($k,$v) = each($tattr)) { // placement dans le cercle
    $tattr[$k]['y'] = $ry + sin($delta*$k)*$ry + 20;
    $tattr[$k]['x'] = $rx + cos($delta*$k)*$rx + 20;
}
$this->lay->setBlockData("ATTR", $tattr); // enregistrement des valeurs
$this->lay->set("dy", $ry*2);
}

```



3.8.4 Vues de consultation spécifiques

Les vues de consultations qui ne sont pas par défaut, n'ont pas les mêmes contraintes que celle par défaut. Elles ne sont pas obligatoirement un corps HTML et peuvent donc revêtir des formats plus variés comme le RTF, XML ou CSV.

Comme pour les vues par défaut, la création d'une nouvelle vue nécessite un squelette et une méthode de remplissage (optionnelle : par défaut Doc::viewDefaultCard()). L'appel à cette vue peut se faire avec l'action IMPCARD de l'application FDL.

[http://zoo.abc.com/what/?
sole=Y&app=FDL&action=IMPCARD&zone=FDL:VIEWBODYCARD&id=1419](http://zoo.abc.com/what/?sole=Y&app=FDL&action=IMPCARD&zone=FDL:VIEWBODYCARD&id=1419)

[http://zoo.abc.com/what/?
sole=Y&app=FDL&action=IMPCARD&zone=FDL:VIEWSOCIETY&id=1419](http://zoo.abc.com/what/?sole=Y&app=FDL&action=IMPCARD&zone=FDL:VIEWSOCIETY&id=1419)

Deux options spéciales peuvent être utilisées lors de l'appel de la zone.

- **S:** signifie que le squelette est autonome, qu'il ne nécessite pas une encapsulation HTML. Cette option est obligatoire pour tous les squelettes non HTML.
- **T :** pour les squelettes HTML, il signifie qu'aucune entête ne sera affichée (pas de titre, ni d'icônes). La barre de menu n'est pas affichée non plus
- **U :** idem T mais en plus pas de CSS définie (version > 2.9.2)
- **V :** pour les squelettes HTML, idem T mais la barre de menu est affichée (version > 2.9.2)
- **B :** pour les squelettes non HTML binaire (tel que fichier openDocumentText .odt) (version > 2.9.2)

FDL:VIEWSOCIETY:	FDL:VIEWSOCIETY:T	FDL:VIEWSOCIETY:S
-------------------------	--------------------------	--------------------------

 ZOO NET Toulouse cedex 8  31176 www.zoo.org 3 rue des girafes Zoo Net 09.89.33.34.90 09.89.33.34.90 standard@zoo.org <p>Vue par défaut avec entête.</p>	Toulouse cedex 8 31176 www.zoo.org 3 rue des girafes Zoo Net 09.89.33.34.90 09.89.33.34.90 standard@zoo.org <p>L'entête n'est pas affichée.</p>	Toulouse cedex 8 31176 www.zoo.org 3 rue des girafes Zoo Net 09.89.33.34.90 09.89.33.34.90 standard@zoo.org <p>Ici le document HTML est incomplet, puisque la balise racine est DIV. Il n'y a pas non plus de css, la représentation du document est brute.</p>
---	---	---

3.8.5 Vues de consultation OpenDocument Text

 (version > 2.9.2) 

Les vues de ODT (OpenDocument Text), permettent de proposer à l'utilisateur un rendu avec plus de possibilités de composition que le rendu HTML. Cette vue est un fichier au format ODT. Celui-ci peut être produit par le traitement de texte [OpenOffice.org Writer](#).

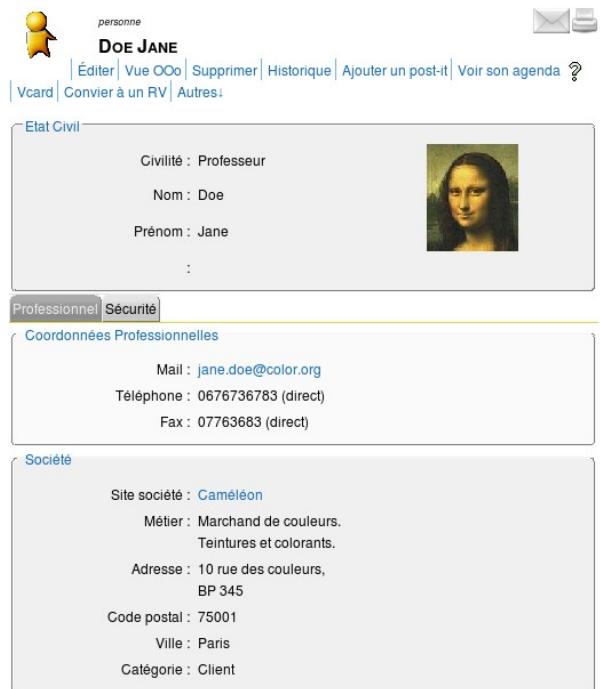
Ce fichier doit être placé, comme pour les squelettes HTML, dans le répertoire Layout de votre application.

Pour référencer le fichier template **fixme internalmedia: freedom_2.14:devel:vue_edition:personne.odt** il faut utiliser la zone : **FDL:personne.odt:B**.

 Ne pas oublier le **B** qui indique que le template est un fichier binaire.

La syntaxe pour les éléments de template est le même que pour le template HTML. Par contre les balises **BLOCK/ENDBLOCK**¹⁸ et **ZONE** ne sont pas prises en compte.

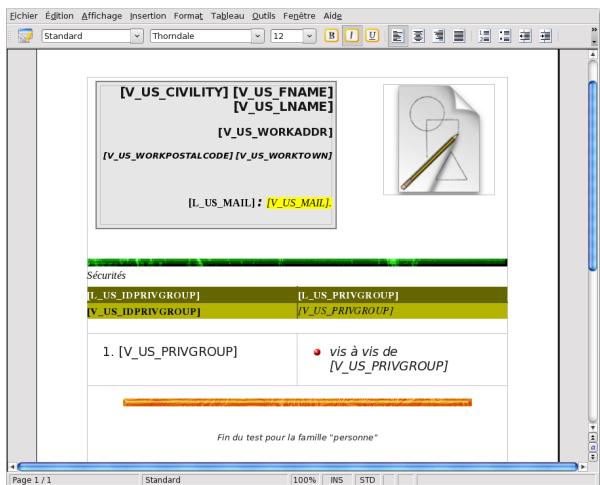
Exemple : Soit le document suivant :



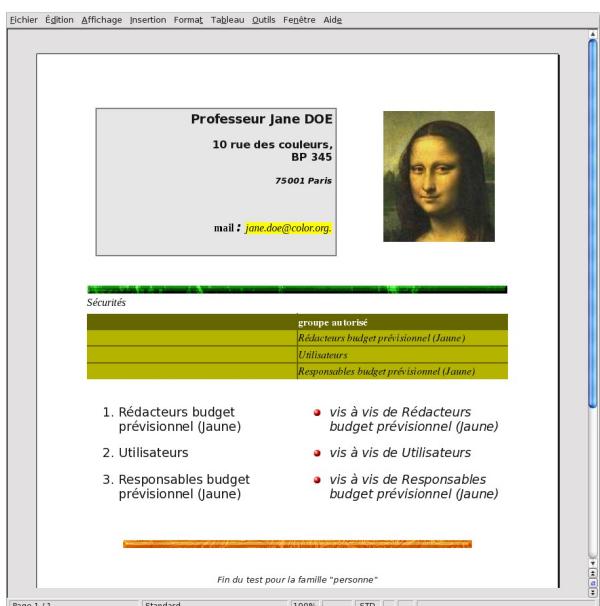
The screenshot shows a person's profile in OpenOffice.org Writer. The top navigation bar includes 'personne', 'Doe JANE', 'Editer', 'Vue Ooo', 'Supprimer', 'Historique', 'Ajouter un post-it', 'Voir son agenda', and 'Vcard'. Below the navigation is a 'Etat Civil' section containing 'Civilité : Professeur', 'Nom : Doe', and 'Prénom : Jane', accompanied by a small portrait of the Mona Lisa. A tab labeled 'Professionnel' is selected, showing 'Coordonnées Professionnelles' with 'Mail : jane.doe@color.org', 'Téléphone : 0676736783 (direct)', and 'Fax : 07763683 (direct)'. At the bottom, the 'Société' section lists 'Site société : Caméléon', 'Métier : Marchand de couleurs.', 'Teintures et colorants.', 'Adresse : 10 rue des couleurs, BP 345', 'Code postal : 75001', 'Ville : Paris', and 'Catégorie : Client'.

18)la méthode `::setBlockData` traditionnelle est par conséquent inopérante

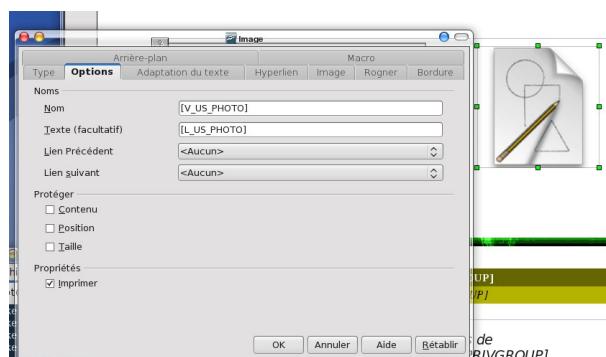
En utilisant le template suivant :



On obtient le résultat suivant : ?
sole=Y&app=FDL&action=FDL_CARD&zone=FDL:personne.odt:B&id=1342



Dans le résultat produit, les attributs V_US_IDPRIVGROUP, ne sont pas affichés car leur visibilité est caché (H - hidden).



Pour incorporer des images, il faut insérer une image quelconque dans le fichier (menu Insertion/Image/A partir d'un fichier (sous Oo writer ! pas de lien)).

Pour référencer l'attribut image, cliquez sur le menu contextuel de l'image et choisissez 'image'; puis renseignez le nom dans l'onglet Options avec l'identifiant de l'attribut freedom entre

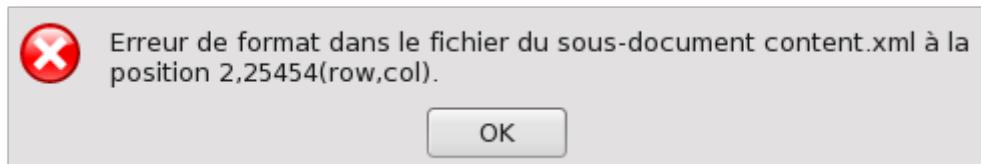
crochets.

En ce qui concerne la taille de l'image, la largeur sera conservée. La hauteur sera calculée en fonction du format de l'image pour ne pas avoir de déformation d'échelle.

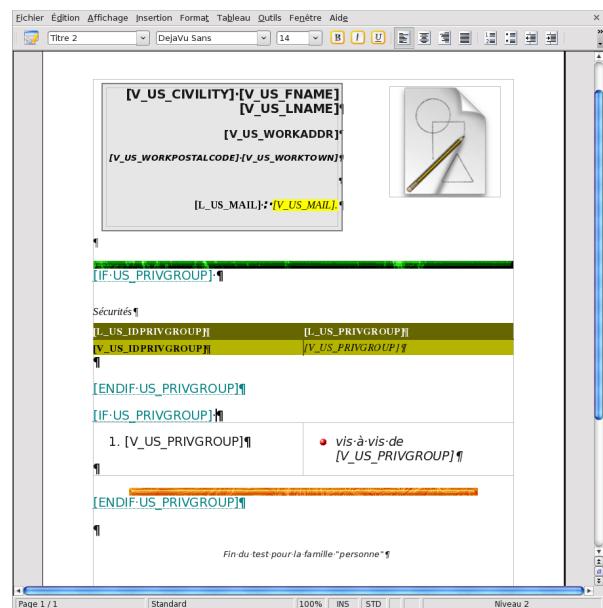
Pour **les attributs inclus dans des tableaux**, indiquez la référence à la valeur dans la rangée d'un tableau déjà présent. Voir dans l'exemple précédent de l'attribut V_US_PRIVGROUP. Ces attributs peuvent aussi être utilisés dans des listes à puces ou numérotées. S'ils sont utilisés hors de ce contexte, la valeur retournée sera l'ensemble des valeurs séparées par une tabulation.

Les conditions IF/ENDIF peuvent être utilisées en respectant la structure du document ODT : il faut que le niveau (structure XML du fichier OpenDocument Text) de la balise IF soit le même que le ENDIF. Par exemple, on ne peut pas mettre un IF au milieu d'un tableau et le ENDIF à l'extérieur du tableau : le IF et le ENDIF doivent être dans la même cellule du tableau. Il est possible de les utiliser dans le même paragraphe ou entre plusieurs paragraphes de même niveau.

En cas d'erreur de structure, un message d'erreur apparaîtra :



Exemple : FIXME internalmedia: freedom_2.14:devel:vue_edition:personne_if.odt



Il est possible d'utiliser une méthode particulière pour instancier le template comme pour les vues HTML. Le nom de la méthode et le nom en minuscule du fichier template sans extension. Exemple pour zone 'FDL:viewpersonne.odt:B', la méthode appelée sera 'viewpersonne'.

Comme pour les templates HTML, il est possible d'appeler la méthode '\$this->lay->set()' afin de renseigner des clefs spécifiques. Il est en plus possible d'utiliser la méthode OOoLayout::setColumn afin de renseigner une liste de valeurs pour d'être insérée dans une liste à puces ou dans un tableau. ! Dans ce cas le nom de la clef doit commencer par 'V_' et être en majuscule. De plus la méthode OOoLayout::setBlockData() peut être utilisé pour renseigner un tableau. Afin d'avoir la même signature que pour Layout::setBlockData(), le nom du bloc est resté mais ne sert à rien. Par contre les clefs des valeurs (pour les colonnes) doivent aussi commencer par 'V_' et être en majuscule.

Exemple :

```
function viewtest() {
```

```

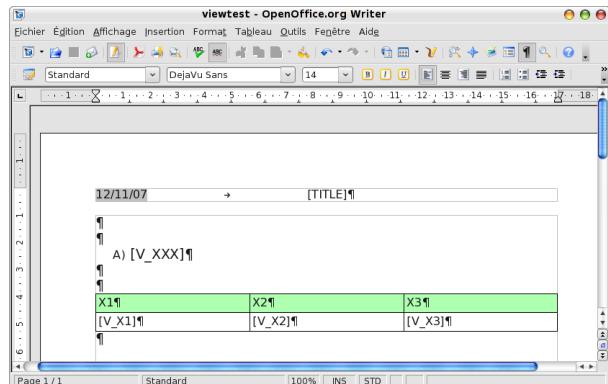
$t[] = array("V_X1"=>'A',
            "V_X2"=>'1',
            "V_X3"=>"La");
$t[] = array("V_X1"=>'B',
            "V_X4"=>'2',
            "V_X3"=>"Si");
$t[] = array("V_X1"=>'C',
            "V_X2"=>'3',
            "V_X3"=>"Do");

$x=array('a','b','c','d');

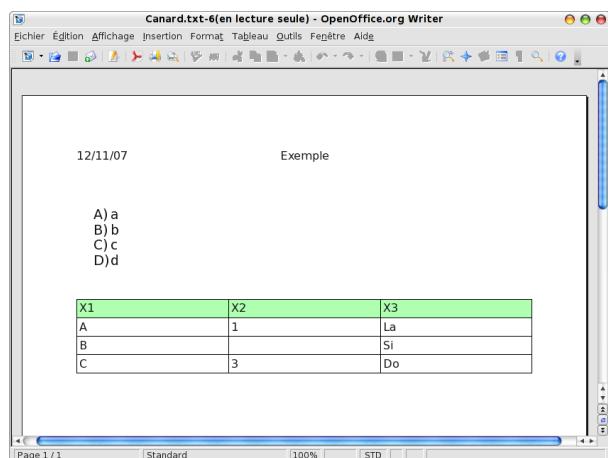
$this->lay->setColumn("V_XXX",$x);
$this->lay->setBlockData("",$t);
}

```

Avec le template suivant : viewtest.ott



On obtient le résultat : (zone = FDL:viewtest.ott:B)



3.8.6 Vues de consultation avec transformation

⚠ (version > 2.9.5) ⚠

Une transformation peut être appliquée après la composition de cette vue. Cette transformation est effectuée par le [module de transformation](#). Pour appliquer une transformation pdf il suffit de spécifier le moteur à la fin de la zone comme ceci : FDL:viewperson.ott:B:pdf.

Si vous voulez faire une transformation [pdfa](#) (pdf pour archivage) il suffit de changer le moteur : FDL:viewperson.odt:B:pdfa. Vous avez alors le fichier pdf avec les polices.

Ceci indique à freedom de lancer une demande de transformation (ici moteur pdf) à partir de la vue produite. L'utilisateur doit alors patienter le temps que la conversion soit effectuée.

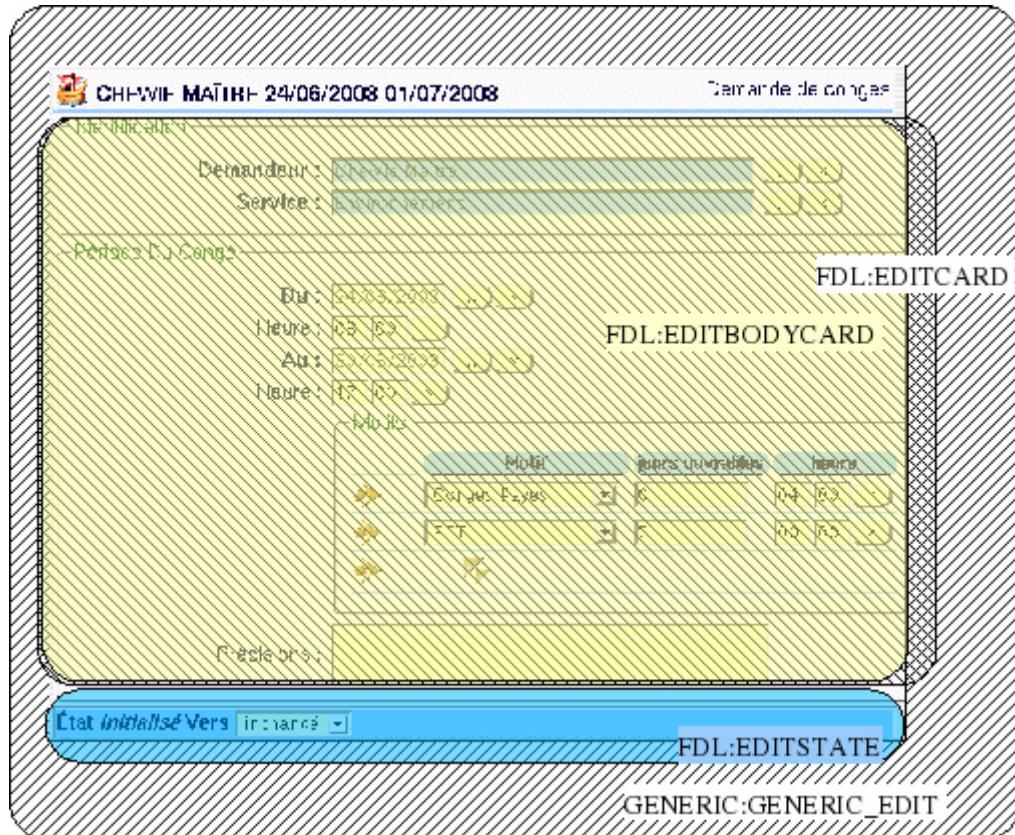
On peut aussi lancer une vue pdf à partir d'un template HTML : exemple : FDL:VIEWBODYCARD::pdf. Dans ce cas le résultat risque de paraître un peu brut par rapport à la vue HTML. En effet le convertisseur PDF utilisé ne comprend pas toutes les possibilités des CSS utilisées. Par contre, pour ce type de rendu à partir de HTML, vous pouvez créer vos propre



template en utilisant du HTML *basique*.

3.8.7 Vues d'édition

Les vues d'éditions sont forcément au format HTML. Les squelettes spécifiques d'édition sont inclus entre des balises FORM pour le formulaire d'envoi. C'est l'action GENERIC_EDIT ou FREEDOM_EDIT qui est utilisé pour afficher les formulaires d'édition.



La vue d'édition modifiable est celle affichée en jaune (par défaut FDL:EDITBODYCARD). Pour modifier la vue d'édition par défaut, il faut associer un fichier METHOD à la famille de document. Dans ce fichier, il suffit de modifier l'attribut \$defaultedit pour lui assigner une autre zone.

```
<html>
  <head>
    <title>[TEXT:edition: ] [TITLE]</title>
    <LINK REL="stylesheet" type="text/css"
      HREF=" [CORE_BASEURL]app=CORE&action=CORE_CSS" >
    <LINK REL="stylesheet" type="text/css"
      HREF=" [CORE_BASEURL]app=FDL&action=FDL_CSS" >
    [CSS:REF]
  </head>
  <body class="freedom"
    onLoad="editOnLoad();autoHresize()"
    onsubmit="selectall()"
    onUnLoad="closechoose();pleaseSave(event) ;autoUnlock('[id]')"
    onResize="resizeInputFields()">

    <form id="fedit"
      class="fborder"
      name="modifydoc">
```

```

onsubmit="document.isSubmitted=true;selectall();"
method="POST" ENCTYPE="multipart/form-data"
action="[CORE_STANDURL]&app=[APPNAME]&action=GENERIC_MOD&id=[id]&dirid=
[dirid]&classid=[classid]" >


|                                                                                                                              |  |  |
|------------------------------------------------------------------------------------------------------------------------------|--|--|
| <IMG border="0" alt="icon" align="absbottom" height="25px" SRC="[iconsrv]"> <span class="FREEDOMTextBigTitle">[TITLE]</span> |  |  |
|                                                                                          |  |  |
| <span class="FREEDOMText">[FTITLE]</span>                                                                                    |  |  |


[ZONE FDL:EDITCARD]



|                                                                                                                                                                                                                                                      |  |  |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--|--|
| <input name="catgid" type="hidden" value="[[catgid]]"/> <input id="iSubmit" type="submit" value="[[editaction]]"/> onmousedown="multiple_for_select(); onclick="if (!canmodify()) return false; document.isSubmitted=true;enableall();return true"/> |  |  |
| <input type="button" value="[[TEXT:Cancel]]"/> onclick="document.isCancelled=true;document.location.href='[CORE_STANDURL]&app=GENERIC&action=GENERIC_LOGO'"                                                                                          |  |  |


```

Comme pour les vues de consultation si la méthode n'existe pas, la méthode de remplissage utilisée sera Doc::viewDefaultCard(). Cette méthode ne construit pas de champ de saisie. Pour utiliser les variables L_<attr> et V_<attr> comme pour la méthode Doc:viewAttr(), on appellera la méthode Doc::editAttr() dans le méthode spécifique d'édition.

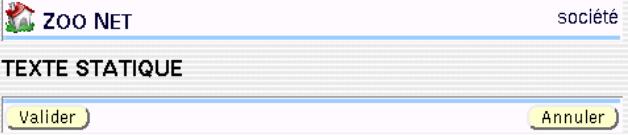
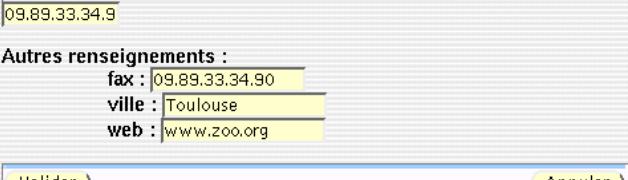
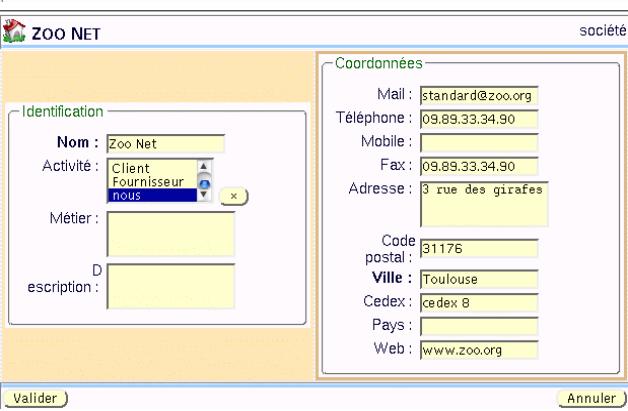
Le fichier Method.Society.php sera :

```

var $defaultedit= "FDL:EDITSOCIETY";

function editsociety() {
    $this->editattr();
}

```

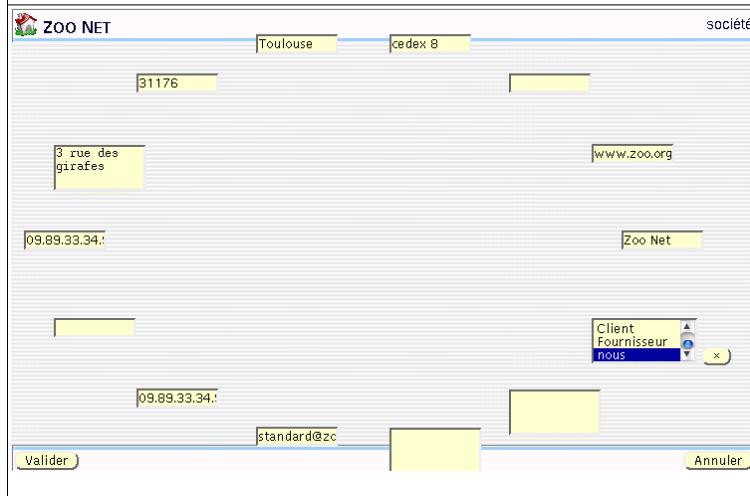
<pre><H1>TEXTE STATIQUE</H1></pre> <pre><H1>TEXTE STATIQUE</H1><P>le numéro de téléphone de [V_SI_SOCIETY] est [V_SI_PHONE] Autres renseignements : [L_SI_FAX] : [V_SI_FAX] [L_SI_TOWN] : [V_SI_TOWN] [L_SI_WEB] : [V_SI_WEB] </pre>	  
<pre><table width="100%" cols="2"><tr> <td style="background-color:moccasin"> <table width="100%" > [ZONE FDL:EDITFRAME?frameid=SI_FR_IDENT] </table></td> <td style="background-color:burlywood"> <table width="100%"> [ZONE FDL:EDITFRAME?frameid=SI_FR_COORD] </table></td> </tr></table></pre>	

Les méthodes d'édition particulières n'ont pas d'argument. Elles offrent les mêmes possibilités de représentation que les vues particulières de consultation. Ces méthodes doivent être utilisées dans des cas d'éditions qui ne sont pas prévus par les types d'attributs offerts par FREEDOM. Elles peuvent être aussi utilisées pour des documents avec beaucoup d'attributs afin de scinder différentes éditions du document.

Pour les attributs de type tableaux, il sont éditables par une zone comme pour les cadres. La syntaxe de la zone est la suivante : **[ZONE FDL:EDITARRAY?arrayid=SI_T_SITES]** par exemple pour le tableau SI_T_SITES.

Vue circulaire spécifique
<pre>viewsociety.xml <div style="height:[dy]px" border=1> [BLOCK ATTR] <P style="position:absolute;left:[x]px;top:[y]px" title="<label>"> [value] </P> [ENDBLOCK ATTR] </div> Method.society.php var \$defaultedit= "FDL:EDITSOCIETY"; function editsociety() { include_once("FDL/editutil.php"); \$rx=300; // rayon X \$ry=200; // rayon Y \$listattr = \$this->GetInputAttributes(); reset(\$listattr); \$tattr=array(); while (list(\$i,\$attr) = each(\$listattr)) { \$value = chop(\$this->GetValue(\$i)); \$tattr[] = array("value"=>getHtmlInput(\$this, \$attr, \$value), "label"=>\$this->getLabel(\$i),</pre>

```
"frame"=>$this->getLabel($attr->fieldSet->id);
}
reset($tattr);
$delta = 2*pi()/count($tattr);
while (list($k,$v) = each($tattr)) {
    $tattr[$k]["y"]=$ry+sin($delta*($k))*$ry+20;
    $tattr[$k]["x"]=$rx+cos($delta*$k)*$rx+20;
}
$this->lay->setBlockData("ATTR", $tattr);
$this->lay->set("dy",$ry*2);
$this->lay->set("ry",$ry/2);
}
```



Comme pour les vues de consultation, les zones définissant les éditions peuvent avoir des options de représentation.

option	Définition	Représentation
	Édition par défaut zone=FDL:EDITBODYCARD	
S	Les entêtes et les boutons « "sauver" » et « "annuler" » ne sont pas affichés. Par contre le document contient bien le formulaire (balise <form>) pour l'édition. Par contre les boutons pour les changements d'état sont présents lorsqu'il y a un cycle de vie. zone=FDL:EDITBODYCARD:S	

V	Idem S sauf que les boutons pour le changement d'état ne sont pas affichés.	
U	Le squelette est affiché de manière brute : pas de formulaire prédéfini. zone=FDL:EDITBODYCARD:U	<p>Identification</p> <p>raison sociale : Zoo Net description : Parc animalier</p> <p>Coordonnées</p> <p>mail : téléphone : 01 02 03 04 01 fax : 01 02 03 04 00 adresse : 3 rue des girafes code postal : ville : Toulouse cedex : web : animaldiversity.umz.ummz.um</p>
T	L'entête n'est pas affiché. Les boutons « " sauver " » et « " annuler " » sont affichés zone=FDL:EDITBODYCARD:T	<p>Identification</p> <p>Raison sociale : Zoo Net Description : Parc animalier</p> <p>Coordonnées</p> <p>Mail : Téléphone : 01 02 03 04 01 Fax : 01 02 03 04 00 Adresse : 3 rue des girafes Code postal : Ville : Toulouse Cedex : Web : animaldiversity.umz.ummz.um</p> <p>Sauver Annuler</p>

3.8.8 Interface spécifique de contrôle d'un formulaire document



Des aides spécifiques sur un formulaire documentaire peuvent être mises en place. Ces aides permettent de remplir des attributs en utilisant une interface plus spécialisée.

Ces aides sont déclarées dans le fichiers de description des familles comme pour les aides à la saisies classiques. La différence est quelles font références à une interfaces dédiée et à un contrôleur d'interface dédiés.

Exemple :

	idattr	idframe	label	T	A	type	ord	vis	need	link	phpfile	phpfunc
ATTR	EN_T_ESPÉCÉS	EN_IDE_NTIFICATION	liste espèce	N	N	array	30	W			zoo.php	ZOO:zoo_searchspecies(A,D,I,D,en_nom):
ATTR	EN_ESP_PRÔTECTED	EN_T_ESPÉCÉS	protégé	N	N	image	40	S				

ATTR	EN_ESP ECÉ	EN_T_E SPÉCÉ S	espèce	N	N	docid("ZOO_E SPECÉ")	50	W			zoo. php	ZOO:zoo_ searchspecies(A,D,I D,en_nom):en_espe ce,en_co mment
ATTR	EN_CO MMENT	EN_T_E SPÉCÉ S	comme	N	N	text	60	W				
ATTR	EN_CA PACITE	EN_IDE NTIFIC ATION	capacit é	N	N	int	70	W				

La notation 'ZOO:zoo_searchspecies(A,D,ID,en_nom):' indique l'interface est le fichier zoo_searchspecies.xml et le contrôleur la fonction zoo_searchspecies() du fichier zoo.php. Puisqu'il n'y a pas de retour défini (après les :), cela indique que c'est l'interface qui explicitement modifiera les attributs voulus.

Fichier zoo.php :

```
function zoo_searchspecies(&$action,$dbaccess,$id,$nom) {
    // print "DB=$dbaccess, NOM=$nom ID=$id";
    $action->lay->set("enclosname",$nom);
    $doc=new_doc($dbaccess,$id);

    if ($doc->isAlive()) {
        $action->lay->set("CAPACITY",$doc->
getValue("en_capacite",__("zoo:Capacity not set")));
    } else {
        $action->lay->set("CAPACITY",__("zoo;Capacity not set"));
    }
}
```

Dans la fonction contrôleur on retrouve les paramètres déclarés dans la famille (A ⇒ &\$action, D ⇒ \$dbaccess, ID ⇒ \$id, en_nom ⇒ \$nom). Pour piloter le template, il est nécessaire de déclarer le paramètre action (A). Si le template n'a pas besoin de contrôleur, il est néanmoins nécessaire de déclarer la fonction dans le fichier indiqué dans phpfile même si elle ne fait rien. Fichier zoo_searchspecies.xml

```
[ZONE FDL:HTMLHEAD?title=[TEXT:Special Species]]
<script type="text/javascript" src="lib/data/fdl-data.js"></script>
<script>
    var C=null; // the main context

    function searchSpecies() {
        C=new Fdl.Context();
        if (! C.isConnected()) {
            alert('error connect:'+C.getLastErrorMessage());
        }
        if (C){
            var sel = document.getElementById("resultTab");
            var d = C.getSearchDocument();
            var f=new Fdl.DocumentFilter({family:'ZOO_ESPECE',
                                         criteria:
                                         [{operator:'~*', left:'title',
                                           right:docume
                                         nt.getElementById('specie').value}]}));
            var dl=d.search({filter:f});
        }
    }
</script>
```

```

        var content=dl.getDocuments();
        for (var i in content) {
            var doc=content[i];
            if (doc && doc.isAlive()) {
                var o = new
Option(doc.getTitle(),doc.getProperty('id'));
                sel.options[sel.options.length]=o;
            }
        }
    }

    function getReference() {
        var ref=Ih.docGetFormValue('en_reference');

        return ref;
    }
    function insertDataFromSelection() {
        var titles=[];
        var icons=[];
        var ids=[];

        var sel=document.getElementById("resultTab");
        for (var i=0;i < sel.options.length;i++) {
            if (sel.options[i].selected){
                var d = C.getDocument({id:sel.options[i].value,
useCache:true});
                console.log(d);
                Ih.docAddTableRow({en_espece:
{id:sel.options[i].value,title:sel.options[i].text,unique:true},
en_comment:'coucou',
en_espprotected:
(d.getValue('es_protegee')=='1')?
{url:'Images/pandared.png',id:'pandared.png'}:{url:'Images/pandagreen.png',
id:'pandagreen.png'},
en_photo:
{url:d.getDisplayValue('es_photo',{url:true}),id:d.getValue('es_photo')}});
            }
        }
        addEvent(window,"load",autoWresize);
        addEvent(window,"load",searchSpecies);

        function setCapacity(inp,delta) {
            if (isNaN(inp.value)) inp.value=0;
            if (delta) inp.value=parseInt(inp.value)+delta;
            Ih.docSetFormValue({en_capacite:inp.value})
        }
    </script>
<h3><b>[TEXT:zoo:Species for] : [enclosname]</b></h3>
<input type="text" id="specie"/>
<input type="button" value="[TEXT:zoo:Search species]"
onclick="searchSpecies()"></input>
<br/>
<select id="resultTab" multiple="multiple">
</select>
<br/>

<input type="button" value="[TEXT:Insert into list !]"
onclick="insertDataFromSelection()"></input>

```

```

<br/>
<input id="capacity" value="[CAPACITY]" />
<input type="button" value="+" onclick="setCapacity(document.getElementById('capacity'),+1)" /></input>
<input type="button" value="-" onclick="setCapacity(document.getElementById('capacity'),-1)" /></input>
<input type="button" value="[TEXT:zoo:Change capacity]" onclick="setCapacity(document.getElementById('capacity'),null)" /></input>

<input type="button" value="[TEXT:zoo:Retrieve reference]" onclick="document.getElementById('reference').value=getReference()" /></input>
<input type="text" id="reference"/>

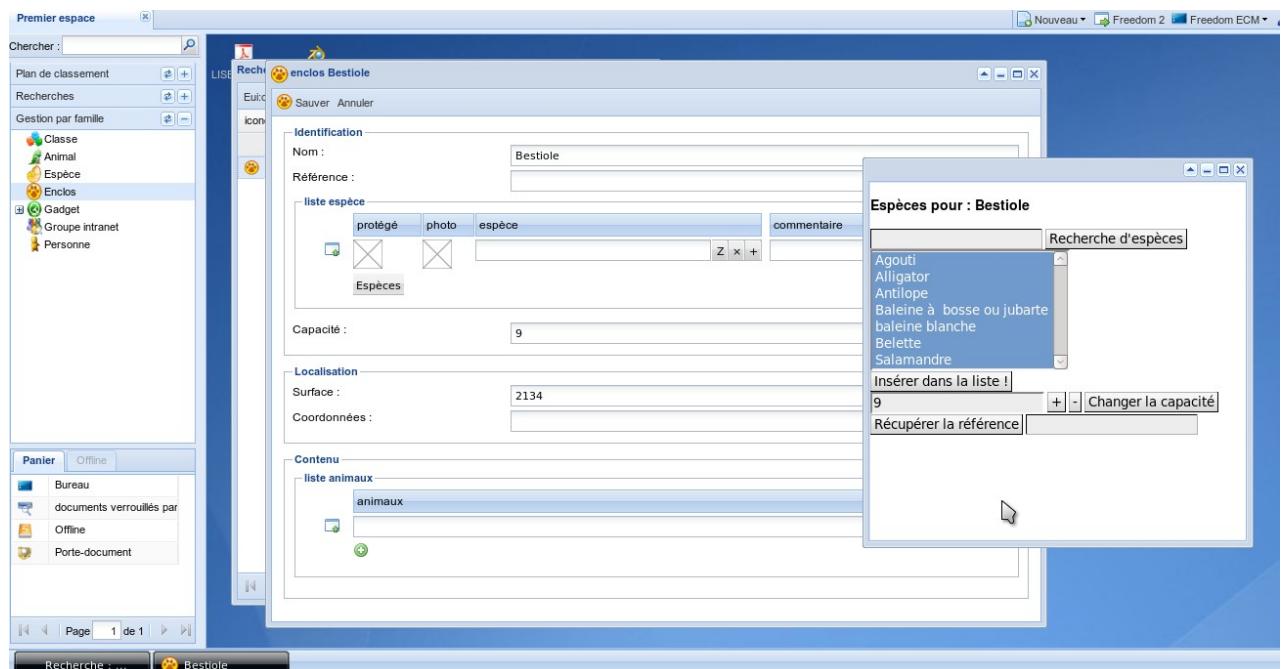
[ZONE FDL:HTMLFOOT]

```

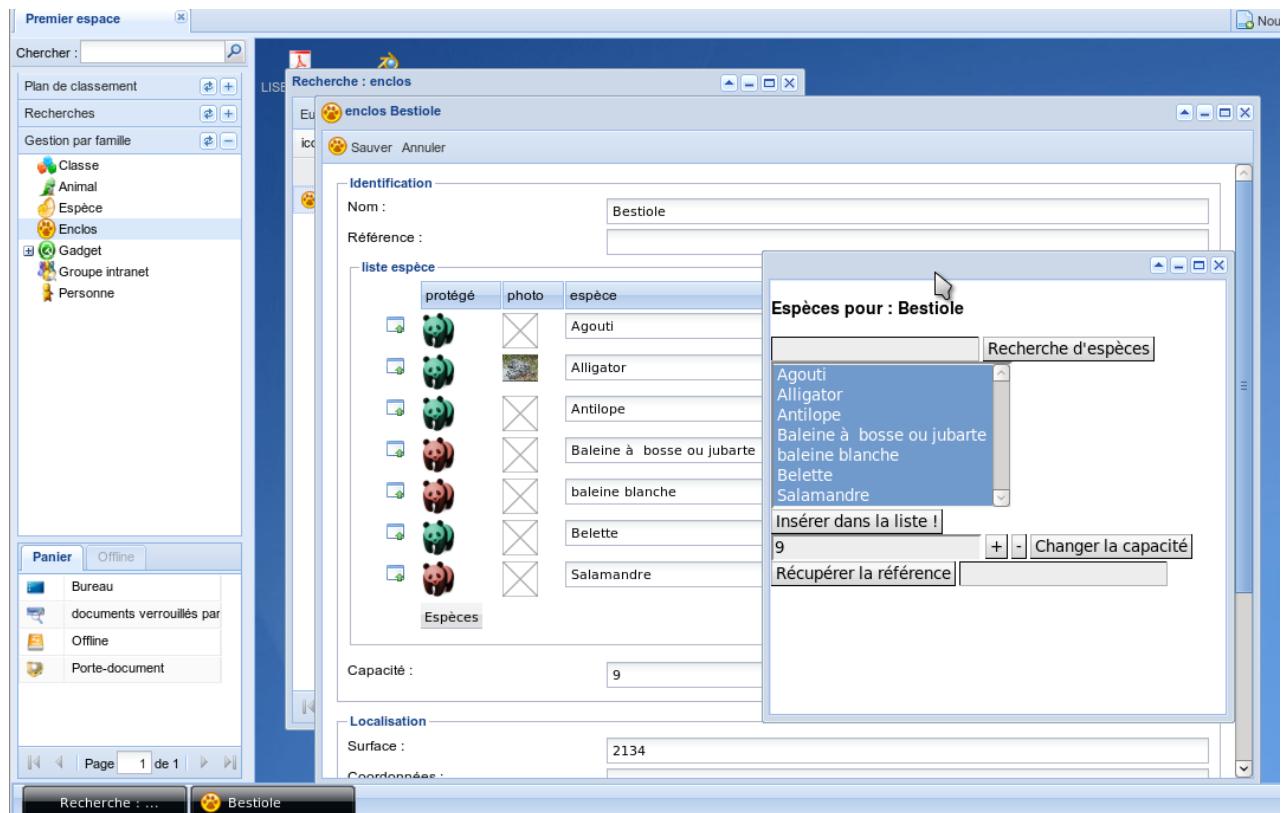
Ce template doit comporter les headers HTML standard de freedom afin d'incorporer automatiquement les scripts nécessaires au contrôle du formulaire. Pour cela il est fortement conseillé d'utiliser la zone [ZONE FDL:HTMLHEAD].

Le résultat par l'image :

Après avoir appuyé sur le bouton “espèces” :



Après avoir appuyé sur le bouton “insérer dans la liste !” :



Cette interface permet d'illustrer 3 aspects des interfaces d'aides à la saisie :

3.8.8.1 Manipulation de tableau

L'interface dispose de la fonction javascript Ih.docAddTableRow permettant d'ajouter une rangée à un tableau. Cette fonction a comme argument un objet de configuration avec comme index le nom de l'attribut et comme valeur la valeur de l'attribut. Cette fonction retourne true si l'ajout a été effectuée.

Deux types ont comme valeur un objet : les types docid et les types "images".

Type docid

Le docid doit déclarer son identifiant et son titre.

```
Ih.docAddTableRow({en_espece:{id:9,title:'racine',unique:true}});
```

De manière optionnelle, il est possible d'ajouter l'argument unique afin d'indiquer que cette colonne ne peut avoir 2 rangée avec le même identifiant. L'option unique ne peut être utilisé que pour une seule colonne 'docid' du tableau.

Type image

Uniquement si l'image est déclarée en static (visibilité S).

L'image doit déclarer son url et son identification interne. L'identification interne est une référence au coffre. Cette référence est obtenu par la valeur d'un attribut image déjà existant.

```
Ih.docAddTableRow(en_imgstatic:{url:d.getValue('es_photo'),{url:true}},id:d.getValue('es_photo'));
```

Il est aussi possible de déclarer une image qui est disponible sur le serveur. Cette image aura pu être déposée par votre module (répertoire images). L'id doit contenir le nom du fichier images (sans chemin) et l'url doit être Le nom précédé du chemin relatif Images.

```
Ih.docAddTableRow(en_imgstatic:  
{url:'Images/confidential.gif',id:'confidential.gif'});
```

Type file

Pas implémenté.

Autre type

On mets directement la valeur :

```
Ih.docAddTableRow({en_reference:'X345',  
en_comment:'coucou',  
en_enumcolor:'c'});
```

 Pour les énumérés il faut mettre la clef.

Exemple complet

```
Ih.docAddTableRow({en_espece:  
{id:sel.options[i].value,title:sel.options[i].text,unique:true},  
en_comment:'coucou',  
en_imgstatic:{url:d.getDisplayValue('es_photo',  
{url:true}),id:d.getValue('es_photo')},  
en_enumcolor:'c'});
```

3.8.8.2 Affectation d'attribut

Il est possible de modifier unitairement un attribut mono-valué dans le formulaire. Les attributs multi-valués ne sont pas pris en compte (attribut dans les tableaux, attribut avec option multiple=yes). Cette fonction return true si l'affectation du formulaire a réussi.

```
Ih.docSetValue({en_capacite:'20'});  
Ih.docSetValue({en_gardien:{id:6789,title:'Jean Garde'}});
```

3.8.8.3 Récupération de valeur d'attribut

Il est aussi possible de récupérer les valeur d'attribut du formulaire. Cette fonction retourne la valeur de l'attribut. Si cette valeur est multi-valués , cela retourne un tableau de valeur.

```
var ref=Ih.docGetFormValue('en_reference');
```

3.8.8.4 Utilisation des attributs de retour définis dans l'aide à la saisie

Il est possible d'utiliser le retour défini dans l'aide à la saisie si on veut faire des interfaces d'aide générique et réutilisable. Pour cela il suffit d'utiliser la fonction Ih.getAttribute(index) qui

retourne l'attribut à sa position indiqué (0 est la première position). Retourne null si position incorrecte. Exemple d'utilisation avec l'aide définie sur l'attribut 'en_espece'.

```
function insertDataUsingReturnAttribute() {
    var titles=[];
    var icons=[];
    var ids=[];
    var sel=document.getElementById("resultTab");
    for (var i=0;i < sel.options.length;i++) {
        if (sel.options[i].selected){
            var d = C.getDocument({id:sel.options[i].value, useCache:true});
            var row={};
            // get first result attribute : the attribute 'en_espece'
            row[Ih.getReturnAttribute(0)]={id:sel.options[i].value,title:sel.options[i].text,unique:true};

            // get second result attribute: the attribute 'en_comment'
            row[Ih.getReturnAttribute(1)]= 'a comment';
            Ih.docAddTableRow(row);
        }
    }
}
```

3.8.9 Mise en forme de rangée de tableau

! Version > 2.11.2 En test

Lorsque l'on souhaite une mise en forme particulière des lignes d'un tableau (effet de présentation ou nombre de colonne important), il est possible d'utiliser un template XML pour spécifier cette mise en forme.

Pour indiquer qu'un tableau a une mise forme spécifique, mettre l'option rowviewzone (pour la vue de consultation) ou roweditzone (pour la vue d'édition) dans la définition de l'attribut de type array. Pour utiliser le même template de ligne, il faut déclarer les 2 options avec le même template.

Le template doit fournir un tableau xml <table>....</table> contenant :

- une section <table-head>...</table-head> utilisée pour la ligne de titre
- une section <table-body>...</table-body> qui est répétée pour chacune des lignes.

Comme un exemple vaut mieux qu'un long discours. Voici un exemple applicable sur la famille "contrôle de vue" CVDOC.

3.8.9.1 Exemple de consultation

id	définition	type	vis	option
CV_T_VIEW	vues	array	W	rowviewzone=FDL: ROWVIEWEDITCV

l'option référence le fichier template se trouvant sur le serveur :
 /usr/share/what/**FDL**/Layout/**rowvieweditcv.xml**

! L'éventuelle méthode rowvieweditcv n'est pas appelée pour renseigner le template comme cela est le cas des templates de document. Les définitions des zones dynamiques sont statiques.

Dans le template XML, on peut utiliser

- les paramètres globaux (exemple [COLOR_B6]),
- les libellés des attributs - commençant par L_ (exemple [L_CV_MSK]),
- les valeurs formatés HTML commençant par 'V_ (exemple [V_CV_KVIEW])
- les valeurs brutes (exemple [CV_KVIEW])

- les traductions commençant par TEXT: (exemple : [TEXT:Mask and zone])

afin d'éviter que les cellules ne contenant pas de texte soient "écrasées", vous pouvez y mettre un espace insécable. Cependant, dans ce cas, il faut absolument déclarer cette entité dans le fichier xml pour qu'il soit correctement encodé: ajoutez

```
<!DOCTYPE root [
<!ENTITY nbsp "&#160;">
]>
```



juste après l'en-tête xml

Pour éviter des problèmes de présentation sur les docid de famille, il est conseillé d'utiliser

```
style="white-space:nowrap;"
```

```
<?xml version="1.0"?>





```

```

        </tr>
      </table>
    </cell>
    <cell style="background-color:[COLOR_C5];vertical-align:middle">
      [V_CV_ORDER]
    </cell>
    <cell style="background-color:[COLOR_C4];vertical-align:middle">
      [V_CV_DISPLAYED]
    </cell>
  </table-body >
</table>

```

En ce qui concerne les éléments de style (CSS). L'attribut style des balises cell sont transmises dans les balises td du HTML produit. Ce qu'il y a à l'intérieur des balises cell doit être du **XHTML** (pas de balises non fermées).

Avant : (sans zone spéciale) :

Id vues	Label	Type	Zone	Masque	Ordre	Affichable
detail	Détails	Edition	FREEDOM:EDITDSEARCH			oui
res	Par ressources	Consultation	FREEEVENT:PLANNER?byres=Y			
evt	Par événement	Consultation	FREEEVENT:PLANNER?byres=N			

Après :

Identification	Type	Mask and zone	Ordre	Affichable
detail Détails	Edition	Masque Zone : FREEDOM:EDITDSEARCH		oui
res Par ressources	Consultation	Masque Zone : FREEEVENT:PLANNER?byres=Y		
evt Par événement	Consultation	Masque Zone : FREEEVENT:PLANNER?byres=N		

3.8.9.2 Exemple d'édition

id	définition	type	vis	option
CV_T_VIEW	vues	array	W	roweditzone=FDL:R OWVIEWEDITCV

Dans ce cas on reprend le même template XML.

Avant :

contrôle de vues

DÉTAIL DES RECHERCHES ÉVÈNEMENT RECHERCHE

ÉVÈNEMENT

[Sauver | Annuler]

Basique

Titre : Détail des recherches évènement																																											
Description :																																											
Famille : recherche événement [...] [X]																																											
vues																																											
<table border="1"> <thead> <tr> <th>id vues</th> <th>label</th> <th>type</th> <th>zone</th> <th>masque</th> <th>ordre</th> <th>affichable</th> </tr> </thead> <tbody> <tr> <td>detail</td> <td>Détails</td> <td>Edition</td> <td>FRE</td> <td>[...] [X]</td> <td></td> <td><input checked="" type="checkbox"/></td> </tr> <tr> <td>res</td> <td>Par ressources</td> <td>Consultation</td> <td>FRE</td> <td>[...] [X]</td> <td></td> <td><input type="checkbox"/></td> </tr> <tr> <td>evt</td> <td>Par événement</td> <td>Consultation</td> <td>FRE</td> <td>[...] [X]</td> <td></td> <td><input type="checkbox"/></td> </tr> <tr> <td colspan="7">[+]</td> </tr> </tbody> </table>									id vues	label	type	zone	masque	ordre	affichable	detail	Détails	Edition	FRE	[...] [X]		<input checked="" type="checkbox"/>	res	Par ressources	Consultation	FRE	[...] [X]		<input type="checkbox"/>	evt	Par événement	Consultation	FRE	[...] [X]		<input type="checkbox"/>	[+]						
id vues	label	type	zone	masque	ordre	affichable																																					
detail	Détails	Edition	FRE	[...] [X]		<input checked="" type="checkbox"/>																																					
res	Par ressources	Consultation	FRE	[...] [X]		<input type="checkbox"/>																																					
evt	Par événement	Consultation	FRE	[...] [X]		<input type="checkbox"/>																																					
[+]																																											
Vues par défauts																																											
Création vue : [...] [X]																																											
dynamique																																											
Famille : [...] [X]																																											

Après :

contrôle de vues

DÉTAIL DES RECHERCHES ÉVÈNEMENT RECHERCHE

ÉVÈNEMENT

[Sauver | Annuler]

Basique

Titre : Détail des recherches évènement																																	
Description :																																	
Famille : recherche événement [...] [X]																																	
vues																																	
<table border="1"> <thead> <tr> <th>Identification</th> <th>Type</th> <th>Masque and zone</th> <th>Ordre</th> <th>Affichable</th> </tr> </thead> <tbody> <tr> <td>detail Détails</td> <td>Edition</td> <td>Masque : [...] [X] Zone : FREEDOM:EDITDSEARCH [...] [X]</td> <td></td> <td><input checked="" type="checkbox"/></td> </tr> <tr> <td>res Par ressources</td> <td>Consultation</td> <td>Masque : [...] [X] Zone : FREEEVENT:PLANNER?by [...] [X]</td> <td></td> <td><input type="checkbox"/></td> </tr> <tr> <td>evt Par événement</td> <td>Consultation</td> <td>Masque : [...] [X] Zone : FREEEVENT:PLANNER?by [...] [X]</td> <td></td> <td><input type="checkbox"/></td> </tr> <tr> <td colspan="5">[+]</td> </tr> </tbody> </table>									Identification	Type	Masque and zone	Ordre	Affichable	detail Détails	Edition	Masque : [...] [X] Zone : FREEDOM:EDITDSEARCH [...] [X]		<input checked="" type="checkbox"/>	res Par ressources	Consultation	Masque : [...] [X] Zone : FREEEVENT:PLANNER?by [...] [X]		<input type="checkbox"/>	evt Par événement	Consultation	Masque : [...] [X] Zone : FREEEVENT:PLANNER?by [...] [X]		<input type="checkbox"/>	[+]				
Identification	Type	Masque and zone	Ordre	Affichable																													
detail Détails	Edition	Masque : [...] [X] Zone : FREEDOM:EDITDSEARCH [...] [X]		<input checked="" type="checkbox"/>																													
res Par ressources	Consultation	Masque : [...] [X] Zone : FREEEVENT:PLANNER?by [...] [X]		<input type="checkbox"/>																													
evt Par événement	Consultation	Masque : [...] [X] Zone : FREEEVENT:PLANNER?by [...] [X]		<input type="checkbox"/>																													
[+]																																	
Vues par défauts																																	
Création vue : [...] [X]																																	
dynamique																																	
Famille : [...] [X]																																	

3.8.9.3 Pour éditer et consulter avec le même template

Dans la colonne "option", mettre :

```
roweditzone=FDL:ROWVIEWEDITCV|rowviewzone=FDL:ROWVIEWEDITCV
```

3.8.10 Vues d'attributs

Les vues d'attributs permettent de modifier la présentation d'un attribut en consultation ou en édition. La définition des ces vues est basée sur les vues de documents.

3.8.10.1 Vue d'attributs de consultation

Les vues de consultation sont déclarées avec l'option viewtemplate. La valeur de cette option est une zone (APPNAME::LAYOUTNAME).

! Ces vues ne sont applicables que sur les vues de consultations par défaut (FDL:VIEWBODYCARD): pas les vues résumés, ni les vues spécifiques, ni les zones FDL:VIEWFRAME. Cette option n'est pas applicable aux types 'tab' ni aux types 'array' ni aux attributs contenus dans un tableau. Pour les templates de tableaux se reporter au chapitre "[Mise en forme de rangée de tableau](#)".

Dans les templates il est possible d'utiliser les clefs d'accès aux valeurs d'attributs ([V_ATTRNAME]) et aux libellés ([L_ATTRNAME]) comme pour les vues de documents. S'il y a une méthode du nom de la vue celle-ci sera aussi appelée. Vous pouvez alors avoir accès à des clefs personnalisées sur le template. L'accès au pilotage du template se fait comme pour les vues de documents avec l'objet \$this->lay.

Si la vue comporte l'option 'S' alors le libellé de l'attribut ne sera pas affiché et le template prendra toute la largeur du document, sinon seule la partie affichage de la valeur sera substituée.

Il est bien sûr possible dans un template d'afficher plusieurs attributs. Si c'est le cas il est possible de mettre le template none (viewtemplate=none) pour les attribut déjà affiché pour ne pas afficher deux fois un même attribut. Cela est différent du fait de mettre la visibilité à 'H' car cela empêcherait aussi de la voir dans template spécifique.

Exemple

	idattr	idframe	label	T	A	type	ord	vis	need ed	link	phpfil e	phpfu nc	elink	const raint	optio n
ATTR	AN_IDENTIFICATION		Identifi cation	N	N	frame	0	W							
ATTR	AN_NOM	AN_IDEN TIFICA TION	nom	Y	N	text	10	W	Y						viewtemplat e=ZOO:ANIMALNAME
ATTR	AN_TATOU AGE	AN_IDEN TIFICA TION	tatou age	N	N	int	20	W							viewtemplat e=ZOO:ANIM ALTATOO:S

Dans cet exemple les attributs an_nom et an_tatouage ont des vues spécifiques. La vue ZOO:ANIMALNAME référence le fichier animalname.xml de l'application ZOO. Fichier animalname.xml :

```
<p style="color:green">[TEXT:zoo:It is a beautifull name for this animal]</p>
[V_AN_NOM]
```

Fichier animaltatoo.xml :

```
<p style="color:red"><b>[L_AN_TATOUAGE]</b> : [TEXT:zoo:The tatoo is
important] : [V_AN_TATOUAGE]</p>
```

Sans les zones d'attributs:

Avec les zones d'attributs :

Il est aussi possible de définir une zone pour les attributs de type frame. La contenu du cadre entier sera alors remplacé par le template. Le cadre reste quant à lui affiché avec son libellé (mettre en plus l'option vlabel=None pour ne pas voir le libellé).

3.8.10.2 Vue d'attributs d'édition

Les vues de consultation sont déclarées avec l'option `edittemplate`. La valeur de cette option est une zone (`APPNAME::LAYOUTNAME`).

! Ces vues ne sont applicables que sur les vues d'édition par défaut (FDL:EDITBODYCARD) et sur l'interface de saisie des paramètres de transitions.

Dans les templates il est possible d'utiliser les clefs d'accès aux champs du formulaire des attributs ([`V_ATTRNAME`]) et aux libellés ([`L_ATTRNAME`]) comme pour les vues d'édition de documents. Si il y a une méthode du nom de la vue celle-ci sera aussi appelée. Vous pouvez alors avoir accès à des clefs personnalisées sur le template. L'accès au pilotage du template se fait comme pour les vues de documents avec l'objet `$this->lay`.

Si la vue comporte l'option 'S' alors le libellé de l'attribut ne sera pas affiché et le template prendra toute la largeur du document, sinon seule la partie affichage de la valeur sera substituée. Si la vue comporte l'option 'U' alors l'input utilisé avec le mot-clef [`V_ATTRNAME`] prendras la largeur disponible. Cette option est à mettre si le template ne comporte qu'un seul champ input.

Il est bien sûr possible dans un template d'afficher plusieurs attributs. Si c'est le cas il est possible de mettre le template `none` (`viewtemplate=none`) pour les attribut déjà affiché pour ne pas afficher deux fois le même champs de formulaire ce qui provoquerais un mauvais enregistrement du document.

Exemple

	idattr	idframe	label	T	A	type	ord	vis	need ed	link	phpfile	phpfunc	ealink	constraint	option
ATTR	AN_IDENTIFICATION	Identification	N	N	frame	0	W								
ATTR	AN_NOM	AN_IDENTIFICATION	nom	Y	N	text	10	W	Y						edittemplat e=ZO O:ANI MALNAME:U
ATTR	AN_TATOUAGE	AN_IDENTIFICATION	tatouage	N	N	int	20	W							edittemplat e=ZO O:ANI MALT ATOO:S

Dans cet exemple les attributs `an_nom` et `an_tatouage` ont des vues spécifiques. La vue `ZOO:ANIMALNAME` référence le fichier `animalname.xml` de l'application `ZOO`. Fichier `animalname.xml` :

```
<p style="color:green">[TEXT:zoo:It is a beautifull name for this animal]</p>
[V_AN_NOM]
```

Fichier `animaltattoo.xml` :

```
<p style="color:red"><b>[L_AN_TATOUAGE]</b> : [TEXT:zoo:The tatoo is
important] : [V_AN_TATOUAGE]</p>
```

Sans les interfaces d'attributs :

Sauver Annuler

Identification

Nom :	Arthur
Tatouage :	23545
Espèce :	Antilope
Sexe :	Masculin
Photo :	
Date naissance :	<input type="text"/> ... ♦ <input type="button" value="x"/>
Date entree :	<input type="text"/> ... ♦ <input type="button" value="x"/>

liste enfant

enfant
Totor Alligator

Avec les interfaces d'attributs :

Sauver Annuler

Identification

Nom :	C'est un joli nom pour cet animal <input type="text" value="Arthur"/>
tatouage : La tatouage est important :	<input type="text" value="23545"/>
Espèce :	Antilope
Sexe :	Masculin
Photo :	
Date naissance :	<input type="text"/> ... ♦ <input type="button" value="x"/>
Date entree :	<input type="text"/> ... ♦ <input type="button" value="x"/>

liste enfant

enfant
Totor Alligator

Il est aussi possible d'avoir des interfaces d'éditions plus spécifiques. Pour les attributs non multivalués, vous pouvez utiliser la fonction 'setFormValue' pour affecter des valeurs d'attributs dans le formulaire. Il est aussi possible d'utiliser la fonction addTableRow pour ajouter une rangée

à un type tableau (voir [manipulation_de_tableau](#)).

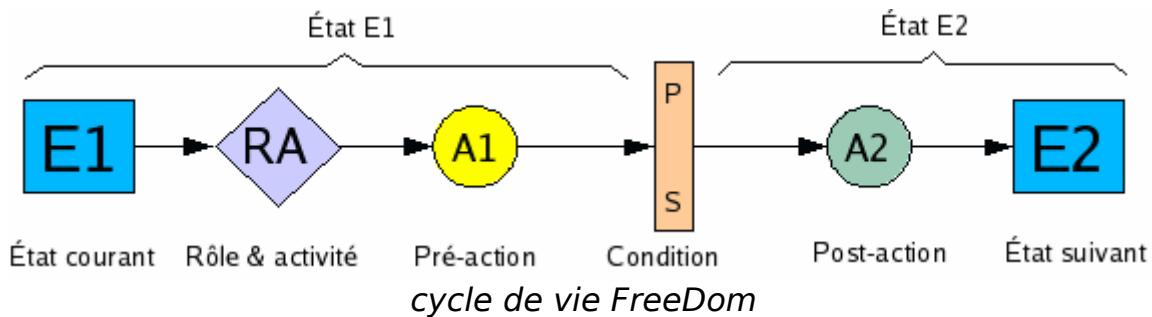
```
[V_AN_TATOUAGE]
<a onclick="setFormValue({an_tatouage:4567})">Set tatoo to 4567</a>;
```

3.9 Cycle de vie

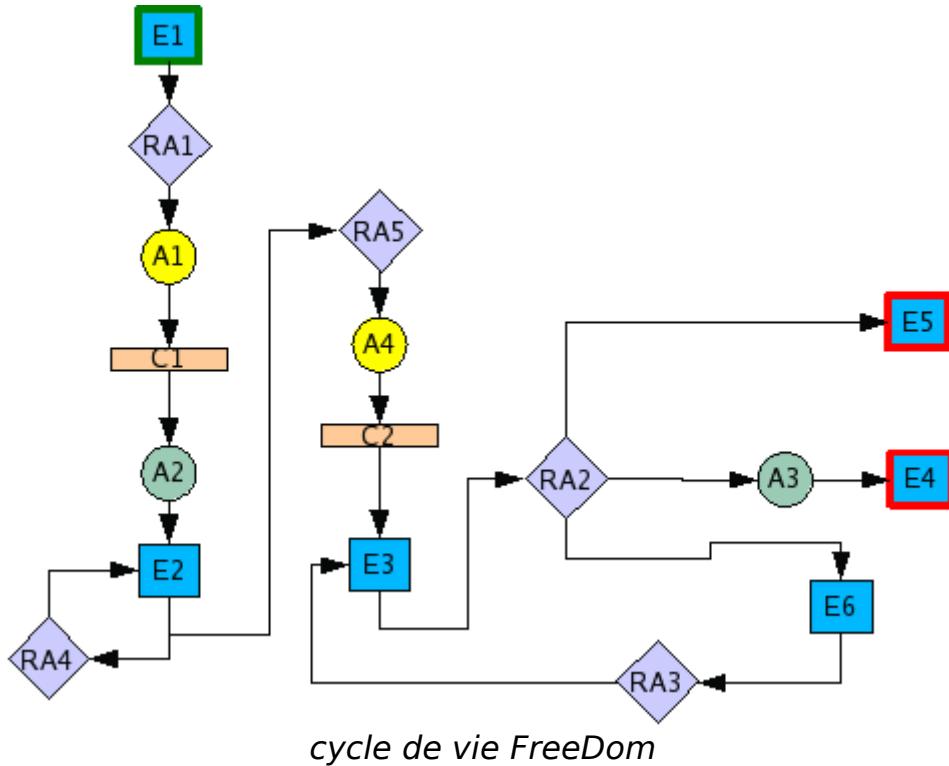
3.9.1 Développement

3.9.1.1 Définition d'un cycle de vie

Le cycle de vie dans FreeDom permet à un document de changer d'état suivant un cycle défini. Pour changer d'état, le document doit franchir une transition. Cette transition s'accompagne d'une pré-action, d'une condition de transition et d'une post-action.



Un document peut aller d'un état E1 à un état E2 seulement si une transition a été définie entre E1 et E2. Le passage de l'état E1 à l'état E2 est soumis à la condition C définie pour la transition. Cette condition est découpé en deux sous-conditions. La première est une condition générale sur les droits de passage de la transition. Ce droit est géré par les profils d'accès utilisateur. La deuxième est une condition spécifique à la transition. Cette dernière est généralement liée au contenu du document.



Les rôles et activités décrivent qui fait quoi avant d'exécuter la transition. L'activité peut être une décision humaine permettant de choisir quelle transition adopter ou simplement une action humaine à exécuter (ex: appeler le directeur, envoyer par recommandé). La notion d'activité est décrite dans un attribut (activité) du cycle. Un attribut 'activité' est créé par état. La notion de rôle n'est pas intégré dans le processus informatique. Il est implicitement donné par [l'utilisateur affecté](#) à l'activité (qui doit modifier le document) et par le [profilage du cycle](#) : qui a le droit de modifier le document, qui a le droit de passer telle ou telle transition.

La pré-action A1 est exécutée systématiquement avant les conditions. Cette pré-action est généralement utilisée pour mettre à jour certains attributs utiles pour la transition et pour l'état suivant. La post-action A2 est exécutée si les conditions de transitions sont réunies. Lorsque A2 s'exécute le document est déjà dans l'état E2. Cette action est généralement utilisée pour créer ou mettre à jour des documents connexes au document courant.

Plusieurs transitions peuvent avoir l'état E1 en état courant ou avoir l'état E2 en état suivant. Le même état peut être à la fois courant et suivant. Par contre on ne peut définir qu'une seule transition de E1 à E2.

Les post-action, pré-action et condition spécifiques étant facultatives, le schéma du cycle ne représente que celles qui sont significatives. Les conditions sur les profils étant systématiques ne sont pas représentées. Les états initiaux, entourés de vert, n'ont pas d'état courant. Un seul état initial est possible. Les états finaux, entourés de rouge, n'ont pas d'état suivant. Lorsqu'un document est dans un état final, il ne peut plus changer d'état. Cela ne veut pas dire que le contenu du document n'est plus modifiable mais seulement que son état n'est plus modifiable.

3.9.1.2 Construction d'un cycle de vie

Le cycle de vie dans FREEDOM est défini dans une classe de document héritant de WDoc (workflow document). Cette classe décrit les états, les transitions, les conditions de transition ainsi que les pré et post-action. Les rôles et les activités ne sont pas décrits pour l'instant.

La famille de cycle de vie est construite en créant une famille hérité de cycle de vie (WDOC) et à l'aide d'une classe documentaire héritée de WDoc.

	fromid	title	id	classname	name
--	---------------	--------------	-----------	------------------	-------------

BEGIN	WDOC	ma famille cycle		WDocTest	MONCYCLE
ICON	cycle.gif				
TYPE	C				
USEFOR	W				
END					



La propriété classname définie le fichier cycle (classe PHP) définissant les états et les transitions. Ici, il identifie le fichier /usr/share/what/FDL/Class.**WDocTest.php**

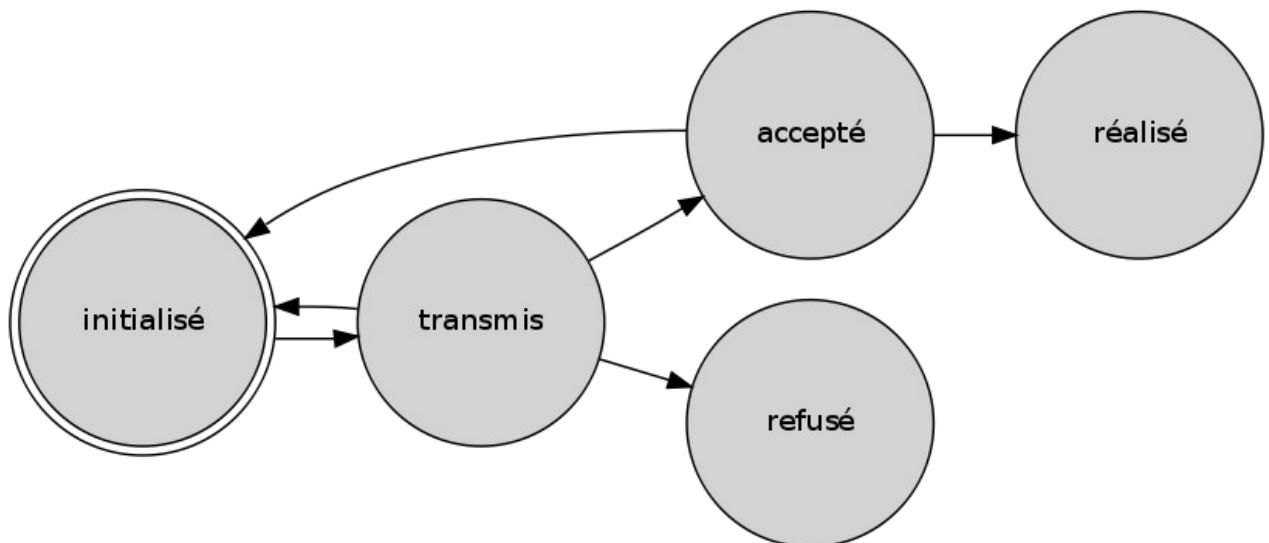
Tous les fichiers de classes PHP doivent être accessibles sous le répertoire /usr/share/what/FDL. Le nom du fichier doit correspondre au nom de la classe PHP.

Pour définir un cycle de vie, seuls les transitions sont obligatoires. Pour définir les transitions la classe WDoc se sert de la notion de *forme de transition*. Les formes de transitions sont nommées et recensées dans l'attribut tableau \$transition de la classe. Une forme transition dans la classe WDoc contient la pré-action, la condition spécifique et la post-action. Elle ne contient pas les informations état courant et état initial. Ces informations sont décrites dans l'attribut \$cycle. Ceci permet de réutiliser une même forme de transition dans plusieurs transitions possibles.

Version > 2.11.12 . Les activités liées à un état sont définis dans le tableau \$stateactivity. Ils peuvent être défini dans la classe ou dans l'instance de cycle. S'ils sont définis dans la classe du cycle ils ne pourront être modifiés par l'interface.



Le nombre de type de transitions est actuellement limité à **20**. Cette limite est due au profil des transitions codé sur un entier de 32 bits (13 bits étant réservés pour les documents en général)



```

<?php
include_once("FDL/Class.WDoc.php");

Class WDocTest extends WDoc {
    public $attrPrefix="WDT";

    const initialised="initialised"; # N_("initialised")
    const transmited="transmited"; # N_("transmited")
}

```

```

const accepted="accepted"; # N_("accepted")
const refused="refused"; # N_("refused")
const realised="realised"; # N_("realised")

const Ttransmited="Ttransmited"; # N_()
const Taccepted="Taccepted"; # N_("Taccepted")
const Trefused="Trefused"; # N_("Trefused")
const Tretry="Tretry"; # N_("Tretry")
const Trealised="Trealised"; # N_("Trealised")
public $firstState=self::initialised;

public $transitions=array( self::Ttransmited =>array(),
    self::Taccepted => array(),
    self::Trefused =>array(),
    self::Trealised=>array(),
    self::Tretry =>array();

public $cycle=array(array("e1"=>self::initialised,
    "e2"=>self::transmited,
    "t"=>self::Ttransmited),
    array("e1"=>self::transmited,
    "e2"=>self::accepted,
    "t"=>self::Taccepted),
    array("e1"=>self::transmited,
    "e2"=>self::refused,
    "t"=>self::Trefused),
    array("e1"=>self::accepted,
    "e2"=>self::realised,
    "t"=>self::Trealised),
    array("e1"=>self::transmited,
    "e2"=>self::initialised,
    "t"=>self::Tretry) );

public $stateactivity=array(self::initialised=>"adoption writting",
    self::transmited=>"adoption verification"); # _("adoption writting")
_("adoption verification")

}
?>

```

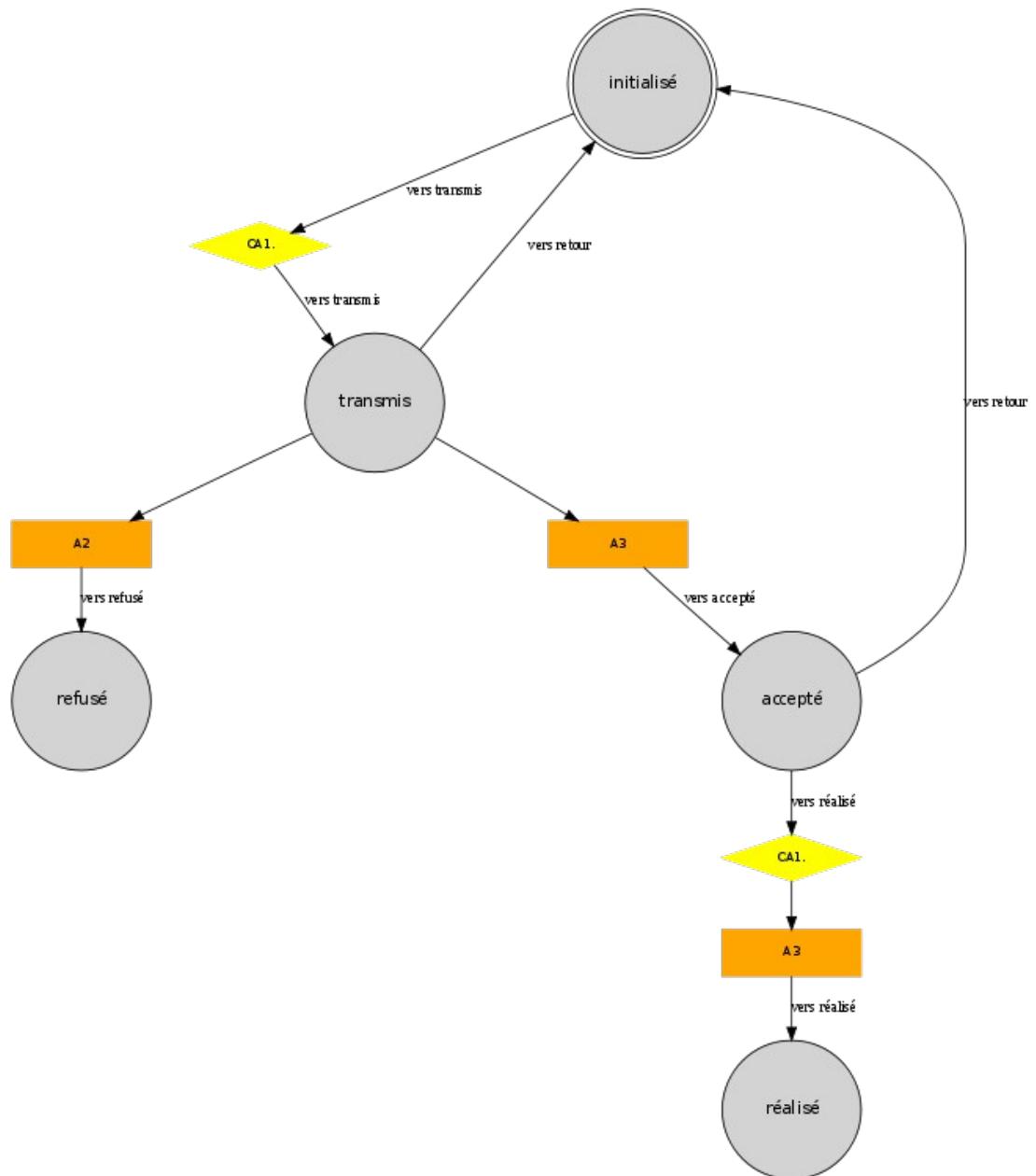
Dans l'exemple ci-dessus, cinq les formes de transitions sont utilisées. Elles sont équivalentes en terme de pré, post actions et de conditions spécifiques de transition. Le fait ici d'avoir trois formes de transitions implique que des droits différents de passage peuvent être appliqués.

Test	Voir	éditer	Supprimer	Vers transmis	Vers accepté	Vers refusé	Vers réalisé	Vers retour	Voir les droits	Modifier les droits
Utilisateurs	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]
Administrateurs	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]
	<input type="button" value="Valider"/>	<input type="button" value="Réinitialisation"/>	<input type="button" value="Seulement les cochés"/>							

Ainsi si la transition 'transmited' vers 'initialised' est autorisée, elle le sera aussi pour 'accepted' vers 'initialised'.

Les pré-actions, les conditions spécifiques et les post-actions sont des méthodes de la classe. Les pré-actions et les conditions spécifiques doivent être regroupés dans une méthode commune qui sera appelé dès la demande de transition. Cette méthode doit retourner la chaîne vide en cas de succès. Si le condition spécifique n'est pas remplie, la méthode doit retourner le texte correspondant à l'erreur. Cette méthode est identifiée par l'index m1 des formes de transitions.

La post-action est la méthode identifiée par l'index m2. Si cette méthode retourne un texte, celui-ci sera affiché comme un message d'avertissement.



```

<?php
include_once("FDL/Class.WDoc.php");

Class WDocTest extends WDoc {
    public $attrPrefix="WDT";
    const initialised="initialised"; # N_("initialised")
    const transmited="transmited"; # N_("transmited")
    const accepted="accepted"; # N_("accepted")
    const refused="refused"; # N_("refused")
    const realised="realised"; # N_("realised")

    const Ttransmited="Ttransmited"; # N_()
    const Taccepted="Taccepted"; # N_("Taccepted")
    const Trefused="Trefused"; # N_("Trefused")
    const Tretry="Tretry"; # N_("Tretry")
    const Trealised="Trealised"; # N_("Trealised")
  
```

```

public $firstState=self::initialised;

public $transitions=array( self::Ttransmited =>array("m1"=>"CA1"),
    self::Taccepted => array("m2"=>"A3"),
    self::Trefused =>array("m2"=>"A2"),
    self::Trealsised=>array("m1"=>"CA1","m2"=>"A3"),
    self::Tretry =>array();

public $cycle=array(array("e1"=>self::initialised,
    "e2"=>self::transmited,
    "t"=>self::Ttransmited),
    array("e1"=>self::transmited,
    "e2"=>self::accepted,
    "t"=>self::Taccepted),
    array("e1"=>self::transmited,
    "e2"=>self::refused,
    "t"=>self::Trefused),
    array("e1"=>self::accepted,
    "e2"=>self::realised,
    "t"=>self::Trealsised),
    array("e1"=>self::transmited,
    "e2"=>self::initialised,
    "t"=>self::Tretry) );

public $stateactivity=array(self::initialised=>"adoption writting",
    self::transmited=>"adoption verification"); # _("adoption writting")
_("adoption verification")

public function A1($newstate,$oldstate="",$comment="") {}
public function C1($newstate) {}
public function CA1($newstate,$oldstate,$comment) {
    $this->A1($newstate,$oldstate,$comment);
    return $this->C1($newstate);
}
public function C2($newstate) {}
public function A2($newstate) {}
public function A3($newstate) {}
}

?>

```

Chacune des méthodes m1 ou m2 doit posséder un argument. Cet argument est le nom de l'état suivant. Pour les méthodes m1, l'état courant est donné par l'attribut state du document courant (\$this->doc).

```

// comment est le commentaire de transition. Il est vide si "nr=>true" (nr :
no raison) dans la description de la transition
// oldstate : état courant
// newstate : état qui va suivre
public function A1($newstate,$oldstate,$comment) {
    $cstate = $this->doc->state; // état courant == oldstate
}

```

Lorsque l'utilisateur change l'état du document (actions : changer d'état ou modifier), la nouvelle révision du document s'affiche à la place de l'ancienne. Dans la définition des méthodes de m1 ou m2, il est possible de rediriger l'URL d'affichage. Cette redirection se fait en modifiant les variables HTTP redirect_app et redirect_act.

```

public function A3($newstate) {
    SetHttpVar("redirect_app","FREEDOM"); //// application WHAT////
    SetHttpVar("redirect_act","HIST0&id=".$this->doc->id);//// action et
argument de l'application////
return "";
}

```

Dans cet exemple, lorsque le document passe de l'état E2 à E3 (forme de transition T3), la post-action A3 indique une redirection afin d'afficher l'historique du document à la place de la vue classique.

Le résultat transition peut être déroutée juste avant le passage de la transition. C'est la condition de transition qui indique la redirection. Si la méthode m1 retourne « → » cela indique que la transition ne sera pas effectué et que la redirection spécifiée par les variables HTTP sera appliquée. Pour effectuer cette transition il faudra forcer le passage (paramètre \$force=true dans la méthode WDoc::changeState()). Cette redirection est généralement utilisée pour demander une confirmation pour la forme de transition.

Soit l'action confirmt qui demande la confirmation de la transition :

confirmt.php

```

<?php
include_once("FDL/Class.Doc.php");

function confirmt(&$action) {
    // Get All Parameters
    // document to change
    $docid = GetHttpVars("id",0);
    // new state
    $nextstate = GetHttpVars("state");

    $action->lay->Set("docid",$docid);
    $action->lay->Set("state",$nextstate);
    $action->lay->Set("tstate",_($nextstate));

}
?>

```

confirmt.xml

```

<form id="fedit"
      class="fborder"
      name="modifydoc"
      method="POST" ENCTYPE="multipart/form-data"

      action="[CORE_STANDURL]&app=FREEDOM&action=MODSTATE">

<input type="hidden" name="fstate" value="yes">
<input type="hidden" name="id" value="[docid]">
<input type="hidden" name="comment" value="[TEXT:confirmed]">
<input type="hidden" name="newstate" value="[state]">
<input type="submit"
       value="[TEXT:Confirm change the document state to] [tstate]" >
<input type="button" value="[TEXT:cancel]"
       onclick="location.href='[CORE_STANDURL]&app=FDL&action=FDL_CARD&id=[do
cid]'">
</form>

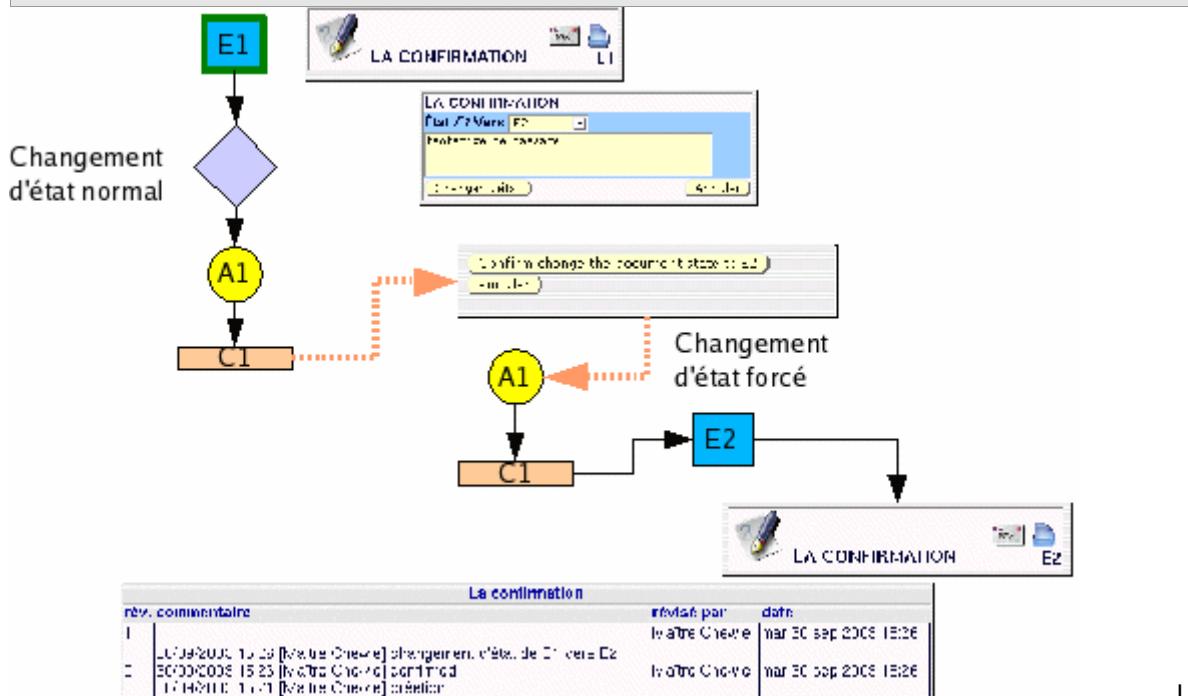
```

Cette action a comme argument l'identificateur de document et l'état suivant du document. Lorsque l'utilisateur appuis sur le bouton de confirmation, une nouvelle demande de changement

d'état a lieu mais cette fois ci avec l'attribut force (appel de l'action MODSTATE avec l'attribut fstate=yes).

Nous reprenons, l'exemple précédent en modifiant les méthodes A1 et C1. Ceci permet de déclencher la confirmation sur la forme T1 de transition.

```
public function A1($newstate) {
    SetHttpVar("redirect_app","TEST");
    SetHttpVar("redirect_act","CONFIRMT&state=$newstate&id=".$this->doc->id);
}
public function C1($newstate) {
    return "->";
}
```



Le changement d'état forcé n'ignore pas la condition C1. Cette condition est recalculée et si le retour est « → » le changement est opéré au contraire du mode normal où une redirection est effectuée. Dans le mode forcé, la redirection des méthodes m1 est ignorée par contre, si une redirection est mise en place dans m2, elle sera prise en compte.

3.9.1.3 Paramètres de transition

Lors de la définition des types de transitions il est possible d'appliquer des paramètres. Ces paramètres seront présentés lors d'un changement d'état et seront accessibles depuis les pré et post-actions. Ces paramètres doivent être définis dans la famille du cycle de vie sous forme d'attribut paramètre.

//	fromid	title	special id	classname name								
BEGIN	WDOC	cycle action de prospection		WProActio	WPROACT							
ICON	cycle.gif											
TYPE	C											
USEFOR	W			//	idattr	idframe	label	T	A	type	ord	vis
										frame	10	W
PARAM	WPROA_FR_TRANS						Paramètres de transition	N	N			
PARAM	WPROA_DATEPEDI	WPROA_FR_TRANS					date de report	N	N	date	20	W
PARAM	WPROA_IDPERS	WPROA_FR_TRANS						N	N	docid	30	H
PARAM	WPROA_PERS	WPROA_FR_TRANS					responsable	N	N	text	40	W
END												

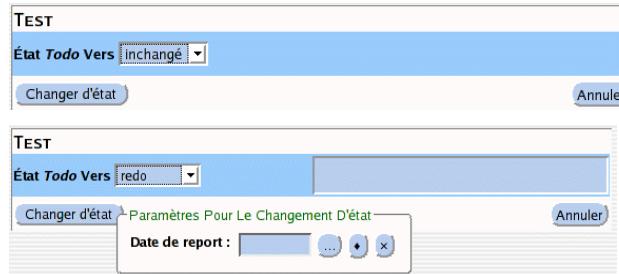
Pour indiquer quels attributs doivent être demandé pour les transition, il faut ajouter l'élément "ask" dans le tableau des transitions comme le montre l'exemple suivant. Cet élément est composé d'un tableau des identificateurs des paramètres. Soit le fichier **Class.WProAction.php** (défini par colonne *classname*):

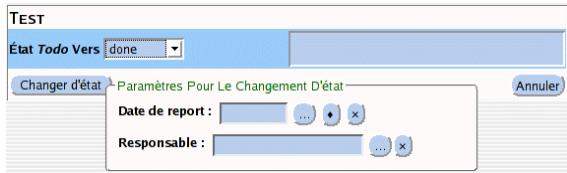
```
<?php
include_once("FDL/Class.WDoc.php");

Class WProAction extends WDoc {
    const todo="todo"; # N_("todo")
    const done="done"; # N_("done")
    const redo="redo"; # N_("redo")
    const Tredo="Tredo"; # N_("Tredo")
    const Tdone="Tdone"; # N_("Tdone")
    public $attrPrefix="WPROA"; // prefix attribute

    public $transitions = array(
        self::Tdone =>array("m1"=>"",
            "ask"=>array("WPROA_DATEREPORT",
                "WPROA_IDPERS","WPROA_PERS"),
            "m2"=>""),
        self::Tredo =>array("m1"=>"askDate",
            "ask"=>array("WPROA_DATEREPORT"),
            "m2"=>""));
    public $cycle = array(
        array("e1"=>self::todo,
            "e2"=>self::done,
            "t"=>self::Tdone),
        array("e1"=>self::todo,
            "e2"=>self::redo,
            "t"=>self::Tredo),
        array("e1"=>self::redo,
            "e2"=>self::redo,
            "t"=>self::Tredo)
    );
    public $firstState=self::todo;
}
public function askDate($newstate) {
    $err="";
    $mydate=$this->getValue("WPROA_DATEREPORT");
    // traitement de la date ...
    return "$err";
}
?>
```

Le résultat de ce cycle donne l'affichage suivant lors d'un changement d'état.





Il est possible d'indiquer les paramètres obligatoires si l'information est indispensable. La récupération des valeurs se fait par la méthode "getValue()" du document cycle de vie. Si l'information des paramètres n'est pas mémorisée dans les méthodes de pré ou post-actions, elles seront perdues.

Depuis la version 2.13.12, il est possible de pré-renseigner les valeurs des paramètres de transitions. Si vous renseignez la valeur du paramètre d'un cycle, cette valeur sera automatiquement proposée lors de la demande de changement d'état. La valeur de ce paramètre peut faire référence à une méthode du cycle (pas du document instancié). Cela permet de proposer des valeur différente suivant le contexte d'exécution.

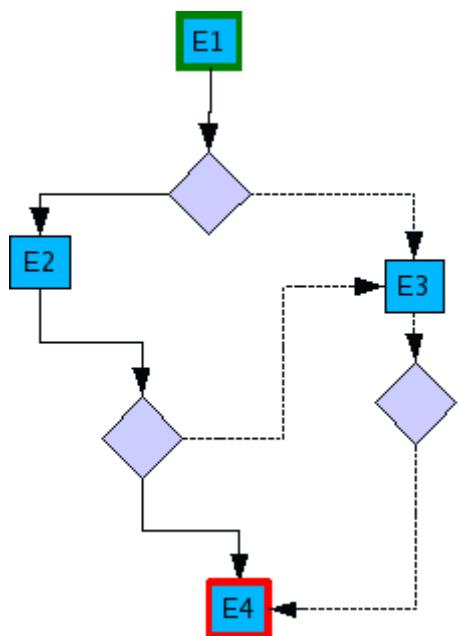
BEGIN	WDOC	Cycle de compte rendu	WMinuteMeet	WMINUTEMEETING
ICON	cycle.gif			
USEFOR	W			
TYPE	C			
//	idattr	idframe	label	T A type ord vis needed
PARAM	WMM_FR_PARAM		Paramètre	frame 10 W
PARAM	WMM_DATE	WMM_FR_PARAM	Une date	N N date 20 W
PARAM	WMM_AUSER	WMM_FR_PARAM	Quelqu'un	N N docid("IUSER") 25 W Y
PARAM	WMM_AVALUE	WMM_FR_PARAM	Le lieu	N N text 30 W
DEFAULT	WMM_AVALUE	::getInstanceValue('SERT_PLACE')		
DEFAULT	WMM_AUSER	::getInstanceValue('SERT_CONTROLLER')		
DEFAULT	WMM_DATE	::getDate(7)		
END				

Dans l'exemple ci-dessus, le paramètre 'WMM_DATE' sera proposée avec la date de la semaine prochaine (>::getDate(7)), WMM_AVALUE aura la valeur du contrôleur du document auquel est attaché le cycle et WMM_AUSER aura la valeur du lieu du document courant.

3.9.1.4 Options de transitions

Transition principale

Dans la définition du cycle, un chemin principal allant de l'état initial vers l'état final peut se dégager. Lors d'un changement d'état il est possible de proposer un état suivant déterminé pour indiquer le chemin « normal ». Par défaut, l'interface de changement propose « inchangé » (pas de transition). Si un état suivant est indiqué dans la spécification du cycle, l'interface proposera cet état en premier (inchangé sera toujours disponible). Ce chemin principal est indiqué dans l'attribut autonext de la classe cycle de vie comme le montre l'exemple suivant.



```

<?php
include_once("FDL/Class.WDoc.php");

Class WTest extends WDoc {
    public $attrPrefix="TWF"; // prefix attribute
    const E1="E1";
    const E2="E2";
    const E3="E2";
    const E4="E2";
    const T1="T1";
    const T2="T2";
    const T3="T2";
    const T4="T2";
    public $firstState=self::E1;
    public $transitions = array(self::T1 =>array(),
        self::T2 =>array(),
        self::T3 =>array(),
        self::T4 =>array());

    public $autonext=array(self::E1=>self::E2, // on propose E2 si l'état est E1
        self::E2=>self::E4); // on propose E4 si l'état est E2

    public $cycle = array(
        array("e1"=>self::E1,
            "e2"=>self::E2,
            "t"=>self::T2),
        array("e1"=>self::E1,
            "e2"=>self::E3,
            "t"=>self::T3));
}
  
```

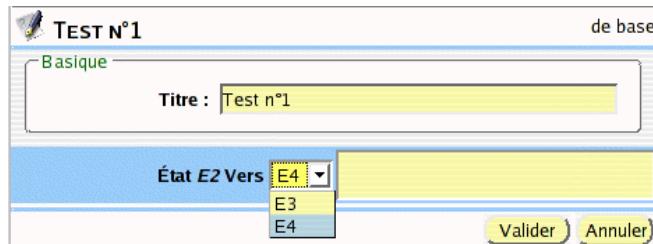
```

    "t"=>self::T3),
array("e1"=>self::E2,
      "e2"=>self::E3,
      "t"=>self::T3),
array("e1"=>self::E2,
      "e2"=>self::E4,
      "t"=>self::T4),
array("e1"=>self::E3,
      "e2"=>self::E4,
      "t"=>self::T4));
}
?>

```



Transition forcée



Afin de simplifier et de forcer le passage d'une transition, il est possible d'indiquer que la sauvegarde sans changement d'état n'est pas possible. Cela implique que l'interface de changement d'état n'indique plus « inchangé » pour le nouvel état.

Cette restriction est indiquée dans l'attribut de type tableau **nosave**. Ce tableau contient tous les états à partir desquels il est obligatoire de changer d'état afin de modifier un document.

Retenant l'exemple précédent, la ligne suivante est rajoutée pour un indiquant que l'état E2 ne peut être sauvegardé sans changement d'état.

```
public $nosave=array("E2");
```

On remarque dans la capture d'écran que « inchangé » n'est pas proposé. L'utilisateur est alors forcé de choisir entre E3 et E4

Transition sans confirmation

Un passage de transition demande toujours la raison du changement d'état. Ceci a pour conséquence un affichage d'une fenêtre de dialogue afin de demander cette raison. Si votre transition n'implique pas de création de document et n'est pas critique, vous pouvez dire que la raison n'est pas obligatoire, et donc que la fenêtre de dialogue ne sera plus présentée, sauf en cas de demande de paramètre de transition (ask). Pour activer l'option sans confirmation il faut indiquer que l'index "nr" (*no reason*) est vrai.

```
public $transitions = array(
    self::Tdone =>array("m1"=>"",
    "nr"=>true, // sans confirmation
```

```
"m2"=>""),  
self::Treдо =>array("m1"=>"askDate",  
"ask"=>array("WPR0A_DATEREPORT"),  
"m2"=>""));
```

Interface boutons

Lorsque les possibilités de transitions sont réduites, l'interface de changement d'état peut être changé pour utiliser des boutons au lieu d'une liste déroulante. Dans ce cadre, il est nécessaire de traduire les états (E1, E2, etc.) et aussi les actions liées à ces états (ToE1, ToE2, etc.). Les actions liées aux états sont nommées avec le préfixe « To » suivi du nom de l'état.

La première capture d'écran reprend les conditions du paragraphe §8.4.1 et la seconde représente la même fonctionnalité que celle montrée au paragraphe précédent, c'est à dire le fait de ne pas pouvoir sauver sans changer d'état.



```
// pour l'i18n  
# N_("E1") N_("E2") N_("E3") N_("E4")  
# N_("ToE1") N_("ToE2") N_("ToE3") N_("ToE4")  
  
public $viewlist="button";
```

Pour indiquer ce mode de présentation, il suffit d'affecter l'attribut \$viewlist à « button » dans la classe du cycle. Par défaut, cet attribut vaut « list ». Il est nécessaire d'indiquer qu'il faut aussi traduire les actions (N_(« **Toxxx** »)).

3.9.2 Paramétrage

3.9.2.1 Créer un Cycle de vie (Workflow)

But de ce document

Le but de ce document est de décrire les étapes pour mettre en place un Workflow.

Dans Freedom un Workflow est appelé “Cycle de vie”.

Principe général

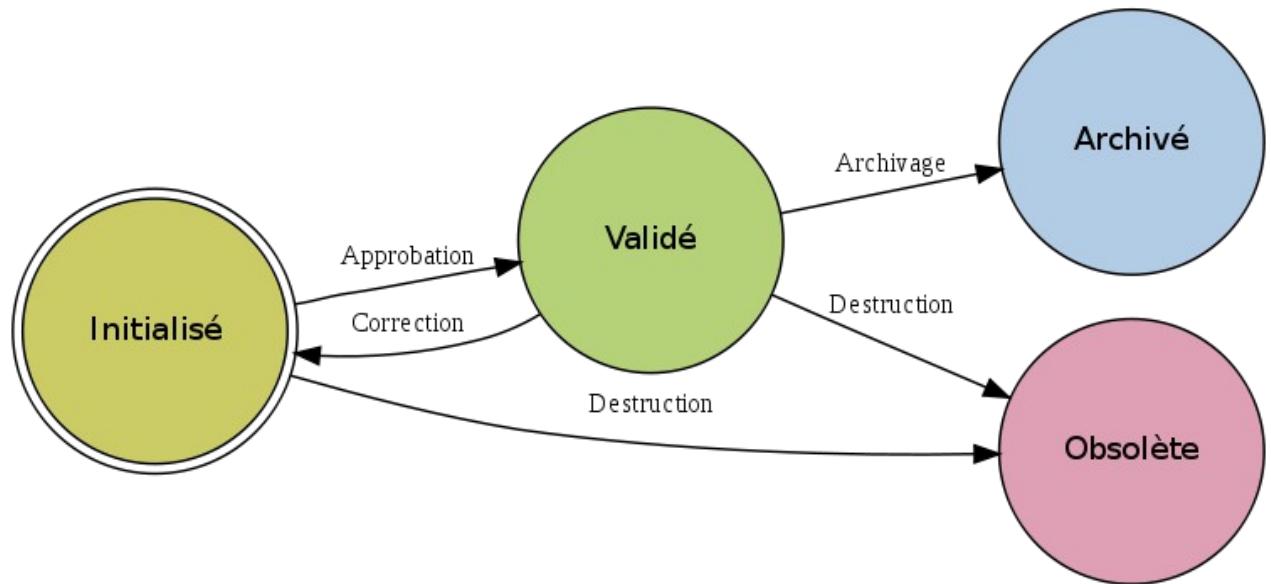
Pour mettre en place un cycle de vie, il faut créer ou modifier plusieurs choses :

- Un fichier PHP qui définit le fonctionnement du cycle de vie (Etats, transitions,...)
- Une famille de cycle de vie utilisant le fichier PHP précédent

- Un document cycle de vie basé sur la famille précédente. Il est possible de créer plusieurs cycles de vie basés sur la même famille de cycle de vie.
- Une famille de documents sur laquelle le cycle de vie sera appliqué. Il est possible d'affecter un cycle de vie à plusieurs familles de documents.
- Un profil général affecté au cycle de vie ou un profil particulier pour chaque étape du cycle de vie. Cela permet de gérer qui a le droit de faire quoi sur chaque étape du cycle de vie.
- Il est possible également de créer un masque particulier pour chaque étape du cycle de vie pour afficher ou masquer certains champs en fonction de l'état d'avancement du document dans le cycle.
- Les modèles de courriel à envoyer.
- Les déclencheurs pour ajouter des alarmes suivant des critères de temps.

Exemple de cycle de vie

Dans ce document, nous allons décrire toutes les étapes nécessaires à la mise en place du cycle de vie ci-dessous :



Dans cet exemple, un document peut avoir 4 états :

- Brouillon
- Validé,
- Archivé
- Poubelle

Les transitions pour passer d'un état à un autre sont nommées :

- Validation : Pour passer de l'état « Brouillon » à l'état « Validé »
- Correction : Pour passer de l'état « Validé » à l'état « Brouillon »
- Archivage : Pour passer de l'état « Validé » à l'état « Archivé »
- Destruction : Pour passer de l'état « Brouillon » à « Poubelle » ou de l'état « Validé » à l'état « Poubelle »

Pour décrire ces états et ces transitions, il faut créer un fichier PHP comme indiqué au chapitre suivant.

Création du fichier PHP de définition du cycle de vie

Le fichier PHP définissant le cycle de vie, sera placé dans le dossier « **/usr/share/what/FDL** » Son

nom sera sous la forme : « Class.**WDocTestCycle1.php** » correspond au nom de la classe de la famille de cycle de vie que nous allons créer au chapitre suivant. Voici le contenu de ce fichier qui permet de définir notre cycle de vie :

```
<?php
include_once("FDL/Class.WDoc.php");

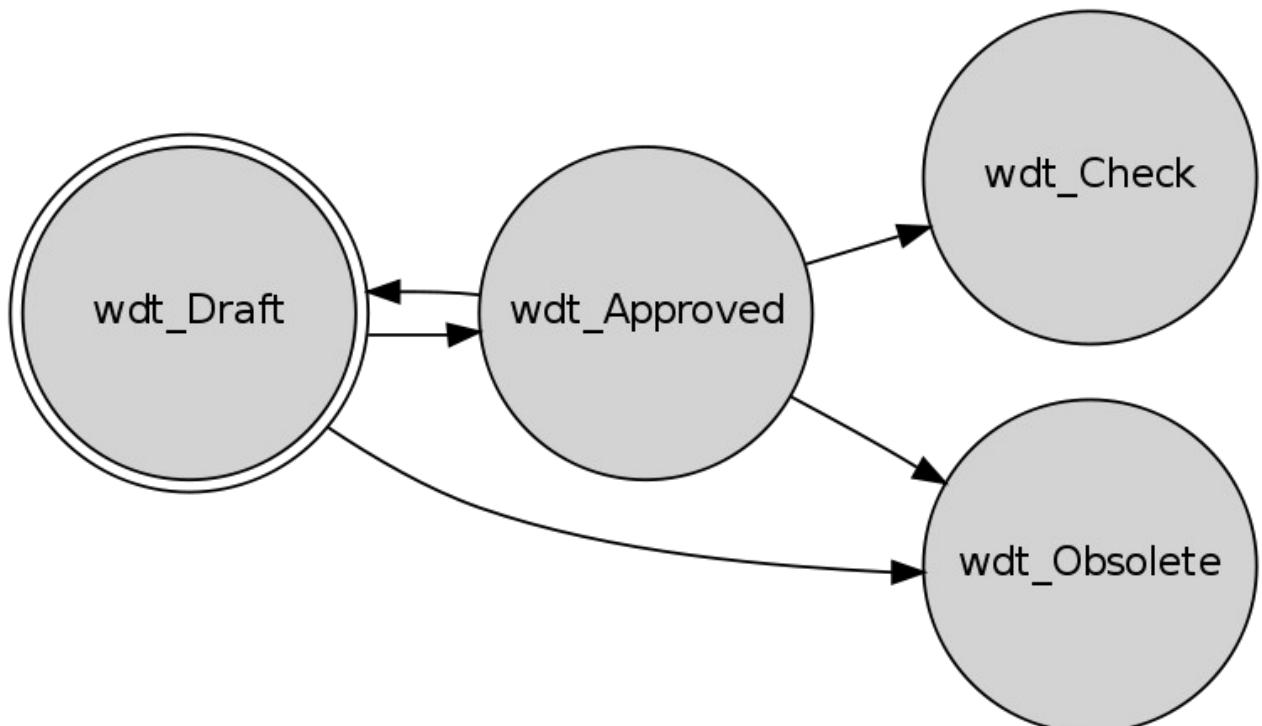
Class WDocTest extends WDoc {

var $attrPrefix="WDT";
var $firstState="wdt_Draft";

var $transitions = array("wdt_Approvement"=>array(),
    "wdt_Correction"=>array(),
    "wdt_Chechin"=>array(),
    "wdt_Destroy"=>array());

var $cycle = array(array("e1"=>"wdt_Draft",
    "e2"=>"wdt_Approved",
    "t"=>"wdt_Approvement"),
    array("e1"=>"wdt_Approved",
    "e2"=>"wdt_Check",
    "t"=>"wdt_chechin"),
    array("e1"=>"wdt_Approved",
    "e2"=>"wdt_Draft",
    "t"=>"wdt_Correction"),
    array("e1"=>"wdt_Draft",
    "e2"=>"wdt_Obsolète",
    "t"=>"wdt_Destroy"),
    array("e1"=>"wdt_Approved",
    "e2"=>"wdt_Obsolète",
    "t"=>"wdt_Destroy"));
}

?>
```



Commentaires sur les variables :

- \$firstState : État initial. Si cette variable n'est pas initialisée, le document sera dans un état indéterminé à la création ce qui en général, n'est pas souhaitable.
- \$cycle : Cette variable permet de définir le nom des états et des transitions. Elle permet également de définir quelle transition utilisée pour passer d'un état à un autre
- \$transition : Cette variable détermine le nom des transitions

Attention :

- Le nom de la « class » doit correspondre au nom du fichier et au nom indiqué dans les attributs de la famille (attention également au respect des minuscules/majuscules).
- Il est possible de modifier ce fichier à tout moment¹⁹. Les modifications sont immédiatement prises en compte dans tous les documents utilisant cette famille de cycle de vie.
- Il ne faut pas utiliser de caractères spéciaux (accents) et d'espaces dans le nom des transitions et des étapes (Pour avoir des accents dans les documents finaux, il faut passer par gettext pour traduire les différents termes)

Création de la famille « Cycle de vie » dans OOo

Il n'est pas possible de créer une famille cycle de vie dans Freedom. Il faut donc créer cette famille dans OOo et l'importer dans Freedom. Vous trouverez dans `FIXME internalmedia: freedom_2.14:admin:family:cycle_famille_testcycle1.ods` la définition de cette famille :

//	héritage	titre	id	class	name
BEGIN	WDOC	Test Cycle		WDocTest	TestCycle
ICON	cycle.gif				
TYPE	C				
USEFOR	W				
END					

Dans ce paramétrage, la seule chose vraiment importante est le nom de la classe et donc du fichier PHP à utiliser. Dans notre cas, c'est « **WDocTest** »

Pour importer dans Freedom `FIXME internalmedia: freedom_2.14:admin:cycle_de_vie:wtest.ods` OpenOffice.org contenant la définition de la famille, il faut se rapporter à la documentation concernant [l'importation de familles](#). Mais en résumé, cette commande, permet d'importer cette famille dans Freedom :

```
/usr/share/what/wsh.php --api=freedom_import --file=wtest.ods
```

Remarques :

- Il faut obligatoirement créer le fichier PHP de définition du cycle avant d'importer la famille sinon l'importation échouera.
- Une fois cette famille de cycle de vie importée, elle apparaît dans la liste des familles mais pas dans la liste des cycles de vie.

Création d'un document « Cycle de vie »

Une fois la famille importée, il faut créer un document basé sur cette famille. Ce document apparaîtra dans la liste des cycles de vie disponible. C'est sur ce document qu'il sera possible d'affecter un profil et donc des droits différents en fonction des besoins.

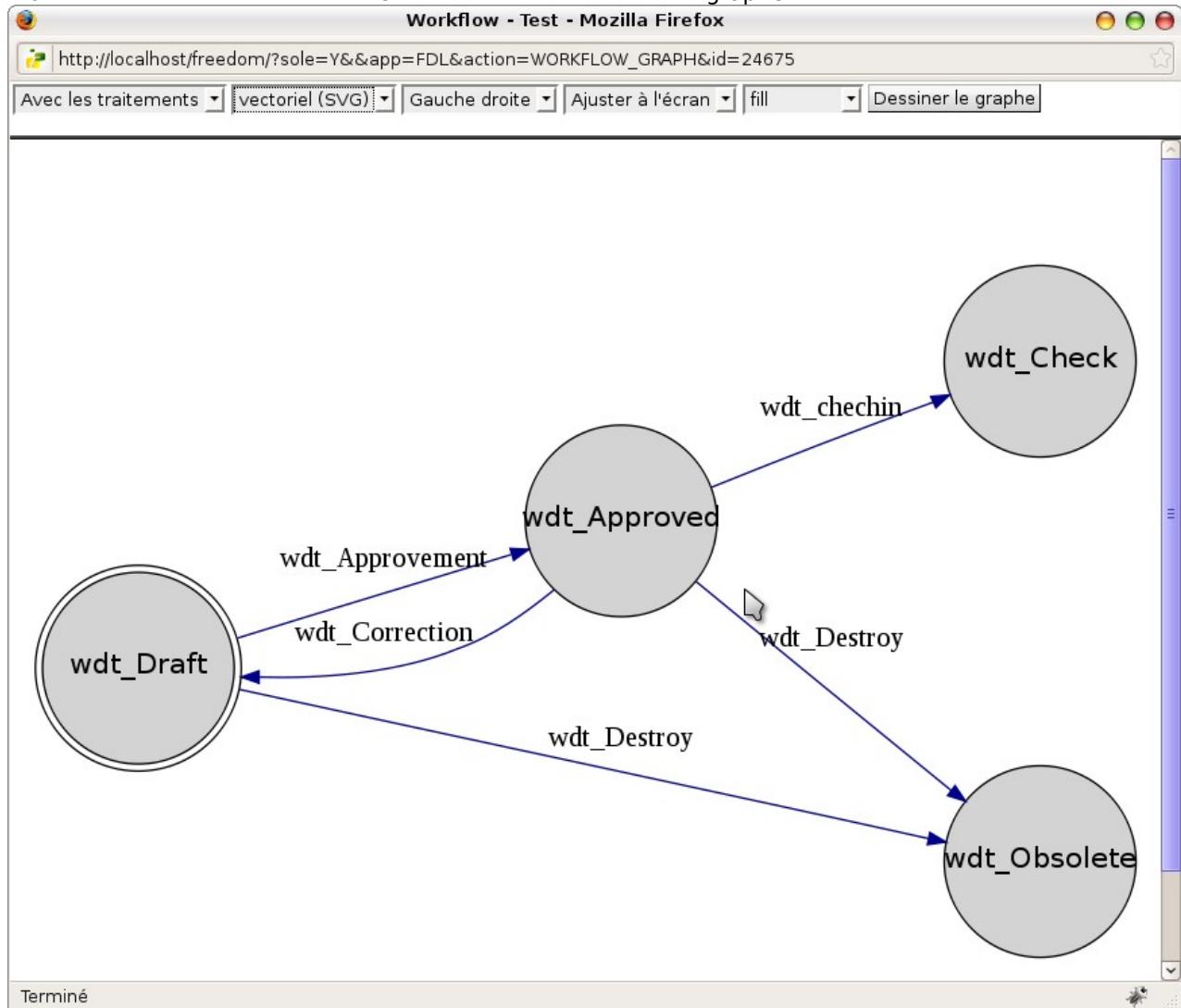
Voici comment créer ce document :

- Sélectionnez la famille de cycle de vie “Test” importée précédemment.
- Faire un clic-droit sur cette famille et sélectionnez “Créer cycle de test”.

¹⁹⁾à éviter sur un site en production

- Donnez un titre à votre document cycle de vie.

Une fois ce document créé, il est possible de voir graphiquement le cycle de vie en cliquant sur le 'Voir le graphe' :



Remarque :

- Il est possible de modifier le fichier PHP « Class.WDocTest.php » et de voir instantanément le changement au niveau du graphe dans ce document cycle de vie et dans tous les documents créés à partir de la famille cycle de vie importée précédemment.
- Il est possible de supprimer, modifier et ajouter des états et des transitions

Affectation d'un cycle de vie à une famille de documents

Le cycle de vie créé doit être associé à une famille de document.

Pour affecter un cycle de vie à une famille, il faut :

20. Sélectionner la famille de documents de votre choix
21. Sélectionner « Choisir un cycle »
22. Sélectionner le cycle de vie créé précédemment

Une fois le cycle choisi, les document de cette famille sont lié à ce nouveau cycle. Si des documents de cette famille étaient déjà liés à un autre cycle, ceux-ci ne sont impactés par ce changement.

Remarque : Il est possible d'utiliser un même cycle de vie pour plusieurs familles de documents.

Initialisation du document cycle de vie

L'initialisation du cycle de vie permet de réinitialiser les attributs du cycle. Ceci est nécessaire lors de la première édition ou lors de la modification du fichier de définition du cycle (classe documentaire). ensuite en éditant ce dernier de sélectionner :

- un profil en fonction de l'état
- une couleur en fonction de l'état
- un masque en fonction de l'état

Pour initialiser un cycle de vie, il faut :

- Sélectionner le document cycle de vie créé précédemment
- Cliquer sur "Initialisation"



Une fois le cycle de vie initialisé, des attributs correspondants aux états sont ajoutés à la famille du cycle de vie.

Basique

Titre : Test

Description :

Famille : ... x

Profil dynamique

Famille : ... x

étais **Transitions**

Paramètres pour l'état wdt_Draft

Profil wdt_Draft :	<input type="text"/>	...	x
Masque wdt_Draft :	<input type="text"/>	...	x +
Couleur wdt_Draft :	<input type="color"/>	...	x
Contrôle de vue wdt_Draft :	<input type="text"/>	...	x +
Modèle de courriel wdt_Draft :	<input type="text"/>	...	x +
Minuteur wdt_Draft :	<input type="text"/>	...	x +
Activité wdt_Draft :	writting		

Type d'affectation wdt_Draft	Utilisateur affecté wdt_Draft	auto-verrouillage wdt_Draft	courriel automatique wdt_Draft
Utilisateur fixe	<input type="text"/> ... x	<input type="checkbox"/>	<input type="checkbox"/>

Paramètres pour l'état wdt_Approved

Pour voir ces attributs ajoutés, il faut :

- Faire un clic droit sur la famille de cycle de vie importée
- Sélectionner "Éditer les attributs"

ATTENTION : Il est conseillé d'initialiser le cycle de vie une fois les états et transitions figés car ensuite, il n'est plus possible de supprimer les attributs ajoutés à la famille du cycle de vie.

Affectation d'un profil dédié au document cycle de vie

L'affectation d'un profil au document cycle de vie permet d'affecter des droits différents pour chaque étape de ce cycle. Pour cela, il faut :

- Sélectionner le document de cycle de vie dans le dossier de recherche « les cycles »
- Sur le document du cycle de vie, sélectionner « Sécurité/Changer de profil » et sélectionner « Contrôle dédié »
- Sur le document du cycle de vie et sélectionner « Sécurité / Accessibilité »
- Dans le tableau qui apparaît, il faut cliquer sur les petits rectangles gris à droite des groupes pour faire apparaître des cases à cocher

* Il faut sélectionner les cases à cocher en fonction de vos besoins et valider

Pour plus d'information sur la gestion des profils, il faut suivre la [documentation correspondante](#).

Test		Voir	éditer	Supprimer	Wdt_Approvement	Wdt_Correction	Wdt_Chechin	Wdt_Destroy	Voir les droits	Modifier les droits
Utilisateurs										
Administrateurs										
		Valider	Réinitialisation	Seulement les cochés						

Terminé

Pour plus d'explications sur la modification des droits du profil, il faut suivre la documentation sur les profils

Remarques :

- Par défaut, un simple utilisateur n'a pas accès aux transitions du cycle de vie. Il faut donc obligatoirement modifier le profil pour autoriser les personnes à modifier les étapes du cycle de vie.
- L'administrateur de Freedom a accès à toutes les étapes d'un cycle de vie et à tout moment et quelque soit les transitions autorisées.
- Il n'est pas possible d'affecter un profil à un document cycle de vie. Il faut obligatoirement créer un « Contrôle dédié ». Cependant, il est possible d'affecter un profil différent pour chaque étape du cycle de vie une fois le cycle initialisé.
- Il est possible de créer un cycle de vie générique avec de nombreuses étapes et transitions. A partir de là, il est possible de créer plusieurs documents de cycles de vie avec chacun un profil différent.

Test du cycle de vie

Avant de tester le cycle de vie, il faut affecter des droits différents à au moins deux utilisateurs au niveau du profil et voir comment évolue le document en fonction des états. En fonction de l'état du document et de l'utilisateur, les boutons permettant de passer d'un état à un autre sont modifiés.

Pour tester le cycle de vie il faut simplement créer un nouveau document basé sur la famille utilisant ce cycle de vie.

The screenshot shows a 'de base' (base) dialog box titled 'CRÉATION DE BASE'. On the right, there is a dropdown menu labeled 'wdt_Draft' containing five items: 'inchangé', 'wdt_Draft', 'wdt_Approved', 'wdt_Check', and 'wdt_Obslete'. The 'wdt_Draft' item is currently selected and highlighted in blue.

The screenshot shows a 'de base' (base) document titled 'TEST N°1'. On the right, there is a dropdown menu containing four items: 'Wdt_Draft', 'Wdt_Approved', 'Wdt_Check', and 'Wdt_Obslete'. The 'Wdt_Draft' item is currently selected and highlighted in blue.

Remarque : L'administrateur de Freedom a toujours accès à tous les boutons de changement d'état quelque soit le paramétrage du cycle de vie.

Affectation d'un profil différent pour chaque étape du cycle de vie

Pour gérer encore plus finement les droits, il est possible d'affecter un profil différent sur chaque étape du cycle de vie.

Pour cela, il faut créer un profil différent pour chaque étape du cycle de vie. Pour créer un profil, il faut suivre [cette documentation](#).

Ensuite, il faut modifier le document cycle de vie et affecter les profils aux différents états

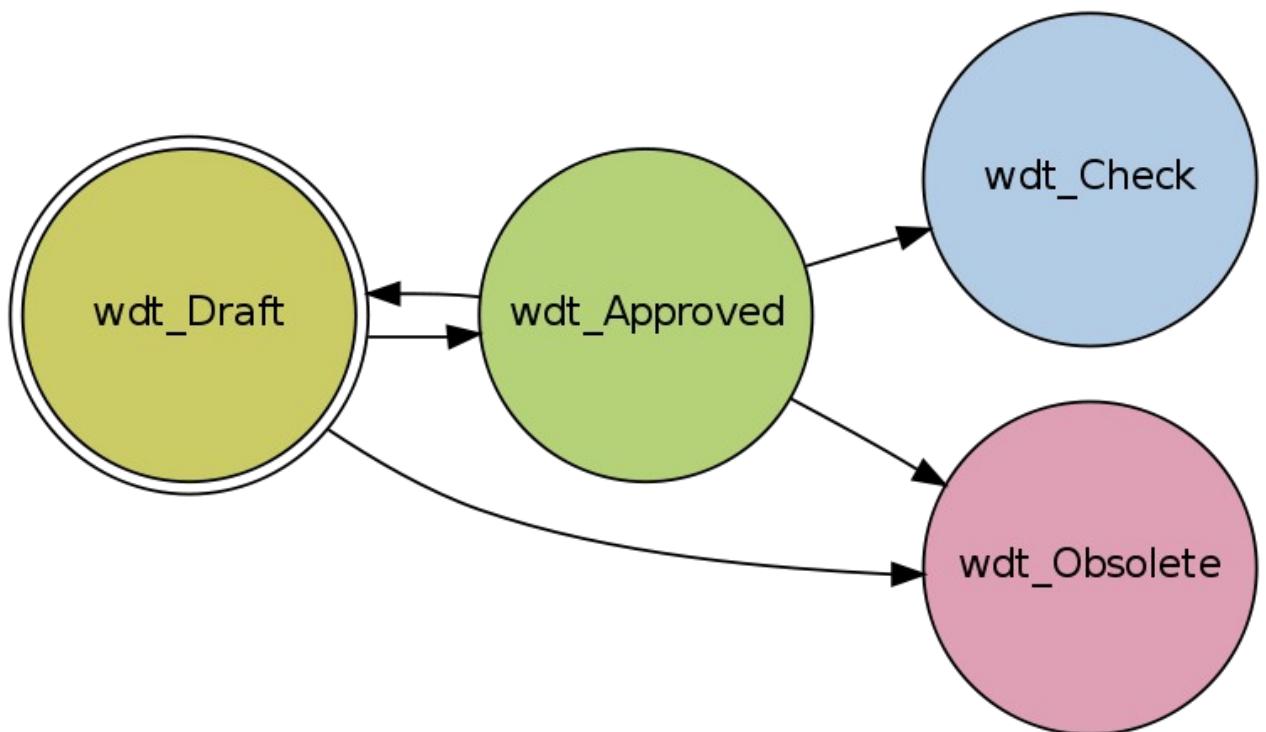
Remarque : Cela est possible seulement si le cycle de vie a été initialisé comme indiqué dans un chapitre précédent.

Ajouter un repère de couleur à chaque étape du cycle de vie

Pour identifier plus facilement dans quel état est le document, il est possible d'affecter des couleurs différentes pour chaque état du cycle de vie. Pour cela, il faut :

- Modifier le document cycle de vie
- Indiquer pour chaque état du cycle de vie, la couleur souhaitée

Une fois les couleurs définies, voici le résultat obtenu dans les différents écrans :



- Changement d'état d'un document sous forme de liste déroulante



- Liste des documents dans le module "Une Famille"



Ajouter un masque différent à chaque étape du cycle de vie

En fonction de l'état du document il est possible d'afficher ou masquer certains champs en appliquant un masque différent à chaque étape.

Pour créer un masque, il faut :

- Depuis le module "Gestion documentaire", sélectionner le menu "Création / Document système".
- Dans la liste "hérite de", sélectionner "masque de saisie".
- Dans la liste "pour" sélectionner une famille.
- Indiquer le nom du masque de saisie
- Pour chaque attribut, il est possible de modifier sa visibilité et son obligation

The screenshot shows the configuration interface for 'TEST CYCLE 1 - FAMILLE'. It includes fields for 'hérite de' (set to 'masque de saisie'), 'pour' (set to 'Test Cycle 1 - Famille'), and a table for defining visibility and obligations for attributes like 'nom' and 'titre' across different states ('basique').

cadre	Nom	nouvelle visibilité	Nouvelle obligation	visibilité par défaut
basique	nom	lecture écriture	N	lecture écriture
basique	titre	statique	N	écriture seule

Sauver Annuler

Pour affecter un masque sur une étape du cycle de vie, il faut :

- Créer un masque différent pour chaque étape comme expliqué ci-dessous.
- Modifier le document cycle de vie.
- Indiquer pour chaque état du cycle de vie, le masque souhaité.

Remarque : Pour retrouver facilement la liste des masques de saisie, il est possible de sauvegarder une recherche simple basée sur la famille "masque de saisie"

Ajouter un contrôle de vue par état

Si le masque ne suffit pas pour renseigner différentes parties du document, il est possible d'associer un [contrôle de vue](#) par état. Le contrôle de vue permettra d'avoir des masques différents par états. Le profilage de chaque contrôle de vue permettra d'appliquer différents masques suivant l'utilisateur.

Définir les activités par états

Version 2.11.12

Les états du cycle définissent l'état du document à un moment donné. Il sont définis par des adjectifs comme 'validé', 'archivé', 'contrôlé'. Pour chacun de ces états, une activité peut être définie afin d'indiquer ce que doit faire celui qui a en charge du document. Cette activité peut être par exemple 'rédition', 'vérification', 'à confirmer'. Les activités peuvent être notées dans les attributs 'activité' du cycle de vie. Ils peuvent ainsi être différents par instance de cycle. Ils peuvent

aussi être définis dans la classe du cycle, dans ce cas les attributs seront non modifiable par l'interface. Dans l'exemple ci-dessous, pour l'état 'validé' nous avons défini l'activité 'en cours d'archivage'.

T2 - Mozilla Firefox

http://localhost/freedom/?sole=Y&&app=FDL&action=FDL_CARD&props=N&abstra...

cycle de test

T2

Éditer | Supprimer | Historique | Ajouter un post-it | Initialisation | Voir le graphe | Autres |

états

Paramètres pour l'état wdt_Draft

Couleur wdt_Draft : #CCCC66

Activité wdt_Draft : writting

Paramètres pour l'état wdt_Approved

Couleur wdt_Approved : #B5D279

Activité wdt_Approved : check in

Paramètres pour l'état wdt_Check

Couleur wdt_Check : #B3CCE6

Paramètres pour l'état wdt_Obsolète

Couleur wdt_Obsolète : #DF9FB5

Voir le graphe

Terminé

Test de base

TEST N°1

Éditer | Changer d'état | Supprimer | Historique | Ajouter un post-it | Autres |

En cours d'archivage (Validé)

Affecter un utilisateur

Lorsqu'on a défini une activité, on peut aussi définir qui doit faire l'activité. Bien qu'une activité puisse être réalisée à plusieurs, un seul responsable peut être désigné. Un document ne peut être affecté qu'à un seul utilisateur à la fois.

Paramètres pour l'état Archivé

Profil Archivé :	<input type="text"/>	...	x
Masque Archivé :	<input type="text"/>	...	x
Couleur Archivé :	#8CBFD9 <input type="button" value="..."/> <input type="button" value="x"/>		
Contrôle de vue Archivé :	<input type="text"/>	...	x
Modèle de courriel Archivé :	<input type="text"/>	...	x
Minuteur Archivé :	<input type="text"/>	...	x
Activité Archivé :	<input type="text"/>	...	x
Type d'affectation Archivé	Utilisateur affecté Archivé	auto-verrouillage Archivé	courriel automatique Archivé
Relation document <input type="button" value="..."/>	bt_idchecker (document) <input type="button" value="..."/> <input type="button" value="x"/>	<input type="checkbox"/>	<input type="checkbox"/>

On peut récupérer l'acteur à partir d'attribut relation (type 'docid') référençant un utilisateur. Cet attribut peut être récupéré à partir du document (attribut ou paramètre de famille) ou à partir du cycle (attribut ou paramètre de famille du cycle). Dans l'exemple ci-dessus, le documentaliste sera affecté comme responsable lorsque le document passera dans l'état archivé. Si vous cochez "auto-verrouillage" le document sera verrouillé pour l'acteur. Si vous cochez "courriel automatique", un courriel d'affectation sera envoyé à l'acteur au moment du changement d'état.

Une fois l'utilisateur affecté, il peut ré-affecter le document s'il a le droit d'édition du document. Dans ce cas le menu "ré-affecter" apparaîtra sur le document. L'affectation est une des propriétés du document. On retrouvera la personne responsable lorsqu'on consulte les propriétés du document. Pour connaître la liste des documents qui vous sont affectés on peut utiliser le service portail "mes documents affectés" depuis l'application portail. Ce service utilise la recherche 'MY_AFFECT_DOCS' qui peut être utilisés aussi dans la gestion documentaire et comme flux RSS.

  recherche détaillée

MES DOCUMENTS AFFECTÉS



Éditer | Supprimer | Historique | Ajouter un post-it | Ouvrir | Autres |

Basique

- Utilisable comme flux oui
- RSS :
- Flux RSS système : oui
- À utiliser dans les menus : non

Critère

- Révision : courante

Conditions

- satisfait toutes les conditions
 - affecté égal ::getMyAttribute(us_whatid)

Pour aller plus loin

Pour aller plus loin dans l'utilisation des cycles de vie, il faut suivre le [chapitre correspondant](#) dans le manuel de programmation.

3.9.2.2 Créer un cycle de vie localisé

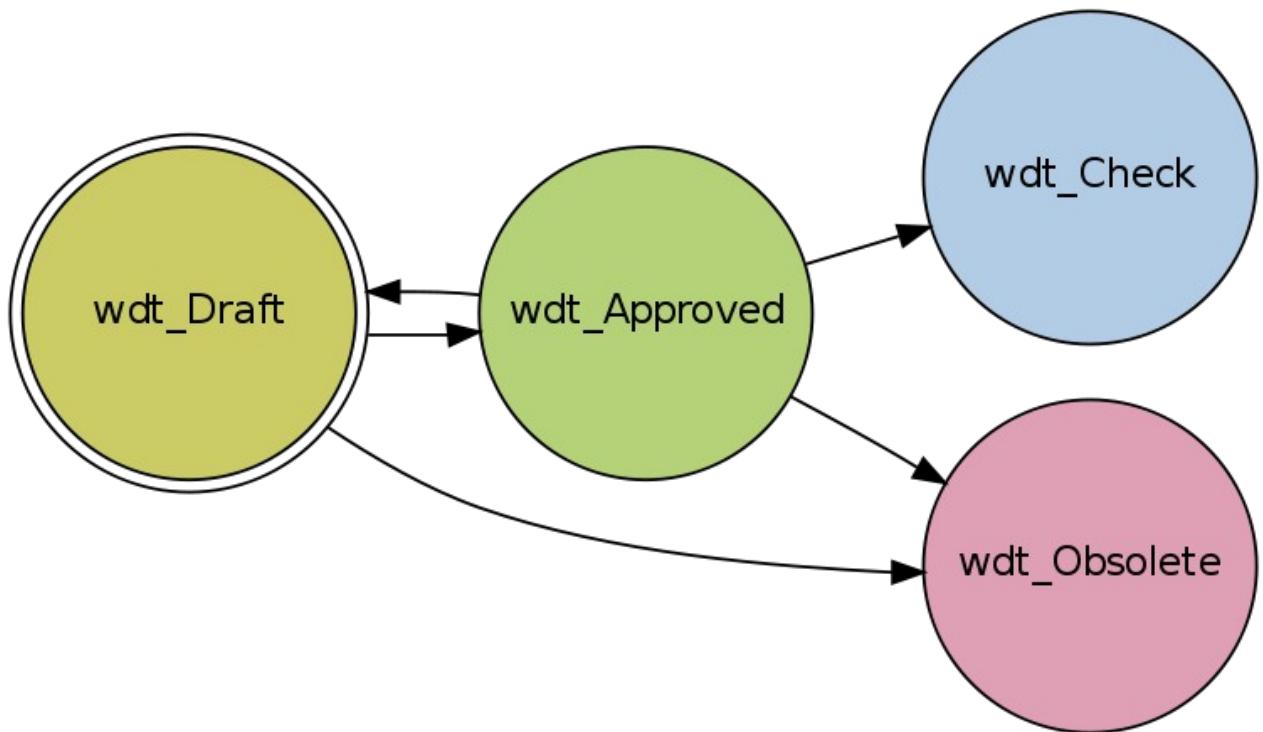
But de ce document

Dans la définition des étapes et des transitons d'un cycle de vie, nous avons vu qu'il n'était pas possible de mettre des espaces et des caractères accentués dans les noms.

En effet, ces noms sont utilisés comme attributs de la famille cycle de vie. Ils correspondent également aux noms des champs de la table dans la base de données.

Le but de ce document est donc d'expliquer comment utiliser la localisation pour traduire ces attributs en textes plus complets dans la langue de votre choix (en français ou en anglais)

Voici l'exemple de cycle que nous allons réaliser dans ce document :



Pré-requis

Avoir déjà mis en place un cycle de vie simple comme indiqué dans cette documentation :

- [Cycle de vie](#)

Exemple de définition de cycle de vie utilisé pour ce document

Pour ce document, nous allons mettre en place un cycle de vie très simple avec seulement deux étapes et deux transitions.

Attention : Les lignes en commentaire avec les `_()` et `N_()` sont importantes car ce sont elles qui nous permettront de localiser les textes dans les différentes langues (français et anglais)

```

<?php
include_once("FDL/Class.WDoc.php");

define ("i18n","i18n");

Class WDocTest extends WDoc {
  public $attrPrefix="WDT";
  
```

```

public $firstState="wdt_Draft";

public $transitions = array("wdt_Approvement"=>array(),
_("wdt_Approvement") _("wdt_Correction") _("wdt_Chechin")
    "wdt_Correction"=>array(),
    "wdt_Chechin"=>array(),
    "wdt_Destroy"=>array());

public $cycle = array(array("e1"=>"wdt_Draft", # _("wdt_Destroy")
_("wdt_Draft") _("wdt_Approved") _("wdt_Check") _("wdt_Obslete")
    "e2"=>"wdt_Approved",
    "t"=>"wdt_Approvement"),
array("e1"=>"wdt_Approved",
    "e2"=>"wdt_Check",
    "t"=>"wdt_Chechin"),
array("e1"=>"wdt_Approved",
    "e2"=>"wdt_Draft",
    "t"=>"wdt_Correction"),
array("e1"=>"wdt_Draft",
    "e2"=>"wdt_Obslete",
    "t"=>"wdt_Destroy"),
array("e1"=>"wdt_Approved",
    "e2"=>"wdt_Obslete",
    "t"=>"wdt_Destroy"));
function postConstructor() {
    $this->stateactivity=array("wdt_Draft"=>N_("writting"),
        "wdt_Approved"=>N_("check in"));
}
?>

```

Traduction des constantes dans différentes langues

La première étape consiste à récupérer le contenu des constantes du fichier .php précédent pour générer un nouveau fichier .po qui nous permettra de renseigner les traductions :

```
# xgettext --keyword='N_' --language=c --keyword='_'
Class.WDocTest.php
```

Ensuite, il faut éditer ce fichier texte « .po », indiquer le bon encodage sur la ligne « Content-Type » et ajouter les traductions. Exemple :

```

msgid ""
msgstr ""
"Project-Id-Version: TEST\n"
"Report-Msgid-Bugs-To: \n"
"POT-Creation-Date: 2009-01-02 23:42+0100\n"
"PO-Revision-Date: 2009-01-02 23:56+0100\n"
"Last-Translator: me\n"
"Language-Team: french <LL2@li.org>\n"
"MIME-Version: 1.0\n"
"Content-Type: text/plain; charset=iso-8859-1\n"
"Content-Transfer-Encoding: 8bit\n"

#: Class.WDocTest.php:10
msgid "wdt_Approvement"
msgstr "Approbation"
```

```

#: Class.WDocTest.php:10
msgid "wdt_Correction"
msgstr "Correction"

#: Class.WDocTest.php:10
msgid "wdt_Chechin"
msgstr "Archivage"

#: Class.WDocTest.php:15
msgid "wdt_Destroy"
msgstr "Destruction"

#: Class.WDocTest.php:15
msgid "wdt_Draft"
msgstr "Initialisé"

#: Class.WDocTest.php:15
msgid "wdt_Approved"
msgstr "Validé"

#: Class.WDocTest.php:15
msgid "wdt_Check"
msgstr "Archivé"

#: Class.WDocTest.php:15
msgid "wdt_Obslete"
msgstr "Obsolète"

#: Class.WDocTest.php:31
msgid "writting"
msgstr "En rédaction"

#: Class.WDocTest.php:32
msgid "check in"
msgstr "En cours d'archivage"

```

Remarques : * Pour mettre en place une traduction en anglais, il faut créer un deuxième fichier .po et indiquer les traductions en anglais.

- Pour éditer ce fichier, il est possible d'utiliser des interfaces graphiques spécialisées comme gtranslator (gnome), kbabel (kde) ou poedit (gtk)

Mise en place de la traduction

Il faut commencer par convertir le fichier texte « .po » en fichier binaire « .mo » :

```
# msgfmt test_fr.po -o /usr/share/what/locale/fr/LC_MESSAGES/test.mo
```

Remarque : La commande « msgunfmt » permet de faire l'inverse :

```
# msgunfmt fichier.mo -o fichier.po
```

Remarque : Pour mettre en place une traduction en anglais, il faut utiliser le dossier « locale/en/LC_MESSAGES/ »

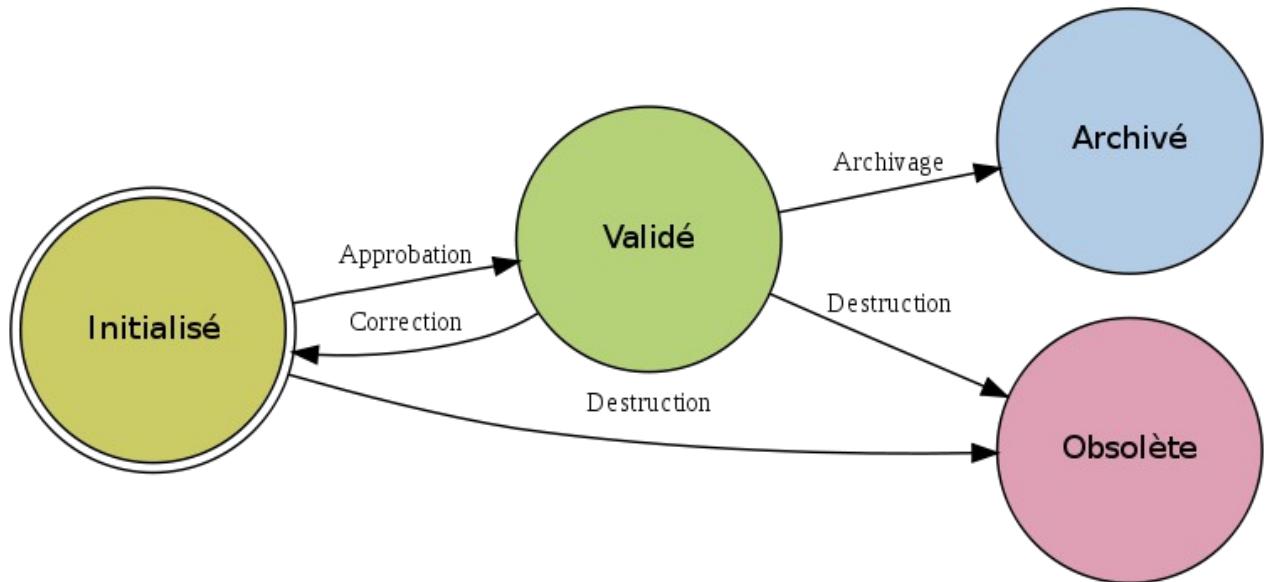
Assembler les différents fichiers « .mo » dans « what.mo » utilisé par Freedom :

```
# /usr/share/what/whattext
```

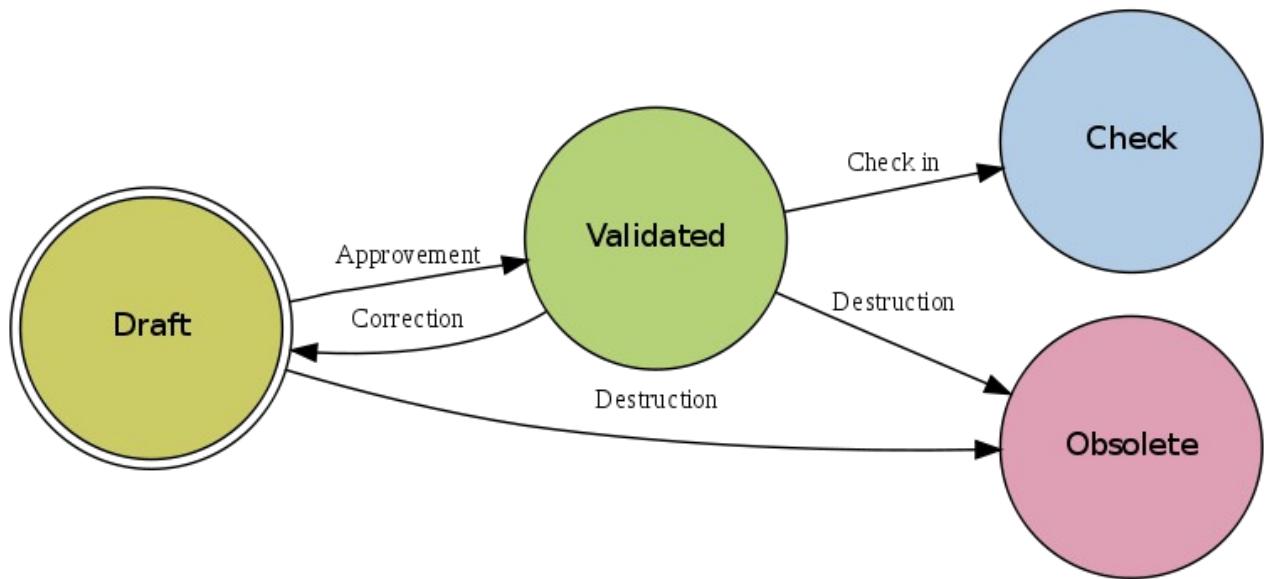
Recharger Apache :

```
# /etc/init.d/apache2 reload
```

Remarque : Le cycle de vie doit apparaître traduit immédiatement.



Pour avoir le graphe en anglais, on procède de la même façon en copiant le fichier de traduction dans le répertoire "locale/en/LC_MESSAGES".



Paramétriser Freedom pour avoir une interface en anglais

Pour avoir l'interface de Freedom en anglais pour tout les utilisateurs, il faut modifier ce paramètre :

- Administration / Gestion des applications / paramètres applicatif / Noyau / langue

Pour avoir l'interface de Freedom en anglais pour un utilisateur particulier, il faut modifier ce paramètre :

- Administration / Gestion des applications / mes paramètres / Noyau / langue

Remarques :

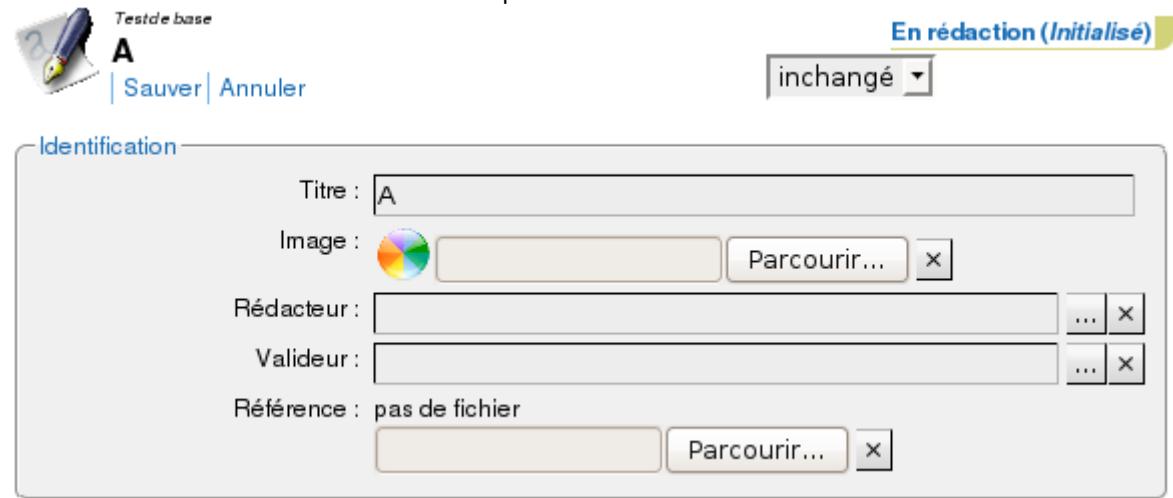
- Après avoir modifié la langue, il est nécessaire de se déconnecter et de se reconnecter à Freedom.

Version > 2.11.12

3.9.2.3 Modèle de courriel pour les cycles

Il est possible de paramétrer le cycle par l'interface afin d'envoyer des courriels lors d'un changement d'état. Les courriels peuvent être définis pour un état ou pour une transition. Si on associe un ou plusieurs courriels à un état, ils seront envoyés à chaque fois qu'un document aura ce nouvel état (quel que soit la transition). Les courriels liés aux transitions seront envoyés lors du passage de la transition.

Soit un document basique associé au cycle de test. Nous prenons comme exemple la famille déclarée dans le fichier FIXME internalmedia: freedom_2.14:admin:cycle_de_vie:wbasictest.ods. Cette famille est associée au cycle 'Cycle de test' défini dans le fichier FIXME internalmedia: freedom_2.14:admin:cycle_de_vie:class.wdoctest.php - à mettre dans le répertoire '/usr/share/what/FDL' du serveur (voir [Création de cycle](#) pour plus de détail). Voici le document de test en édition. Il comprend un rédacteur et un validateur.



Le but est d'envoyer un courriel pour avertir le documentaliste d'archiver le document lorsque celui-ci est validé (passage à l'état validé).

Pour cela on édite le cycle de test et clique sur le '+' de l'attribut 'Modèle de courriel Validé'. Cela affiche le document suivant :

éditer Demande d'archivage - Mozilla Firefox

http://localhost/freedom/?sole=Y&&app=GENERIC&action=GENERIC_EDIT&rzone=&id=24701

modèle de mail

DEMANDE D'ARCHIVAGE

| Sauver | Annuler

Entête

Titre : Demande d'archivage

Famille : Test de base

Famille cycle : cycle de test

Emetteur

type	De
Attribut relation	bt_idredac (rédacteur)

Destinataires

-	type	destinataire
→	À	bt_idchecker (documentaliste)
→	Cc	bt_idvalid (validateur)
→	Cc	wdt_mailchief (Courriel du chef)
+ (ajouter)		

Sujet : Demande d'archivage [TITLE]

Contenu

Avec lien :

Corps :

De [V_BT_IDREDAC],
 Bonjour veuillez archiver le document suivant [V_TITLE].
[WCOMMENT]
 Il a été validé par [V_BT_IDVALID].
 Cordialement [V_BT_IDREDAC]

Attachments

Attachement
bt_fileref (Référence)
+ (ajouter)

Terminé

Le modèle de courriel permet de définir l'émetteur, les destinataires et le corps du message.

Définir l'émetteur

L'émetteur doit faire référence à une adresse email valide. Si l'émetteur reste vide, c'est l'adresse de l'utilisateur qui fait la transition qui sera utilisée.

Plusieurs possibilités de choix de l'émetteur sont possibles:

- adresse fixe : choisir dans le carnet d'adresse une personne. C'est l'adresse mail de cette personne qui sera utilisée. (attribut US_MAIL). L'adresse indiquée doit uniquement contenir l'adresse "pure" : préférez **jean.dupont@mail.fr** à **jean dupont jean.dupont@mail.fr**.
- attribut texte : un attribut du document portant le cycle. Cet attribut doit renseigner une adresse email. Il est possible d'utiliser la notation ':' (TST_MYID:THE_MAIL) pour aller chercher des valeurs sur les documents liés.
- attribut relation : un attribut de type 'docid' du document portant le cycle. Cet attribut doit renseigner une personne ou un groupe. Ce sera l'attribut 'US_MAIL'/'GRP_MAIL' de la personne/groupe lié à cet attribut qui sera utilisé pour renseigner l'émetteur.
- attribut paramètre : un attribut paramètre (PARAM) de la famille du document portant le cycle. Cet attribut doit renseigner une adresse email. Il est possible d'utiliser la notation ':' pour aller chercher des valeurs sur les documents liés.
- attribut cycle : un attribut du document cycle de vie. Cet attribut doit renseigner une adresse email. Il est possible d'utiliser la notation ':'.
- relation cycle : un attribut de type 'docid' du document cycle de vie. Ce sera l'attribut 'US_MAIL'/'GRP_MAIL' de la personne/groupe lié à cet attribut qui sera utilisé pour renseigner l'émetteur.
- paramètre cycle : un attribut paramètre (PARAM) de la famille du cycle de vie portant le cycle. Cet attribut doit renseigner une adresse email. Il est possible d'utiliser la notation ':' pour aller chercher des valeurs sur les documents liés.

Définir les destinataires

La définition des destinataires se fait comme pour l'émetteur. Vous pouvez choisir ensuite le mode d'envoi :

- à : destinataire principal
- cc : destinataire en copie
- bcc : destinataire en copie cachée

Définir le sujet

Le sujet est un texte libre. Il peut contenir des parties variables issues du document qui va être envoyé. On notera les attributs (en majuscules) entre crochets. Exemple [BT_APPROVDATE] si le document à un attribut 'BT_APPROVDATE'. La partie entre crochets sera remplacée par la valeur de l'attribut.

Définir le corps du message

Le corps du messages est un texte HTML. Il peut contenir des parties variables qui sont les attributs du document et les [paramètres de transitions](#) du cycle. Ces parties variables peuvent être notées de 2 formes :

1. [MY_ATTR] : cela affichera la valeur *brute*, c'est à dire la valeur inscrite en base de données.
2. [V_MY_ATTR] : cela affichera la valeur *formatée*, c'est à dire telle qu'elle est présentée à l'utilisateur sur l'interface web.

Concrètement, par exemple, la valeur brute d'un énuméré sera la clef, la valeur formaté sera la traduction du libellé. Le barre de menu de l'éditeur de texte propose les différentes possibilités pour les attributs formatés du document. Le commentaire de transition peut être récupéré par le mot clef '[WCOMMENT]'. Les propriétés du document peuvent aussi être affichée : [ID] [TITLE] [CDATE]. Par contre, les [V_ID] ne sont pas possibles; ce ne sont pas des attributs. Seul [V_TITLE] est possible pour afficher un lien vers le document.

Si on utilise les valeurs formatées, des hyperliens peuvent apparaître dans le courriel à destination du serveur d'envoi. Si vous ne souhaitez pas avoir de liens il faut décocher la case 'avec liens'. Ceci est à faire si vous destinez votre courriel à des personnes extérieures à votre système

d'information.

Tout mail sans corps ne sera pas envoyé.

Ajouter des fichiers attachés

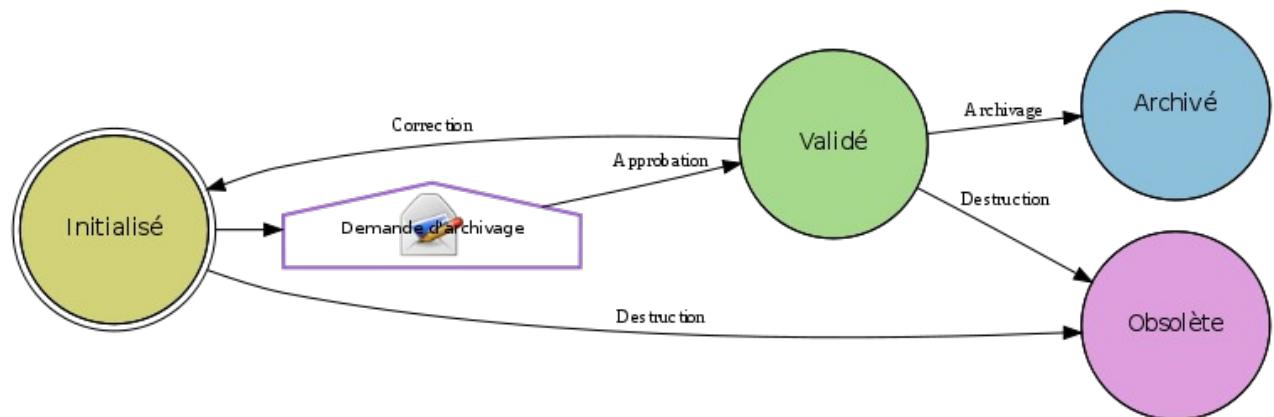
Les fichiers attachés font référence à des attributs de type fichier (ou image) du document. Les fichiers seront alors en pièce jointe du courriel. Bien sûr ces attribut peuvent être des listes de fichiers (attribut fichier ou image dans un tableau).

Enregistrer le messages

Depuis la version 2.14.2 de freedom, vous pouvez conserver l'enregistrement du message envoyé. Celui-ci sera stocké dans la famille "message envoyé". Son profil sera celui du document servant à l'envoi. Pour stocker le message il faut cliquer sur le champ "Enregistrer une copie".

3.9.2.4 Ajout de courriel pour les cycles

Une fois le modèle de courriel créé, il faut l'associer au cycle (utiliser les '...' pour sélectionner le modèle. Les envois de courriel sont visibles dans le graphe du cycle avec l'option 'avec les traitements'.



Les modèles de courriels peuvent être associés à un état ou à une transition. Lorsqu'on mets le courriel sur un état il sera envoyé dès que le document changera vers cet état, même si la transition est non définie (cas possible lorsque l'utilisateur 'admin' utilise le cycle).

Version > 2.11.12

3.9.2.5 Minuteurs

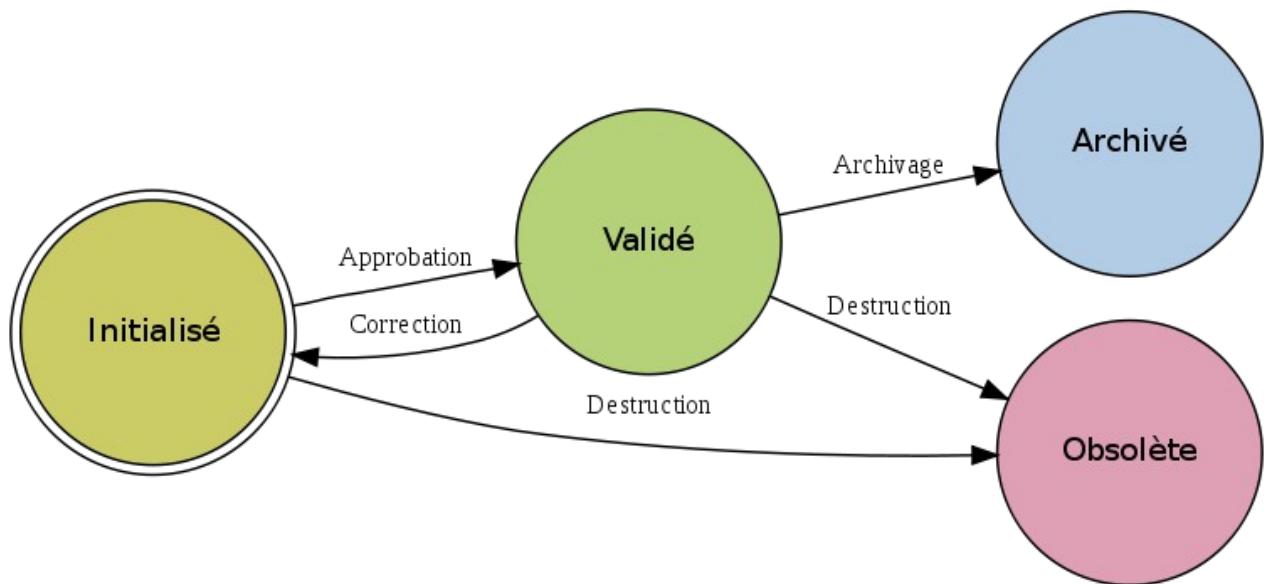
Les minuteurs ("timers" in english language) permettent de déclencher des actions sur des documents à des moments définis.

Définition

Le minuteur dans le cadre du cycle de vie va permettre par exemple de donner une durée limite pour l'activité à réaliser dans un état défini.

Exemple : "vous avez une semaine pour valider ce contrat sinon il sera automatiquement déclaré comme obsolète."

Nous reprenons ici le même cycle que pour les paragraphes précédents.



Pour mettre en place l'exemple, nous allons créer un minuteur qui fera un changement d'état vers "obsolète" 7 jours après le passage en "initialisé" (ou ici dès que le document est créé). Pour avertir le rédacteur, nous allons l'avertir 3 jours avant en lui précisant qu'il faudrait y penser et ensuite un jour avant, avec en copie le valideur, avant afin qu'il finisse son travail (non non on lui mets pas la pression 😊).

Création

Pour faire cela nous allons créer un minuteur. Nous éditons le cycle et avec le bouton "+" de l'attribut "Minuteur".

• CRÉATION MINUTEUR

Identification

Titre :	Obsolescence une semaine
Famille :	Test de base
Famille cycle :	cycle de test

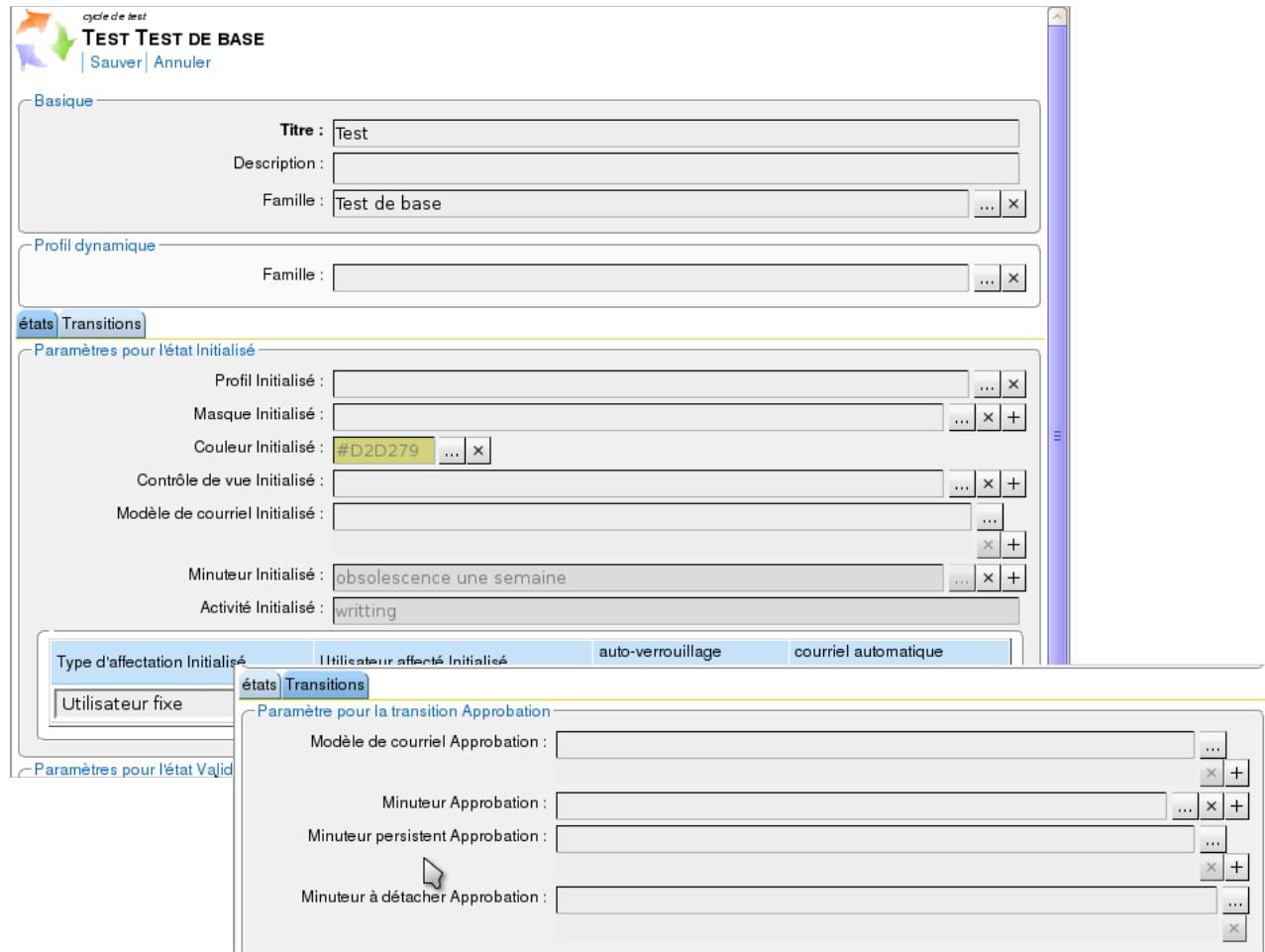
Configuration

Délai (en jours)	Délai (en heures)	Nombre d'itérations	Modèle de mail	Nouvel état	Méthode
4		1	Plus que 3 jours		
2		1	Reste un jour		
1		1		wdt_Obsolete	
Actions					
<input type="checkbox"/> Envoyer un courriel <input type="checkbox"/> Changer l'état <input type="checkbox"/> Appeler une méthode					

Nous pouvons indiquer les actions à faire une fois que le minuteur sera attaché au document. Trois types d'actions sont possibles :

1. envoyer un courriel (ou plusieurs) en utilisant les modèles de courriel
2. changer d'état
3. appeler une [méthode du document](#).

Lors d'un changement d'état le minuteur associé à l'état précédent est enlevé. S'il y a un autre minuteur pour l'état suivant il sera activé. Le minuteur peut être mis sur un état ou sur une transition.

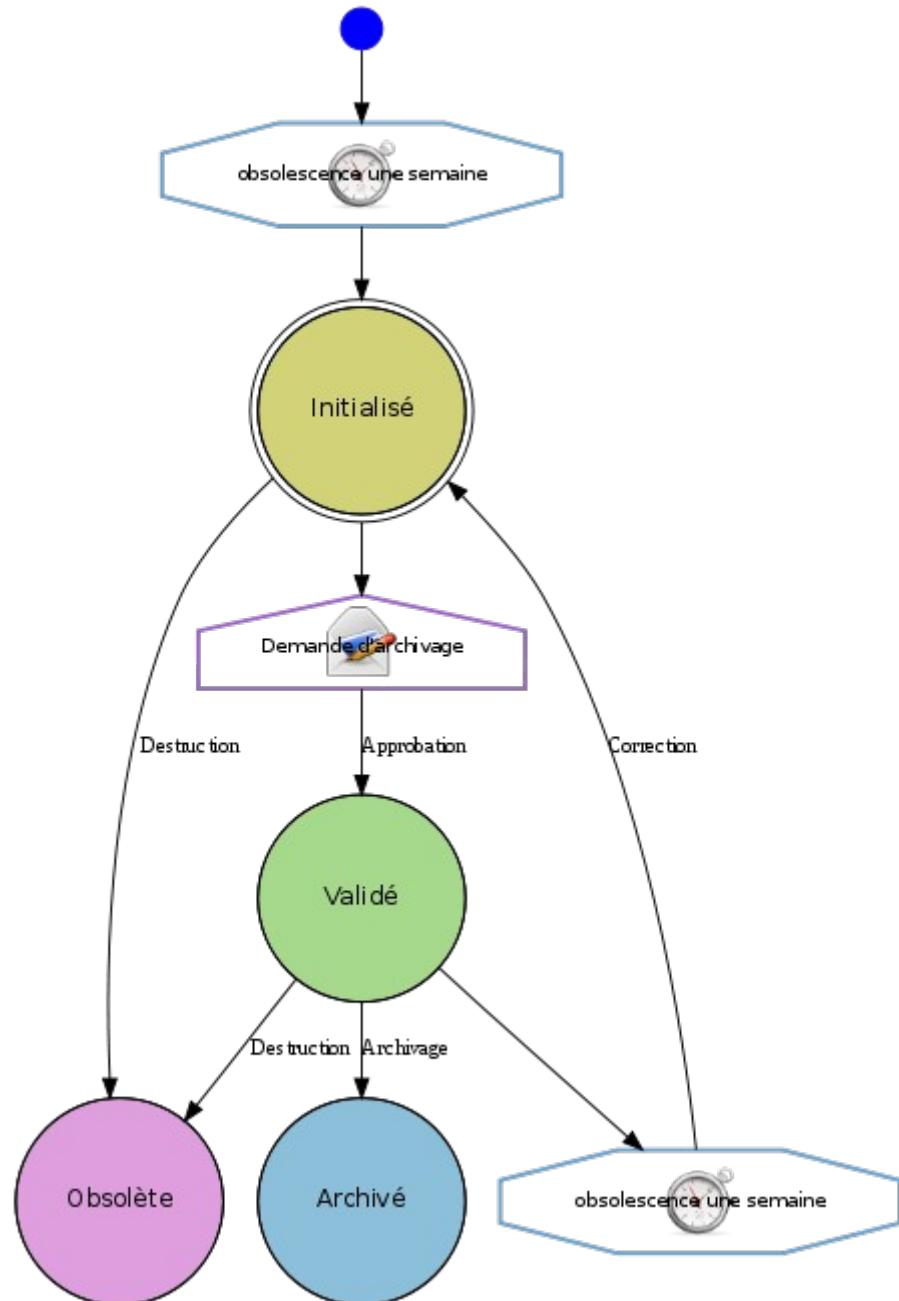


Pour associer un minuteur à un état, il faut éditer le cycle de vie et renseigner les attributs minuteurs dans le cadre état ou dans le cadre de la transition voulu. Sur les transitions, les minuteurs "persistants" ne sont pas détachés de manière implicite lors d'un changement d'état. Ils sont détachés lorsqu'ils sont indiqués dans les attributs "Minuteur à détacher". Ceci permet de lancer des actions qui peuvent durer sur plusieurs transitions.

Lorsqu'un niveau est exécuté (le délai est écoulé), le niveau suivant (rangée suivante du tableau 'configuration') est armé s'il existe. Il est possible de faire des répétitions. Je veux envoyer le même courriel tous les jours pendant 7 jours. Pour faire cela on mettra le nombre d'itérations à 7.

S'il n'y a plus de niveau le minuteur est détaché du document.

Dans notre exemple, nous avons associé le minuteur à l'état "initialise". Dès que l'on crée un document "test de base", le minuteur sera activé.

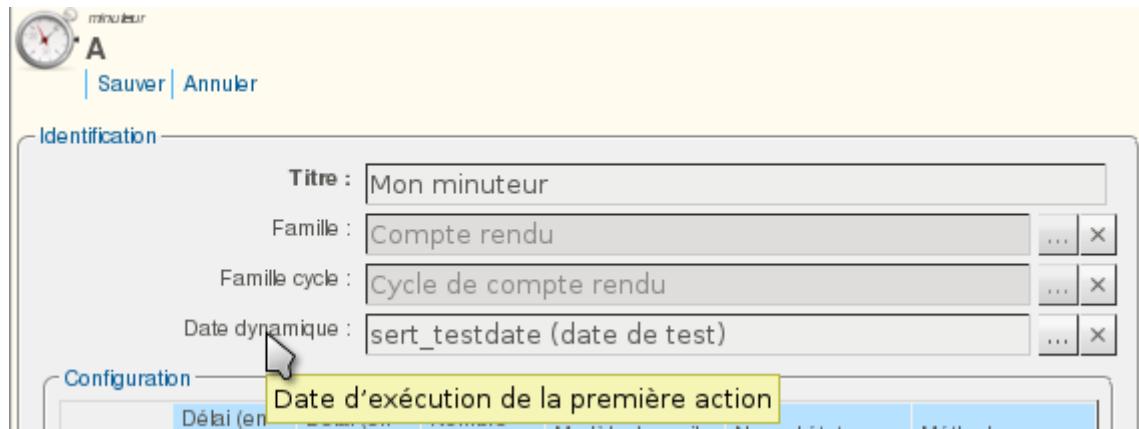


Si l'on regarde l'historique après la création de notre document nous y voyons l'affectation du minuteur.

http://localhost - historique F - Mozilla Firefox			
F			
version	rev&etat	auteur	date
0	En rédaction (Initialisé)		Date de dernière modification mar 06 jan 2009 17:52
	Attachement déclencheur obsolescence une semaine [24727]	Default Master	06/01/2009 17:52:22
	Création du document	Default Master	06/01/2009 17:52:22
Terminé			

Date dynamique

 version > 2.13.12



La date de déclenchement de la première action peut être liée à un des attributs de type 'date' ou 'timestamp' du document auquel est attaché un timer. Ainsi, on peut indiquer que le changement d'état vers 'diffusée' sera exécuté à la date marquée dans l'attribut 'sert_testdate'. Si on veut que ce soit 3 jours après cette date, il faudra ajouter au document un attribut 'sert_threedayafter' qui sera calculé en fonction de 'sert_testdate'.

Administration

L'interface d'administration des minuteurs est disponibles à partir de l'application 'Administration'.

Prochains minuteurs	Document	Action	Durée restante
Historique des minuteurs	F	envoi courriel avec le modèle Plus que 3 jours [24725]	11/01/2009 18:36:00 3 jours 23 heures 44 minutes
Minuteurs dépassés	A	envoi courriel avec le modèle Plus que 3 jours [24725]	11/01/2009 18:36:00 3 jours 23 heures 44 minutes
Nettoyage de l'historique	B	envoi courriel avec le modèle Plus que 3 jours [24725]	11/01/2009 18:37:00 3 jours 23 heures 45 minutes
jours			
Désactiver les minuteurs			
Filtre sur le document:			



niveau	minuteur	date	durée restante	actions
0	obsolescence une semaine	2009-01-11 18:36	3 jours 23 heures 55 minutes	<ul style="list-style-type: none"> Envoi d'un courriel avec le modèle : Plus que 3 jours
1	obsolescence une semaine	2009-01-13 18:36	5 jours 23 heures 55 minutes	<ul style="list-style-type: none"> Envoi d'un courriel avec le modèle : Reste un jour
2	obsolescence une semaine	2009-01-14 18:36	6 jours 23 heures 55 minutes	<ul style="list-style-type: none"> Changement d'état vers : Obsolète

Lorsque l'on clique sur le titre d'un document, cela affiche les prochaines actions qui vont être exécutées.

Prochains minuteurs

Cela affiche par ordre chronologique les prochaines actions qui vont s'exécuter. freedom vérifie toutes les 5 minutes les minuteurs qui doivent être exécuté. L'heure d'activation a donc une précision de 5 minutes. Si l'heure d'activation est dépassé de plus de 2 heures (paramètre FDL_TIMERHOURLIMIT) le minuteur sera ignoré.

Historique des minuteurs

Cela affiche les actions, issus de minuteurs, déjà exécutés. Il sont ordonnés par ordre chronologique d'exécution.

Minuteurs dépassés

Liste les minuteurs qui aurait dû se déclencher depuis plus de 2 heures (FDL_TIMERHOURLIMIT). Cela peut arriver si le serveur a été arrêté pendant plus de 2 heures. Le paramètre applicatif FDL_TIMERHOURLIMIT est modifiable avec l'application de paramétrage :administration/paramètre de configuration/paramètres applicatifs/Bibliothèque freedom.

Nettoyage de l'historique

Permet d'effacer les traces des minuteurs déjà exécutés depuis un certain nombre de jours. Ce nombre de jour est par défaut de 7. Il peut être modifié dans la zone de saisie contigüe. Le nettoyage prend en compte le filtre sur les documents : ne seront nettoyer que ceux dont le titre contient le filtre.

Désactiver les minuteurs

Cela permet d'annuler des minuteurs actifs. Si le filtre est vide tous les minuteurs seront annulés. Si le filtre n'est pas vide seul les minuteurs attachés aux documents filtrés seront pris en compte.

3.9.2.6 Les accords

Sur certains états d'un document, vous pouvez associer des questions telles que 'Approuvez-vous le document ?' ou 'Donnez votre appréciations'. La réponse à ces questions est un ensemble de choix fini.

Lors de l'édition d'un cycle vous associez un ou plusieurs "Accord" à un état. L'accord est un document de la famille "accord" qui définit la question et les réponses possibles.

Configurer un accord

The screenshot shows the 'accord' configuration interface for a document titled 'COULEUR'. The top bar includes a blue speech bubble icon, the title 'accord COULEUR', and buttons for 'Sauver' (Save) and 'Annuler' (Cancel).

Identification

- Référence : Couleur (highlighted in yellow)
- Question : Quelle est votre couleur préférée ?

Réponses possibles

Clef	Libellé
red	rouge
blue	bleu
green	vert
+ (add new)	

Profil dynamique

- Famille : (empty input field with browse and delete buttons)

Pour définir qui doit répondre à la question associées on utilise le mécanisme de profil du document accord. Premièrement vous changez le profil en "contrôle dédié" (menu sécurité/changer de profil). Ensuite vous lancer l'interface de configuration du profil (menu sécurité/accessibilités).

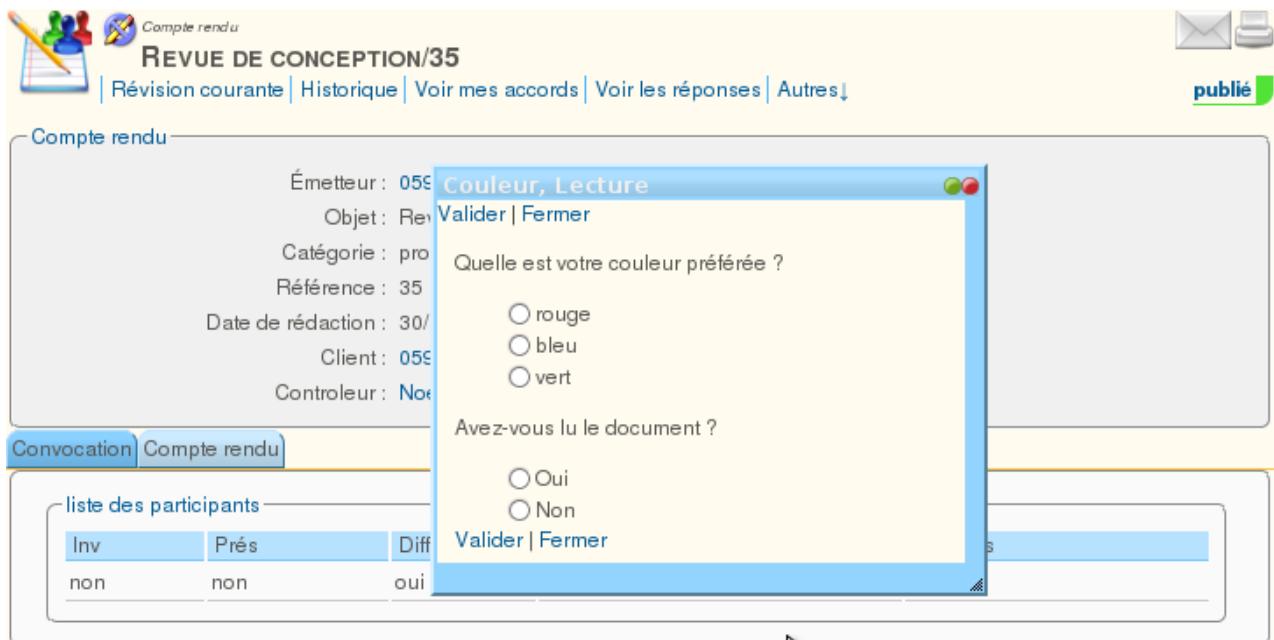
Couleur	Voir	éditer	Supprimer	Envoyer	Déverrouiller	Confidentiel	Forum	Voir les réponses	Répondre	Voir les droits	Modifier les droits
Approbateurs	<input type="checkbox"/>	<input checked="" type="checkbox"/>							<input checked="" type="checkbox"/>		
Chef de pôle/département (CD)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>						<input checked="" type="checkbox"/>	<input type="checkbox"/>	
Chef de section (CS)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>						<input checked="" type="checkbox"/>	<input type="checkbox"/>	
Utilisateurs	<input type="checkbox"/>	<input checked="" type="checkbox"/>									

Le droit 'Répondre' ('answer') indique quels sont les groupes et les utilisateurs qui doivent répondre à la question. Ceux qui n'ont pas ce droit n'ont pas à répondre au questionnaire. Le profil peut être dynamique si vous renseigné l'attribut "famille" dans la cadre "Profil dynamique". Dans ce cas les attributs relations de la famille défini apparaîtront dans le profil. Vous pouvez ainsi indiquer en fonction du document qui doit répondre si vous avez défini dans votre document des attributs relations qui contient les questionnés.

Fonctionnement des accords

Une fois l'accord créé et configuré, nous pouvons l'associer à un état du cycle. Lorsqu'un document consulte un document par l'interface, il consulte généralement la dernière révision du document, celle qui vie et qui, peut-être, a été modifiée depuis le dernier changement d'état. **La question ne peut être posée que sur un document figé.** On ne peut demander un avis sur un document qui peut être modifié. Imaginons que l'on mette un accord sur l'état "publié", la question de l'accord sera posé que sur les documents "publiés". Si le document est dans sa dernière révision, une fenêtre avertira l'utilisateur qu'il doit fournir une réponse sur le document publié (dans sa dernière version).

Lorsque vous cliquez sur "aller au dernier publié" alors le document qui fait l'objet de la question apparaît ainsi que le questionnaire. Une fois que l'utilisateur a répondu aux questions, il peut revenir à la dernière révision en cliquant sur "Révision courante". La fenêtre d'avertissement concernant les accords ne s'affiche plus s'il a déjà répondu. Pour changer sa réponse, il faut passer par l'historique afin d'avoir le dernier document publié puis ensuite cliquer sur "Voir mes accords" pour changer son avis.



Gérer les réponses

Dans les profils de document vous avez le droit 'Voir les réponses' (wask). Si vous disposez de ce droit dans le document où porte l'accord, le menu "Voir les réponses" sera disponible dans le document. Ce menu liste les utilisateurs qui ont répondu (trié par réponse) et ceux qui n'ont pas répondu. Cette liste est issue de la configuration du profil de l'accord : ceux qui ont le droit 'Répondre'.

3.9.2.7 Cycle de vie de publication de documents

But de ce document

Expliquer comment mettre en place un cycle de vie de publication de documents classique du genre « Bouillon → Validé » mais avec la possibilité de repasser à l'état brouillon tout en laissant la dernière version validée accessible aux utilisateurs.

Règles à mettre en place

Voici les règles de fonctionnement que devra respecter notre famille de documents :

- Un document à l'état brouillon n'est ni visible ni modifiable par le groupe « Lecteurs »
- Un document à l'état brouillon est modifiable uniquement par le groupe « Rédacteurs »
- Seule le groupe « Rédacteur » peut changer l'état du document pour le passer à l'état « Validé » ou le repasser à l'état « Brouillon »
- Un document à l'état « Validé » est visible du groupe « Rédacteur » et « Lecteur », mais n'est pas modifiable
- Pour modifier un document « Validé », le groupe « Rédacteur » doit le repasser à l'état « Brouillon »

Cycle de vie et profils

Notre cycle de vie aura donc deux états avec deux profils associés :

- Brouillon : Le profil associé donne accès au document uniquement au groupe « Rédacteurs »
- Validé : Le profil associé donne un accès en lecture seule à tout le monde

Droits sur les changements d'états :

- Les rédacteurs seront les seules à accéder au cycle de vie pour changer le document d'état et seront également les seules à pouvoir modifier le document quand il sera à l'état « Brouillon »

Au final, il faut donc :

- Créer la famille de cycle de vie
- Créer le document cycle de vie
- Associer le cycle de vie la famille
- Créer les deux profils
- Associer les deux profils au cycle de vie

Rapports permettant d'accéder aux documents

Un rapport pour le groupe « Lecteur » permettra de consulter la liste de tous les documents à l'état « Publié » et d'y accéder. Un autre rapport pour le groupe « Rédacteur » permettra d'accéder aux documents à l'état « Brouillon »

Liste des profils nécessaires

- Profil pour la famille
- Profil pour le document à l'état « Brouillon »
- Profil pour le document à l'état « Publié »
- Profil du cycle de vie (Contrôle dédié)

Envoi d'un mail Un mail sera envoyé aux personnes du groupe « Lecteurs » lors de la publication du document.

Menu personnalisés pour la famille

- Un menu dans le document permettra d'accéder au rapport affichant la liste des documents publiés
- Un autre menu accessible uniquement par le groupe « Administrateur » affichera le rapport des documents à l'état Brouillon
- Le menu « Révision courante » sera accessible uniquement au groupe « Rédacteur ».

Installation des documents de l'exemple

Voici la liste des documents à installer :

-

```
FIXME internalmedia: freedom_2.14:admin:cycle_de_vie:class.wpublication.php.txt
```

contient la Class de notre cycle de vie. Il faut enregistrer le contenu de ce fichier dans “FDL / Class.WPublication.php”

-

```
FIXME internalmedia: freedom_2.14:admin:cycle_de_vie:method.publication.php.txt
```

contient la Méthode associé à notre famille pour limiter l'accès aux menus. Il faut enregistrer le contenu de ce fichier dans “FDL / Method.Publication.php”

-

FIXME internalmedia: freedom_2.14:admin:cycle_de_vie:mail_publication.xml.txt

contient le contenu du message envoyé aux lecteurs lors du passage à l'état "Publié". Il faut enregistrer le contenu de ce fichier dans "FDL / Layout / mail_publication.xml"

FIXME internalmedia: freedom_2.14:admin:cycle_de_vie:publication_0.1.ods

permet d'importer dans Freedom tous les documents nécessaires

Une fois le fichier OOo importé, vous devriez avoir dans le dossier "Freedom Master / Publication", les fichiers suivant :



Remarques :

- Après avoir importé une première fois le document OOo, il faut initialiser le document cycle de vie et importer une deuxième fois le document OOo pour que le cycle de vie soit correctement paramétré.
- Il faut également activer les profils et modifier les accessibilités pour que le système soit opérationnel (ex : Il faut compléter le groupe des rédacteurs pour que le rapport permettant d'accéder aux brouillons soit accessible)

Résultats

Exemple de document à l'état publié :

The screenshot shows the Freedom software interface. At the top, there's a navigation bar with icons for 'Publication', 'CYCLE DE VIE DE PUBLICATION DE DOCUMENTS AVEC FREEDOM', 'Liste des publications', 'Autres...', and 'Publie'. Below the navigation bar, there's a sidebar with a tree view showing 'Publication' expanded. In the main content area, the document title is 'Cycle de vie de publication de documents avec Freedom'. Below the title, there's a section titled 'But de ce document' with the following text: 'Expliquer comment mettre en place un cycle de vie de publication de documents classique du genre « Brouillon -> Validé » mais avec la possibilité de repasser à l'état brouillon tout en laissant la dernière version validée accessible aux utilisateurs.'.

Exemple de document à l'état brouillon en édition :

Publication
CYCLE DE VIE DE PUBLICATION DE DOCUMENTS AVEC
FREEDOM
| Sauver | Annuler

inchangé ▾

Titre : Cycle de vie de publication de documents avec Freedom

Contenu :

Cycle de vie de publication de documents

But de ce document

Rapport affichant la liste des documents publiés :

rapport
LISTE DES PUBLICATIONS

Éditer | Supprimer | Historique | Ajouter un post-it | Calc | Modifier | Ouvrir | Version imprimable | Autres ↗

Liste des publications

Titre	date de modification
Cycle de vie de publication de documents avec Freedom	24.05.2008 19:50:35
Test 02	24.05.2008 17:08:36
2	-

3.10 Créer une nouvelle application Freedom

3.10.1 But de ce document

Freedom Webdesk

- Mon application
- Coffre
- Accessibilités
- Carnet d'adresses
- Une famille
- Utilisateurs
- Gestion documentaire
- Agenda de groupe
- Demande d'achat
- Documentation électronique
- Plan d'action
- Espace de travail
- Mon application
- Mon portail
- Mon compte
- Administration

Montrer comment créer de nouvelles applications dans Freedom. Une application, permet de regrouper des familles ou des actions.

3.10.2 Arborescence des fichiers d'une application

Pour créer une nouvelle application, il faut au minimum deux fichiers :

/usr/share/what/MONAPPLI/MONAPPLI.app

et

/usr/share/what/MONAPPLI/MONAPPLI_init.php

Si l'application est visible, il faut ajouter l'icône de l'application (image carrée entre 48 et 68 px, format png si possible) :

/usr/share/what/MONAPPLI/Images/votre_image.png

Si l'application contient des actions, il faut ajouter les

fichiers .php et/ou .xml des différentes actions :

```
/usr/share/what/MONAPPLI/action1.php
/usr/share/what/MONAPPLI/Layout/action1.xml
```

3.10.3 MONAPPLI.app

Ce fichier PHP décrit l'application : éléments de présentation \$app_desc = array(...), actions de l'application \$action_desc = array(...) et les acl (droits) \$app_acl = array (...). Exemple de contenu minimum :

```
<?php

$app_desc = array (
    "name"          =>"MONAPPLI",
    "short_name"    =>"Mon application",
    "description"   =>"Mon application de test",
    "access_free"   =>"Y",
    "icon"          =>"cycle.gif",
    "displayable"   =>"Y",
    "with_frame"    =>"Y",
    "childof"       =>"ONEFAM"
);

?>
```

Pour un aperçu plus complet de ce fichier, rendez-vous à la page [actions particulières](#).

Variables	Descriptions
name	Nom de l'application tel qu'il apparaîtra dans les menus de configuration de Freedom. Ce nom doit également correspondre exactement au nom du dossier contenant l'application.
short_name	Nom de l'application qui apparaîtra dans le menu général pour les utilisateurs (si l'application est visible).
description	Nom qui apparaît en info-bulle en laissant la souris sur le nom de l'application dans le menu général (si l'application est visible).
access_free	Indique que les droits d'accès à l'application ne sont pas contrôlés
icon	Nom du fichier disponible dans « /usr/share/what/Images » ou dans « /usr/share/what/MONAPPLI/Images ». Remarque : Le script « /usr/share/what/wstart » créera automatiquement un lien de cette image dans le dossier « /usr/share/what/Images »
displayable	Si « Y », l'application apparaîtra dans le menu général pour les utilisateurs sinon, elle sera invisible. Remarque : Une application invisible peut être utilisée pour y stocker des actions. Par exemple l'application « GENERIC » est invisible mais dispose de nombreuses actions utilisées par l'application « ONEFAM »
with_frame	n'est plus utilisé mais obligatoire pour le fonctionnement des anciennes applications
childof	Indique si cette application hérite d'une autre. Il est possible de faire dériver une application de n'importe quelle autre application existante (ex : ONEFAM , WGCAL ,..). Cette technique est

principalement utilisée pour créer des applications basées sur ONEFAM pour regrouper les familles par fonctionnalités.

3.10.4 MONAPPLI_init.php

Ce fichier PHP contient les variables déclarées par l'application (globales, applicatives, utilisateurs). Il contient au moins la version de l'application :

```
<?php
global $app_const;
$app_const = array(
    "VERSION"=>"0.1-2"
);
?>
```

Ces variables sont accessibles par l'administrateur, via l'application Administration.

Remarque : Il n'est pas obligatoire d'augmenter le numéro de version pour que la mise à jour soit prise en compte avec la commande "-method=update". Par contre la commande "wcheck" vérifie que la version a changé pour effectuer la mise à jour.

Déclaration de paramètres applicatifs : Une ou plusieurs valeurs, de type string, integer, ... peuvent être nécessaires pour le fonctionnement d'une application, tout en restant modifiables par l'administrateur. Pour déclarer ces paramètres, il faut ajouter un tableau dans le code du fichier MONAPPLI_init.php.ini. Exemple :

```
<?php
global $app_const;
$app_const= array(
    "VERSION" => "@VERSION@-@RELEASE@",
    "Name_of_the_parameter"      =>array("val"=>"1",
                                            "descr"=>N_("Description of the
parameter"),
                                            "global"=>"Y",
                                            "user"=>"N")
);
?>
```

Variables	Descriptions
val	indique la valeur par défaut du paramètre. Cette valeur est modifiable par l'administrateur (Menu "Administration"/"Paramètres applicatifs"/ puis choisir l'application).
descr	permet de noter une description, ce texte étant affiché dans le champs du même menu que ci-dessus.
global	indique si le paramètre est global.
user	indique si la valeur du paramètre est personnalisable pour chaque utilisateur ayant accès à l'application : "Y" ou "N". En cas de "Y", le choix apparaîtra, pour l'administrateur, dans le menu "Administration"/"Paramètres utilisateurs"/ puis choisir l'utilisateur).

3.10.5 MONAPPLI_post

Ce script batch (facultatif) permet d'effectuer des opérations supplémentaires lors de l'initialisation, la mise à jour ou la suppression de l'application.

Attention : Ce script est lancé uniquement avec la commande "wcheck" et non pas la commande

“-method=update” et seulement si le numéro de version de l'application a changé.

Par exemple, le script ci-dessous permet d'importer les documents contenus dans le fichier “monappli.ods” lors de l'initialisation et la mise à jour de l'application.

```
#!/bin/bash

if [ "$dbpsql" == "" ]; then
    . /etc/freedom.conf
    wchoose -b
fi

#-----
#post installation
#-----
if [ "$1" = "I" ] ; then
    echo
fi

#-----
#post update
#-----
if [ "$1" = "U" ] ; then
    $wpub/wsh.php --api=freedom_import --file=$wpub/MONAPPLI/monappli.ods
fi

#-----
#post uninstallation
#-----
if [ "$1" = "D" ] ; then
    echo
fi
```

Remarque : Vous trouverez d'autres exemples dans les fichiers suivants :

- /usr/share/what/FREEEVENT/FREEEVENT_post
- /usr/share/what/FREEDOM/FREEDOM_post

3.10.6 Initialisation ou mise à jour de l'application

Pour initialiser l'application, on peut utiliser l'utilitaire wcheck. Si les fichiers .app et _init.php sont présents et corrects la nouvelle application doit être listée par cet utilitaire.

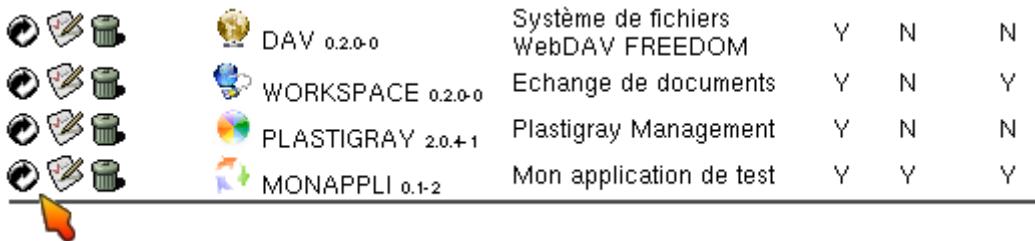
L'utilitaire wcheck ne fait les mises à jours d'application que si la version est supérieure à celle déjà enregistrée. Pour forcer une mise à jour, vous devez lancer la commande suivante :

```
/usr/share/what/wsh.php --api=appadmin --method=update --appname=MONAPPLI
```

Attention : Si vous utilisez un script “MONAPPLI_post”, il faut obligatoirement augmenter le numéro de la version de l'application et utiliser “wcheck” pour effectuer l'initialisation ou la mise à jour de l'application. Sinon ce script ne sera pas pris en compte.

Remarque : Une fois l'initialisation effectuée, il sera possible de faire les mises à jour avec cette même commande ou en passant par l'interface graphique avec ce menu :

- Menu “Administration / Les applications” et cliquer sur l'icône « mise à jour » en forme de flèche qui tourne.



	DAV 0.2.0-0	Système de fichiers WebDAV FREEDOM	Y	N	N
	WORKSPACE 0.2.0-0	Echange de documents	Y	N	Y
	PLASTIGRAY 2.0.4+1	Plastigray Management	Y	N	N
	MONAPPLI 0.1-2	Mon application de test	Y	Y	Y

3.10.7 Liens pour avoir d'autres informations

- Le chapitre [Actions particulières](#) vous donnera un exemple de création d'une application contenant des actions.
- L'application [Web Externe](#) (FREEDOM-URL) utilise ce principe pour intégrer des sites Web dans Freedom.

3.10.8 Actions particulières

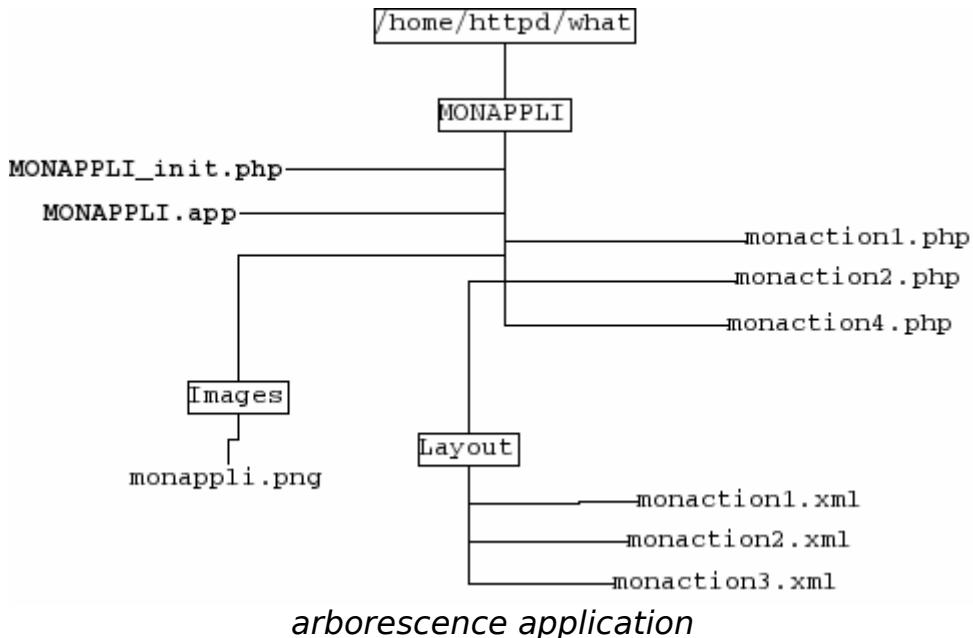
les actions particulières des documents font appel à des actions WHAT. Une action est le déroulement d'un code PHP et la production d'une sortie (en général HTML), l'action peut être vue comme le traitement d'une requête particulière :

3.10.8.1 Principe et spécification des actions particulières

3.10.8.2 Introduction

Les actions particulières des documents font appel à des actions WHAT. Ces actions doivent être définies au sein d'une application.

Pour créer une application, il faut créer une arborescence de fichier comme décrit ci-dessous :



La description de l'action se fait dans le fichier .app. Ce fichier contient la liste des actions ainsi que la liste des droits d'accès gérés par cette l'application.

Le fichier d'exemple MONAPP.app définit 3 droits d'accès et 4 actions. Une action est définie à

l'aide d'un fichier PHP et d'un layout XML. Le fichier PHP ou le layout est optionnel, mais il faut au moins un des deux. A moins que l'application soit déclarée libre d'accès, chaque action doit définir son droit d'accès (ACL). Si l'utilisateur n'a pas le droit défini, l'exécution de l'action sera interdite.

On définit généralement un fichier PHP par action. Ce fichier contient la fonction action dont le nom est le même que celui du fichier. Les fichiers PHP et XML sont calculés par la fonction d'interprétation du fichier .app. Le nom de l'action ne doit pas dépasser 30 caractères!

action	fichier PHP	fichier XML
MONACTION1	MONAPP/monaction1.php fonction : monaction1(&\$action)	MONAPP/Layout/monaction1.xml
MONACTION2	MONAPP/monaction2.php fonction : monaction2(&\$action)	MONAPP/Layout/monaction2.xml
MONACTION3	MONAPP/ monaction1 .php fonction : monaction3(&\$action)	MONAPP/Layout/monaction3.xml
MONACTION4	MONAPP/monaction4.php fonction : monaction4(&\$action)	MONAPP/Layout/ monaction2 .xml

La fonction d'action appelée a un seul paramètre qui est la référence à l'objet action. Cet objet contient la référence au layout (Action:::lay).

3.10.8.3 Fichier 'MONAPP.app'

```
<?php
$app_desc = array (
    // nom
    "name" => "MONAPP",
    // description court
    "short_name" => N_("mon application"),
    // description complète
    "description" => N_("mon application de test"),
    // "N" => L'application n'apparaîtra pas, par défaut, dans le
    // menu des utilisateurs.
    // Il faudra alors spécifier l'accès, au cas par cas, dans
    // le menu "Accessibilités".
    "access_free" => "N",
    // Icône
    "icon" => "monappli.png",
    // Doit être affiché dans le bandeau en haut (Y,N)
    "displayable" => "N",
    // "N" => les balises <html><head> seront rajoutées au
    // layout de l'application
    "with_frame" => "N",
    // héritage d'une application
    "childof" => ""
);

$app_acl = array (
    array(
        "name" =>"NORMAL",
        "description" =>N_("Access to common action"),
        // "Y" => le groupe "Utilisateurs" sera affecté à l'ACL
        "group_default"      =>"Y"),
    array(
        "name" =>"EDIT",
        "description" =>N_("Access to edit action"),
        "group_default"      =>"Y"),
    array(
        "name" =>"EXPORT",
        "description" =>N_("For export functions"),
        "group_default"      =>"N")
);
```

```

);
$action_desc = array (
  array(
    "name" =>"MONACTION1",
    "short_name" =>N_("action one"),
    "acl" =>"NORMAL",
    // action initiale lorsque l'utilisateur appuis sur
    // l'icone de l'application
    "root" =>"Y" ),
  array(
    "name" =>"MONACTION2",
    "short_name" =>N_("action two"),
    "acl" =>"EXPORT" ),
  array(
    "name" =>"MONACTION3",
    "short_name" =>N_("action three"),
    "acl" =>"EDIT",
    "script" =>"monaction1.php",
    "function" =>"monaction3" ),
  array(
    "name" =>"MONACTION4",
    "short_name" =>N_("action four"),
    "acl" =>"NORMAL",
    "layout" =>"monaction2.xml")
);
?>

```

Le fichier <NOMAPP>_init.php est obligatoire. Il contient la liste des paramètres propres à l'applicatif. Parmi ces paramètres, VERSION est obligatoire car il est utilisé pour les mises à jour éventuelles de l'application.

3.10.8.4 Fichier 'MONAPP_init.php'

```

global $app_const;
$app_const= array(
"VERSION" =>"0.0.1"
);

```

3.10.8.5 Fichier 'monaction1.php'

Le fichier *monaction1.php* doit être au nom de l'action, de préférence en minuscule. Il contient au moins une fonction du même nom que l'action, avec comme paramètre *&\$action* (passage par référence) :

```

function monaction1(&$action) {
//...
}

```

Cette fonction peut en appeler d'autres, définies elles aussi dans ce fichier php.

Récupérer des paramètres

Un paramètre applicatif :

```
$action->getParam("nom_du_parametre");
```

La base de données :

```
$db=getParam("FREEDOM_DB");
```

L'utilisateur courant :

```
$action->user ;
```

Pour avoir accès à l'applicatif (retourne un objet de la classe applicatif) :

```
$app=$action->parent
```

Récupérer des valeurs passées dans l'URL

L'action peut être lancée par l'activation d'un lien hypertexte. L'URL appelée peut contenir des paramètres, dont les valeurs sont récupérées par l'intermédiaire de la fonction getHttpVars :

```
$val = getHttpVars("url_val","val_par_defaut");
```

Afficher/Transmettre des données dans le layout

La valeur `val_to_be_sent` sera affichée à la place du code `[xml_data]` (du fichier xml) :

```
$action->lay->set("xml_data","val_to_be_sent");
```

Paramètre	Signification
<code>xml_data</code>	Nom du champ destinataire dans le fichier XML
<code>val_to_be_sent</code>	Valeur à afficher/transmettre

Warning/Astuces

- L'appel à `$this->` ne fonctionne pas,
- Pour manipuler un objet de classe documentaire, ne pas oublier :

```
include_once("FDL/Class.Doc.php");
```

- Sortir de la méthode :

```
$action->exitError("message");
```

3.10.8.6 Fichier monaction1.xml

Le layout de l'action peut faire référence au layout FDL. Pour cela, le contenu du fichier est inséré entre les 2 balises :

```
[ZONE FDL:HTMLHEAD]
[...]
```

[ZONE FDL:HTMLFOOT]

Dans la balise [ZONE FDL : HTMLHEAD], on peut rajouter

&title=mon titre

ce qui changera le titre de la page web.

Afficher une zone de texte composée d'un libellé et d'une zone de saisie :

```
<input type="text" name="texte_du_libelle"
       value="Texte_a_afficher_dans_le_libelle">
```

Afficher un bouton "Valider" :

```
<input type="submit" value="Valider">
```

3.10.8.7 Gestion des droits sur les actions

Certaines actions (par ex : exécuter MONACTION2) peuvent être réservées à un (ou plusieurs) groupe(s) d'utilisateurs. Pour limiter cet accès, vous pouvez l'indiquer par import de tableau dont une ligne respecte la mise en forme suivante : L'accès d'un groupe d'utilisateurs à une ACL peut être indiqué lors de l'importation des groupes utilisateurs. Pour cela, il faut ajouter une ligne dans le tableau d'import selon :

mot clé	identifiant du groupe	identifiant de l'application	nom de l'acl
ACCESS	Group_A	MONAPP	EXPORT

3.10.8.8 Initialisation de l'application et des actions

Pour initialiser l'application, on peut utiliser l'utilitaire wcheck. Si les fichiers .app et _init.php sont présents et corrects la nouvelle application doit être listée par cet utilitaire.

L'utilitaire wcheck ne fait les mises à jours d'application que si la version est supérieure à celle déjà enregistrée. Pour forcer une mise à jour, vous devez lancer la commande suivante :

```
# /usr/share/what/wsh.php --api=appadmin --method=update --appname=MONAPP
MONAPP...updateLOG:::(I)::Init : MONAPP
LOG:::(I)::Acl Modify : NORMAL, Access to common action
LOG:::(I)::Acl Modify : EDIT, Access to edit action
LOG:::(I)::Acl Modify : EXPORT, For export functions
LOG:::(I)::Update Action MONACTION1
LOG:::(I)::Update Action MONACTION2
LOG:::(I)::Update Action MONACTION3
LOG:::(I)::Update Action MONACTION4
```

3.10.8.9 Action dans les menus contextuels

L'appel à une action se fait à l'aide de l'URL

%S%app=<appname>&action=<actname>&...<autres paramètres>.

Par exemple, on ajoute l'attribut menu suivant dans la famille société :

Id	Description	Vis	lien
SI_MONMENU	Les sites	W	%B %app=MONAPP&action=MONACTION1&docid=%I%

L'action MONACTION1 revoit la liste des sites de la société. Les sites sont présentés avec la vue résumé. L'utilisateur peut cliquer sur le nom du site pour avoir le descriptif complet du site.

```
<?php
include_once("FDL/Lib.Dir.php");
function monaction1(&$action) {
    // les paramètres HTTP
    $docid = GetHttpVars("docid",0); // document société
    if ($docid == 0) $action->exitError("identifiant non spécifié");
    $dbaccess = $action->GetParam("FREEDOM_DB");
    $doc= new_Doc($dbaccess, $docid);
    $tdoc=getChildDoc($dbaccess,
        0,           // dans toute la base
        "0","ALL", // tous les résultats
        array("si_idsoc=$docid"), // filtre société
        $action->user->id, // utilisateur courant
        "TABLE",      // retourne liste de structure document
        "SITE");     // famid : société
    // remplissage des données pour le bloc
    $tsites=array();
    while (list($k, $v) = each($tdoc)) {
        $tsites[] = array("site"=>$v["title"],
            "idsite"=>$v["id"]);
    }
    // on renseigne le layout
    $action->lay->set("societe",$doc->title);
    $action->lay->setBlockData("SITES",$tsites);
}
?>
```

monaction1.xml

Exemple : liste des sites

```
Liste des sites de [societe]

<UL>
[BLOCK_SITES]
<LI><A href="[CORE_STANDURL]app=FDL&action=FDL_CARD&id=[idsite]" target="s[idsite]">
[site] </A>
<iframe name="s[idsite]" width="100%" src="[CORE_STANDURL]app=FDL&action=IMPCARD&id=[idsite]&zone=FDL:VIEWABSTRACTCARD:T"></iframe>
</LI>
[ENDBLOCK_SITES]
<UL>
```

Le résultat de cette action sur la société zoo net montre ses deux sites. La deuxième capture d'écran montre la représentation lorsque l'utilisateur a cliqué pour voir les vues complètes.

résultat de cette action sur la société zoo

vue complète

Cette action peut être aussi utilisée dans la famille site :

Id	Description	Vis	lien
SI_MONMENU	Les autres sites	W	%S %app=MONAPP&action=MONACTION1&docid=%SI_SOCID%

Ce type d'action peut aussi ce faire avec une vue. On préférera créer une action lorsqu'elle celle-ci n'a pas de rapport avec une famille de document précise et lorsqu'elle peut être réutilisée dans d'autre contexte.

La visibilité de l'attribut menu peut être contrôlée par une méthode de l'objet documentaire.

Id	Description	Vis	lien	phpfunc
SI_MONMENU	Les autres sites	W	%S %app=MONAPP&action=MONACTION1&docid=%SI_SOCID%	::controle_site()

Pour cela on indique la méthode utilisée dans la colonne phpfunc. Cette méthode doit retourner la visibilité voulue :

- MENU_ACTIVE : affiche l'action dans le menu
- MENU_INVISIBLE : n'affiche pas l'action dans le menu
- MENU_INACTIVE : affiche l'action mais n'est pas activable dans le menu

Ces trois valeurs sont des énumérés²⁰. Il faut donc retourner cette valeur sans guillemets.

```
/**
 * return true if it is the latest revision
 * @return enum
 */
public function controle_site() {
    if ($this->locked != -1) return MENU_ACTIVE;
```

20)constante PHP, pas des chaînes de caractères

```

        return MENU_INVISIBLE;
    }
}

```

3.11 Localisation (Traduction de freedom dans d'autres langues)

3.11.1 But de ce chapitre

Expliquer comment mettre en place la localisation dans **freedom**.

Pour développer dans de bonnes conditions il est indispensable d'utiliser les techniques énoncées dans ce document même si **freedom** ne sera utilisé qu'en Français pour plusieurs raisons :

- Il est déconseillé de mettre des accents dans les fichiers utilisés par **freedom** (.php, .xml, .app,...)
- Il n'est pas possible de mettre des accents aux noms des étapes des cycles de vie sans passer par ces techniques de localisation

3.11.2 Quel mécanisme utilise freedom pour la localisation ?

freedom utilise le système le plus répandu dans le monde des logiciels libres à savoir [Gettext](#)

3.11.3 Localiser son application

3.11.3.1 Localisation des familles

Les familles freedom sont décrites via les fichiers .ods. Les différentes informations issues de ces fichiers et présentées à l'utilisateur (libellé, énumérés, etc...) peuvent être traduites via la localisation. Aucune syntaxe particulière, ni mise en évidence des textes à traduire n'est nécessaire.

3.11.3.2 Localisation des fichiers .php (Methode, Class, Action, Api,...)

A chaque fois qu'une chaîne de caractères est susceptible d'être traduite ou contient des accents il est fortement conseillé d'utiliser la fonction "gettext()" ou sa forme simplifiée "_(())"

Exemple avec un extrait du fichier "Method.Report.php" :

```

function _getInternals() {
    return array("title" => _("doctitle"),
                "revdate" => _("revdate"),
                "revision" => _("revision"),
                "state" => _("state"));
}

```

Autre exemple avec la fonction "sprintf" :

```
$err=sprintf(_("Unable to connect to LDAP server %s"),$ldaphost);
```

En conclusion, il suffit de placer à l'intérieur de la fonction "_(())" chaque chaîne de caractères à localiser.

3.11.3.3 Localisation des fichiers .xml (Layout et Action)

Comme il n'est pas possible dans un fichier .xml d'utiliser une fonction PHP, il existe une autre

technique. Il faut placer la chaîne à traduire entre crochets et la faire précédé du mot clé réservé "TEXT". Exemple :

```
<h1>[TEXT:Title of document]</h1>
```

Remarque : Vous trouverez de nombreux exemples dans les fichiers .xml fourni par défaut dans Freedom.

3.11.3.4 Localisation des fichiers .app (Application)

Le contenu des fichiers .app est enregistré dans la base de données et ne doit pas être localisé.

Pour ces fichiers, il faut utiliser la fonction spécifique à Freedom "N_()". Cette fonction retourne la chaîne elle même sans la traduire. Cette fonction permet d'identifier les chaînes à traduire pour pouvoir utiliser les traductions dans les différentes interfaces d'administration. Exemple :

```
$app_desc = array (
    "name" => "ONEFAM", //Name
    "short_name" => N_("Onefam"), //Short name
    "description" => N_("One Family Management"), //long description
```

3.11.3.5 Localisation des cycles de vie

Pour localiser un cycle de vie, c'est un peu particulier car nous ne pouvons pas utiliser la fonction "_" directement dans les tableaux de définition des étapes et des transitions.

Il faut donc placer cette fonction en commentaire mais en utilisant uniquement le signe "#" et sans le placer en début de ligne car dans le cas contraire il ne serait pas analysé.

Voici un exemple pour les transitions :

```
var $transitions = array ( #_("T1") _("T2") _("T3")
    "T1" =>array(),
    "T2" =>array(),
    "T3" =>array());
```

Et un autre pour les étapes :

```
var $cycle = array( #_("Etape1") _("Etape2") _("Etape3")
    array("e1" => "Etape1",
        "e2" => "Etape2",
        "t" => "T1"),
    ...
```

3.11.3.6 Localisation des attributs des familles

Pour activer la localisation des libellés d'attribut, il est nécessaire d'indiquer dans le Makefile principal les fichiers ods à scruter. Ceci se fait en ajoutant les chemins des fichiers dans la variable TRANSODS.

```
# =====
# $Id: Makefile.in,v $
# =====
PACKAGE = @PACKAGE@
VERSION = @VERSION@
```

```
...
include $(utildir)/PubRule
TRANSODS += mesfamilles.ods
...
```

Ensuite cela produit des entrées dans les fichiers po comme ci-dessous :

```
#, fuzzy
msgid "ENTITE#title"
msgstr "Entité"

#, fuzzy
msgid "ENTITE#ae_type"
msgstr "Type"

#, fuzzy
msgid "ENTITE#ae_campagne"
msgstr "Campagne"

#, fuzzy
msgid "ENTITE#ae_type#yes"
msgstr "Type"

#, fuzzy
msgid "ENTITE#ae_type#no"
msgstr "Type"
```

l'identificateur est composé du nom de la famille, suivi de dièse puis de l'identifiant de l'attribut. Le titre de la famille est aussi traduisible avec la clef nom de la famille suivi de #title. Les libellés des valeurs d'énuméré sont aussi traduisibles. Elle se retrouve à l'aide du nom de la famille, du nom de l'attribut puis de la clef de l'énuméré.

3.11.3.7 Recherche des chaines à traduire (Génération fichier .po)

Pour rechercher les chaines à traduire dans les différents fichiers, il faut utiliser make po. Cela va mettre à jour les fichiers <APP>_fr.po et <APP>en.po.

3.11.3.8 Comment ça marche

Voici le code permettant de récupérer les données de tous nos fichiers et de générer un fichier .pot (Ce code est inclus dans votre fichier PubRule) :

```
xmlfiles=`find . -name "*.xml" -o -name "*.js"`
grep TEXT: $xmlfiles | sed -e's/\[TEXT:/\n\[TEXT:/g' | sed
-e's/\(\.*\)\TEXT:\([^\[]*\)\(\.*\)/N_\("\2"/g' > textxml.php

phpfiles=`find . -name "*.app" -o -name "*.php"`
xgettext --no-location $phpfiles --keyword='_'
--keyword='text'
--keyword='Text' --language=c -o $appname.pot
```

Ensuite, il faut à partir de ce fichier .pot générer un fichier .mo pour chaque langue à définir. Voici un extrait du code :

```
for lang in $LANGS; do
destpo=$appname"_"$lang.po"
msgmerge -v --force-po $destpo $appname.pot > $destpo.new;
mv $destpo.new $destpo;
msgfmt $destpo -o $wpub/locale/$lang/LC_MESSAGES/$appname.mo
done
```

3.11.4 Traduire un module Freedom dans une autre langue

Pour traduire un module Freedom dans une autre langue, on fera un module qui ne fournira que le fichier

Exemple pour la traduction de `freedom-core' en espagnol :

On initialisera un module `freedom-core-langpack-es'

```
$ mkdir freedom-core-langpack-es
$ cd freedom-core-langpack-es

$ vi VERSION
1.0.0

$ vi RELEASE
1

$ vi info.xml.in
<?xml version="1.0"?>
<module name="freedom-core-langpack-es" version="@VERSION@"
release="@RELEASE@" basecomponent="no">

    <description lang="en">freedom-core spanish (es_ES) locale</description>

    <requires>
        <module name="freedom-core" />
    </requires>

    <post-install>
        <process command="programs/update_catalog"><label lang="en">Generate
traduction catalog</label></process>
    </post-install>

    <post-upgrade>
        <process command="programs/update_catalog"><label lang="en">Generate
traduction catalog</label></process>
    </post-upgrade>

    <changelog>
        <version number="1.0.1-1" date="2010-05-17">
            <change title="Initial release" />
        </version>
    </changelog>

</module>

$ vi Makefile
PACKAGE = @PACKAGE@
VERSION = @VERSION@
utildir=@PUBRULE@
appname = @APPNAME@
```

```
pubdir = @prefix@
srcdir = @srcdir@
rootprefix=$(RPM_BUILD_ROOT)

export pubdir utildir appname pidir

TAR = gtar
GZIP_ENV = --best

export targetdir
export utildir

pages_not_xml = info.xml
pages_not_php = fam2po.php po2js.php lang_es.php

LANGS=es

include $(utildir)/PubRule

$ vi configure.in
AC_PREREQ(2.13)
AC_INIT("./Makefile.in")
AC_SUBST(VERSION)
VERSION=`cat VERSION`
AC_SUBST(RELEASE)
RELEASE=`cat RELEASE`
AC_SUBST(PACKAGE)
PACKAGE=freedom-core-langpack-es
AC_SUBST(APPNAME)
APPNAME=WHAT

ac_default_prefix=/usr/share/what
AC_SUBST(PUBRULE)
PUBRULE=
AC_ARG_WITH(pubrule, [ --with-pubrule=dir      Path to PubRule],
PUBRULE=$withval)
if test "x$PUBRULE" != "x"; then
  PUBRULEDIR=$PUBRULE
else
  if test "x$PUBRULEDIR" == "x"; then
    AC_CHECK_FILE($HOME/anakeen/devtools/PubRule,
PUBRULEDIR=$HOME/anakeen/devtools/)
    if test "x$PUBRULEDIR" = "x"; then
      PUBRULEDIR=.
    fi
  fi
fi
AC_CHECK_FILE($PUBRULEDIR/PubRule, PUBRULE=$PUBRULEDIR)
if test "x$PUBRULE" = "x"; then
  AC_MSG_ERROR([Could not find PubRule])
fi
AC_MSG_NOTICE([PubRule located at $PUBRULE])

AC_OUTPUT(Makefile info.xml)

$ vi WHAT_es.po
# SOME DESCRIPTIVE TITLE.
# Copyright (C) YEAR Free Software Foundation, Inc.
# FIRST AUTHOR <EMAIL@ADDRESS>, YEAR.
```

```
#  
msgid ""  
msgstr ""  
"Project-Id-Version: TEST locale\n"  
"Report-Msgid-Bugs-To: \n"  
"POT-Creation-Date: 2010-03-12 16:27+0100\n"  
"PO-Revision-Date: 2010-03-12 16:55+0100\n"  
"Last-Translator: Eric Martin <eric.martin@somewhere.com>\n"  
"Language-Team: spanish-anakeen <distrib@anakeen.com>\n"  
"MIME-Version: 1.0\n"  
"Content-Type: text/plain; charset=UTF-8\n"  
"Content-Transfer-Encoding: 8-bit\n"  
\n  
msgid "userlogin  
msgstr "Inicio"  
  
msgid "username"  
msgstr "Nombre"  
  
msgid "permissions"  
msgstr "derecho"  
  
msgid "none"  
msgstr "Ninguno"  
  
msgid "user"  
msgstr "usuario"  
  
msgid "object"  
msgstr "objeto"
```

[...]

```
$ vi lang_es.php  
<?php  
  
$lang["es_ES"] = array(  
    "label" => "Español",  
    "flag" => "",  
    "locale" => "es",  
    "dateFormat" => "%m/%d/%Y",  
    "dateTimeFormat" => "%m/%d/%Y %H:%M",  
    "timeFormat" => "%H:%M:%S",  
);  
  
?>
```

3.11.5 Outils pour lire les fichiers .po et traduire les chaînes

Une fois le fichier .po généré, il faut traduire les chaînes de caractère dans les différentes langues.

Vous pouvez le faire avec un simple éditeur de texte, mais il est fortement recommandé d'utiliser un outil spécialisé :

- kbabel (KDE)

- gtranslator (Gnome)
- PoEdit (multiplateforme)
- emacs (geek 😊)

3.11.6 Génération des fichiers .mo (binaires)

La commande 'msgfmt' permet de compiler les fichiers .po en fichier binaire .mo :

```
msgfmt $destpo -o $wpub/locale/$lang/LC_MESSAGES/$appname.mo
```

3.11.7 Initialisation de freedom et d'Apache pour prendre en compte les localisations

Pour finir, il faut assembler les fichiers .mo des différentes applications et les rendre disponible pour Apache :

```
whattext  
etc/init.d/apache2 reload
```

3.11.8 Script pour automatiser les tâches

Si vous avez suivis la formation développeur proposé par Anakeen, vous aurez appris à créer vos propres paquets .rpm ou .deb pour fabriquer vos applications basées sur **freedom**. Et la partie localisation est incluse dans la génération du paquet.

Le but de ce document n'est pas de vous expliquer comment fabriquer un paquet mais simplement vous permettre de localiser vos applications.

FIXME internalmedia: freedom_2.14:devel:make_app.sh.txt reprend le code évoqué précédemment et y apporte quelques améliorations pour vous permettre de localiser une application.

Pour utiliser ce script, il faut le placer dans "/usr/share/what" et l'exécuter en donnant comme argument le nom de l'application traiter :

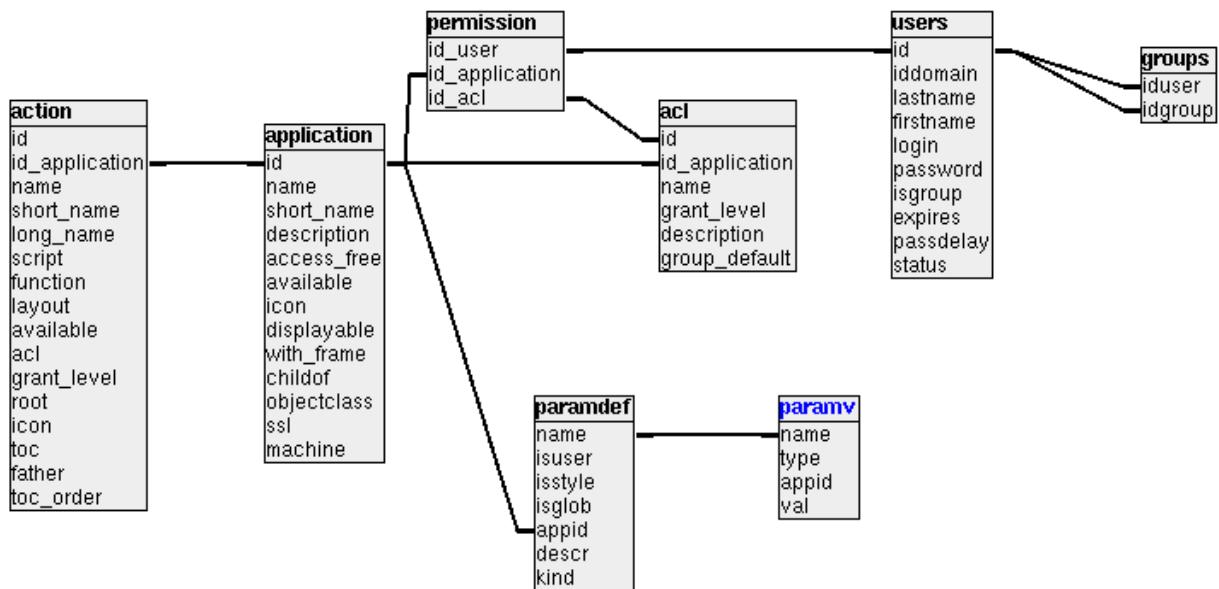
```
/usr/share/what/make_app.sh MONAPPLI
```

Voici les fonctionnalités de ce script :

- Création des liens vers les Method, Class et Images
- Génération des fichiers .mo
- Compilation des .mo en .po
- Intégration des traductions dans Apache avec whattext

3.12 Bases de données PostgreSQL

3.12.1 Tables de la base anakeen



Les tables de la base 'anakeen' sont utilisées par le cœur de Freedom et en particulier tout ce qui concerne l'authentification des utilisateurs.

3.12.1.1 acl

Contenu des acl des fichiers .app :

- id : id de l'acl
- id_application : id de l'application (lien avec la table 'application')
- name : nom de l'acl tel qu'il a été indiqué dans le fichier .app
- grant_level :
- description : Description de l'application indiqué dans le fichier .app
- group_default : Si=Y, le groupe utilisateur sera affecté à l'ACL.

3.12.1.2 action

Liste des actions (Contenu des actions des fichiers .app)

- id : id de l'action
- id_application : id de l'application (Lien avec la table 'application')
- name : Nom de l'action indiqué dans le fichier .app
- short_name : Nom court de l'action indiqué dans le fichier .app
- long_name : Nom long
- script : Fichier .php contenant l'action
- function :
- layout : Fichier .xml (layout) de l'action
- available : Y = l'action est disponible
- acl : acl permettant d'utiliser l'action

- grant_level :
- root :
- icon :
- toc :
- father :
- toc_order :

3.12.1.3 application

Liste des applications (Contenu des fichiers .app)

- id : id de l'application
- name : nom
- short_name : nom court
- description : description
- access_free : Accès libre (pas d'acl)
- available : Y = Application disponible
- icon : Icône de l'application
- displayable : Y = Application visible dans le menu général par les utilisateurs. Certaines applications sont invisibles et servent seulement à fournir des actions à d'autres applications (ex : GENERIC)
- with_frame : N'est plus utilisé
- childof : Indique de quel parent hérite l'application. Il est par exemple courant de créer une application basée sur ONEFAM.
- objectclass :
- ssl :
- machine : Peut-être utilisé pour répartir des applications sur plusieurs serveurs
- iorder :

3.12.1.4 docfrom

3.12.1.5 docname

3.12.1.6 domain

3.12.1.7 groups

Liens entre les id des utilisateurs et les id des groupes. Cette table est répliquée dans la base Freedom à chaque modification. Si la table n'est pas synchronisé cela peu poser des problèmes.

- iduser : id de l'utilisateur (lien avec un utilisateur de la table 'users')
- idgroup : id du groupe (lien avec un groupe de la table 'users')

3.12.1.8 mailaccount

3.12.1.9 paramdef

Table de définition des paramètres

- name : Nom du paramètre
- isuser : Y = Paramètre utilisateur
- issstyle : Y = Paramètre utilisé dans les styles
- isglob : Y = Paramètre global
- appid :
- descr : Description du paramètre
- kind : Format du paramètre

3.12.1.10 paramv

Table contenant les valeurs des paramètres.

- name :
- type : A=Applicatif, G=Global, U1 = Paramètre de l'utilisateur N°1
- appid :
- val : Valeur du paramètre

Remarque : Tous ces paramètres sont accessibles dans les Layout.

3.12.1.11 permission

Permissions sur les actions

- id_user : id de l'utilisateur
- id_application : id de l'application
- id_acl : Valeur de de l'acl sous forme binaire

3.12.1.12 session_conf

3.12.1.13 sessions

Sessions des personnes connecté dans la journée.

- id : Cookie enregistré
- userid : id de l'utilisateur

3.12.1.14 style

Liste des thèmes installés.

- name : Nom du thème
- Description du thème

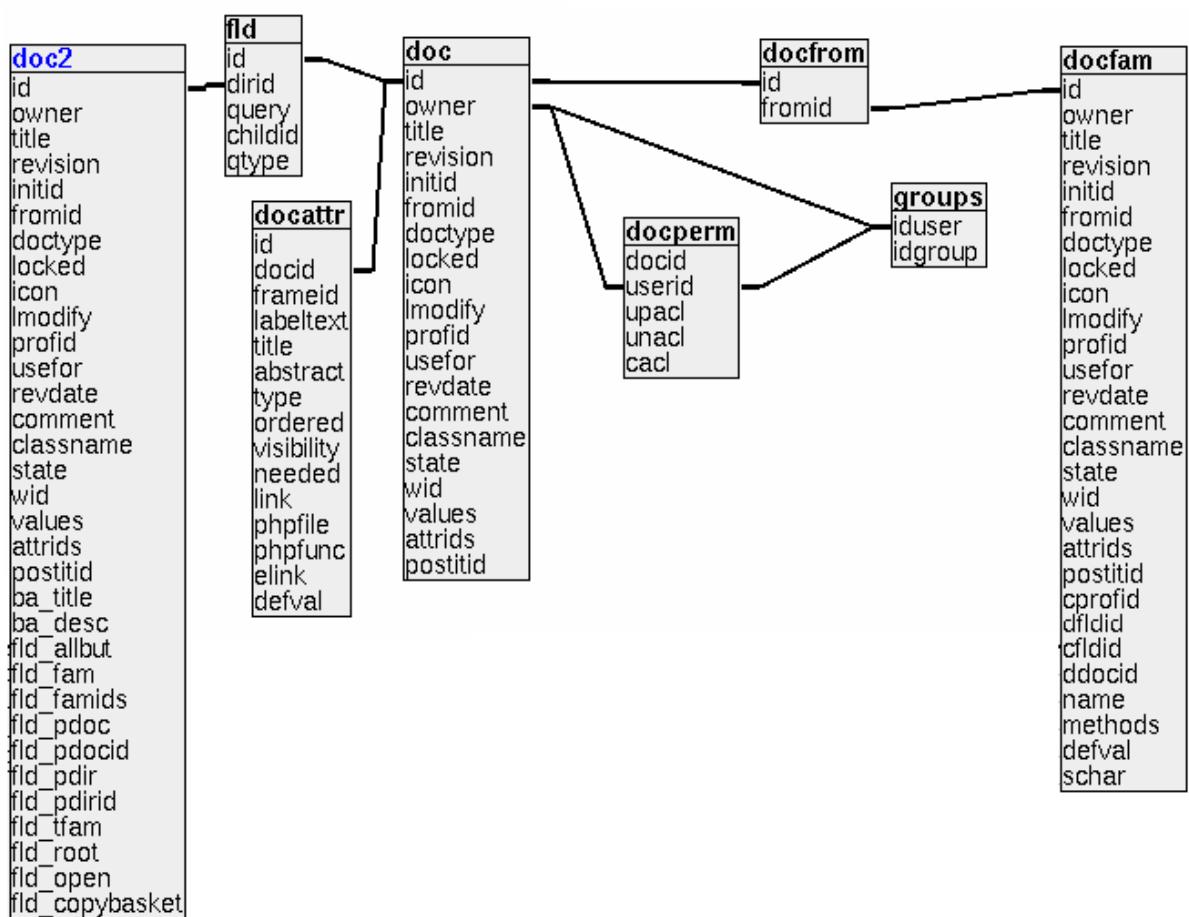
3.12.1.15 users

Utilisateurs et groupes. Table servant à l'authentification.

- id : id de l'utilisateur ou du groupe
- iddomain :
- lastname : Nom de l'utilisateur ou du groupe
- firstname : prénom
- login : login

- password : mot de passe
- isgroup : Y = C'est un groupe
- expires :
- passdelay :
- status :
- mail : mail de l'utilisateur
- ntpasswordhash :
- Impasswordhash :
- fid : lien vers l'id du document Freedom

3.12.2 Tables de la base freedom



3.12.2.1 Héritage entre les tables

Freedom utilise les possibilités d'héritage entre les tables de Postgres. Exemples :

- La table « doc », contient (virtuellement) tous les documents créés dans Freedom. Cette table contient toutes les propriétés par défaut d'un document (Profil, Cycle,...)
- La table « doc1 » contient tous les documents de la famille « de base ». Elle hérite de la table « doc ».
- La table « doc2 » contient tous les dossiers. Elle hérite de la famille « de base », donc de la table « doc1 » comme la plupart des familles standards de Freedom

Si une famille A hérite d'une famille B, chaque enregistrement sera disponible dans 3 tables :

- doc,
- docA
- docB

La suppression ou la modification d'un enregistrement dans l'une des tables le supprimera dans les 3.

Remarque : pgAdmin permet de consulter ce mécanisme de dépendance entre les tables.

3.12.2.2 doc

Cette table ne contient physiquement aucune donnée mais contient virtuellement grâce au mécanisme d'héritage des tables de Postgresql toutes les propriétés de tous les documents Freedom.

- id, owner, title, revision, initid, fromid, doctype, locked, allocated, icon, Imodify, profid, usefor, revaldate, version, cdate, adate, comment, classname, state, wid, values, attrids, postitid, cvid, name, dprofid, prelid, atags, confidential, ldapdn, svalues, fulltext, forumid

3.12.2.3 doc<idfam>

Chaque famille de Freedom est stockée dans une table séparée 'doc<idfam>' où '<idfam>' correspond à l'id de la famille.

De plus, si une famille hérite d'une autre famille, le mécanisme d'héritage des tables de Postgresql sera utilisé.

3.12.2.4 docattr

Cette table contient la définition des attributs de chaque famille. Cette table est lue uniquement par par fdl_adoc.

3.12.2.5 docattrldap

Cette table permet de faire le lien entre les attributs d'une famille Freedom et les attributs d'un annuaire ldap. Cette table est utilisée en particulier pour le fonctionnement de la famille 'Utilisateur réseau'

3.12.2.6 docfam

Cette table contient la liste des familles.

3.12.2.7 docfrom

Cette table permet de retrouver à quelle famille (fromid) appartient un document (id). Cette table est construite construite automatiquement avec des triggers pour améliorer les performances des requêtes.

- id
- fromid

3.12.2.8 dochisto

Cette table contient l'historique des modifications de tous les documents. Cette table est utilisée en particulier pour afficher le contenu du lien 'historique' d'un document Freedom.

3.12.2.9 docname

Cette table permet de faire le lien entre le nom logique d'un document et son id.

- name : Nom logique du document
- id : id du document
- fromid :

En cas de problème avec cette table, Freedom ne fonctionnera plus correctement mais il est possible de la reconstruire complètement en cas de besoins.

3.12.2.10 docperm

Table des permissions.

Remarques :

- Les points verts sont visibles dans la table.
- Les points gris sont calculés.
- Pour chaque point jaune un group virtuel est créé dans la table vgroup.
- Le userid 100000 est un utilisateur virtuel utilisé pour calculer les droits sur les profils dynamique.
- Le droit est calculé par la fonction postgresql « getuperm » qui indique si l'utilisateur à le droit de voir le document. Si le droit n'est pas encore calculé, il va le faire et remplir les boules grises. Les boules grises sont calculés par « getuperm » au moment de l'accès au document et ajouté dans la table docperm. Une fois la boule grise ajouté, elle n'est plus recalculée. La réinitialisation est faite également à chaque changement de groupe.

3.12.2.11 docread

Cette table gérée automatiquement par les triggers de Postgresql contient le contenu complet de tous les documents de Freedom (svalue) y compris le contenu des pièces jointes si le moteur de transformation est utilisé (fulltext).

- fulltext : Recherche avec * dans ONEFAM
- svalues : Recherche avec ~ dans ONEFAM

Son but est de pouvoir effectuer rapidement des recherches plein texte sur le contenu de tous les documents. Elle est utilisée dans les recherches ou aucune famille n'est précisée. Si une famille est précisée, la recherche sera effectuée dans la table de la famille.

3.12.2.12 docrel

Table alimentée par des triggers utilisée pour connaître les relations entre les documents. Cette table est utilisée en particulier par le menu 'Autre / Relations du document' des documents de Freedom.

- sinitid : Source
- cinitid : Cible
- type : folder = Relation de type folder

3.12.2.13 docutag

Table contenant les tags des documents par utilisateur.

Sur chaque document, il est possible d'appliquer un ou plusieurs tags.

Exemples :

- VIEWED : tag ajouté automatiquement sur tous les documents. Il permet de connaître la date, l'heure et la dernière personne ayant consulté le document.
- TOVIEW : Tag indiquant que la personne doit voir le document. Utilisé par Workspace (Mes documents à voir)

SetUTag : Méthode Permettant d'ajouter un TAG.

Cela peut-être utilisé pour mémoriser des variables sur le document utilisateur.

3.12.2.14 docvaultindex

Table de correspondance entre les id des documents et les numéros des fichiers. Cette table permet uniquement de rechercher les orphelins.

- docid
- vaultid

3.12.2.15 fld

Table permettant de retrouver les documents par dossier (Relations de type folder). Table faisant un peu redondance avec la table docrel mais uniquement pour les dossiers. Cette table contient également les recherches stockés

3.12.2.16 groups

Lien entre les id des utilisateurs et les id des groupes

Cette table est normalement la réplique exacte de la table 'groups' de la base 'anakeen'

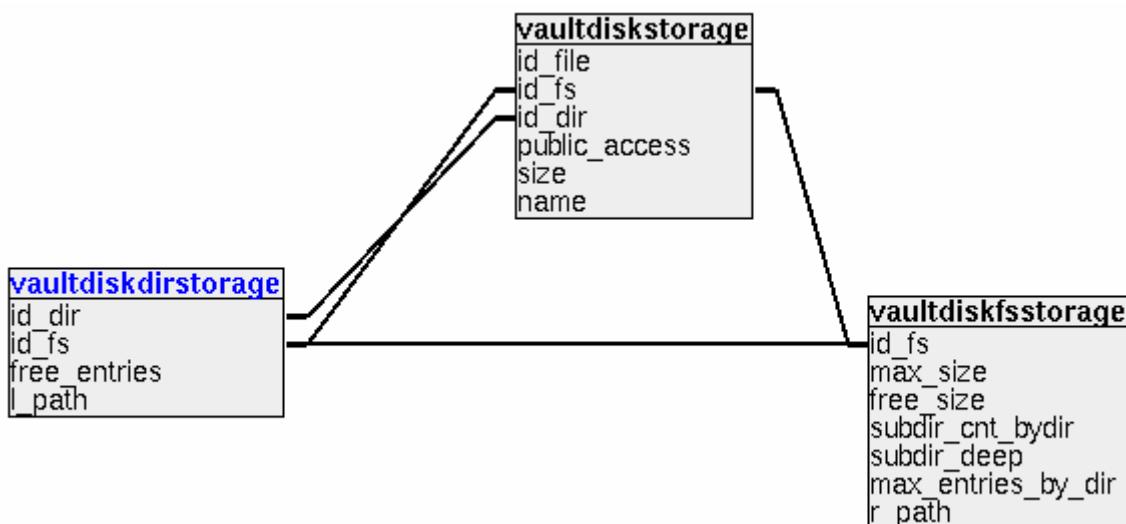
Elle existe pour des raisons historique

3.12.2.17 pg_ts_cfg, pg_ts_cfgmap, pg_ts_dict et pg_ts_parser

Tables internes à tsearch de Postgresql utilisées pour la recherche plein texte

3.12.2.18 vgroup

3.12.3 Tables du vault (base freedom)



3.12.3.1 vaultdiskdirstorage

Sous-dossier de chaque Vault

3.12.3.2 vaultdiskfsstorage

Liste des coffres

3.12.3.3 vaultdiskstorage

Table permettant de retrouver un fichier dans le Vault à partir de son numéro.

- id_file : id du fichier dans le Vault
- id_fs :
- id_dir :
- public_access :
- size : Taille du fichier
- name : Nom réel du fichier
- mime_t : Type mime (ex : Microsoft Office Document)
- mime_s : Type mime (ex : application/vnd.ms-excel)
- cdate :
- mdate :
- adate :
- teng_state : Utilisé par le moteur de transformation
- teng_lname : Utilisé par le moteur de transformation
- teng_id_file : Utilisé par le moteur de transformation
- teng_comment : Utilisé par le moteur de transformation

4 Exploitation

4.1 dbaccess.php

4.1.1 Fichier "dbaccess.php"

```

$pgservice_core="ged";
$pgservice_freedom="ged";
$freedom_context="default";
$dbpsql=$pgservice_core;

$freedom_authtype = 'html';

$freedom_authtypeparams = array(
    'html' => array (
        'cookie' => 'freedom_auth',
        'authurl' => 'guest.php?sole=A&app=AUTHENT&action=LOGINFORM',
        'username' => 'auth_user',
        'password' => 'auth_pass',
    ),
    'open' => array(),
    'basic' => array(
        'realm' => 'freedom',
    ),
);
$freedom_authprovider = 'freedom';

$freedom_providers = array(
    'freedom' => array(
        'connection' => 'service='.$pgservice_core,
    ),
    'file' => array(
        'authfile' => '/var/www/ged/.freedompwd',
    ),
);

```

\$freedom_authtype	Le mode de l'interface utilisateur d'authentification. Le paramètre peut prendre les valaurs : apache, basic, html ou open.
\$freedom_authprovider	Ce paramètre spécifie le, ou les, backend d'authentification à utiliser pour valider les mot de passes. Par défaut Freedom est livré avec les backend suivants : freedom, ldap et file.

4.1.1.1 freedom_authtype

Ce paramètre permet de spécifier quelle interface d'authentification est utilisé pour demander les informations de connexion à l'utilisateur (login/password la plupart du temps) et lui transmettre le résultat de l'authentification.

apache

Ce mode spécifie que toute la mécanique d'authentification est déléguée à Apache en mode HTTP Basic. Freedom ne s'occupe pas d'authentifier les utilisateurs, et Apache lui fournit les utilisateurs qui viennent de se connecter via la variable PHP `$_SERVER['PHP_AUTH_USER']`.

Dans ce mode, le paramètre `$freedom_authprovider` n'est pas utilisé puisque c'est Apache qui gère toute l'authentification.

basic

Le mode basic permet à Freedom de fournir un mécanisme d'authentification HTTP Basic.

Dans ce mode, Freedom gère l'authentification au format HTTP Basic et l'utilisateur rentre son login et son mot de passe dans la boîte de dialogue affiché par le navigateur.

Paramètres :

realm	Le realm de l'authentification HTTP Basic.
-------	--

html

Le mode html présente un formulaire HTML pour demander le login et le mot de passe de l'utilisateur. Une fois le login et le mot de passe validé auprès du `freedom_authprovider`, une session par cookie est ouverte afin de valider les accès suivants.

Paramètres :

cookie	Le nom du cookie contenant la clef associé à la session de l'utilisateur.
authurl	L'URI d'accès au formulaire de login.
username	L'id du champ input HTML contenant le nom d'utilisateur.
password	L'id du champ input HTML contenant le mot de passe de l'utilisateur.

open

Le mode open ne présente aucune interface pour la saisie du login et du mot de passe, mais se base sur un jeton présent dans l'URI.

Dans ce mode, le paramètre `$freedom_authprovider` n'est pas utilisé.

Le jeton est contenu dans la variable `privateid`.

Exemple d'URI avec authentification `open` et jeton :

<http://<server>/freedom/index.php?authtype=open&privateid=e2bb65612c70e7ac78d5ccbfe12aa234&app=FOO&action=BAR>

L'action `BAR` de l'application `FOO` doit être déclarée `openaccess=Y` pour pouvoir être accessible avec cette authentification par jetons.

Le jeton doit être présent dans la table `usertoken` qui stocke les jetons, avec l'utilisateur associé au jeton et une date d'expiration du jeton.

Voir [Class.UserToken.php](#).

4.1.1.2 freedom_authprovider

Le paramètre `$freedom_authprovider` permet de spécifier le sous-mécanisme utilisé pour la

validation du login et du mot de passe.

Ces providers prennent en entrée un login et un mot de passe, et retourne s'ils sont valide ou non.

Chaque provider contient un ensemble de paramètres de configuration qui lui sont propre.

freedom

Le provider `freedom` est le mécanisme historique de Freedom qui utilise la base Postgres et la table user pour la validation des login et mots de passe.

On parlera dans ce cas d'authentification « interne », puisqu'elle ne se base sur aucun élément externe à Freedom.

Paramètres :

connection	Contient la chaîne de connexion à la base de donnée Postgresql contenant la table user. Exemple : service=freedom
------------	--

ldap

Le provider `ldap` permet de valider le login et le mot de passe auprès d'un serveur LDAP, ou d'un serveur Active Directory, en effectuant un « bind LDAP » auprès de celui-ci.

Ce provider supporte une création basique d'utilisateurs « à la volé ». Dans ce cas, si l'authentification du est validé auprès du LDAP mais qu'il n'y a pas de compte utilisateur Freedom correspondant à ce login, alors le provider peut créer un compte utilisateur Freedom (IUSER) avec ce login. Cela permet de pouvoir « peupler » dynamiquement la base des comptes utilisateurs Freedom au fur et à mesure que les personnes se connectent. Le paramètre `dGroup` permet de spécifier le nom logique, ou l'id, du groupe Freedom (IGROUP) dans lequel seront créé ces nouveaux utilisateurs Freedom.

Paramètres :

host	Le nom, ou l'adresse IP, du serveur LDAP.
port	Le port du serveur LDAP.
ssl	y pour utiliser une connexion SSL au serveur LDAP, n pour ne pas utiliser de connexion SSL.
options	Options de la librairie OpenLDAP pour la connexion au serveur LDAP.
dn	 Le DN de connexion utilisé pour valider le mot de passe de l'utilisateur. La chaîne %s est remplacé par le login, et le mot de passe fourni par l'utilisateur est utilisé pour effectuer un bind avec le DN ainsi construit.
dGroup	Le nom logique du groupe Freedom sous lequel seront créés les utilisateurs. Voir
allowAutoFreedomUserCreation	yes pour activer la création des comptes utilisateurs par le provider, no dans le cas contraire (par défaut no).

file

Le provider file permet de valider un couple de login/mot de passe auprès d'un fichier à plat au format htpasswd Basic d'Apache.

Ce provider ne supporte pas la création d'utilisateurs « à la volé ».

Paramètres :

authfile	Le chemin d'accès au fichier à plat de mot de passe au format htpasswd.
----------	---

freedomNu

Le provider freedomNu est fourni par l'application `freedom-networkuser', et permet de valider le login et le mot de passe auprès d'un serveur LDAP ou d'un serveur Active Directory.

Ce provider supporte la création d'utilisateurs « à la volé ». Dans ce cas, si l'authentification est validé auprès du LDAP mais qu'il n'y a pas de compte utilisateur Freedom correspondant à ce login, alors le provider peut créer un compte utilisateur Freedom (LDAPUSER) avec ce login. Cela permet de pouvoir « peupler » dynamiquement la base des comptes utilisateurs Freedom au fur et à mesure que les personnes se connectent.

Paramètres :
Voir [documentation](#) [paramètres](#) de [networkuser](#)

Les paramètres de connexion au LDAP/AD sont pris dans les paramètres applicatifs de l'application Freedom `NU' (voir [documentation configuration Freedom de networkuser](#)).

4.2 Développer un provider

Un provider doit implémenter les méthodes définies dans la classe `WHAT/Class.Provider.php`

Les méthodes à implémenter obligatoirement sont :

- `abstract function validateCredential($username, $password)`
 - `abstract function validateAuthorization($opt)`
- Les méthodes optionnelles sont :
- `public function __construct($authprovider, $parms)`
 - `public function initializeUser($username)`

4.2.1 validateCredential()

Cette méthode prend en entrée deux arguments qui sont :

- `$username` : le login entré par l'utilisateur
- `$password` : le mot de passe entré par l'utilisateur.

La méthode doit retourner :

- `true` si le couple login/mot de passe est correct
- `false` si le couple login/mot de passe est incorrect.

4.2.2 validateAuthorization()

Une fois le couple login/password est validé, cette méthode permet de contrôler si l'utilisateur est autorisé à se connecter.

Cette méthode prend entrée un argument :

- \$opt : une structure contenant le nom de l'utilisateur.

```
$opt = array(
    'username' => $username
);
```

La méthode retourne :

- true si l'utilisateur est autorisé à se connecter
- false dans le cas contraire.

4.2.3 `__construct()`

C'est le constructeur du Provider que l'on peut étendre si celui-ci nécessite une initialisation particulière.

Cette méthode prend en entrée deux arguments qui sont :

- \$authprovider :
- \$parms :

4.2.4 `initializeUser()`

Si le compte de l'utilisateur n'existe pas dans Freedom, cette méthode est utilisé pour créer le compte de l'utilisateur dans Freedom.

Cette méthode prend en entrée le login de l'utilisateur :

- \$username : le login de l'utilisateur.

La méthode retourne :

- "" : une chaîne vide s'il n'y a pas eu d'erreur à la création du compte.
- "Error message ..." : une chaîne non vide contenant le message d'erreur rencontré.

Cette méthode spécifique implémente la recherche des informations de l'utilisateur à partir du login sur le système d'authentification utilisé, et la création du compte utilisateur Freedom avec les informations obtenus.

4.2.5 Exemple

Dans l'exemple ci-dessous, nous allons écrire un provider pour valider les mots de passe des utilisateurs auprès d'un service PAM à l'aide de la commande [checkpassword-pam](#). Ce module ne supportera pas la création d'utilisateurs à la volée.

4.2.5.1 Fichier `context/default/dbaccess.php'

On va déclarer dans le `dbaccess.php' que l'on utilise en premier notre provider `pam', et ensuite le provider `freedom' si l'utilisateur n'est pas reconnu par `pam'.

Notre provider `pam' aura un paramètre nommé "service" qui contiendra le nom du service PAM auprès duquel seront validé les login et mot de passe (c'est le nom du fichier situé dans `/etc/pam.d').

```
[...]
$freedom_authprovider = 'pam,freedom';
```

```
$freedom_providers = array(
    [...]
    'pam' => array(
        'service' => 'freedom'
    )
);
[...]
```

4.2.5.2 Fichier `WHAT/providers/Class.pamProvider.php`

Notre provider est donc décrit dans le fichier `Class.pamProvider.php` situé dans le sous-répertoire `WHAT/providers` de l'installation Freedom.

Celui-ci fournit une classe `pamProvider` qui étend la classe `Provider`, et nous allons décrire nos méthodes spécifiques `validateCredential()` et `validateAuthorization()`.

Dans `validateCredential()` nous allons récupérer le paramètre `service` de notre provider, et utiliser la commande `checkpassword-pam` pour valider le username et le password reçu.

Pour simplifier l'exemple, la méthode `validateAuthorization()` retournera `true` systématiquement (on supposera que l'autorisation du compte est établi dans la phase de validation du mot de passe).

```
<?php
include_once('WHAT/Class.Provider.php');

Class pamProvider extends Provider {

    public function validateCredential($username, $password) {
        $service = 'freedom';
        if( array_key_exists('service', $this->parms) ) {
            $service = $this->parms['service'];
        }

        return $this->checkpassword_pam($username, $password, $service);
    }

    public function validateAuthorization($opt) {
        $username = $opt{'username'};

        return true;
    }

    public function checkpassword_pam($username, $password, $service) {
        $cmd = sprintf("checkpassword-pam --service %s --no-chdir-home --noenv",
        escapeshellarg($service)
        );

        $proc = proc_open($cmd, array(3=>array('pipe', 'r')), $pipes);
        if( ! is_resource($proc) ) {
            error_log(__CLASS__."::__FUNCTION__".
            sprintf("Error running checkpassword-pam"))
```

```

);
    return false;
}

fwrite($pipes[3], sprintf("%s\0%s\0timestamp\0",
    $username,
    $password
));
fclose($pipes[3]);

$ret = proc_close($proc);
if( $ret === 0 ) {
    return true;
}

error_log(__CLASS__."::".__FUNCTION__." ".
sprintf("Authentication failed for user '%s' and service '%s'.",
    $username,
    $service
)
);
return false;
}

?>

```

4.2.5.3 Fichier `/etc/pam.d/freedom'

Pour finir, le fichier associé au paramètre "service" de notre provider. Celui-ci contiendra les règles que l'on souhaite voir appliquée pour la validation des comptes. Dans cet exemple, on utilisera une authentification des comptes sur la base locale des utilisateurs Unix du serveur.

```

# auth      sufficient      pam_securityserver.so # uncomment for OS X
auth      sufficient      pam_unix.so
auth      required       pam_deny.so
account   required       pam_permit.so
session   required       pam_permit.so

```

4.3 Debuggage

4.3.1 Affichage des avertissements PHP

Fichier php.ini (souvent sous /etc/php.ini)

```

; - Show all errors, except for notices and coding standards warnings
;
error_reporting = E_ALL & ~E_NOTICE
;
; Print out errors (as a part of the output). For production web sites,
; you're strongly encouraged to turn this feature off, and use error logging

```

```
; instead (see below). Keeping display_errors enabled on a production web
site
; may reveal security information to end users, such as file paths on your Web
; server, your database schema or other information.
display_errors = On
```

 Ensuite il faut redémarrer apache.

Ceci permet de voir les erreurs directement sur le navigateur. Sinon les log sont écrits dans fichier de log apache (souvent sous /var/log/httpd/error.log).

4.3.2 Affichage des requêtes PostgreSQL

Fichier `postgresql.conf` :

```
log_destination = 'syslog'
# These are relevant when logging to syslog:
#syslog_facility = 'LOCAL0'
#syslog_ident = 'postgres'
log_min_duration_statement = 0 # -1 is disabled, 0 logs all statements
```

Il faut aussi configurer syslog pour afficher les log en cohérences avec `postgresql.conf`.

Exemple avec `/etc/syslog.conf` :

```
# Log all the postgresql messages in one place.
local.* -/var/log/pgsql
```

Pour n'afficher que les erreurs mettre -1 dans `log_min_duration_statement`.

 Pour activer les changements il faut redémarrer postgresql.

4.3.3 freedom en mode debug

Il est aussi possible d'activer la trace de nombreuses vérifications en positionnant le paramètre CORE_LOGLEVEL du noyau freedom. Pour cela, il faut se connecter admin, aller dans l'application administration puis dans gestion des applications/paramètres applicatif/noyau/CORE_LOGLEVEL. Il faut mettre la valeur IWEFD pour tout tracer. Le 'D' indique le mode debug. Ceci entraîne des performances moindre. Ne pas oublier d'enlever ce mode lorsque le problème est résolu.

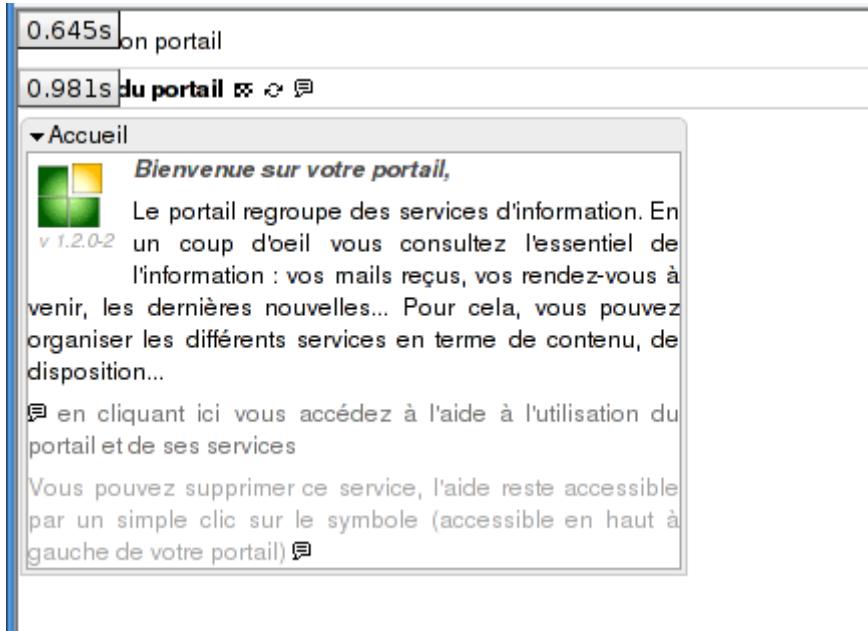
L'affichage de ces traces se trouve dans le même fichier de log que celui d'apache.

4.3.4 Tracer le temps d'exécution des actions dans Freedom

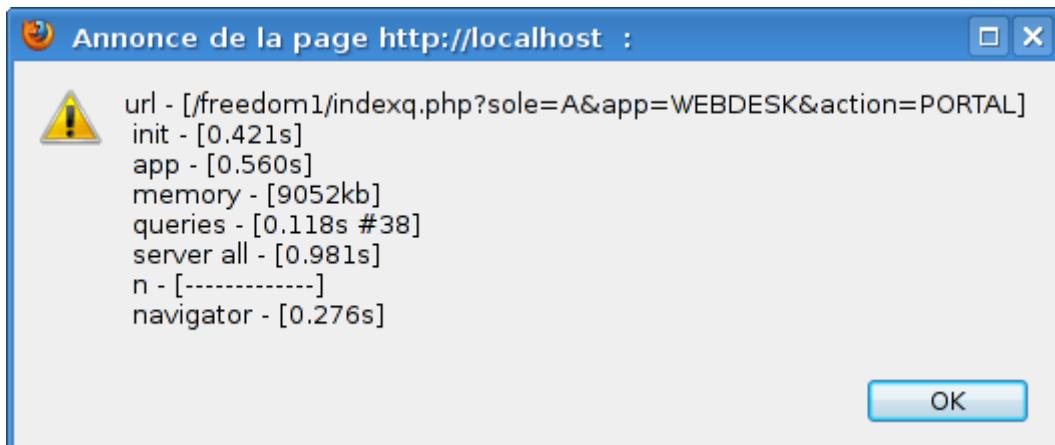
En pré-requis, il faut installer le paquet 'php5-xdebug'

Ensuite, il faut se connecter à l'adresse suivante : → <http://localhost/freedom/indexq.php>

Cela permet d'afficher en haut à gauche de chaque fenêtre Freedom le temps d'exécution des actions :



Et en cliquant sur l'un des chiffres, cela affiche le détail des temps :



Détail des temps indiqués :

- init : tps inclusion des fichiers : PHP : include + authentification
- app : code PHP hors init : ce que fait l'action
- memory : mémoire utilisée
- server all : somme des temps
- navigator : tps d'affichage par le navigateur

Remarque : Un clic droit sur le chiffre permet de le faire disparaître.

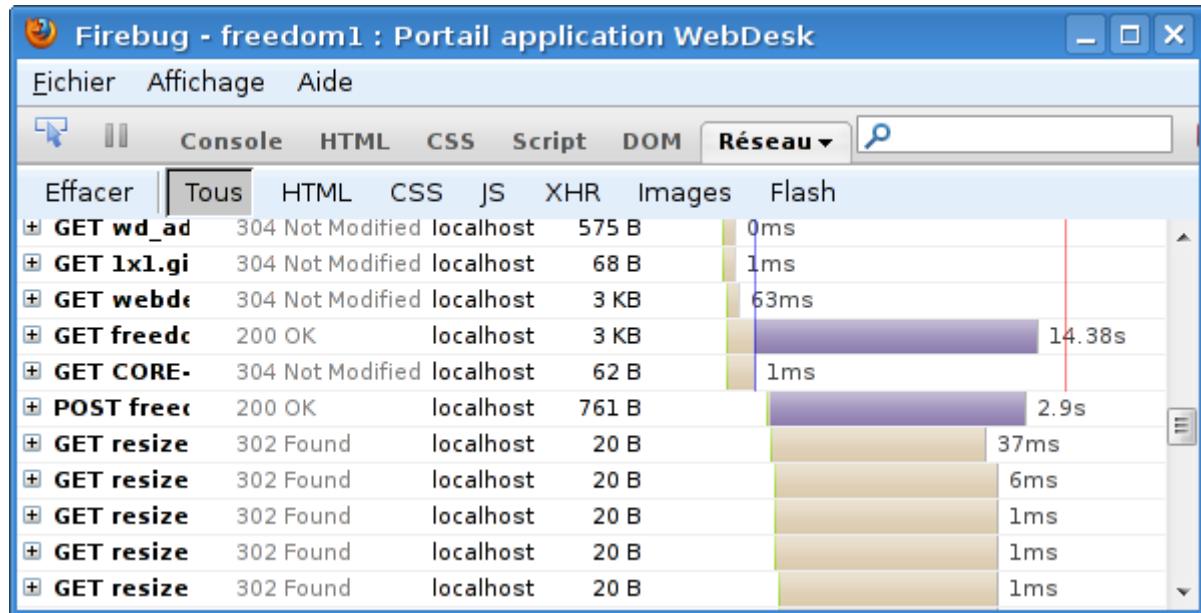
4.3.5 Firebug

Le plugin 'Firebug' disponible pour Firefox permet :

- de débuguer les applications javascript
- d'analyser ses pages Web
- de connaître précisément le temps de chargements de chaque élément d'une page Web

Firebug est téléchargeable ici : <https://addons.mozilla.org/fr/firefox/addon/1843>

Voici un exemple de résultat :



4.4 Diagnostic des bases de données

En cas de problème sur le fonctionnement de FREEDOM, un diagnostic de premier niveau peut être effectué sur les bases de données.

Ce diagnostic est accessible via l'url suivante

```
http:<votre serveur>/freedom/admin
```

Cette url vous demandera le mot de passe de l'administrateur (Celui qui a servi à la création).

Cliquez sur **Check List**

4.4.1 main connection db

Signification :

Remède :

Vérifier que Postgresql est bien lancé

4.4.2 unreference user in group

Signification :

Remède :

Supprimer les références :

```
DELETE FROM groups where iduser not in (select id from users);
```

4.4.3 user as group

Exemple :

```
user as group1 users detected as group 25
```

Signification :

Peut arriver par un import sql qui écraserait les utilisateurs

Remède :

Soit mettre les utilisateurs comme groupe (mettre l'attribut `isgroup` à Y dans la table `users`) ou supprimer la référence dans la table `groupe` :

```
anakeen=# begin;
anakeen=# DELETE FROM groups WHERE idgroup not in (select id from users where
isgroup='Y');
DELETE 110
anakeen=# commit;
COMMIT
```

Et exécuter cette commande :

```
# wsh --api=freedom_groups
```

4.4.4 unreference actions

Signification :

Peut arriver lors de suppression d'applications

Remède :

Appliquer le fichier sql `/usr/share/what/WHAT/what_clean.sql` sur la base principale

4.4.5 unreference parameters

Signification :

Peut arriver lors de suppression d'applications

Remède :

Appliquer le fichier sql `/usr/share/what/WHAT/what_clean.sql` sur la base principale

4.4.6 unreference acl

Signification :

Remède :

Appliquer le fichier sql `/usr/share/what/WHAT/what_clean.sql` sur la base principale

4.4.7 unreference permission

Signification :

Remède :

Appliquer le fichier sql `/usr/share/what/WHAT/what_clean.sql` sur la base principale

4.4.8 connection db freedom

Signification :

Remède :

4.4.9 double doc id

Exemple :

```
1 double id detected
Array
(
    [1178] => 2
)
```

Signification :

Plusieurs documents ont le même identifiant numérique. Peut arriver lors d'importation avec des identificateurs fixe ou lors d'importation avec des familles inexistantes.

Remède :

Supprimer l'un des doublons (ou les deux)

```
freedom=# select id,count(*) from doc group by id having count(*)>1;
id | count
---+---
1178 | 2
```

Détail des enregistrements posant problème :

```
freedom=# select id,title,fromid,locked from doc where id=1178;
id | title | fromid | locked
---+-----+-----+
1178 | Fiche de suivi | 0 | 0
1178 | Dossier EI auto | 1169 | 0
```

Suppression de l'enregistrement incorrecte :

```
freedom=# delete from doc where id=1178 and fromid=0;
DELETE 1
```

4.4.10 double doc name**Signification :**

Plusieurs documents ont le même identifiant logique (attribut name)

Remède :

Supprimer l'un des doublons (ou les deux)

4.4.11 family inheritance**Exemple :**

```
Family [44232]: fromid = 0, pg inherit=
Family [44237]: fromid = 0, pg inherit=
Family [44241]: fromid = 0, pg inherit=
...

```

Signification :

L'héritage déclaré par la famille ne correspond pas à celle défini dans postgresql. Peut arriver si on change l'héritage d'une famille ou si une famille est créée avant la famille père

Remède :

Identifier les enregistrement posant problème :

```
freedom=# select id,title,fromid,locked,doctype from doc where fromid=0 and doctype is null;
   id | title | fromid | locked | doctype
-----+-----+-----+-----+
 44232 |      |     0 |      0 |
 44237 |      |     0 |      0 |
 44241 |      |     0 |      0 |
 ...

```

Supprimer les enregistrements :

```
freedom=# delete from doc where fromid=0 and doctype is null;
DELETE 14
```

Ou supprimer le table correspondante et lancer :

```
wsh --api=fdl_adoc --docid=<N°famille>
```

4.4.12 user group synchro

Exemple

```
inheritance tables group are differents between user=anakeen dbname=anakeen
and user=anakeen dbname=freedom connection db webdavdbnam
```

Signification :

Les groupes définis dans la base principale ne sont pas les mêmes que celle de la base FREEDOM

Remède :

Lancer :

```
wsh --api=freedom_groups
```

4.4.13 connection db webdav

Signification :

Si le module WORKSPACE est installé

Remède :

4.4.14 connection USER LDAP

Signification :

Si le module NETWORKUSER est installé

Remède :

4.5 Exécution périodique

La famille Processus (EXEC) permet de définir des tâche/traitements à exécuter de manière périodique ou récurrente.

Le traitement peut-être :

- le lancement d'une action d'une application donnée ;

- ou, le lancement d'un script d'API.

Il est possible ensuite de spécifier une répétition pour l'exécution périodique du processus. Dans ce cas, la date/horaire de la prochaine exécution sera planifiée à la date de fin d'exécution du processus précédent + la période d'exécution spécifié.

Le déclenchement manuel du processus, par l'utilisation du "Exécuter maintenant", désactivera l'exécution périodique du processus. Pour ré-activer l'exécution périodique, il faudra alors éditer et sauver le processus.

Exemple de déclaration d'un processus pour l'import périodique d'un fichier CSV toutes les 2 heures :

The screenshot shows the configuration dialog for a process named 'IMPORT NOUVEAUX PRODUITS' under the 'FREEDOM_IMPORT' category. The dialog is divided into several sections:

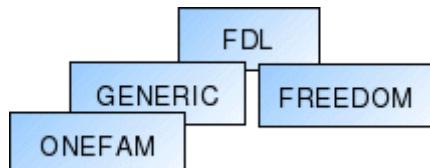
- Identification:** Exécutant: Master Default, Titre: Import nouveaux produits
- Traitement:** Application: (empty), Action: (empty), Api: freedom_import
- paramètres:** A table with a single row: variable file, valeur /tmp/nouveaux_produits.csv
- Dates:** À exécuter le: 03/11/2009 16:34, Période en jours: (empty), Période en heures: 2, Période en minutes: (empty)

4.6 Paramétrage de Freedom

4.6.1 Les modules de Freedom

FREEDOM comprend plusieurs modules applicatifs de base :

- CORE - Noyau
- FDL - Freedom Library
- FREEDOM - Gestion documentaire
- GENERIC - Manipulation d'une famille
- ONEFAM - Interface d'utilisation de GENERIC
- VAULT - Coffre de stockage des fichiers



Chacune des applications comprend des paramètres.

4.6.2 Accès au paramétrage des modules de Freedom

Le paramétrage est accessible via les menus :

- Freedom Master / Paramètres applicatif
- Freedom Master / Mes paramètres

Les paramètres sont de plusieurs types :

- Type A : Paramètre local à l'application
- Type G : Paramètre global pouvant être utilisés par d'autres applications
- Type U : Paramètre pouvant être modifié par chaque utilisateur via le menu "Freedom Master / Mes paramètres"

Remarques :

- Les paramètres en gras indiquent qu'ils sont globaux (Type = G)
- Pour connaître le nom interne du paramètre, il faut placer la souris au dessus de sa description

The screenshot shows the Freedom application's administration interface. On the left, there is a sidebar with various menu items like Mail Administration, Suivi d'affaires, Commercial, Production, Vente, Achat, Prospect, Agenda de groupe, Espace de travail, Coffre, Site freedom, Accessibilités, Camel d'adresses, Une famille, Utilisateurs, Gestion documentaire, Recherche générale, Mon portail, and Mon compte. The 'Administration' item is highlighted with a red oval. On the right, there is a main window titled 'Administration'. It has a toolbar with icons for mail, messaging, security, and help. Below the toolbar, there are three tabs: 'paramètres applicatif', 'paramètres utilisateur', and 'mes paramètres'. The 'paramètres applicatif' tab is selected and shows a list titled 'Liste des paramètres applicatif' with a red header bar. The list contains items with icons and names: Accessibilités, Achat, Suivi d'affaires, Gestion des applications, Authent, Bug Tracker, and Commercial. The 'Gestion des applications' item is highlighted with a red oval. A red arrow points from the 'Gestion des applications' item in the main window to the 'Gestion des applications' item in the list.

4.6.3 ACCESS - Accessibilités

Nom du paramètre	Description	Type	Valeur par défaut
INIT	INIT	G	yes
VERSION	version application		2.6.0-2
APPNAME	APPNAME		ACCESS

4.6.4 APPMNG - Gestion des applications

Nom du paramètre	Description	Type	Valeur par défaut
INIT	INIT	G	yes
VERSION	version application		2.6.0-2
APPNAME	APPNAME		APPMNG

4.6.5 AUTHENT - Authent

Nom du paramètre	Description	Type	Valeur par défaut
INIT	INIT	G	yes
VERSION	version application		2.6.0-2
APPNAME	APPNAME		AUTHENT

4.6.6 CORE - Noyau

Nom	Définition	Type	Défaut
Système			
CORE_URLINDEX	url d'accès à l'index de FREEDOM Exemple : http://192.168.34.12/free-dom/index.php Cette URL est utilisé notamment pour les liens dans les mails	G	Note Version > 2.7.0
MEMORY_LIMIT	limite mémoire par processus	G	32M
CORE_DBCONNECT	mode de connexion aux bases données. Mettre en persistant peut permettre une accélération des requêtes. Dans ce cas il faut penser à modifier les 2 paramètres PHP (fichier php.ini) : De plus il est nécessaire de bien paramétrier le fichier postgresql.conf <pre># - Connection Settings - max_connections = 100</pre> Le nombre de connection doit être approximativement 3 fois le nombre max de requêtes en parallèles. La prise en compte de ce paramètre n'est effectif qu'après avoir lancé le script /usr/share/what/wstart. La modification en persistant ne doit être fait que si nécessaire après des tests de performances. Plus de détails	G	unpersistant Version > 2.7.1

USE_FREEDOM_USER	utilise FREEDOM comme interface de modification des utilisateurs	G	yes
Personnalisation client			
STYLE	Style par défaut	GU	default
FONTSIZE	Taille par défaut	GU	normal
CORE_CLIENT	nom du client	G	
CORE_LOGOCLIENT	logo du client	G	
CORE_LANG	langue par défaut	GU	fr_FR
Installation			
CORE_DB	accès base de donnée principale	G	dbname=anakeen user=anakeen
CORE_PUBDIR	dépôt d'installation	G	/usr/share/what
CORE_REALM	nom du royaume de connection (voir conf HTTP)	G	FREEDOM Connection

4.6.7 FDL - Bibliothèque FREEDOM

L'application FDL définit un ensemble de fonctions pour la manipulation des documents. Ces paramètres sont tous globaux et sont utilisés par les autres applications : FREEDOM, GENERIC et ONEFAM.

Nom	Définition	Type	Défaut
FREEDOM_DB	base de données Freedom paramètres d'accès à la base de données postgresql pour la base documentaire.	G	host=localhost user=anakeen port=5432 dbname=freedom
ENUM_TITLE_SIZE	taille maximum de chaînes de caractère pour les choix.	GU	40
FDL_BCC	copie envoi de mail. Mettre 'yes' si vous voulez recevoir une copie des mails que vous envoyez via le serveur. Dans ce cas, vous êtes mis en destinataire caché (Blind Carbon Copy).	GU	No
FDL_MAX_FGEXPORTDOC	Nombre maximum de document pouvant être importé directement, c'est à dire pas en tâche de fond. Cela dépend de la puissance du serveur, un timeout de 30s est déclenché si le serveur n'a pas pu traiter tous les documents. Seuls les documents traités dans les 30s seront importés.	G	20
SMTP_HOST	nom ou adresse ip du serveur SMTP	G	localhost
SMTP_PORT	numéro de port du serveur SMTP	G	25
SMTP_LOGIN	en cas de connexion avec authentification. Identifiant de connexion	G	

SMTP_PASSWORD	mot de passe pour connexion authentifiée	G	
SMTP_FROM	adresse email de retour en cas d'envoi de mail par utilisateur sans mail	G	
TE_HOST	nom du serveur de transformation	G	localhost
TE_PORT	numéro port de la connection vers le serveur de transformation	G	10000



FREEDOM n'a pas par défaut un serveur SMTP pour envoyer les mail. Il est nécessaire de configurer l'accès à un serveur SMTP pour bénéficier des envois de documents par email.

4.6.8 GENERIC - Manipulation d'une famille

L'application GENERIC n'est pas utilisée directement. Ses paramètres sont transmis aux applications héritant de GENERIC. Il n'est pas nécessaire de modifier ces paramètres.

Nom	Définition	Type	Défaut
DEFAULT_FAMILY	famille de document par défaut	A	
DEFAULT_FLD	n° dossier par défaut	A	
CARD_SLICE_LIST	nombre de cartes par page	AU	8

4.6.9 FREEDOM - Gestion documentaire

Nom	Définition	Type	Défaut
ROOT_FLD	Identificateur du dossier racine Ceci permet de définir le dossier racine du plan de classement commun	A	9

4.7 Quelques commande PostgreSQL pratiques

4.7.1 Voir la taille des tables/index

```
freedom=> SELECT relname, reltuples, relpages * 8 / 1024 AS "MB" FROM pg_class
 ORDER BY relpages DESC;
   relname      | reltuples | MB
-----+-----+-----+
 i_docrelc    | 60868    | 492
 i_docrels    | 60868    | 488
 full_docread | 16353    | 48
 pg_toast_103473 | 23626    | 32
 idx_perm     | 178001   | 28
 pg_toast_16853 | 19210    | 28
 docread      | 16353    | 20
```

[...]

Cette commande est pour un block size de 8 Ko. Pour connaître la valeur de votre block size :

```
# LC_ALL=C pg_controldata /var/lib/pgsql/data/
[...]
Database block size: 8192
[...]
```

4.7.2 Autres ressources

[Optimisation de Postgresql](#)

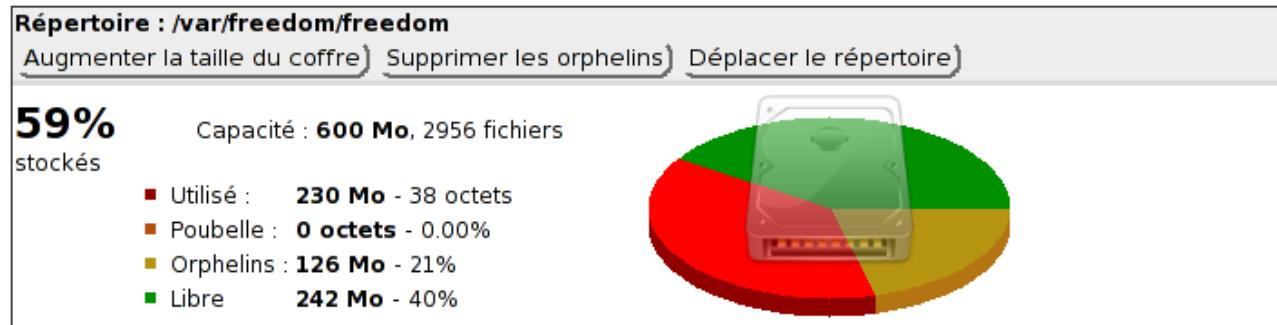
4.8 Sauvegarde et restauration des bases de données

4.9 VAULT - Coffre de stockage des fichiers

Le VAULT (Coffre) est un module qui gère le stockage et l'accès aux fichiers attachés aux documents.

La localisation du coffre et sa capacité est modifiable par l'interface web via l'application Coffre (accessible si connecté en tant qu'administrateur).

4.9.1 Interface



Utilisation du vault :

- Utilisé volume utilisé par les fichiers liés aux documents freedom
- Poubelle volume utilisé par les fichiers liés aux documents freedom supprimés
- Orphelins sont ceux qui ne sont rattachés à aucun document FREEDOM. Il ne sont plus accessibles via l'interface classique. Ces orphelins sont créés lors des documents ont été supprimés de la base de données lors du vider de la poubelle ou par une opération manuelle sur la base de données. Ils peuvent à tout moment être supprimés par l'administrateur pour récupérer de la place.
- Libre : espace disponible

4.9.2 Création d'un coffre

Le bouton **Nouveau coffre** situé au dessus de la liste des coffres existants, permet la création d'un coffre.

Les informations à fournir sont :

- le chemin complet de la racine du coffre. Ce dossier doit exister, être accessible en lecture et écriture par le user sous lequel tournent les processus apache et être vide.
- la taille allouée.

4.9.3 Modification du volume d'un coffre

Pour modifier sa capacité il suffit de cliquer sur 'Augmenter la taille du coffre', puis de saisir la nouvelle taille. FREEDOM n'a pas en charge de vérifier la capacité réelle de stockage. L'administrateur doit au préalable vérifier que cette capacité est réellement disponible. La capacité renseignée est une contrainte logique. Dès que la somme des tailles des fichiers stockées est dépassée, FREEDOM bloque l'ajout de nouveaux fichiers.

4.9.4 Déplacement d'un coffre

Pour modifier la localisation, il faut procéder à deux opérations. L'administrateur doit déplacer physiquement le répertoire (généralement utilisation de la commande mv unix). Attention, il faut que le répertoire, et l'ensemble de ses fichiers et sous-répertoires soient accessibles en lecture et écriture par le compte utilisateur unix qui a lancé le serveur web (généralement http ou apache). Ensuite, il doit indiquer via l'interface web le nouveau répertoire.

4.9.5 Commande shell freedom

- `./wsh.php -api=VaultIndexInit` : reconstruit la table d'index entre le(s) coffre(s) et la base de données FREEDOM. Utile en cas de modification des bases de données directement. Cette table d'index est utilisée lors de la suppression des orphelins.
- `./wsh -api=VaultExamine -cmd=clean-unref` : Supprime les orphelins
 - * `./wsh -api=vault_init -size=500000000 -path=/var/freedom/chemin` : Crée un nouveau coffre de 500Mo dans le répertoire /var/freedom/chemin.



A compter de la version 3.1.0 du VAULT, il est possible de fournir un **nom logique** au nouveau coffre. La commande wsh est la suivante :
`./wsh.php -api=vault_init -name=nom_logique_du_coffre - size=500000000 -path=/var/freedom/chemin'

4.10 Type MIME

Pour la détection du type MIME d'un fichier, Freedom utilise les mécanismes suivant :

- Association type MIME par l'extension du fichier
 - En utilisant un fichier d'association local s'il est présent (\$CONTEXT_ROOT/admin/mime-user.conf)
 - En utilisant un fichier d'association global (\$CONTEXT_ROOT/admin/mime.conf)
- Utilisation de la commande système `file`

L'ordre d'application des règles d'association est le suivant :

- 1) Si le fichier `mime-user.conf` existe, alors appliquer les règles de détection de `mime-user.conf`
- 2) Si le type n'a pas été trouvé, alors appliquer les règles de détection de `mime.conf`
- 3) Si le type n'a pas été trouvé, alors appliquer l'utilisation de la commande système `file`

4.10.1 Fichier d'association local `\$CONTEXT_ROOT/admin/mime-user.conf`

Le fichier `\$CONTEXT_ROOT/admin/mime-user.conf` n'est pas fourni par Freedom, et permet donc à l'administrateur local du système de le créer et d'y insérer ses propres règles d'association.

Ce fichier est donc optionnel, et ne sera pas écrasé par les mises-à-jour de Freedom.

Vous pouvez créer ce fichier à partir du modèle fournit dans `'\$CONTEXT_ROOT/admin/mime-user.conf.sample' :

```
$ cp admin/mime-user.conf.sample admin/mime-user.conf
```

Le format XML du fichier est le suivant :

```
<?xml version="1.0"?>
<mimes>
    <mime ext="foo" sys="application/foo" text="Foo files" />
    <mime ext="bar" sys="application/bar" text="Bar files" />
</mimes>
```

Les règles d'association sont décrites par des éléments <mime/> avec les attributs suivants :

- 'ext' : l'extension du fichier sans le point
- 'sys' : le type MIME de ce type de fichier
- 'text' : la description textuelle de ce type de fichier

4.10.2 Fichier d'association global `'\$CONTEXT_ROOT/admin/mime.conf'

Le fichier `'\$CONTEXT_ROOT/admin/mime.conf' est fourni par Freedom.

Ce fichier ne doit pas être modifié car il sera remplacé lors des mises-à-jour de Freedom. Pour déclarer vos propres règles d'association, utiliser le fichier `'mime-user.conf' décrit ci-dessus.

Le format XML du fichier est identique à celui du fichier `'mime-user.conf' décrit ci-dessus.

fin du document



Ce document est publié sous licence CC.

<http://creativecommons.org/licenses/by-nc-sa/2.0/fr/>



Vous êtes libres :

- de reproduire, distribuer et communiquer cette création au public
- de modifier cette création

Selon les conditions suivantes :

- **Paternité** — Vous devez citer le nom de l'auteur original de la manière indiquée par l'auteur de l'oeuvre ou le titulaire des droits qui vous confère cette autorisation (mais pas d'une manière qui suggérerait qu'ils vous soutiennent ou approuvent votre utilisation de l'oeuvre).
- **Pas d'Utilisation Commerciale** — Vous n'avez pas le droit d'utiliser cette création à des fins commerciales.
- **Partage des Conditions Initiales à l'Identique** — Si vous modifiez, transformez ou adaptez cette création, vous n'avez le droit de distribuer la création qui en résulte que sous un contrat identique à celui-ci.

