



aide du fichier manifest.xml et de l'utilisation XML+XSL  
version 1.03

<https://intranet.insa-toulouse.fr/index.jsp>

# Utilisation du manifest.xml Utilisation de XML + XSL



Conception et réalisation du document :  
Alain Bérard (alain.berard@insa-toulouse.fr tel : 94.29)

## Résumé :

Il est possible d'envoyer sur l'intranet pédagogique les contenus « zipé » accompagné d'un fichier xml « manifest.xml » afin d'automatiser des renseignements sur les contenus.

# 1. Uploader le fichier Zip & du fichier manifest.xml

Depuis la partie « gérer mes cours » cliquer sur « Ajouter un contenu ».



Cliquez sur l'icône « ajouter un zip avec manifest.xml »

Ajout d'un cours zippé avec un manifest XML.

*Veuillez insérer le chemin du fichier zip du contenu à mettre en ligne (contenant un fichier descripteur en XML). Cela permettra à l'application de l'uploader sur le serveur et de le décompresser.*

Chemin du fichier :  Parcourir...

VALIDER

Cliquez sur le bouton :

Parcourir...

Choisir le fichier Zip sur votre ordinateur.

Cliquez sur le bouton « valider »

VALIDER

Le fichier Zip est alors « envoyé » sur l'intranet pédagogique pour y être « dézipé » avec analyse du fichier manifest.xml.

Le fichier manifest.xml contient les informations sur :

Les dossiers et leurs hiérarchies.

Les fichiers et leurs hiérarchies.

Les boutons.

Les droits sur les dossiers et les fichiers.

Il est ainsi possible d'indiquer dans le fichier, la hiérarchie ainsi que les droits sur les contenus évitant l'étape manuel qui doit être fait quand les contenus ne sont pas accompagnés du fichier de configuration.

En l'absence d'information sur la structure des dossiers et le type de fichiers, l'intranet utilisera directement les informations du fichier zip.

Pour résumé, il n'est pas utile de renseigner les champs sur les dossiers et les types de fichier, le moteur de l'intranet reconstruira les informations à partir des fichiers et de la structure des dossiers du fichier zip.

Exemple d'un fichier manifest.xml :

```
<?xml version="1.0" encoding="iso-8859-1"?>
<root>
<directory uri="/content" >
<button type="2" >Accueil du cours</button>
</directory>
<directory uri="/content/sous_dossier" >
<userright>aberard;enrici</userright>
</directory>
<directory uri="/content/sous_dossier/toto" >
<button type="1" >toto</button>
<right>public;</right>
</directory>
<file uri="/content/sous_dossier/toto/ico.gif" >
<property name="getcontenttype" ><![CDATA[image/gif]]></property>
</file>
</root>
```

Ceci donnant une structure :



## 2. Structure du fichier manifest.xml

```
<?xml version="1.0" encoding="iso-8859-1"?>
<root>
|
| Tag xml
|
</root>
```

Les Balises :

`<?xml version="1.0" encoding="iso-8859-1"?>` Entête du fichier « manifest.xml ».

`<root></root>` Début et fin du fichier xml

`<directory uri="/content" > </directory>` Définition d'un dossier ( = répertoire)  
`uri="/content"` Nom du dossier

les « sous dossier » au dossier « content » base obligatoire des répertoires des cours se déclare ainsi :

`<directory uri="/content/sous_dossier" > « sous_dossier »` et ici le nom d'un sous dossier du dossier racine « content »

`<button type="2" >`Accueil du cours `</button>` Définition de la première page d'un cours et donc du bouton accueil du cours. Le tag `<button>` doit faire partie du tag `<directory>` :

`<directory uri="/content" ><button type="2" >Accueil du cours</button></directory>`

Le `type="2"` définit la page d'accueil

Le `type="1"` définit les autres boutons

Le nom du bouton étant défini comme ceci : `<button type="1" >toto</button>`

Ici « toto » est le nom du bouton.

La balise `<right>public;</right>` permet de définir les droits, il doit être encadré dans une balise `<directory>` pour des droits sur un dossier ou une balise `</file>` pour les droits sur un fichier.

La balise `<userright>` contient :

Public ; pour un élément public,

le nom des utilisateurs autorisés, `<userright>aberard;enrici</userright>` ici les utilisateurs « aberard » et « enrici ».

La balise `<file>` permet de définir les fichiers, leur type, leur droit et leur place dans la hiérarchie des dossiers.

```
<file uri="/content/sous_dossier/toto/ico.gif" >
<property name="getcontenttype" ><![CDATA[image/gif]]></property>
</file>
```

Ici le fichier « ico.gif » est dans un dossier toto qui est lui-même dans un dossier « sous\_dossier » a la racine du cours ( le dossier « content »). La balise `<property name=` » permet d'indiquer le type de fichier :

<code>&lt;![CDATA[image/gif]]&gt;</code>	Image au format gif
<code>&lt;![CDATA[image/jpg]]&gt;</code>	Image au format jpg
<code>&lt;![CDATA[image/png]]&gt;</code>	Image au format png
<code>&lt;![CDATA[text/xxx]]&gt;</code>	fichier texte, ici format css peut être :
<code>&lt;![CDATA[text/rtf]]&gt;</code>	document rtf
<code>&lt;![CDATA[text/html]]&gt;</code>	document htm, html
<code>&lt;![CDATA[text/xml]]&gt;</code>	document xml
<code>&lt;![CDATA[text/css]]&gt;</code>	document css (feuille de style)
<code>&lt;![CDATA[text/plain]]&gt;</code>	document txt (notepad sous windows)
<code>&lt;![CDATA[application/xxx]]&gt;</code>	fichier provenant d'une application comme :
<code>&lt;![CDATA[application/msword]]</code>	fichier provenant de Word.
<code>&lt;![CDATA[application/pdf]]&gt;</code>	fichier provenant d'Adobe Acrobat.
<code>&lt;![CDATA[application/mspowerpoint]]&gt;</code>	fichier provenant de Powerpoint.
<code>&lt;![CDATA[application/msexcel]]&gt;</code>	fichier provenant d'excel.

`<![CDATA[application/octet-stream]]>` fichier inconnu de la plateforme.

La balise `<publish-date>` et la balise `<unpublish-date>` permet d'indiquer une date de publication et une date de fin de publication. Les deux balises doivent être inclus dans une balise `<directory>` ou une balise `<file>` selon le cas.

Exemple :

```
<directory uri="/content/sous_dossier/toto" >
<property name="source" ><![CDATA[]]></property>
<unpublish-date>29/05/2005</unpublish-date>
<publish-date>29/04/2005</publish-date>
<right>public;</right>
</directory>
```

Le dossier « toto » a une date de début de publication au 29/04/2005 et une fin de publication au 29/05/2005.

La valeur `name="user-revocation"` d'une balise `<property>` permet d'indiquer des utilisateurs dont on ne veut pas accorder l'accès à une ressource.

Exemple l'utilisateur « enrici » n'a pas accès au dossier « content » et à ce qu'il contient :

## Aide sur l'utilisation des fichiers xml sur l'intranet pédagogique.

```
<directory uri="/content" > <property name="user-revocation" > <![CDATA[enrici;]]>
</property> </directory> <property name="source" > <![CDATA[]]> </property>
```

### 3. Contenus minimums :

Comme déjà indiqué l'intranet va recréer les informations non disponibles dans le fichier manifest.xml via la structure et le contenu du fichier zip.

Il est ainsi possible d'avoir un fichier manifest.xml minimaliste :

```
<?xml version="1.0" encoding="iso-8859-1"?>
<root>
<directory uri="/content" >
<userright>c2p;</userright>
<publish-date>21/03/2005</publish-date>
<right>public;</right>
</directory>
<file uri="/content/forum.htm" ><button type="1" >Forum de Discussion</button>
</file>
<directory uri="/content/images" ><button type="1" >Telechargement</button>
</directory>
<file uri="/content/dgei.htm" ><button type="2" >Accueil du cours</button>
</file>
<file uri="/content/fiche.htm" ><button type="1" >Contact</button>
</file>
</root>
```

En cas de contradiction entre le fichier manifest.xml et la logique du fichier zip, le moteur de l'intranet pédagogique utilisera les informations qu'il trouvera dans le fichier zip en lieu et place des informations du fichier manifest.xml.

## 4. Tableau des Balises.

**<root>**

Début du fichier manifest.xml, encadre les autres balises.

**<directory uri="/content" >**

Description des dossiers (= répertoire).

**uri="/xxx"**

Nom et chemin du dossier.

**<button type="2" >**Accueil du cours **</button>**

Description des boutons.

**type="1"** Bouton standard

**type="2"** Bouton de la page d'accueil

**<right>**

Description des droits.

**<file>**

Description des fichiers.

**<property name="getcontenttype" >** Propriété des fichiers.

**<![CDATA[xxx/yyy]]>** Description du type de fichier.

**<property name="getlastmodified" ><![CDATA[Tue, 29 Mar 2005 11:27:59 GMT]]></property>** Permet d'utiliser le système de gestion des versions.

**<property name="creationdate" ><![CDATA[2005-03-29T11:27:59Z]]></property>** Permet d'utiliser le système de gestion des versions.

**<unpublish-date>**

Description des dates de fin de publication.

**<publish-date>**

Description des dates de début de publication.

**<property name="user-revocation" ><![CDATA[enrici;]]></property>**

Description des utilisateurs non autorisé a visualiser une ressource.



## 5. Utilisation d'un fichier XML associé a un fichier XSL

Un fichier XSL est une feuille de styles, dédiée au XML, et elle est elle-même un fichier XML. Les règles d'un document XML s'appliquent donc à un document XSL. Le XSL permet de transformer un document XML en un document spécialement formaté pour une application, HTML ou PDF par exemple.

Pour que l'intranet pédagogique utilise vos fichiers xml associés a un fichier xsl,, vous devez uploader sur le serveur un fichier xsl associé a des fichiers xml.

Depuis la partie « gérer mes cours » cliquer sur « Ajouter un contenu ».



Cliquez sur l'icône « ajouter un cours XML/XSL »

**Ajout d'un cours zippé au format XML/XSL.**

*Veuillez insérer le chemin du fichier zip du contenu à mettre en ligne. Cela permettra à l'application de l'uploader sur le serveur et de le décompresser. Le contenu de ce cours doit être au format XML/XSL, avant de continuer vérifier bien que tout vos fichiers XML font bien référence à une feuille de style XSL via un chemin relatif.*

Chemin du fichier :

Vous devez indiquer le fichier zip contenant les fichiers xml et le fichier xsl.

## Aide sur l'utilisation des fichiers xml sur l'intranet pédagogique.

### Fichiers XML à transformer.

Choisissez les fichiers XML qui doivent être transformés ou retransformés. Un fichier XML dont la feuille de style XSL est modifiée doit être retransformé si vous voulez que les modifications sur la feuille XSL soient prises en compte.

**VALIDER**

content  
simple.xml  
test.xml

Puis une fois le fichier analysé par le serveur, le message suivant vous demande d'indiquer le ou les fichiers xml qui doivent être transformés par avec les informations du fichier xsl. Cliquez sur le bouton Valider.

Puis comme pour un cours « traditionnel » le serveur vous demande d'indiquer le point d'entrée du cours, puis éventuellement les boutons que vous voulez associé.

### Définition des droits et boutons

#### Droit pour un utilisateur

#### Droit pour une filière

Definissez le point d'accès de votre cours en cliquant sur le fichier correspondant.

content  
simple.xml  
test.xml

Le serveur a créé une page avec l'extension html pour chaque fichier xml transformé. C'est cette page html qui doit être utilisée comme page d'accueil :

Definissez le point d'accès de votre cours en cliquant sur le fichier correspondant.

content  
xml.xml  
xsl.xsl  
xml.xml.html

Il est possible d'indiquer ultérieurement au serveur un autre fichier xsl via le bouton « Régénérer des fichiers xml » :

Gestion du cours					
 Paramètres du cours	 Sauvegarde du cours	 Restauration du cours	 Statistiques	 Supprimer le cours	 Ajouter Des Concepteurs

  

Gestion du contenu						
 Ajouter un contenu	 Remplacer le contenu	 Droits et boutons	 Gestion des fichiers	 Versions des documents	 Régénérer des fichiers XML	 Liens cassés

Le serveur vous montrera les fichiers de votre cours, afin d'indiquer le fichier xsl qui servira pour la transformation des fichiers xml :

Fichiers XML à transformer.

*Choisissez les fichiers XML qui doivent être transformés  
être retransformés si vous voulez que les modifications*

**VALIDER**

- ☒ content
- ☐ simple.xsl
- ☐ test.xml

Table des matières.

1.	Uploader le fichier Zip & du fichier manifest.xml.....	2
2.	Structure du fichier manifest.xml .....	4
3.	Contenus minimums :.....	7
4.	Tableau des Balises. ....	8
5.	Utilisation d'un fichier XML associé a un fichier XSL.....	9