05 - Migration de données uPortal 3.2 vers uPortal 4.0

Cette page est destinée aux intégrateurs.

Références

- Documentation officielle pour la migration vers la version 4 d'uPortal : https://wiki.jasig.org/display/UPM40/Upgrading

Outils

- Script Shell permettant la mise en conformité des channel-type pour une migration d'uPortal 3.2 vers 4.0: Migration-channel-type.sh
- Script Shell permettant la mise en conformité des fragment-definition pour une migration d'uPortal 3.2 vers 4.0: Migration-fragment-definition.sh

Contexte

Dans le cadre de la maintenance et de l'évolution de leur portail, le consortium ESUP a souhaité s'intéresser davantage quant à la migration de données d' un portail uPortal contenues dans une instance de base de données en version 3.2 vers une instance de base en version 4.0. Ce processus n'étant pas totalement défini et validé par Jasig, il est important d'en définir ses limites ainsi que la liste exhaustive des données traitées.

La description de la procédure générique décrite ci-dessous a été testée et validée sur l'environnement suivant :

- Système d'exploitation du serveur : Windows XP
- Serveur d'application : Tomcat 6.0.32
- Serveur de base de données des bases uPortal : PostgreSQL 9.0.4
- Migration réalisée depuis une version 3.2.2 d'uPortal vers la version 4 présente sur le repository Git d'ESUP (https://github.com/EsupPortail/esup-uportal)
- Les différents outils développés ont également été testés sur un environnement Unix afin d'en valider leur bon fonctionnement.

Prérequis

Comme indiqué sur la page de Jasig, le processus de migration nécessite une installation d'une version 4.0 servant de socle pour l'import des données exportées depuis le portail en version 3.2.

A l'heure actuelle, il semble en effet impossible de réaliser ce genre de migration avec la démarche inverse, consistant à installer une version 4.0 du portail sur une instance de base de données d'une version 3.2.

Il est donc nécessaire pour réaliser cette migration d'installer la version 4.0 cible en créant :

- une nouvelle base de données configurée avec le même utilisateur que la version 3.2.
- une nouvelle instance du serveur d'application qui « contiendra » l'application uPortal en version 4.0.

Export des données de la version 3.2

Après installation de la version 4.0, il est nécessaire de réaliser l'export des données de la base de la version 3.2. Ceci se fait via la commande Ant suivante:

```
ant.sh db.export -Ddir="PATH/TO/DIRECTORY/WHERE/DATA/WILL/BE/SAVED" -Dtype=all
```

, avec :

- « *PATH/TO/DIRECTORY/WHERE/DATA/WILL/BE/SAVED* » correspondant au répertoire dans lequel les données exportées seront sauvegardées.
- « *-Dtype=all* » correspondant aux éléments qui seront exportés. Dans ce cas précis, voici la liste des éléments exportés :
    - all-layouts
    - all-profiles
    - all-permission_sets
    - all-channels
    - all-channel-types
    - all-users
    - all-themes
    - all-structures
    - all-entity-types
    - all-group_memberships
    - all-fragment-definitions

Après export de ces éléments, les répertoires suivants devraient être créés dans le répertoire d'export:

- channel
- channel-type
- entity-type
- fragment-layout
- group_membership
- layout
- permission_set
- profile
- structure
- theme
- user

Remarque : La liste des "entity-type" pouvant être exportés depuis la commande d'export et plus précisément via les arguments : "-Dtype" et "-DsysId" (optionnel) est décrite sur la page suivante: https://wiki.jasig.org/display/UPM32/Import+Export+Data+Migration+Tools du manuel Jasig uPortal (Cf. paragraphe "Table of Entity Types")..

Après cette procédure d'export, il est fortement recommandé de vérifier les traces de la console afin de s'assurer que la procédure s'est déroulée correctement. Si celle-ci a échoué le message "BUILD FAILED" sera explicité en fin de la "sortie Console", et si au contraire l'export s'est déroulé avec succès alors le message "BUILD SUCCESSFUL" sera affiché à la fin des traces écrites dans la console.

Il est également recommandé, une fois l'export terminé, de copier l'ensemble du répertoire d'export obtenu vers un répertoire de travail qui fera l'objet de différentes modifications nécessaires à la procédure de migration vers une version 4.0.

Import des données dans la version 4.0

Import des « channel-type »

Afin de réaliser un import de données cohérent, il est nécessaire de ne conserver dans le répertoire de travail que les fichiers « .channel-type » personnalisés, c'est-à-dire tout ceux qui ne sont pas natifs à une version 3.2 non-personnalisée. Tous les autres fichiers peuvent alors être supprimés.

Afin de réaliser cette opération avec précaution, nous proposons d'appliquer une méthode de comparaison entre le répertoire « channel-type » exporté depuis la version 3.2 « personnalisée » et le répertoire « channel-type » situé dans « *UPORTAL_ROOT/uportal-impl/src/main/resources/properties/db /entities* » d'un portail en version 3.2 et non-personnalisé.

Transformation des « channel-type »

Pour les « channel-type » restant, c'est-à-dire ceux à importer en version 4.0, il est alors nécessaire de les transformer en fichiers « .portlet-type.xml » tout en modifiant la définition contenu dans chacun des fichiers (Cf. https://wiki.jasig.org/display/UPM40/Upgrade+Data+Import).

Pour cela un script Shell Unix : « ***Migration-channel-type.sh*** » utilisant SED a été développé. Celui-ci permet de :

- prendre en argument le répertoire contenant tous les « channel-type » à transformer.
  Exemple de commande en résultant:

  ```
  Migration-channel-type.sh « ./V3-EXPORT/channel-type »
  ```

- ajouter l'entête de définition XML dans le fichier.
- modifier la balise ouvrante :

```
<channel-type script="classpath://org/jasig/portal/io/import-channel-type_v2-6.crn">
```

par la balise:

```
<portlet-type xmlns="https://source.jasig.org/schemas/uportal/io/portlet-type" xmlns:ns2="https://source.
jasig.org/schemas/uportal/io/permission-owner" xmlns:ns3="https://source.jasig.org/schemas/uportal/io
/stylesheet-descriptor" xmlns:ns4="https://source.jasig.org/schemas/uportal/io/portlet-definition" xmlns:
ns5="https://source.jasig.org/schemas/uportal" xmlns:ns6="https://source.jasig.org/schemas/uportal/io
/user" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" version="4.0" xsi:schemaLocation="
https://source.jasig.org/schemas/uportal/io/portlet-type https://source.jasig.org/schemas/uportal/io
/portlet-type/portlet-type-4.0.xsd">
```

- modifier la balise fermante : *</channel-type>* par *</portlet-type>*
- modifier les balises : *<desc></desc>* par les balises *<description></description>*
- supprimer la ligne contenant la balise *<type>* qui n'est plus nécessaire pour les « portlet-type »
- modifier l'extension du fichier *« .channel-type »* par l'extension *« .portlet-type.xml »* utilisée en version 4.0

Exemple pour le channel-type « Bookmarks_Portlet.channel-type »

Fichier *« Bookmarks_Portlet.channel-type »* avant transformation :

```
<channel-type script="classpath://org/jasig/portal/io/import-channel-type_v2-6.crn">
          <name>BookmarksPortlet</name>
        <type>org.jasig.portal.channels.portlet.CSpringPortletAdaptor</type>
        <desc>UWisc Bookmarks Portlet</desc>
        <uri>/edu/wisc/my/portlets/bookmarks/BookmarksPortlet.cpd</uri>
</channel-type>
```

Fichier *« Bookmarks_Portlet.portlet-type.xml »* résultant de la transformation:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<portlet-type xmlns="https://source.jasig.org/schemas/uportal/io/portlet-type" xmlns:ns2="https://source.jasig.
org/schemas/uportal/io/permission-owner" xmlns:ns3="https://source.jasig.org/schemas/uportal/io/stylesheet-
descriptor" xmlns:ns4="https://source.jasig.org/schemas/uportal/io/portlet-definition" xmlns:ns5="
https://source.jasig.org/schemas/uportal" xmlns:ns6="https://source.jasig.org/schemas/uportal/io/user" xmlns:
xsi="http://www.w3.org/2001/XMLSchema-instance" version="4.0" xsi:schemaLocation="https://source.jasig.org
/schemas/uportal/io/portlet-type https://source.jasig.org/schemas/uportal/io/portlet-type/portlet-type-4.0.xsd">
        <name>Bookmarks Portlet</name>
        <description>UWisc Bookmarks Portlet</description>
        <uri>/org/jasig/portal/portlets/bookmarks/BookmarksPortlet.cpd.xml</uri>
</portlet-type>
```

Import des « channel-type » dans uPortal v4.0

Après transformation des fichiers « channel-type » l'import peut être réalisé dans le portail en version 4.0 via la commande :

```
ant data-import -Ddir="./V3-EXPORT/channel-type"
```

, avec Ddir ayant pour valeur le chemin vers le répertoire contenant les « channel-type » exportés et transformés.

Si l'import s'est déroulé avec succès, un message « Build successful » devrait alors être affiché dans la console. Dans le cas échéant, un message « Build failed » sera alors affiché et les traces détaillant les différentes erreurs rencontrées seront alors visibles dans le répertoire « *UPORTAL_ROOT/target/data-import-reports* » du portail en version 4.0.

Import des « entity-type »

Dans le but de réaliser un import de données cohérent, il est nécessaire pour les « entity-type » de ne conserver seulement ceux qui sont personnalisés et qui ont été définis « manuellement » dans le portail 3.2 servant de socle de migration. Ainsi, identiquement aux « channel-type », tous les « entity-type » inclus nativement dans la version 3.2 pourront être supprimés du répertoire de travail utilisé pour l'import des données en base en version 4.0.

Afin de réaliser cette tâche de manière rigoureuse, il est également conseillé d'effectuer une comparaison entre le répertoire « entity-type » obtenu depuis l'export et les répertoires « entity-type » du portail en version 3.2 situés dans *« UPORTAL_ROOT/uportal-impl/src/main/resources/properties/db/entities /entity-type »* et *« UPORTAL_ROOT/uportal-impl/src/main/resources/properties/db/base_entities/entity-type »*.

Les « entity-type » personnalisés peuvent alors être importés dans le portail v4.0 via la commande Ant suivante :

```
ant data-import -Ddir="./V3-EXPORT/entity-type"
```

, avec Ddir ayant pour valeur le chemin vers le répertoire contenant les « entity-type » exportés.

Si l'import s'est déroulé avec succès, un message « Build successful » devrait alors être affiché dans la console. Dans le cas échéant, un message « Build failed » sera alors affiché et les traces détaillant les différentes erreurs rencontrées seront alors visibles dans le répertoire « *UPORTAL_ROOT/target/data-import-reports* » du portail en version 4.0.

Import des « user »

Pour effectuer l'import dans le portail v4.0 des utilisateurs depuis les données exportées en v3.x, il suffit juste de jouer la commande Ant ci-dessous :

```
ant data-import -Ddir="./V3-EXPORT/user"
```

, avec Ddir ayant pour valeur le chemin vers le répertoire contenant les « user » exportés depuis la version 3.2.

Si l'import s'est déroulé avec succès, un message « Build successful » devrait alors être affiché dans la console. Dans le cas échéant, un message « Build failed » sera alors affiché et les traces détaillant les différentes erreurs rencontrées seront alors visibles dans le répertoire « UPORTAL_ROOT/target/data-import-reports » du portail en version 4.0.

Import des « group-membership »

Afin de pouvoir importer dans la version 4.0 les groupes exportés depuis la version 3.2 il est nécessaire de réaliser deux modifications majeures dans la configuration d'uPortal v4.0.

Modification de la configuration de la version 4.0

Les modifications permettant la mise en conformité des groupes concernent deux fichiers XML :

- Le premier : « PAGSGroupStoreConfig.xml », situé dans « *UPORTAL_ROOT/ uportal-war/src/main/resources/properties/groups* », permet de configurer la gestion des attributs des utilisateurs appartenant aux différents groupes.
- Le deuxième : « personDirectoryContext.xml », situé dans le répertoire « *UPORTAL_ROOT/ uportal-war/src/main/resources/properties/contexts* », permet quant à lui de définir les attributs des utilisateurs en fonction de leur source respective (utilisateur enregistré en base, utilisateur LDAP etc…)

Concernant le premier fichier « PAGSGroupStoreConfig.xml », le travail consiste à ajouter dans le fichier l'ensemble des groupes personnalisés (groupes LDAP entre autres) définis dans le fichier « PAGSGroupStoreConfig.xml » de la version 3.2. Il est donc juste nécessaire de réaliser une comparaison entre les deux fichiers afin d'inclure dans le fichier de la version 4.0 tous les groupes ajoutés dans le fichier source de la version 3.2

Concernant le fichier « personDirectoryContext.xml », il est nécessaire également de réaliser une comparaison du fichier avec celui de la version 3.2. Dans le cas présent, il suffit de modifier le bean *« uPortalLdapAttributeSource »* pour y configurer la gestion des attributs des utilisateurs synchronisés avec le LDAP.

Voici un exemple de bean définissant les attributs LDAP utilisés pour les utilisateurs synchronisés avec un LDAP :

```
    <bean id="uPortalLdapAttributeSource" class="org.jasig.services.persondir.support.ldap.
LdapPersonAttributeDao">
        <property name="contextSource" ref="defaultLdapContext" />
        <property name="queryAttributeMapping">
            <map>
                <entry key="username" value="${ldap.uidAttr}"/>
            </map>
        </property>

        <property name="resultAttributeMapping">
            <map>
                <entry key="eduPersonPrimaryAffiliation">   <value>eduPersonPrimaryAffiliation</value><
/entry>
                <entry key="eduPersonAffiliation">          <value>eduPersonAffiliation</value><
/entry>
                <entry key="cn">                            <value>cn</value></entry>
                <entry key="description">                   <value>description</value></entry>
                <entry key="displayName">                   <value>displayName</value></entry>
                <entry key="facsimileTelephoneNumber">      <value>facsimileTelephoneNumber</value></entry>
                <entry key="givenName">                     <value>givenName</value></entry>
                <entry key="mail">                          <value>mail</value></entry>
                <entry key="postalAddress">                 <value>postalAddress</value></entry>
                <entry key="sn">                            <value>sn</value></entry>
                <entry key="telephoneNumber">               <value>telephoneNumber</value></entry>
                <entry key="${ldap.uidAttr}">
                    <set>
                        <value>${ldap.uidAttr}</value>
                        <value>username</value>
                        <value>user.login.id</value>
                    </set>
                </entry>
                <entry key="supannCodeINE">                 <value>supannCodeINE</value></entry>
                <entry key="supannEtuId">                   <value>supannEtuId</value></entry>
                <entry key="supannEmpId">                   <value>supannEmpId</value></entry>
                <entry key="eduPersonAffiliation">          <value>eduPersonAffiliation</value></entry>
                <entry key="supannaffectation">             <value>supannaffectation</value></entry>
                <entry key="objectclass">                   <value>objectclass</value></entry>
                <entry key="supannorganisme">               <value>supannorganisme</value></entry>
            </map>
        </property>
    </bean>
```

Une fois ces modifications effectuées, il est recommandé de redéployer et redémarrer l'application en version 4.0 afin de s'assurer que les modifications apportées n'ont pas affecté son bon fonctionnement.

Import des données

Suite à la vérification précédente, il suffit d'importer les données issues de l'export de la version 3.2 via la commande Ant suivante :

```
ant data-import -Ddir="./V3-EXPORT/group_membership"
```

, avec Ddir ayant pour valeur le chemin vers le répertoire « group_membership » contenant les données exportées depuis la version 3.2.

Import des « channel »

NOTE : Si la servlet ExternalURLStats était utilisée dans votre ancien portail, les URL des channels de type InlineFrame seront très probablement de cette forme : "*ExternalURLStats?fname=myService&service=https://my.univ.fr/iframe*". Il est donc nécessaire de modifier cette valeur puisque la servlet ExternalURLStats n'est pas transposée dans cette v4. Pour cela, il suffit de lancer la commande suivante avant d'effectuer l'import : *sed -i "s /ExternalURLStats.*service=//g" ./V3-EXPORT/channel/*channel*

Afin de réaliser l'import des « channels », il est seulement nécessaire d'importer les fichiers exportés correspondants dans la version 4.0 via la commande Ant :

```
ant data-import -Ddir="./V3-EXPORT/channel"
```

, avec Ddir ayant pour valeur le chemin vers le répertoire contenant les « channel » exportés.

Si l'import s'est déroulé avec succès, un message « Build successful » devrait alors être affiché dans la console. Dans le cas échéant, un message « Build failed » sera alors affiché et les traces détaillant les différentes erreurs rencontrées seront alors visibles dans le répertoire « *UPORTAL_ROOT/target/data-import-reports* » du portail en version 4.0.

ATTENTION : Les channels correspondants à des IChannel de la version 3.x n'existent plus dans les versions 4.0. De ce fait, ceux-ci ne seront pas importés dans la nouvelle version et le message d'avertissement suivant sera affiché pour chacun d'eux dans la console durant la phase d'import :

<channel-xxx> is not a portlet. It was likely an IChannel from a previous version of uPortal and will not be imported.

Ce message n'est cependant pas bloquant et les channel restants seront tout de même traités par la procédure d'import.

Import des « channel » et problème de Duplicate Key

Il se peut que durant la phase d'import des channels, certaines définitions de portlets ne puissent être importées à cause d'un problème de "Duplicate Key". Ceci signifie que les champs *fname* et *name* de la portlet à importer existent déjà dans la table de la base de données et ceci lève donc une exception suite à une contrainte d'unicité définie sur ces champs là. Une procédure manuelle a été cependant élaborée afin de palier à cette anomalie : elle consiste à réaliser une comparaison entre les deux fichiers causant l'anomalie (fichier exporté et fichier de la nouvelle version) pour en cibler les principales modifications :

- si les fichiers sont identiques (ou presque : champ "*iconURL*" seulement qui diffère)
    - supprimer le fichier du dossier d'export pour utiliser celui déjà existant en base car sa définition est plus fidèle à la version 4.0 d'uPortal.
- sinon : renommer la zone causant l'anomalie.

Voici un exemple illustrant cette procédure de correction :

Anomalie tracée dans les logs : "portlet recherche : "name" identique -> (portlet_name)=(Recherche) already exists."

Or les deux portlets sont différentes : l'une permet de faire la recherche sur Google alors que l'ancienne utilise une application de recherche spécifique.

La solution dans ce cas consiste à renommer le champ *"name"* de l'ancienne portlet afin que celui-ci soit unique en base. Nous proposons dans ce cas de la renommer de façon plus explicite en fonction de la fonctionnalité liée à la portlet, par exemple en la renommant : *"Recherche de contenu"* à la place de *" Recherche"*.

Import des « fragment-layout »

Identiquement aux « channel », la procédure d'import des « fragment-layout » est relativement basique. Celle-ci se fait via la commande Ant :

```
ant data-import -Ddir="./V3-EXPORT/fragment-layout"
```

, avec Ddir ayant pour valeur le chemin vers le répertoire contenant les « fragment-layout » exportés.

Si l'import s'est déroulé avec succès, un message « Build successful » devrait alors être affiché dans la console. Dans le cas échéant, un message « Build failed » sera alors affiché et les traces détaillant les différentes erreurs rencontrées seront alors visibles dans le répertoire « UPORTAL_ROOT/target/data-import-reports » du portail en version 4.0.

ATTENTION : Si des IChannel sont utilisés dans certains « fragment-layout » alors ces fichiers ne pourront être importés dans la version 4.0. Deux cas sont alors envisageables :

- Si ces IChannels ont été développés sous forme de nouvelles portlets alors il est nécessaire de modifier les fichiers « fragment-layout » associés afin de remplacer les IChannel dans les balises <folder ID…> par les portlets associées.
- Dans le cas contraire, si l'IChannel causant l'anomalie n'a pas été portée sous forme de portlet, il est alors nécessaire de :
    - Supprimer le bloc *<channel>* correspondant au IChannel à intégrer dans le layout *si et seulement si* plusieurs channels sont définis dans ce même bloc *<folder ID...>.*
    - Supprimer la totalité du bloc *<folder ID...>* si le channel correspondant au IChannel est le seul intégré dans le bloc *<folder ID...>* englobant.

Les « fragments-layout » corrigés peuvent alors être importés de façon unitaire avec la commande Ant :

```
ant data-import -Dfile="./V3-EXPORT/fragment-layout/IChannel1.fragment-layout"
```

, avec « IChannel1.fragment-layout » correspondant au fichier « fragment-layout » contenant un ou plusieurs IChannel et n'ayant donc pu être importé lors de la première tentative d'import.

Import des « permission_set »

L'import des « permission_set » ne devrait nécessiter aucune étape supplémentaire de traitement ou modification. Il suffit juste d'importer tout le contenu du répertoire dans la version 4.0 via la commande Ant :

```
ant data-import -Ddir="./V3-EXPORT/permission_set"
```

, avec Ddir ayant pour valeur le chemin vers le répertoire contenant les « permission_set » exportés.

Si l'import s'est déroulé avec succès, un message « Build successful » devrait alors être affiché dans la console. Dans le cas échéant, un message « Build failed » sera alors affiché et les traces détaillant les différentes erreurs rencontrées seront alors visibles dans le répertoire « *UPORTAL_ROOT/target/data-import-reports* » du portail en version 4.0.

Migration des « fragment-definition »

A partir de la version 4.0.4 d'uPortal, la définition des fragments a été revue. En effet, désormais celle-ci n'est plus gérée par défaut via le fichier « dlm.xml » situé dans le répertoire « UPORTAL_ROOT/uportal-war/src/main/resources/properties » mais directement en base de données via un bean Java associé.

Afin que ces « fragment-definition » exportés depuis la version 3.2 soient bien intégrés lors de la migration vers la 4.0 deux possibilités sont proposées par Jasig :

- la première solution consiste à modifier le comportement d'uPortal v4.0 afin de revenir à une gestion des définitions de fragment via le fichier « dlm.xml ».
- la deuxième consiste à laisser la gestion des « fragment-definition » en base des données et à créer les fichiers « fragment-definition » associés à ceux de la version 3.2.

Import des « fragment-definition » via dlm.xml

Comme indiqué précédemment, depuis la version 4.0.4 d'uPortal la gestion des fragments est persistée en base et n'est plus gérée via le fichier « dlm.xml » tel que c'était le cas auparavant. Cependant Jasig permet toujours dans ces dernières versions de basculer la gestion des fragments via le fichier « dlm.xml » afin de faciliter notamment les procédures de migration telles que celle-ci. Pour effectuer cette manipulation deux actions sont alors à effectuer :

- La première consiste à modifier le fichier « layoutContext.xml » (situé dans « UPORTAL_ROOT/uportal-war/src/main/resources/properties /contexts ») comme suit :
  - commenter le bean lié à la classe « RDBMConfigurationLoader » :

```
<!-- Mise en commentaire du bean lié à la gestion des fragments en base
<bean id="dlmConfigurationLoader" class="org.jasig.portal.layout.dlm.RDBMConfigurationLoader">
        <property name="fragmentDao" ref="fragmentDefinitionDao" />
    </bean>
-->
```

  - Il faut alors décommenter dans ce même fichier le bean lié à la classe « LegacyConfigurationLoader » permettant de gérer les définitions de fragments via le fichier « dlm.xml » :

```
<bean id="dlmConfigurationLoader" class="org.jasig.portal.layout.dlm.LegacyConfigurationLoader">
        <property name="configurationFile" value="classpath:/properties/dlm.xml" />
</bean>
```

  - Désormais le portail utilisera la gestion des « fragment-definition » via le fichier « dlm.xml » qu'il est nécessaire de récupérer depuis la version 3.2.
- Afin d'utiliser dans la version 4.0 du portail les « fragment-definition » définis dans la version 3.2, il est alors nécessaire de copier le fichier « dlm.xml » situé dans le répertoire « *UPORTAL_ROOT/uportal-impl/src/main/resources/properties* » du portail 3.2 vers le répertoire « *UPORTAL_ROOT/uportal-war/src/main/resources/properties* » du portail 4.0.

L'application est désormais configurée pour utiliser convenablement les « fragment-definition » de la version 3.2. Il est juste nécessaire de redéployer uPortal via la commande Ant suivante afin que la modification soit répercutée sur le répertoire « uPortal » placé dans le serveur d'application :

```
ant clean deploy-war -Dmaven.test.skip=true
```

Import des « fragment-definition » via base de données

La deuxième méthode d'import des « fragment-definition » permet quant à elle de conserver la gestion des fragments en base de données telle qu'elle a été mise en place à partir de la version 4.0.4 d'uPortal.

Il est nécessaire pour cela d'importer chaque « fragment-definition » associé en base afin que l'ensemble de celles-ci soient disponibles depuis l' application en version 4.0. En voici la procédure à suivre :

- Edition du fichier « *dlm.xml* » de la version 3.2 situé dans le répertoire « UPORTAL_ROOT/uportal-impl/src/main/resources/properties ».
- Copie de chaque bloc (élément) <*dlm:fragment*> dans un nouveau fichier ayant pour extension « .fragment-definition.xml » et ayant pour nom le nom correspondant au fragment défini.
- Ajout d'un bloc englobant <*fragment-definition*></*fragment-definition*> autour de l'élément XML : <*dlm:fragment*>.

- Sauvegarde du nouveau fichier obtenu : « *xxx.fragment-definition.xml* ».
- Import de ces nouveaux fichiers « *.fragment-definition.xml* » via la commande Ant :

```
ant data-import -Ddir="path/to/fragment-def.dir"
```

, avec ***« path/to/fragment-def.dir »*** correspondant au répertoire contenant l'ensemble des nouveaux fichiers « .fragment-definition.xml »

Afin d'automatiser ces différentes tâches un script Shell Unix : « ***Migration-fragment-definition.sh*** » a été développé. Celui-ci utilise AWK afin de « découper » chaque fragment contenu dans le fichier *dlm.xml* vers un nouveau fichier ayant pour convention de nommage : « *fragmentDLMXXX.fragment-definition.xml* » où XXX correspondant au numéro du fragment traité.

Pour lancer ce script, il suffit alors de jouer la commande suivante dans la console :

```
Migration-fragment-definition.sh « ./PATH/TO/3.2/DLM/FILE/dlm.xml »
```

, avec *« ./PATH/TO/3.2/DLM/FILE/dlm.xml »* correspondant au chemin versle fichier dlm.xml de la version 3.2.

Autant de fichiers « .fragment-definition.xml » que de fragments définis dans le fichier « dlm.xml » seront alors créés dans le répertoire où est situé le script. Il ne suffit donc plus qu'à importer chacun de ces fichiers via la commande d'import Ant suivante :

```
ant data-import -Dfile="path/to/fragmentDLMXXX.fragment-definition.xml"
```

Voici un exemple décrivant les tâches effectuée par ce script pour le premier fragment défini dans le fichier « dlm.xml » de la version 3.2 :

- La première étape consiste en la lecture du fichier « dlm.xml » :

```
<?xml version="1.0"?>
<managedLayoutFragments xmlns:dlm="http://org.jasig.portal.layout.dlm.config">

  <dlm:property name='defaultLayoutOwner' value='fragmentTemplate'/>

  <!-- Controls clearing of dlm fragment cache.  This allows changes  made to layout
  owners to be reflected once the cache has been updated.  Specified in minutes. -->

  <dlm:property name='org.jasig.portal.layout.dlm.RDBMDistributedLayoutStore.fragment_cache_refresh'
value="5"/>

    <dlm:fragment name='Admin' ownerID='admin-lo' precedence='3'>
        <dlm:audience evaluatorFactory='org.jasig.portal.layout.dlm.providers.PersonEvaluatorFactory'>
          <paren mode="AND">
            <attribute name="username" mode='equals' value='admin'/>
          </paren>
        </dlm:audience>
    </dlm:fragment>

    <dlm:fragment name='Guests' ownerID='guest-lo' precedence='3'>
        <dlm:audience evaluatorFactory='org.jasig.portal.layout.dlm.providers.GuestUserEvaluatorFactory'
/>
    </dlm:fragment>

    <dlm:fragment name='All' ownerID='all-lo' precedence='2'>
      <dlm:audience evaluatorFactory='org.jasig.portal.layout.dlm.providers.
GroupMembershipEvaluatorFactory'>
        <paren mode="AND">
          <attribute mode='memberOf' name='Tout le monde'/>
          <paren mode="NOT">
            <attribute mode='memberOf' name='Proprietaires de fragment'/>
          </paren>
          <paren mode="NOT">
            <attribute mode='memberOf' name='Administrateurs Portail'/>
          </paren>
          <paren mode="NOT">
            <attribute mode='memberOf' name='Anonymes'/>
          </paren>

        </paren>
      </dlm:audience>
```

```
    </dlm:fragment>

    <dlm:fragment name='AdministrateurCentral' ownerID='administrateurCentral-lo' precedence='10'>
        <dlm:audience evaluatorFactory='org.jasig.portal.layout.dlm.providers.
GroupMembershipEvaluatorFactory'>
          <paren mode="AND">
            <attribute mode='deepMemberOf' name='Administrateurs Centraux'/>
          </paren>
        </dlm:audience>
    </dlm:fragment>

    <dlm:fragment name='AgentTechnique' ownerID='agentTechnique-lo' precedence='10'>
        <dlm:audience evaluatorFactory='org.jasig.portal.layout.dlm.providers.
GroupMembershipEvaluatorFactory'>
          <paren mode="AND">
            <attribute mode='deepMemberOf' name='Agents techniques'/>
          </paren>
        </dlm:audience>
    </dlm:fragment>

    <dlm:fragment name='AssistantEducation' ownerID='assistantEducation-lo' precedence='10'>
        <dlm:audience evaluatorFactory='org.jasig.portal.layout.dlm.providers.
GroupMembershipEvaluatorFactory'>
          <paren mode="AND">
            <attribute mode='deepMemberOf' name='Assistants education'/>
          </paren>
        </dlm:audience>
    </dlm:fragment>

    <dlm:fragment name='ChefTravaux' ownerID='chefTravaux-lo' precedence='10'>
        <dlm:audience evaluatorFactory='org.jasig.portal.layout.dlm.providers.
GroupMembershipEvaluatorFactory'>
          <paren mode="AND">
            <attribute mode='deepMemberOf' name='Chefs de travaux'/>
          </paren>
        </dlm:audience>
    </dlm:fragment>

    <dlm:fragment name='ConseillerEducation' ownerID='conseillerEducation-lo' precedence='10'>
        <dlm:audience evaluatorFactory='org.jasig.portal.layout.dlm.providers.
GroupMembershipEvaluatorFactory'>
          <paren mode="AND">
            <attribute mode='deepMemberOf' name='Conseillers education'/>
          </paren>
        </dlm:audience>
    </dlm:fragment>

    <dlm:fragment name='ConseillerOrientation' ownerID='conseillerOrientation-lo' precedence='10'>
        <dlm:audience evaluatorFactory='org.jasig.portal.layout.dlm.providers.
GroupMembershipEvaluatorFactory'>
          <paren mode="AND">
            <attribute mode='deepMemberOf' name='Conseillers orientation'/>
          </paren>
        </dlm:audience>
    </dlm:fragment>

    <dlm:fragment name='Directeur' ownerID='directeur-lo' precedence='10'>
        <dlm:audience evaluatorFactory='org.jasig.portal.layout.dlm.providers.
GroupMembershipEvaluatorFactory'>
          <paren mode="AND">
            <attribute mode='deepMemberOf' name='Directeurs'/>
          </paren>
        </dlm:audience>
    </dlm:fragment>

    <dlm:fragment name='Documentaliste' ownerID='documentaliste-lo' precedence='10'>
        <dlm:audience evaluatorFactory='org.jasig.portal.layout.dlm.providers.
GroupMembershipEvaluatorFactory'>
          <paren mode="AND">
            <attribute mode='deepMemberOf' name='Documentalistes'/>
          </paren>
```

```
        </dlm:audience>
    </dlm:fragment>

    <dlm:fragment name='Eleve' ownerID='eleve-lo' precedence='10'>
        <dlm:audience evaluatorFactory='org.jasig.portal.layout.dlm.providers.
GroupMembershipEvaluatorFactory'>
          <paren mode="AND">
            <attribute mode='deepMemberOf' name='Eleves'/>
          </paren>
        </dlm:audience>
    </dlm:fragment>

    <dlm:fragment name='Enseignant' ownerID='enseignant-lo' precedence='10'>
        <dlm:audience evaluatorFactory='org.jasig.portal.layout.dlm.providers.
GroupMembershipEvaluatorFactory'>
          <paren mode="AND">
            <attribute mode='deepMemberOf' name='Enseignants'/>
          </paren>
        </dlm:audience>
    </dlm:fragment>

    <dlm:fragment name='Inspecteur' ownerID='inspecteur-lo' precedence='10'>
        <dlm:audience evaluatorFactory='org.jasig.portal.layout.dlm.providers.
GroupMembershipEvaluatorFactory'>
          <paren mode="AND">
            <attribute mode='deepMemberOf' name='Inspecteurs'/>
          </paren>
        </dlm:audience>
    </dlm:fragment>

    <dlm:fragment name='PersonnelAdministratif' ownerID='personnelAdministratif-lo' precedence='10'>
        <dlm:audience evaluatorFactory='org.jasig.portal.layout.dlm.providers.
GroupMembershipEvaluatorFactory'>
          <paren mode="AND">
            <attribute mode='deepMemberOf' name='Personnels administratif'/>
          </paren>
        </dlm:audience>
    </dlm:fragment>

    <dlm:fragment name='PersonnelEtablissementAutre' ownerID='personnelEtablissementAutre-lo'
precedence='10'>
        <dlm:audience evaluatorFactory='org.jasig.portal.layout.dlm.providers.
GroupMembershipEvaluatorFactory'>
          <paren mode="AND">
            <attribute mode='deepMemberOf' name='Personnels etablissement autre'/>
          </paren>
        </dlm:audience>
    </dlm:fragment>

    <dlm:fragment name='PersonnelLaboratoire' ownerID='personnelLaboratoire-lo' precedence='10'>
        <dlm:audience evaluatorFactory='org.jasig.portal.layout.dlm.providers.
GroupMembershipEvaluatorFactory'>
          <paren mode="AND">
            <attribute mode='deepMemberOf' name='Personnels laboratoire'/>
          </paren>
        </dlm:audience>
    </dlm:fragment>

    <dlm:fragment name='PersonnelMedical' ownerID='personnelMedical-lo' precedence='10'>
        <dlm:audience evaluatorFactory='org.jasig.portal.layout.dlm.providers.
GroupMembershipEvaluatorFactory'>
          <paren mode="AND">
            <attribute mode='deepMemberOf' name='Personnels medicaux'/>
          </paren>
        </dlm:audience>
    </dlm:fragment>

    <dlm:fragment name='PersonnelServAcademique' ownerID='personnelServAcad-lo' precedence='10'>
        <dlm:audience evaluatorFactory='org.jasig.portal.layout.dlm.providers.
GroupMembershipEvaluatorFactory'>
          <paren mode="AND">
```

```
            <attribute mode='deepMemberOf' name='Personnels service academique'/>
          </paren>
        </dlm:audience>
    </dlm:fragment>

    <dlm:fragment name='PersonnelTOS' ownerID='personnelTOS-lo' precedence='10'>
        <dlm:audience evaluatorFactory='org.jasig.portal.layout.dlm.providers.
GroupMembershipEvaluatorFactory'>
          <paren mode="AND">
            <attribute mode='deepMemberOf' name='Personnels TOS'/>
          </paren>
        </dlm:audience>
    </dlm:fragment>

    <dlm:fragment name='PersonnelRelation' ownerID='personneRelation-lo' precedence='10'>
        <dlm:audience evaluatorFactory='org.jasig.portal.layout.dlm.providers.
GroupMembershipEvaluatorFactory'>
          <paren mode="AND">
            <attribute mode='deepMemberOf' name='Parents'/>
          </paren>
        </dlm:audience>
    </dlm:fragment>

     <!-- definition du fragment pour le personnel de collectivité territoriale -->
      <dlm:fragment name='PersonnelCollectivite' ownerID='personnelCollectivite-lo' precedence='10'>
        <dlm:audience evaluatorFactory='org.jasig.portal.layout.dlm.providers.
GroupMembershipEvaluatorFactory'>
          <paren mode="AND">
            <attribute mode='deepMemberOf' name='Personnels collectivite'/>
          </paren>
        </dlm:audience>
      </dlm:fragment>

</managedLayoutFragments>
```

- Le premier bloc <dlm :fragment » suivant est alors extrait de celui-ci :

```
<dlm:fragment name='Admin' ownerID='admin-lo' precedence='3'>
    <dlm:audience evaluatorFactory='org.jasig.portal.layout.dlm.providers.PersonEvaluatorFactory'>
      <paren mode="AND">
        <attribute name="username" mode='equals' value='admin'/>
      </paren>
    </dlm:audience>
</dlm:fragment>
```

Il est alors copié dans un nouveau fichier XML nommé « fragmentDLM1.fragment-definition.xml » (avec ajout de l'entête XML : *<?xml version=" 1.0" encoding="UTF-8"?>*)

- L'élément *<fragment-definition>* est alors ajouté autour du bloc *<dlm :fragment>* :

```
<?xml version="1.0" encoding="UTF-8"?>
<fragment-definition xmlns:dlm="http://org.jasig.portal.layout.dlm.config" script="classpath://org/jasig
/portal/io/import-fragment-definition_v3-1.crn">
<dlm:fragment name='Admin' ownerID='admin-lo' precedence='3'>
        <dlm:audience evaluatorFactory='org.jasig.portal.layout.dlm.providers.PersonEvaluatorFactory'>
          <paren mode="AND">
            <attribute name="username" mode='equals' value='admin'/>
          </paren>
        </dlm:audience>
    </dlm:fragment>
</fragment-definition>
```

- Le nouveau fichier « fragmentDLM1.fragment-definition.xml » est alors sauvegardé et il peut être alors importé de manière dans la base de données 4.0 via la commande Ant :

```
ant data-import -Dfile="PATH/TO/ fragmentDLM1.fragment-definition.xml"
```

, avec ***–Dfile*** ayant pour valeur le chemin vers le fichier « fragmentDLM1.fragment-definition.xml »

ou de manière groupé avec tous les autres fichiers « fragmentDLMXXX.fragment-definition.xml » via la commande Ant :

```
ant data-import -Ddir="path/to/fragment-def.dir"
```

, avec ***« path/to/fragment-def.dir »*** correspondant au répertoire contenant l'ensemble des fichiers « .fragment-definition.xml »

- Si l'import est effectué avec succès le message « build successful » devrait alors être affiché dans la console, le fragment défini précédemment est désormais présent dans le portail uPortal v4.0.

Migration des « theme », « structure », « profile » et « layout »

Tous ces éléments ne correspondant pas à des « données brutes » mais plutôt à la présentation, aucune procédure n'a été définie. De ce fait, aucun de ces éléments ne sera migré vers la version 4.0 d'uPortal.

Migration des données spécifiques aux développements

Dans le cas où le portail ESUP 3.2 comprendrait des données en base liées à des développements spécifiques réalisés par ESUP, il sera nécessaire de définir des procédures de migration propre à chacun de ces développements. En effet, la procédure décrite dans la page courante permet de traiter l' ensemble des données génériques gérées par uPortal et non les données qui peuvent être issues de développements spécifiques.