

01 - les DLM uPortal

Le DLM (ou gestion de fragments)

Le DLM (*distributed layout management*) est le nouveau mode de fonctionnement des environnements utilisateur (précédemment ALM - *aggregated layout management*).

- Principe
- Le fichier dml.xml
- Détails d'un fragment
 - nom [name]
 - propriétaire [ownerID]
 - emplacement de gauche a droite [precedence]
 - ciblage [audience]
- Gestion des fragments
 - Utilisation de fichiers XML
 - Fichier d'import de fragment
 - Import/Export ANT
 - Utilisation de l'interface graphique

Principe

On définit dans un fichier XML (**dml.xml**) le ciblage fragment <=> utilisateur.

Chaque fragment appartient a un utilisateur particulier : le *layout owner* (LO). Les propriétaires des fragments sont des utilisateurs du portail et sont automatiquement créés dans la base (table **UP_USER**) lors de la lecture de **dml.xml** au démarrage du portail

Lors de la connexion d'un utilisateur "normal" le portail regarde le fichier *dml.xml* pour savoir quel fragment lui attribuer ; ensuite il va voir l'environnement de chaque propriétaire d'environnement (*layout owner*) concerné et attribue la partie d'environnement (onglet, groupe d'onglet) a l'utilisateur connecté.

Le fichier dml.xml

Il se trouve dans le dossier **properties** du portail. On définit dans ce fichier la position des fragments et leur visibilité pour les utilisateurs. Le contenu des fragments (canaux) est précisé ailleurs, dans les fichiers d'import de fragments ou par l'interface graphique (voir **gestion des fragments**)

```

<?xml version="1.0"?>
<!-- $Header$ -->

<managedLayoutFragments xmlns:dlm="http://org.jasig.portal.layout.dlm.config">
  <dlm:property name='defaultLayoutOwner' value='fragmentTemplate' />
  <!-- <dlm:property name='layoutDecorator' value='someClass' /> -->

  <!-- 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='Guests' ownerID='guest-lo' precedence='20'>
    <dlm:audience evaluatorFactory='org.jasig.portal.layout.dlm.providers.GuestUserEvaluatorFactory' />
  </dlm:fragment>

  <dlm:fragment name='Admin' ownerID='admin-lo' precedence='10'>
    <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='Default' ownerID='default-lo' precedence='10'>
    <dlm:audience evaluatorFactory='org.jasig.portal.layout.dlm.providers.GroupMembershipEvaluatorFactory'>
      <paren mode="AND">
        <attribute mode='deepMemberOf' name='Etablissement' />
      </paren>
    </dlm:audience>
  </dlm:fragment>
</managedLayoutFragments>

```

On retrouve dans ce fichier :

- L'utilisateur par défaut (dont tout le monde hérite) : **defaultLayoutOwner**
- le temps pendant lequel ce fichier est caché : **org.jasig.portal.layout.dlm.RDBMDistributedLayoutStore.fragment_cache_refresh**
- tous les fragments **<dlm:fragment> [...] </dlm:fragment>**

Détails d'un fragment

```

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

```

On retrouve :

nom [name]

Uniquement utilisé dans ce fichier pour les distinguer

propriétaire [ownerID]

C'est le *layout owner* (par convention on suffixe le login de ces utilisateurs par **-lo** pour les distinguer des utilisateurs classiques).

emplacement de gauche a droite [precedence]

Détermine la place du fragment de gauche a droite par rapport aux autres.
C'est un numérique ; plus il est grand plus il se trouve a gauche.
Les *precedences* sont évaluées entre différents fragments.

```
<dml:fragment name='Testing' ownerID='test-lo' precedence='90'>
...
<dml:fragment name='Other' ownerID='other-lo' precedence='100'>
```

Dans ce cas on retrouve dans l'ordre : Other puis Testing.

Les fragments sont des groupes d'onglets ; si on prend l'exemple de 2 fragments contenant chacun 2 onglets, on peut positionner le fragment complet, pas les onglets.

Il n'est donc pas possible d'intercaler un onglet d'un fragment puis un onglet d'un autre et ainsi de suite.

En cas de conflit c'est l'ordre de définition dans le fichier **dml.xml** qui tranchera.

ciblage [audience]

Va déterminer à qui doit être attribué le fragment.

On paramètre une classe java qui va faire l'évaluation [evaluatorFactory].

On dispose de plusieurs classes java de base dans le portail :

- **AllUsersEvaluatorFactory** : évalue tous les utilisateurs
- **GroupMembershipEvaluatorFactory** : évalue l'appartenance a un groupe
- **GuestUserEvaluatorFactory** : évalue l'utilisateur guest
- **PersonEvaluatorFactory** : évalue sur des attributs uPortal

On peut aussi faire des combinaisons a base de directive *parent* AND, OR, NOT.

La partie mode d'égalité dépend de la classe **evaluatorFactory** :

- **PersonEvaluatorFactory**
 - equals
 - exists
 - contains
 - startsWith
 - endsWith
- **GroupMembershipEvaluatorFactory**
 - memberOf (Membre direct)
 - deepMemberOf (Membre Recusif)

Gestion des fragments

Un fragment consiste en un ou des onglets, contenant un ou des canaux disposés en colonnes.

Vous pouvez gérer les fragments de deux manières :

- Utilisation de fichiers XML contenant l'environnement de l'utilisateur (fichiers d'import de fragment) puis importation/exportation via des tâches ANT.
- Connexion directement avec l'utilisateur propriétaire et création de son environnement par l'interface web du portail.

Utilisation de fichiers XML

Fichier d'import de fragment

Les fichiers d'import de fragment sont stockés dans le répertoire **properties/layouts**. Il y a un fichier par fragment. Par convention, on donne au fichier le nom de son propriétaire (**guest-lo.xml** pour le propriétaire **guest-lo**)

```
<?xml version="1.0" encoding="UTF-8"?>
<layout user="guest-lo">
  <root-folder>
    <folder mutable="True" removable="True" type="header">
      <name>Header folder</name>
      <channel fName="header" />
      <channel fName="portal/login/general" />
      <channel fName="session-locales-selector" />
    </folder>
    <folder mutable="True" removable="True" type="regular">
      <name>Accueil</name>
      <folder mutable="True" removable="True" type="regular">
        <name>Column</name>
        <channel fName="accueil-esup-portail" />
      </folder>
    </folder>
    <folder mutable="True" removable="True" type="footer">
      <name>Footer folder</name>
      <channel fName="footer" />
    </folder>
  </root-folder>
</layout>
```

On retrouve dans ce fichier la structure complète de l'environnement utilisateur :

- le nom de l'utilisateur [user] : nom de l'utilisateur propriétaire du fragment
- dossier racine [root-folder] : contient l'ensemble des folders

Un dossier de premier niveau est un onglet, un dossier de second niveau (folder/folder) est une colonne.

- les onglets [folder]
 - name : Nom de l'onglet affiché à l'écran
 - mutable : [true|false] est-il déplaçable ?
 - removable : [true|false] est-il supprimable ?
 - type : [header|regular|footer] : type de l'onglet (*header* et *footer* sont réservés par le portail)
- Les colonnes [folder/folder] : idem onglets (les noms des colonnes ne s'affichent pas)
- Les canaux contenus dans une colonne : le **fName** correspond au *functional name* utilisé lors de la publication du canal.

Import/Export ANT

Importation du fichier uPortal/properties/layouts/guest-lo.xml (appartenant à guest-lo) dans la base du portail :

```
ant layout.import -Dusername=guest-lo
```

On peut aussi importer tous les environnements contenus dans le dossier layouts

```
ant layout.import -Dusername=all
```

NB : En base, les fragments sont stockés dans les tables **UP_LAYOUT_STRUCT** et **UP_LAYOUT_PARAM**

Opération inverse : Exportation de l'environnement de guest-lo dans uPortal/properties/layouts/guest-lo.xml (utile par exemple si on a créé/modifié l'environnement de guest-lo avec l'interface graphique)

```
ant layout.export -Dusername=guest-lo
```

Utilisation de l'interface graphique

Pour modifier un fragment avec l'interface graphique, il faut pouvoir se connecter sous le nom de l'utilisateur propriétaire. Il faut donc préalablement le créer dans la table servant à l'authentification locale :

```
ant user.add -Dusername=guest-lo
```

Puis indiquer son mot de passe.

NB : Ceci ajoute l'utilisateur dans la table **UP_PERSON_DIR**

Une fois l'utilisateur créé dans le portail vous pouvez vous connecter sous son nom. Attention il s'agit d'une authentification locale, le paramètre **esup.db.auth** de config.properties doit donc être égal à **true**. On devrait aussi pouvoir passer par une authentification distante (CAS => LDAP), mais je n'en voit pas l'intérêt.

Vous pouvez comme tout utilisateur ajouter des onglets, des colonnes et des canaux.

Ensuite si vous voulez gérer des restrictions, allez dans le canal préférence (Menu en haut a droite "Préférence")

En cliquant sur un onglet vous pourrez indiquer si on peut :

- Déplacer l'onglet
- Editer les propriétés
- Ajouter des colonnes
- Supprimer l'onglet

En cliquant sur une colonne vous pourrez indiquer si on peut :

- Déplacer la colonne
- Editer les propriétés
- Ajouter des canaux
- Supprimer la colonne

En cliquant sur un canal vous pourrez indiquer si on peut :

- Déplacer le canal
- Supprimer le canal

Toutes les modifications seront répercutées sur le layout, donc sur les utilisateurs autorisés à voir ce fragment.