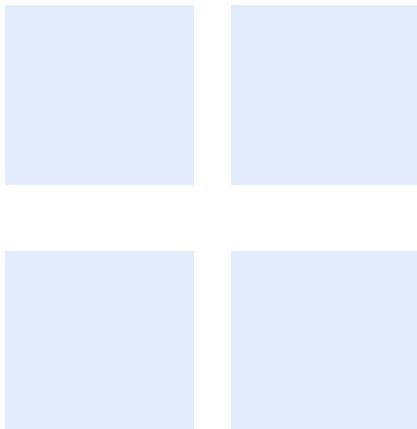
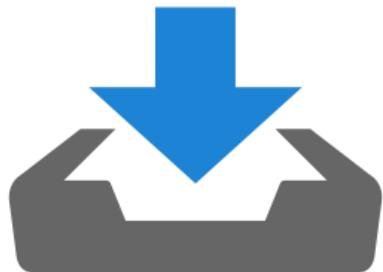


Référence SPIP/X/2024.001 – Ed. 1.0



GUIDE CONCEPTION DU PLUGIN ARCHIVAGE DE CONTENUS



FICHE D'IDENTIFICATION

| | |
|----------------------|--|
| Rédacteur | Eric Lupinacci |
| Projet | SPIP |
| Étude | Conception du plugin Archivage de Contenus |
| Nature du document | Guide |
| Date | 16/06/2024 |
| Nom du fichier | Guide - Le plugin Archivage Contenus.docx |
| Référence | SPIP/X/2024.001 – Ed. 1.0 |
| Dernière mise à jour | 16/06/2024 15:09:18 |
| Langue du document | Français |
| Nombre de pages | 21 |

TABLE DES MATIERES

| | | |
|------------|---|-----------|
| 1. | INTRODUCTION | 5 |
| 2. | CONCEPTS | 5 |
| 2.1 | LE CONCEPT D'ARCHIVAGE | 5 |
| 2.2 | LES OBJETS ARCHIVABLES | 5 |
| 2.3 | LE CONTEXTE D'ARCHIVAGE | 5 |
| 2.4 | LES ACTIONS RELATIVES A L'ARCHIVAGE | 6 |
| 2.4.1 | L'ARCHIVAGE | 6 |
| 2.4.2 | LE DESARCHIVAGE | 6 |
| 2.4.3 | L'EXPLICATION | 6 |
| 2.4.4 | L'EFFACEMENT | 6 |
| 2.4.5 | GRAPHE ETAT-TRANSITION DE L'ARCHIVAGE | 7 |
| 2.5 | LA GESTION DES BRANCHES | 7 |
| 3. | PERIMETRE DU PLUGIN ARCHIVAGE DE CONTENUS | 8 |
| 3.1 | L'API DE GESTION DE L'ARCHIVAGE | 8 |
| 3.2 | LES CRITERES | 9 |
| 3.3 | LES BALISES | 10 |
| 3.4 | LES PIPELINES | 10 |
| 3.5 | L'INTERFACE UTILISATEUR DANS L'ESPACE PRIVE | 10 |
| 3.5.1 | LE SUIVI DES ARCHIVES | 10 |
| 3.5.2 | MODIFICATIONS DES AFFICHAGES & COMPORTEMENTS STANDARD | 11 |
| 4. | FONCTIONNEMENT DU PLUGIN | 12 |
| 4.1 | LE MECANISME D'ARCHIVAGE | 12 |
| 4.1.1 | LE PIPELINE POST_BOUCLE | 12 |
| 4.1.2 | LE CRITERE {ARCHIVE} | 12 |
| 4.1.3 | LE CRITERE {IGNORER_ARCHIVAGE} | 12 |
| 4.1.4 | LE CRITERE {DESARCHIVE} | 12 |
| 4.1.5 | LE PIPELINE POST_ARCHIVAGE_OBJET | 13 |
| 4.1.6 | LA PROPAGATION DES ACTIONS | 13 |
| 4.2 | L'INSTALLATION ET LA CONFIGURATION | 13 |
| 4.3 | IMPACTS SUR LA GESTION DES OBJETS | 13 |
| 4.3.1 | LE CHANGEMENT DE STATUT D'UN OBJET | 13 |
| 4.3.2 | LA CREATION D'UN OBJET | 14 |
| 4.3.3 | LA MODIFICATION D'UN OBJET | 14 |
| 4.3.4 | LE DEPLACEMENT D'UN OBJET | 14 |

| | | |
|------------|--|-----------|
| 4.4 | LES AUTORISATIONS | 15 |
| 4.5 | L'INTERFACE PRIVEE | 16 |
| 4.5.1 | LE SUIVI DES ARCHIVES | 16 |
| 4.5.2 | LE PARAMETRAGE DU PLUGIN | 17 |
| 4.5.3 | LES ACTIONS SUR LES OBJETS | 17 |
| 5. | DONNEES DE ARCHIVAGE DE CONTENUS | 18 |
| 5.1 | LE CONTEXTE D'ARCHIVAGE | 18 |
| 5.2 | LA CONFIGURATION DU PLUGIN | 18 |
| 5.3 | LES ESPACES DE STOCKAGE | 18 |
| 6. | UTILISATION DU PLUGIN | 19 |
| 6.1 | PERSONNALISER LA LISTE DES TABLES ARCHIVABLES | 19 |
| 6.2 | PERSONNALISER LES MOTIFS | 19 |
| 6.2.1 | LA LISTE DES MOTIFS | 19 |
| 6.2.2 | LES MOTIFS PAR DEFAUT | 20 |
| 6.3 | PERSONNALISER LES LISTES D'ARCHIVES | 20 |
| 7. | REGLES DE CODAGE | 21 |
| 7.1 | NOMMAGE DES FONCTIONS | 21 |
| 7.2 | ARGUMENTS ET VARIABLES STANDARDISES | 21 |
| 7.3 | TYPAGE DES ARGUMENTS | 21 |

1. INTRODUCTION

Ce document a pour but de décrire les principes de base et les éléments de conception du plugin Archivage de Contenus (version 0.5.4 et ultérieures) dont l'objectif est de fournir un mécanisme d'archivage des objets SPIP sans modification du statut.

Le plugin Archivage de Contenus fournit également une interface d'administration des archives dans l'espace privé de SPIP.

2. CONCEPTS

2.1 Le concept d'archivage

L'archivage est une action destinée à ranger des objets SPIP dans une zone de stockage virtuelle pour les extraire de la publication (espace public) voire des différents affichages standard de l'espace privé. La différence avec d'autres plugins comme Archive est que l'**action d'archivage ne modifie pas le statut de l'objet** concerné. Un objet archivé peut-être restauré dans son état initial en une action appelée **désarchivage**.

Étant donné que certains objets sont, comme les rubriques, des conteneurs d'autres objets du même type ou d'un type différent, les actions d'archivage ou de désarchivage sont **propagées automatiquement** à l'ensemble des enfants de la branche.

2.2 Les objets archivables

Le plugin peut fonctionner avec tout type d'objet SPIP, c'est-à-dire un enregistrement d'une table déclarée par le pipeline `declarer_tables_objets_sql`. Par défaut, le plugin peut s'appliquer à tous les types d'objet déclarés par le pipeline.

Ces types d'objet et leurs tables SQL associées sont qualifiés de **archivables**. Il est possible de réduire cette liste en utilisant le pipeline `archivage_exclusion_tables` ou la constante `_ARCHIVAGE_TABLES_EXCLUES`.

Par défaut, les objets archivables ne sont pas d'emblée autorisés à l'archivage : un paramétrage utilisateur est nécessaire pour choisir parmi les objets archivables ceux qui deviendront effectivement des objets autorisés à l'archivage dans le contexte du site utilisateur.

2.3 Le contexte d'archivage

Le contexte d'archivage est un ensemble de données qui informe sur l'archivage d'un objet. Il est composé :

- d'un **indicateur d'état d'archivage** qui prend les valeurs :

- archivé,
- non archivé, sachant que cet état peut être précisé pour distinguer un objet « hors archive » d'un objet désarchivé si la consignation du désarchivage est activée.

- d'une **date**, celle de d'archivage ou de désarchivage ;
- de l'éventuel **objet racine** sur lequel l'action a été déclenchée, et ce uniquement pour les enfants de ladite racine ;
- et d'un **motif** facultatif, expliquant la raison de l'archivage voire du désarchivage.

Le contexte d'archivage est donc matérialisé par plusieurs champs qui sont ajoutés à chaque table archivable. Le plugin Archivage de Contenus ajoutent ces champs lors de son installation pour les tables archivables par défaut.

Dans certains cas, le plugin manipule un contexte d'archivage étendu, composé du contexte d'archivage standard complété par un indicateur de descendance et un état calculé qui peut prendre les valeurs 'archive', 'desarchive' et '' (hors archive).

2.4 Les actions relatives à l'archivage

2.4.1 L'archivage

L'action de base est l'archivage d'un objet non archivé. Cette action s'accompagne toujours de la mise à jour de la date du contexte d'archivage et, le cas échéant pour les enfants, de la consignation de l'objet racine sur lequel l'archivage a eu lieu.

En outre, si le motif d'archivage est utilisé, ce motif est rempli automatiquement par un motif par défaut qui peut être modifié ultérieurement.

2.4.2 Le désarchivage

Le désarchivage, appliqué uniquement à un objet archivé, permet de rétablir le comportement normal de l'objet (affichage public, affichage privé et fonctionnement) et de ses éventuels enfants. Néanmoins, si la consignation du désarchivage est activée par paramétrage, l'objet obtenu est considéré comme **désarchivé : non archivé avec trace du désarchivage**.

De même que pour l'archivage, si le motif est utilisé, ce motif est aussi rempli automatiquement et peut aussi être modifié à posteriori via l'action d'explication du désarchivage. Cette option n'est possible que si la consignation du désarchivage est activée.

2.4.3 L'explication

L'explication est l'action qui permet de modifier le motif d'archivage. Elle n'est accessible que si le paramétrage du plugin autorise l'utilisation du motif. Cette action n'est jamais propagée aux enfants car elle ne concerne que l'objet propagateur.

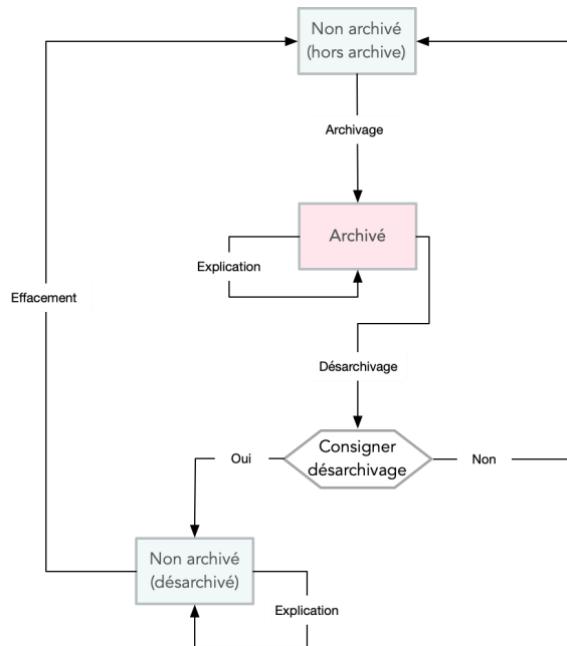
Le motif est choisi parmi une liste prédéfinie fournie par le plugin et qui peut être complétée ou amendée en utilisant le pipeline `archivage_liste_motifs`.

2.4.4 L'effacement

Dans le cas où la consignation du désarchivage est activée, il est possible de rétablir l'objet concerné dans l'état initial « hors archive » : c'est l'action d'effacement des données d'archivage qui s'applique uniquement aux objets désarchivés et, le cas échéant, à leurs enfants.

2.4.5 Graphe état-transition de l'archivage

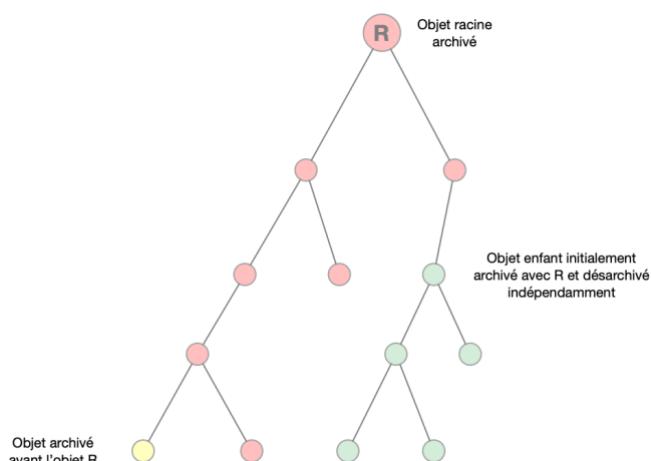
Le fonctionnement du plugin peut être résumé dans le graphe état-transition qui suit.



2.5 La gestion des branches

Toutes les actions à l'exception de l'explication entraînent automatiquement la **propagation aux enfants** de l'objet concerné. Ce mécanisme n'est pas débrayable. Néanmoins, il est possible de le contourner quelque peu en autorisant, par paramétrage, les enfants à vivre leur vie d'après indépendamment de celle de la racine initiatrice.

Ainsi, un enfant archivé à la suite d'un archivage de sa racine pourra être désarchivé indépendamment avec ses éventuels enfants. Tout lien d'archivage avec son ancienne racine est alors rompu et toute action future sur son ancienne racine sera inopérante sur cet enfant.



3. PERIMETRE DU PLUGIN ARCHIVAGE DE CONTENUS

Le plugin propose :

- une API de gestion de l'archivage composée de fonctions PHP, de critères et de balises SPIP ;
- des pipelines facilitant l'utilisation et la personnalisation du plugin ;
- une administration des archives dans l'espace privé.

3.1 L'API de gestion de l'archivage

L'API concerne la gestion du contexte d'archivage, lecture et modification, la gestion des motifs et la gestion des tables archivables et de leurs champs.

API ARCHIVAGE : INC/ARCHIVAGE.PHP

Gestion du contexte d'archivage

| | |
|-------------------------------------|--|
| archivage_objet_est_autorise | Vérifie si un type d'objet est autorisé à l'archivage en consultant le paramétrage du plugin. |
| archivage_objet_lire | Renvoie les informations d'archivage d'un objet connu par son identifiant. Outre le contexte d'archivage, la fonction calcul l'état d'archivage (archivé, désarchivé, hors archive) et l'indicateur d'existence d'une descendance. |
| archivage_objet_modifier | Modifie l'état d'archivage d'un objet. Plusieurs actions sont possibles : archiver, désarchiver, expliquer (modification du motif) ou effacer (tout le contexte d'archivage). Sauf pour l'explication, cette action est propagée aux enfants éventuels de l'objet. Le pipeline <code>post_archivage_objet</code> permet de lancer des post-traitements. Cette fonction ne doit jamais être utilisée pour traiter des enfants lors d'une propagation. |
| archivage_objet_repertorier | Répertorie les objets archivés ou désarchivés d'un type donné et éventuellement enfants d'un objet parent direct. Chaque objet retourné, quel que soit le type, possède une structure standard. |

Gestion des motifs

| | |
|-------------------------------|--|
| archivage_motif_lister | Renvoie la liste des motifs d'archivage ou de désarchivage pour un type d'objet donné. Le tableau de sortie est de la forme <code>[id_motif] = traduction</code> . La fonction utilise le pipeline <code>archivage_liste_motifs</code> pour les plugins qui souhaitent ajouter des motifs. |
|-------------------------------|--|

| | |
|---|--|
| archivage_motif_defaut | Renvoie l'identifiant du motif par défaut pour l'archivage ou le désarchivage d'un type d'objet donné. La fonction utilise le pipeline <code>archivage_defaut_motif</code> pour les plugins qui souhaitent choisir leur motif par défaut. |
| archivage_motif_traduire | Renvoie pour un objet donné, la traduction du motif associé à l'objet. |
| Gestion des tables SQL | |
| archivage_table_reinitialiser | Supprime tout ou partie des traces d'archivage dans la base de données. Passe en revue toutes les tables autorisées ou une liste de tables précisées. |
| archivage_table_lister | Renvoie la liste des tables archivables ainsi que leur configuration si elles possèdent des enfants. On peut filtrer cette liste en ne retenant que les tables autorisées par l'utilisateur (paramétrage du plugin). Utilise le pipeline <code>archivage_exclusion_tables</code> ou la constante <code>_ARCHIVAGE_TABLES_EXCLUES</code> pour réduire la liste. |
| archivage_table_upgrade_contexte | Renvoie le tableau des instructions SQL permettant de créer chaque champ du contexte d'archivage d'une table archivable. Utilisée dans la fonction de mise à jour de la base (<code>upgrade</code>). |
| archivage_table_ajouter_contexte | Ajoute les champs nécessaires à l'archivage pour une table SQL donnée. Cette fonction est appelée dans la configuration si une table a été créée après l'installation du plugin |
| archivage_table_supprimer_contexte | Supprime, pour une table SQL archivable, les champs nécessaires à l'archivage. Utilisée dans la fonction de vidage de la base (<code>vider_tables</code>). |

3.2 Les critères

Les critères suivants sont mis à disposition des boucles sur les types d'objets archivables :

- `{archive}` et sa négation `{!archive}` permet de réduire les objets de la boucle à ceux archivés ou non archivés ;
- `{ignorer_archivage}` qui permet de désactiver le mécanisme d'exclusion automatique des objets archivés dans les boucles.
- `{desarchive}` permet, si la consignation du désarchivage est activée, de sélectionner plus simplement les objets désarchivés.

3.3 Les balises

Les balises suivantes permettent d'exposer une partie de l'API précédente pour une utilisation dans les squelettes :

- `#ARCHIVAGE_TABLES` qui renvoie la liste des tables d'objets archivables ou autorisées à l'archivage suivant l'argument fourni (encapsulation de `archivage_table_lister()`) ;
- `#ARCHIVAGE_CONTEXTE` qui fournit, dans un tableau, le contexte d'archivage étendu d'un objet (encapsulation de `archivage_objet_lire()`) ;
- `#ARCHIVAGE_LISTE` qui fournit, les objets archivés ou désarchivés d'un type donné, éventuellement enfants d'un parent (encapsulation de `archivage_objet_repertorier()`) ;
- `#ARCHIVAGE_MOTIF` qui renvoie le texte traduit du motif d'archivage ou de désarchivage (encapsulation de `archivage_motif_traduire()`).

Les prototypes des balises sont :

```
#ARCHIVAGE_TABLES{[tables_autorisees_seules = false]}.
#ARCHIVAGE_CONTEXTE{objet, id_objet}.
#ARCHIVAGE_LISTE{objet, etat[, parent]}.
#ARCHIVAGE_MOTIF{objet, id_objet}.
```

3.4 Les pipelines

Pour faciliter l'utilisation et la personnalisation des concepts, le plugin propose les pipelines suivants :

- `archivage_exclusion_tables`, pour exclure des tables de la liste des tables archivables ;
- `archivage_liste_motifs`, pour ajouter des motifs d'archivage et de désarchivage pour les types d'objet souhaités ;
- `archivage_defaut_motif`, pour définir le motif d'archivage et de désarchivage par défaut à utiliser pour chaque type d'objet ;
- `post_archivage_objet`, pour compléter les traitements de base de chaque action (à utiliser avec précaution).

3.5 L'interface utilisateur dans l'espace privé

3.5.1 Le suivi des archives

Le plugin Archivage de Contenus propose une page de suivi des archives qui liste, type d'objet par type d'objet autorisé à l'archivage, les objets archivés ou désarchivés (si la consignation du désarchivage est activée).

Cette page (`exec=archives`) est accessible à partir du menu « Publication » via l'entrée « Suivi de l'archivage ».

En outre, pour chaque page d'un objet dont le type est autorisé à l'archivage, la boîte d'informations est complétée par un bouton d'action permettant d'archiver ou de désarchiver ledit objet.

3.5.2 Modifications des affichages & comportements standard

Étant donné que l'espace privé n'est pas toujours prévu pour être extensible, ou du moins sans surcharger des pages entières, les modifications opérées par le plugin à ce stade de développement de l'espace privé de SPIP restent limitées.

En premier lieu, le mécanisme de changement de statut, formulaire d'institution et menu rapide, est bloqué pour les objets archivés. Cependant, la modification de contenus archivés reste possible ainsi que la création ou le déplacement de contenus enfants d'un contenu parent archivé.

Comme pour l'espace public, les boucles SPIP excluent automatiquement les objets archivés. C'est le cas pour les articles et les auteurs. Néanmoins, les rubriques étant affichées au travers d'un code PHP qui ne fait pas appel aux boucles SPIP, les affichages ne sont pas modifiés pour le moment. Ainsi, il serait intéressant d'afficher la liste des objets archivés dans un onglet des pages récapitulatives des objets comme articles, rubriques ou auteurs, mais cela demande de surcharger complètement la page. Ce n'est donc pas disponible aujourd'hui.

Enfin, pour les types d'objets conteneur comme les rubriques, le plugin affiche les listes des enfants archivés via le pipeline `affiche_enfants`.

4. FONCTIONNEMENT DU PLUGIN

4.1 Le mécanisme d'archivage

Le mécanisme d'archivage est basé sur deux assertions :

- par défaut, les objets archivés sont exclus des boucles (sauf condition particulière) ;
- l'archivage d'un objet conteneur est propagé automatiquement à ses objets enfants.

4.1.1 Le pipeline `post_boucle`

Le principe du plugin est, par défaut, d'exclure des boucles les objets archivés. Ce mécanisme est orchestré par le pipeline `post_boucle` qui est appelé systématiquement à la fin de la compilation d'une boucle SPIP. Si aucune condition impliquant le débrayage du mécanisme est identifiée, la boucle SPIP est mise à jour avec la condition supplémentaire, `est_archive=0`.

Les conditions qui débrayent le mécanisme d'exclusion automatique sont les suivantes :

- La table n'est pas autorisée à l'archivage ;
- La requête concerne le calcul de la hiérarchie ;
- Le critère `{ignorer_archivage}` est présent ;
- Un critère sur le champ `est_archive` est utilisé ;
- Le critère `{archive}` ou `{!archive}` est utilisé ;
- Un critère d'égalité ou d'appartenance à une liste sur l'id de la table est présent.

En outre, si le critère conditionnel `{est_archive ?}` est utilisé dans une boucle sans que le contexte ne fournit la valeur, la condition obtenue est `est_archive=0`, c'est-à-dire que les objets archivés restent toujours exclus.

4.1.2 Le critère `{archive}`

Le critère `{archive}` est une expression plus lisible du critère `{est_archive=1}`. Sa négation, `{!archive}` est donc un alias du critère `{est_archive=0}`.

4.1.3 Le critère `{ignorer_archivage}`

Le critère `{ignorer_archivage}` permet, pour la boucle concernée, de lever la restriction sur les objets archivés. Toute restriction explicite ou implicite concernant l'archivage ne sera pas prise en compte.

4.1.4 Le critère `{desarchive}`

Le critère `{desarchive}` permet, pour la boucle concernée et si la consignation du désarchivage est activée, de sélectionner les objets désarchivés. Si la consignation du désarchivage est désactivée, le critère ne retourne aucun objet. Sinon, il correspond à la conjonction des critères `{est_archive=0}` et `{date_archive>0}`.

4.1.5 Le pipeline `post_archivage_objet`

Le pipeline `post_archivage_objet` est appelé à la suite du traitement complet d'une action sur un objet autorisé à l'archivage. Dans le cas où l'objet est un conteneur, la propagation sur les enfants est effectuée avant l'appel du pipeline qui ne concerne donc que l'objet racine initiateur. Le pipeline reçoit le type et l'id de l'objet initiateur, l'identifiant de l'action, la date et l'identifiant du motif si utilisé.

4.1.6 La propagation des actions

Toute action à l'exception de l'explication (modification du motif) est propagée sur les enfants si l'objet concernée est un conteneur et que sa configuration identifie bien les informations nécessaires sur ces enfants. La propagation est réalisée par la fonction `archivage_objet_modifier_enfants()`. Les enfants ne sont modifiables que s'ils respectent des conditions d'archivage liées à l'action déclenchée :

- demande d'archivage : on propage à tous les objets enfants non archivés, y compris ceux pouvant être repérés comme désarchivés ;
- demande de désarchivage ou d'effacement :
 1. si l'objet propageant le désarchivage est la racine d'un archivage, on propage aux descendants de cet objet ;
 2. si l'objet propageant le désarchivage est déjà un descendant d'une autre racine d'archivage R, on propage aux descendants de cet objet ayant pour consignation de racine, la racine R.Les traitements du désarchivage et de l'effacement diffèrent par le fait que le premier s'applique aux enfants archivés alors que le deuxième s'applique aux enfants non archivés.

Si le motif est utilisé et que l'action n'est pas un effacement, les enfants se voient attribués un motif standard qui rappelle la propagation par l'objet racine à l'initiative de l'action.

4.2 L'installation et la configuration

A l'installation du plugin Archivage de Contenus, les champs du contexte d'archivage (voir 5.1) sont ajoutés à chacune des tables d'objets (pipeline `declarer_tables_objets_sql`). Si, par la suite, un autre plugin crée un nouveau type d'objet celui-ci ne bénéficiera pas d'emblée du contexte d'archivage. Néanmoins, le nouveau type étant proposé à l'archivage dans le formulaire de paramétrage du plugin, le choisir provoquera la création des champs du contexte.

En outre, chaque changement de configuration est suivi par une réinitialisation du contexte d'archivage des objets concernés de façon à conserver une cohérence nécessaire à l'application des mécanismes d'archivage.

4.3 Impacts sur la gestion des objets

4.3.1 Le changement de statut d'un objet

Le parti pris par le plugin Archivage de Contenus est de **bloquer le changement de statut** d'un objet archivé. En effet, il paraît improbable voire irréaliste de modifier le statut si l'objet est archivé. Le blocage est réalisé via le pipeline `formulaire_charger` en rendant le formulaire non éditable.

Néanmoins, il pourrait être envisagé dans le futur de rendre ce comportement paramétrable.

4.3.2 La création d'un objet

En suivant la même logique que pour le changement de statut, il serait naturel d'interdire toute création d'un objet au sein d'un objet conteneur archivé : on ne crée pas d'objet dans une poubelle ou dans une boîte d'archive.

Néanmoins, le plugin se devait de rester générique, il doit utiliser un mécanisme standard pour bloquer ce type de création, mécanisme qui n'existe pas actuellement. L'autorisation serait le meilleur moyen mais rien ne permet de savoir qu'elle est l'autorisation à utiliser (aucune information n'est utilisable au travers de la déclaration des objets).

De fait, le **plugin Archivage de Contenus ne fait rien** pour le moment.

4.3.3 La modification d'un objet

En suivant la même logique que pour la création, si ce n'est que la modification concerne tous les objets conteneurs ou pas, il serait aussi pertinent de bloquer toute modification du contenu d'un objet archivé. Mais là encore, il n'y a aucun moyen générique de connaître l'autorisation de modification d'un objet.

De fait, le **plugin Archivage de Contenus ne fait rien** pour le moment, ce qui reste une option toutefois plus acceptable que la création.

4.3.4 Le déplacement d'un objet

Il est possible de déplacer un objet enfant d'un objet conteneur vers un nouvel objet conteneur. Cela est réalisé en modifiant l'identifiant du parent de l'objet enfant.

Dans l'absolu, on pourrait de même que pour les autres actions, interdire par autorisation, le déplacement d'un objet archivé. Mais là encore, on ne connaît pas cette autorisation de façon standard. En outre, que faire du contexte d'archivage de l'objet déplacé sachant que celui-ci a toutes les chances d'avoir changé ?

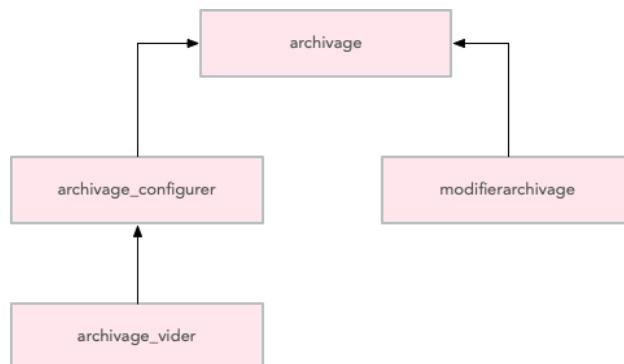
Il est aussi possible de déplacer un objet non archivé dans un conteneur archivé : la question est alors que devient le contexte de l'objet déplacé ?

On voit donc que pour l'action déplacer, la solution est plus complexe et nécessite d'aller au-delà de la seule autorisation en activant des modifications du contexte d'archivage au travers des pipelines `pre_edition` et `post édition`. Mais là encore, les mécanismes de SPIP manquent de générativité.

De fait, le **plugin Archivage de Contenus ne fait rien** pour le moment, ce qui est plutôt problématique.

4.4 Les autorisations

Toutes les actions liées à l'archivage des données sont conditionnées à des autorisations sur les auteurs. Les autorisations sont construites de façon arborescente afin de minimiser les répétitions de code comme l'illustre le diagramme ci-dessous.



Les autorisations sont décrites dans le tableau suivant :

| AUTORISATIONS | |
|-----------------------------|--|
| archivage | <p>Autorisation minimale d'accès à la gestion du plugin Archivage de Contenus. Par défaut, seuls les administrateurs sont autorisés à utiliser le plugin. Cette autorisation est à la base de la plupart des autres autorisations du plugin.</p> <p>Appel : <code>autoriser('archivage')</code></p> |
| archivage_configurer | <p>Autorisation d'accès à la page de configuration du plugin. Il faut être administrateur complet (autorisation par défaut de la configuration)</p> <p>Appel : <code>autoriser('configurer', '_archivage')</code></p> |
| modifierarchivage | <p>Autorisation d'archiver, de désarchiver, de modifier le motif d'archivage ou d'effacer les données d'archivage d'un objet donné :</p> <ul style="list-style-type: none"> ○ l'utilisateur doit être un administrateur ; ○ l'archivage doit être autorisé sur le type d'objet ; ○ l'objet ne doit pas être dans l'état consécutif à l'action demandée ; ○ la configuration doit être cohérente avec l'action d'archivage demandée <p>Appel : <code>autoriser('modifierarchivage', \$type_objet, \$id_objet, null, ['action' => 'archiver'])</code></p> |
| archivage_vider | <p>Autorisation d'effacer toutes les données d'archivage de tous les objets archivables. Il faut avoir l'autorisation de configurer le plugin car cette action n'est accessible que dans la page de configuration.</p> <p>Appel : <code>autoriser('vider', '_archivage')</code></p> |

4.5 L'interface privée

4.5.1 Le suivi des archives

Le suivi des archives est principalement réalisé au travers de la page `?exec=archives`. Cette page regroupe, type d'objet par type d'objet, les listes d'objets archivés. La structure des listes proposées par le plugin est uniforme quel que soit le type d'objet concerné. Néanmoins, il est possible de personnaliser ces listes pour un ou plusieurs types d'objet (voir le paragraphe 6.3).

Liste des contenus archivés

Articles

5 articles

| # | N° | Titre | Depuis le | Racine |
|-------------------------------------|------|-------------------------------|-----------|--------|
| <input type="checkbox"/> | 5535 | Article 748-1 pour archivage | 1er juin | 748 |
| <input checked="" type="checkbox"/> | 5536 | Article 748-2 pour archivage | 1er juin | 748 |
| <input type="checkbox"/> | 5537 | Article 2207-1 pour archivage | 1er juin | 748 |
| <input type="checkbox"/> | 5539 | Article 2 du sousous | 1er juin | 748 |
| <input checked="" type="checkbox"/> | 5538 | Article 1 du sousous | 1er juin | 748 |

Rubriques

3 rubriques

| # | N° | Titre | Depuis le | Racine |
|--------------------------|------|------------------------------|-----------|--|
| <input type="checkbox"/> | 748 | Paquets | 1er juin | <input type="button" value="Désarchiver"/> |
| <input type="checkbox"/> | 2207 | Sous rubrique pour archivage | 1er juin | 748 |
| <input type="checkbox"/> | 2208 | Rubrique sous-sous | 1er juin | 748 |

Si la consignation du désarchivage est activée, la page `?exec=desarchives` affiche selon le même principe la liste des objets désarchivés.

Liste des contenus désarchivés

Articles

1 article

| # | N° | Titre | Depuis le | Racine |
|--------------------------|------|-------|-----------|---|
| <input type="checkbox"/> | 5540 | essai | 2 juin | <input type="button" value="Archiver"/> |

Rubriques

Aucune rubrique

Sites référencés

Aucun site

Ces listes permettent de repérer les objets archivés ou désarchivés comme enfants d'une racine archivée ou désarchivée. La racine est identifiée par son identifiant et une bulle d'information permet d'en connaître le type et le titre exact. La date d'archivage ou de désarchivage est également fournie.

Pour les types d'objet conteneur, des listes identiques sont affichées pour chaque type enfant dans la fiche de l'objet conteneur. Pour les pages liste des objets d'un type donné (par exemple, `?exec=rubriques` et `?exec=articles`), il n'est pas actuellement possible d'afficher la liste des archives. En revanche, les listes d'objets sont, elles, filtrées sur les objets non archivés.

4.5.2 Le paramétrage du plugin

Le paramétrage est réalisé au travers d'un formulaire inclus dans la page `?exec=configurer_archivage`. Le formulaire permet de définir la liste des tables autorisés à l'archivage ainsi que quelques paramètres de fonctionnement dont l'explication est donnée au paragraphe 5.2.

La boite d'information de la page intègre, si l'autorisation de l'utilisateur est suffisante, un bouton qui permet d'effacer toutes les traces d'archivage quel qu'elles soient : l'archivage est ainsi remis à zéro dans tout le site.

4.5.3 Les actions sur les objets

Les actions d'archivage et de désarchivage sont lancées via des boutons intégrés soit dans la boite d'information d'un objet, soit dans une liste d'archives. Les boutons se nomment 'archiver' ou 'désarchiver'.

Quand l'option d'utilisation du motif est activée, un bouton 'Modifier le motif' apparaît dans la boite de notice sur l'archivage incluse dans la fiche de chaque objet. Ce bouton ouvre un formulaire de changement de motif affichant la liste des motifs disponibles pour le type d'objet et l'état d'archivage courant.

5. DONNEES DE ARCHIVAGE DE CONTENUS

5.1 Le contexte d'archivage

Le contexte d'archivage est défini par plusieurs champs valorisés intégrés à chaque table archivable. Les champs sont les suivants :

TABLE ARCHIVABLES : CONTEXTE D'ARCHIVAGE

| | | |
|-----------------------------|--|--------------------------|
| est_archive | Indique si l'objet est archivé (valeur 1) ou non archivé (valeur 0) | 0 |
| date_archive | Précise la date d'archivage ou de désarchivage si la consignation du désarchivage est activée. | '0000-00-00 00:00:00' |
| motif_archive | Identifiant d'un motif d'archivage ou de désarchivage | '' |
| objet_racine_archive | Type de l'objet parent racine ayant propagé son archivage ou son désarchivage | '' |
| id_racine_archive | Identifiant de l'objet parent racine | 0 |

5.2 La configuration du plugin

La configuration du plugin est principalement réalisée par un paramétrage utilisateur stocké dans la meta `archivage`. Les paramètres sont les suivants :

META : ARCHIVAGE

| | | |
|-------------------------------|--|-----|
| tables_autorisees | Liste des tables SQL autorisées à l'archivage. Elles sont choisies parmi la liste des tables SQL archivables | [] |
| modifier_enfant | Indique si l'utilisateur peut modifier l'état d'archivage d'une sous-branche d'une branche archivée. | '' |
| consigner_desarchivage | Indique si le désarchivage d'un objet est consigné. Dans ce cas, l'objet désarchivé possède une date et son état d'archivage est désarchivé et pas hors archive. | '' |
| utiliser_motif | Indique si un motif peut être affecté à chaque archivage (ou désarchivage si celui-ci est consigné) | '' |

Il est aussi possible de réduire la liste des tables archivables en utilisant la constante `_ARCHIVAGE_TABLES_EXCLUES`.

5.3 Les espaces de stockage

Le contexte d'archivage est inclus dans chaque table SQL archivable. Le paramétrage du plugin est lui aussi stocké en base de données dans la variable `archivage` de la table `spip_meta`.

6. UTILISATION DU PLUGIN

6.1 Personnaliser la liste des tables archivables

Par défaut, Archivage de Contenus considère que toutes les tables déclarées au travers du pipeline `declarer_tables_objets_sql` sont éligibles à l'archivage. Néanmoins, certains objets n'ont aucun intérêt de faire partie de cette liste car ils ne seront jamais concernés par l'archivage. C'est le cas, par exemple, des objets plugins, paquets, dépôts de SVP. Il est donc possible d'exclure des tables de la liste, tables qui ne seront jamais présentées dans le formulaire de configuration du plugin :

- soit en utilisant la constante `_ARCHIVAGE_TABLES_EXCLUES`,
- soit en utilisant le pipeline `archivage_exclusion_tables`.

L'exemple ci-dessous illustre l'utilisation de la constante.

```
// Exclusions de l'archivage
define('_ARCHIVAGE_TABLES_EXCLUES', ['spip_depots', 'spip_plugins', 'spip_paquets']);
```

6.2 Personnaliser les motifs

6.2.1 La liste des motifs

Par défaut, le plugin Archivage de Contenus propose deux motifs, l'un pour l'état `archive`, l'autre pour l'état calculé `desarchive`, si utilisé. Ces motifs sont communs à tous les types d'objet.

Un plugin utilisateur peut déclarer de multiples motifs pour chaque état mais aussi pour chaque type s'il souhaite différencier les motifs par type d'objet. Pour cela, il doit utiliser le pipeline `archivage_liste_motifs` dont les arguments sont le type d'objet et l'état calculé.

L'exemple suivant illustre l'utilisation du pipeline : on définit une identifiant de motif pour chaque état et chaque type d'objet.

```
function galactic_contrib_archivage_liste_motifs(array $flux) : array {
    // On rajoute des motifs pour Contrib
    $etat = $flux['args']['etat'];
    $objet = $flux['args']['objet'];
    $motifs_contrib = [
        "{$etat}_{$objet}_obsolescence"
    ];
    $flux['data'] = array_merge($flux['data'], $motifs_contrib);

    return $flux;
}
```

Pour traduire chaque motif, il est ensuite nécessaire de définir chaque item de langue dans un module `archivage`, un item de langue étant toujours de la forme `motif_{$identifiant_motif}`.

```
'motif_archive_article_obsolescence_label' => 'article archivé car obsolement',
'motif_archive_rubrique_obsolescence_label' => 'rubrique archivée car obsolement',
'motif_desarchive_article_obsolescence_label' => 'article désarchivé suite à une erreur d\'archivage',
'motif_desarchive_rubrique_obsolescence_label' => 'rubrique désarchivée suite à une erreur d\'archivage',
```

6.2.2 Les motifs par défaut

Le plugin Archivage de Contenus utilise les deux motifs qu'il a défini comme motif par défaut pour chaque état quel que soit le type d'objet.

Un plugin utilisateur peut déclarer, parmi ses listes de motifs, ceux qui doivent être utilisés par défaut. Pour cela il doit utiliser le pipeline `archivage_motif_defaut` dont les arguments sont le type d'objet et l'état calculé.

L'exemple suivant illustre l'utilisation du pipeline.

```
function galactic_contrib_archivage_defaut_motif(array $flux) : array {
    // On rajoute des motifs pour Contrib
    $etat = $flux['args']['etat'];
    $objet = $flux['args']['objet'];
    $flux['data'] = "{$etat}_{$objet}_obsolescence";

    return $flux;
}
```

6.3 Personnaliser les listes d'archives

Pour éviter de multiplier les squelettes de liste en fonction du type d'objet, le plugin Archivage de Contenus propose une liste générique `prive/objets/liste/archives.html`. Celle-ci est motorisée par une boucle DATA sur le tableau retourné par la balise `#ARCHIVAGE_LISTE` et permet d'afficher les objets archivés mais aussi les objets désarchivés quand la consignation du désarchivage est activée.

Néanmoins, il est possible de surcharger cette liste pour un type d'objet donné en créant un squelette `prive/objets/liste/{$objets}_archives.html`. Par exemple, pour les articles, il faudrait créer une liste nommée `articles_archives.html`.

Cette liste possède en entrée un contexte avec au minimum les éléments suivants :

- l'état d'archivage sous la forme canonique (`est_archive`) et calculé (`archive, desarchive`) ;
- un critère `where` dans le cas des objets désarchivés ;
- le type d'objet ;
- le titre éventuel si aucun objet n'est répertorié ;
- le tri, par défaut, par date d'archive.

La boucle doit posséder des critères conditionnels indispensables, à savoir, `{est_archive ?}` et `{where ?}` comme illustré dans l'exemple suivant pour les articles :

```
<BOUCLE_liste_art(ARTICLES)
{est_archive?}
{where?}
{id_?}
{statut?}
{recherche?}
{tri #ENV{par, date_archive}, #GET{defaut_tri}}{par titre}{pagination #ENV{nb,20}}
{!lang_select}>
```

7. REGLES DE CODAGE

7.1 Nommage des fonctions

Le nommage des fonctions appartenant à l'API du plugin suit des règles strictes qui simplifient l'identification de l'objet et de l'action appliquée. Le nom de chaque fonction est donc composé ainsi : `archivage_<sujet>_<verbe_infinitif>[_<complement>]`.

Le sujet peut prendre les valeurs objet, motif et table. Par exemple, la fonction des données d'archivage d'un objet donné se nomme `archivage_objet_lire()` et la fonction de traduction d'un motif se nomme `archivage_motif_traduire()`.

7.2 Arguments et variables standardisés

Les arguments ou variables locales dépendent de chaque fonction mais leur nommage est toujours le même d'une fonction à une autre.

Par exemple, l'argument ou la variable `$objet` désigne toujours le type d'objet et `$id_objet` son identifiant.

L'argument ou la variable `$action` désigne le type d'action d'archivage, soit 'archiver', 'desarchiver', 'effacer' ou 'expliquer'.

L'argument `$table_sql` ou `$table` désigne toujours le nom de la table SQL de l'objet, comme 'spip_articles' ou 'spip_rubriques'. Les variables commençant par `$table` désignent toujours une table SQL ou un tableau de tables SQL.

Le préfixe `$type_` est en général utilisé pour désigner un type d'objet comme `$type_parent` ou `$type_objet`.

7.3 Typage des arguments

Toutes les fonctions du plugin ont un prototype dont les arguments et le retour sont typés.