

Formation à logidee-tools



Ce document est sous Licence de Documentation Libre GNU.

Support de cours

Formation à logidee-tools

Auteurs : Raphaël Hertzog, Stéphane Casset.

La reproduction dans ce manuel de noms de marques déposées, de désignations des marques, etc. même sans stipulation explicite que de tels noms et marques sont protégés par la loi, n'autorise pas à supposer qu'ils soient libres et puissent être utilisés par quiconque.

Cet ouvrage a été mis en page avec L^AT_EX, à partir de fichiers au format XML convertis à l'aide de xsltproc développé par *Daniel Veillard* (<http://xmlsoft.org/XSLT/>).

Les feuilles de styles, la DTD et les outils associés ont été développés par *Logidée* (<http://www.logidee.com>) pour rédiger et maintenir ses supports de formations. Ces outils sont bien sûr sous licence libre et disponibles sur le site web mentionné ci-dessus.

Permission est accordée de copier, distribuer et/ou modifier ce document selon les termes de la Licence de Documentation Libre GNU (GNU Free Documentation License disponible à <http://www.gnu.org/copyleft/fdl.html>), version 1.1 ou toute version ultérieure publiée par la Free Software Foundation.

Veuillez vous référer aux sources XML de ce document (que vous pouvez obtenir auprès de l'auteur) pour connaître le détail de la licence avec la liste des Sections Invariables, des Textes de Première de Couverture, et des Textes de Quatrième de Couverture.

Sommaire

Thème 1 : Références de la DTD et de logidee-tools	4
1.1 Description d'un support de cours en XML	5
1.1.1 La structure globale	6
Formations, thèmes et modules	6
L'élément <code>formation</code>	6
L'élément <code>theme</code>	6
L'élément <code>module</code>	7
L'élément <code>slideshow</code>	7
1.1.2 L'élément <code>info</code>	9
La structure	9
Détails du bloc d'information	10
1.1.3 La mise en page du texte	11
Les balises pour le texte	11
Les tableaux	12
Les formules de mathématiques	13
1.1.4 Les balises de structure	14
La structure du contenu	14
L'élément <code>page</code>	15
L'élément <code>slide</code>	16
L'élément <code>section</code>	16
L'élément <code>para</code>	16
L'élément <code>note</code>	17
L'élément <code>exercice</code>	17
L'élément <code>glossary</code>	17
L'élément <code>xi:include</code>	17
1.1.5 Les caractères spéciaux	19
Les entités pour les caractères spéciaux	19
1.1.6 Annexe : la DTD	20
La DTD des supports de cours	20
1.2 Les outils logidee-tools	24
1.2.1 Présentation générale	25
Vue d'ensemble	25
Changements	25
1.2.2 Installation des outils	26
Dépendances	26
Pour les utilisateurs de Debian	26
1.2.3 Générer les supports de cours	27
Installer les makefiles	27
Utiliser les makefiles	28
1.2.4 Rédiger des supports de cours	30
Avec un éditeur de texte	30
Ce qu'il faut savoir en rédigeant	30
1.2.5 Conversions	32
Passage de la version 1.0 à 1.2	32
Récupération de documents docbook-xml	32
1.2.6 Fonctionnalités avancées	33
Les chartes graphiques	33
Localisation d'une charte	34

1. Références de la DTD et de logidee-tools



1.1. Description d'un support de cours en XML

- Formations, thèmes et modules
- L'élément **formation**
- L'élément **theme**
- L'élément **module**
- L'élément **slideshow**

Formations, thèmes et modules

Chaque formation est composée de plusieurs thèmes. Eux-mêmes se décomposent en différents modules.

L'élément **formation**

L'élément **formation** se présente de la manière suivante :

```
<formation xmlns:xi="http://www.w3.org/2001/XInclude">
  <info>
    ...
  </info>
  <theme>
    ...
  </theme>
  <xi:include href="fichier_theme.xml"/>
</formation>
```

L'élément **formation** contient un élément **info** décrit plus bas et peut contenir des thèmes ou des éléments inclus si ceux-ci correspondent à des thèmes.

Cet élément sert généralement à rassembler tout le matériel pédagogique d'un cursus.

L'élément **theme**

L'élément **theme** se présente de la même manière que l'élément **formation** :

```
<theme xmlns:xi="http://www.w3.org/2001/XInclude">
  <info>
    ...
  </info>
  <module>
```



```

    ...
  </module>
  <xi:include href="fichier_module.xml"/>
</theme>

```

L'élément `theme` permet de rassembler tous les modules d'un même thème.

L'élément `module`

L'élément `module` suit une structure similaire :

```

<module>
  <info>
    ...
  </info>
  <page>
    ...
  </page>
  <page>
    ...
  </page>
</module>

```

L'élément `module` constitue une entité indivisible d'enseignement. Il ne traite en général qu'un seul sujet.

L'élément `slideshow`

Au lieu d'écrire un module de cours complet (avec support de cours imprimé), on peut se limiter à la création des seules diapositives grâce à l'élément `slideshow`. Il est très simple à utiliser :

```

<slideshow>
  <shortinfo>
    <title>Titre du diaporama</title>
    <description><para>Une description</para></description>
    <version number="1.0">
      <author>Nom de l'auteur</author>
      <email>mon_email@mondomaine.tld</email>
      <comment>...</comment>
      <date>2001-08-09</date>
    </version>
  </shortinfo>
  <slide>
    (première diapo)
  </slide>
  <slide>
    (deuxième diapo)
  </slide>
  ...

```



```
<slide>
  (Nième diapo)
</slide>
</slideshow>
```



- La structure
- Détails du bloc d'information

La structure

Cet élément est présent dans les formations, les thèmes et les modules et se présente comme suit :

```
<info>
  <title>Le titre</title>
  <ref>Une référence</ref>
  <description><para>Une brève description</para></description>
  <objectives>
    <item>Objectif 1</item>
    <item>Objectif 2</item>
  </objectives>
  <ratio value="théorie/pratique"/>
  <duration value="duree" unit="m ou h ou j"/>
  <prerequisite><para>Les pré-requis</para></prerequisite>
  <dependency>
    <ref>une référence à un élément nécessaire</ref>
    <ref>une référence à un élément nécessaire</ref>
  </dependency>
  <suggestion>
    <ref>une référence à un élément suggéré</ref>
    <ref>une référence à un élément suggéré</ref>
  </suggestion>
  <version number="version">
    <author>Auteur du texte</author>
    <email>mon_email@mondomaine.tld</email>
    <comment>Un commentaire de mise à jour</comment>
    <date>2001-08-09</date>
  </version>
  <level value="niveau"/>
  <state finished="true" proofread="true"/>
</info>
```



```
<proofreaders>
  <item>Nom d'un relecteur</item>
</proofreaders>
</info>
```

Détails du bloc d'information

- ❑ **<title>** : contient le titre que l'on veut donner à l'élément, il peut contenir les balises correspondant aux balises de texte (voir plus loin).
- ❑ **<description>** : contient une brève description de l'élément, elle doit être concise mais bien préciser le sujet traité.
- ❑ **<objectives>** : contient une liste des objectifs pédagogiques de l'élément, il peut servir à la génération automatique des conducteurs de stage.
- ❑ **<ratio>** : indique la répartition entre la théorie et la pratique, par exemple s'il y a 60% de théorie et 40% de pratique on notera : `<ratio value="60/40"/>`
- ❑ **<duration>** : indique la durée prévue pour l'élément, l'attribut *value* indique la durée et l'attribut *unit* précise l'unité de temps utilisée, par exemple :
 - ❑ `<duration value="30" unit="m"/>` indique une durée de 30 minutes et
 - ❑ `<duration value="2" unit="h"/>` indique une durée de 2 heures.
- ❑ **<level>** : permet de préciser le niveau de l'élément.
- ❑ **<version>** : permet de tenir un historique des modifications, l'attribut *number* permet de donner un numéro de version. Les sous-éléments **<author>** et **<comment>** permettent de préciser qui a effectué les modifications et sur quoi elles portent. Les deux attributs **<email>** et **<date>** permettent de préciser respectivement l'email de l'auteur et la date de modification du document. Ces deux balises sont optionnelles.
- ❑ **<ref>** : définit un identifiant unique pour le document afin de pouvoir le référencer ailleurs (en tant que dépendance ou autres).
- ❑ **<prerequisite>** : permet de préciser les pré-requis nécessaires à la compréhension de l'élément pédagogique.
- ❑ **<suggestion>** : permet de donner une liste d'éléments pouvant servir à mieux comprendre le contenu de l'élément, ou pouvant être des suites.
- ❑ **<dependency>** : permet de donner une liste d'éléments nécessaires avant l'accès à l'élément.
- ❑ **<state>** : permet de préciser l'état du document grâce aux deux attributs **finished** (fini ou pas) et **proofread** (relu ou pas) qui peuvent être à "true" ou "false".
- ❑ **<proofreaders>** : permet d'indiquer le nom des relecteurs, un nom par sous-élément **item**.



- Les balises pour le texte
- Les tableaux
- Les formules de mathématiques

Les balises pour le texte

Une autre catégorie de balises permet de mettre des éléments du texte en évidence, ces balises peuvent être utilisées dans tous les champs où l'on s'attend à trouver du texte.

- ** : permet de mettre un texte en évidence (généralement en italique), c'est généralement le cas pour les termes en langues autre que le français (ie, via, boot disks).
exemple : `texte en évidence`
- <menu>** : permet de mettre en évidence une entrée dans un menu.
exemple : `<menu>Fichier</menu>`
- <cmd>** : permet de mettre en évidence une commande.
exemple : `<cmd>ls -al</cmd>`
- <file>** : permet de mettre en évidence un nom de fichier.
exemple : `<file>/etc/passwd</file>`
- <url>** : permet de mettre en évidence une URL.
exemple : `<url href="http://linux-france.org"/>`
- <code>** : permet de citer un morceau de code ou un extrait d'un fichier.
exemple :

```
<code>
...
void main (void) {
printf("Hello World.\n");
}
</code>
```
- <image>** : permet d'inclure une image dans le texte. On peut y adjoindre une légende avec la balise `legend`. On peut également préciser une échelle pour redimensionner l'image avec l'attribut `scale`.
exemple :

```
<image src="logo.eps" scale="0.5"/>
<image src="schema.eps">
```



```
<legend>Schéma d'interconnexion</legend>
</image>
```

Note

Ces deux derniers éléments disposent d'un attribut `visible` qui permet de définir s'il doit apparaître dans une diapositive spécifique lors de la présentation. Si l'attribut vaut « oui » alors l'élément est repris dans une diapositive spécifique, sinon il n'est pas repris. La valeur par défaut de l'attribut est « oui » pour l'image et « non » pour le bloc de code. C'est à dire qu'une image est automatiquement reprise dans la présentation si l'attribut `visible` n'est pas explicitement positionné à « non ».

Note

Pour éviter les problèmes avec les caractères spéciaux `>`, `<`, et `&` dans les blocs de code, on prend souvent la précaution de protéger ces blocs de texte en les mettant dans un bloc CDATA comme ci dessous :

```
<code><![CDATA[
  ici on utilise < et > et & sans aucun risque ...
]]></code>
```

Les tableaux

Il est possible de définir des tableaux simples avec ou sans bordures. Il n'est pas possible d'avoir des cellules s'étalant sur plusieurs colonnes ou lignes. Conceptuellement le tableau (élément `table`) est une suite de lignes (élément `row`) et les lignes sont des suites de cellules (élément `col`).

Chacun de ces 3 éléments dispose d'un attribut `border` qui peut être soit à « 0 » soit à « 1 ». Pour l'élément `table`, il est à « 1 » par défaut, c'est-à-dire que les bordures extérieures du tableau sont présentes par défaut. Pour l'élément `row`, il est également à « 1 » par défaut, c'est-à-dire qu'il y a un trait sous chaque ligne sauf si `border` est explicitement positionné à « 0 ». Enfin pour l'élément `col`, sa valeur par défaut est soit celle obtenue de l'élément `col` correspondant de la première ligne soit « 1 ».

L'élément `col` dispose également d'un attribut `head` qui permet de préciser si la cellule joue le rôle d'intitulé de colonne. Concrètement, si la valeur de `head` est à « 1 » alors la cellule est mise en avant avec du gras.

Exemple :

```
<table>
  <row>
    <col> ... </col>
    ...
  </row>
  <row>
    ...
  </row>
  ...
</table>
```



Par exemple :

col1	col2
col1	col2

Note

Seule la feuille de style LaTeX gère correctement toutes les combinaisons de bordures. La feuille de style HTML gère uniquement la bordure au niveau du tableau en entier.

Les formules de mathématiques

Etant donné que MathML n'est pas encore au point et que cela peut rendre service à certaines personnes, il est possible d'insérer des formules mathématiques rédigées en TeX en utilisant la balise `math`. L'emploi de celle-ci n'est pas recommandé puisqu'elle ne fonctionne que pour une sortie TeX (c'est à dire celle produisant les PDF) et qu'il n'y aura pas de formule correspondante dans la version HTML.

La balise `math` a un attribut obligatoire `notation` qui indique si son contenu est du TeX ou du contenu à destination de la sortie HTML. Cela permet de mettre un mot signalant l'absence de la formule dans la version HTML à défaut de pouvoir y reproduire une formule.

Par exemple :

$$E = MC^2$$

```
<math notation="tex"><![CDATA[
$ E = MC^2 $
]]></math>
<math notation="html">
[ Energie = Masse * Célérité au carré ]
</math>
```

Note

Ceux pour qui la sortie HTML est importante et qui veulent des formules pour toutes les sorties sont encouragés à insérer leurs formules sous forme d'images (éventuellement vous pouvez créer cette image avec LaTeX et sa sortie postscript que vous convertissez ensuite en EPS).



- La structure du contenu
- L'élément `page`
- L'élément `slide`
- L'élément `section`
- L'élément `para`
- L'élément `note`
- L'élément `exercice`
- L'élément `glossary`
- L'élément `xi:include`

La structure du contenu

Afin de structurer le document, les balises suivantes sont là pour le découper selon le schéma suivant :

```

formation
  |-> theme
      |-> module
          |-> page
              |-> slide, section, exercice
                  |-> section, para, note, exercice
                      |-> para, note, exercice

```

et aussi pour les listes :

```

list
  |-> list, item
      |-> item

```

Les `item` peuvent contenir tous les attributs de texte.



Note

Les éléments `page`, `section`, `exercice`, `para` et `note` disposent d'un attribut `restriction` qui permet de préciser si l'élément est spécifique à quelque chose. On pourra ainsi le positionner à des mots-clés. Cela permettra de générer des documents spécifiques par sélection de mots-clés jugés pertinents. Ainsi si le document traite de Linux, ce système peut être mis à profit pour inclure des informations sur différentes distributions Linux et générer des documents spécifiques à une distribution. D'une manière générale ce système permet de gérer des documents similaires destinés à plusieurs publics.

Exemple :

```
<note restriction="debian">
  Tout ce qui est dit là est spécifique à Debian.
</note>
```

Note

Les éléments `exercice`, `para` et `note` disposent d'un attribut `icon` permettant de rajouter une petite image informative comme un point d'exclamation pour signaler un point important. La valeur de cet attribut est un nom de fichier `.eps` ou alors un code symbolique (comme "attention"). Le code symbolique doit être défini dans la section `icons` de la charte graphique (`charte/default/charte.xml`).

L'élément page

L'élément `page` est à l'intérieur des modules et se subdivise ainsi :

```
<page>
  <title></title>
  <slide>
    <list>
      <item></item>
    </list>
  </slide>

  <section>
    <title></title>
    <section>
      <title></title>
      <para>
      </para>
    </section>
    ...
  </section>
  ...
</page>
```

L'élément `page` est composé de diapositives (ou slides) et de sections, ces sections peuvent être subdivisées par d'autres sections.

Un élément `page` peut aussi contenir des exercices (voir plus loin).



Un élément page se compose obligatoirement d'un élément `title`, suivi d'une succession quelconque d'éléments `slide` et de sections.

L'élément `slide`

Si cet élément est totalement absent de l'élément `page`, il est alors généré en prenant les titres des sections et sous sections de la page en cours.

L'élément `slide` contient un ou plusieurs paragraphes qui peuvent être préfixés par des sous-titre (élément `subtitle`). Les paragraphes peuvent contenir tout ce qu'ils contiennent habituellement.

Exemple :

```
<slide>
  <title></title>
  <para>
    <list>
      ...
    </list>
    ...
  </para>
  <subtitle>Sous-titre</subtitle>
  <para>
    ...
  <image src="mon_image.eps"/>
  ...
</para>
</slide>
```

Note

L'élément `slide` dispose d'un attribut `background` qui permet de préciser l'image de fond (fichier eps) que l'on souhaite pour la diapositive de la présentation. Ici aussi il est possible d'indiquer des noms symboliques qui seront transformés en noms réels grâce à la section `backgrounds` de la charte graphique (`charte/default/charte.xml`).

L'élément `section`

Cet élément permet de découper la « page » en sections et sous-sections si l'on imbrique des balises `section` (dans la limite de 2 niveaux).

Les sections sont composées d'un élément `title` et de `para` (paragraphes), on peut aussi leur ajouter des notes (`note`). Enfin, elles peuvent également accueillir des éléments `slide`.

L'élément `para`

Cet élément peut contenir n'importe quel élément de texte ainsi que des éléments de structure basique (listes, tableaux, images, formules).

exemple :

```
<para>Ceci est du texte normal, <em>du texte en évidence</em>,
une URL <url href="http://linux-france.org"/>, etc. </para>
```



L'élément note

Cet élément permet de mettre son contenu en note, il se compose de la même manière qu'un élément `para`.

Il dispose en plus d'un attribut `trainer` qui peut prendre les valeurs « true » et « false » (valeur par défaut). Si la valeur est à « true » alors la note ne sera intégrée que dans un support de cours destiné au formateur (cela peut servir à indiquer des endroits où le formateur doit aborder un thème précis ou faire un exercice particulier, etc.).

L'élément exercise

Cet élément est prévu pour les exercices et les travaux pratiques et se compose de 2 sous-éléments : un élément question et un élément answer (réponse). Ces deux sous-éléments peuvent contenir un ou plusieurs paragraphes. La réponse est optionnelle ; si elle est disponible elle sera incluse à la fin du support de cours dans une section à part.

Exemple :

```
<exercise>
  <question>
    <para>
      combien font 2 plus 2 ?
    </para>
  </question>
  <answer><para>4</para></answer>
</exercise>
```

L'élément glossary

Certains termes compliqués peuvent bénéficier d'une explication. En marquant ce terme avec la balise `glossary`, il sera intégré dans un glossaire global. Le mot référencé est précisé avec l'attribut `name`, et le contenu de la balise est la définition du terme. Si le mot a déjà été défini une première fois il n'est pas nécessaire de remettre la définition.

Exemple :

```
Un réseau <glossary name="IP">Internet Protocol</glossary> est ...
Les technologies <glossary name="IP"/> ...
```

L'élément `xi:include`

Cet élément sert à inclure un autre fichier XML, il permet ainsi de partager des thèmes ou des modules entre formations et/ou à ne pas tout avoir dans un même fichier.

exemple :

```
<xi:include src="fichier_a_inclure.xml"/>
```



Pour que cela soit possible, il est impératif d'avoir défini l'espace de noms *xi* dans une balise XML "parente" de celle-ci. Pour éviter tout problème, on a l'habitude de définir l'espace de nom dans la balise racine du document :

```
<module xmlns:xi="http://www.w3.org/2001/XInclude">  
...  
</module>
```

Cette balise n'est pas définie dans notre DTD pour une simple raison, c'est qu'elle est substituée par le contenu du fichier dès que possible par le parser XML respectant la norme *XInclude*. Il existe différentes options pour l'inclusion d'un fichier. Pour les détails, consultez la norme à l'adresse <http://www.w3.org/TR/xinclude>.



- Les entités pour les caractères spéciaux

Les entités pour les caractères spéciaux

Un certain nombre de caractères ne sont pas accessibles au clavier ou sont réservés en XML, pour pallier cela, certaines entités ont été définies.

- ` ` : l'espace insécable ;
- `&tir;` : le tiret d'intervalle '—';
- `"` : le caractère '"';
- `&` : le caractère '&';
- `<` : le caractère '<';
- `>` : le caractère '>';
- `Œ` : le caractère OE ligaturé (Œ) ;
- `œ` : le caractère oe ligaturé (œ) ;
- `®` : le caractère '®';
- `€` : le caractère pour l'euro (€).



- ❑ La DTD des supports de cours

La DTD des supports de cours

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<!--
# Stephane Casset, Raphael Hertzog, Logidee 2000-2001
# See LICENSE file for copyright notice
-->

<!-- DTD for training documents -->

<!-- Entities -->
<!ENTITY nbsp    " ">
<!ENTITY tir     "—">
<!ENTITY reg     "©">
<!ENTITY quot    """>
<!ENTITY ctrl    "^">
<!ENTITY OElig   "Œ">
<!ENTITY oelig   "œ">
<!ENTITY euro    "€">

<!-- Old entities for compatibilty only, don't use them -->
<!ENTITY ud      "_">
<!ENTITY di      "#">
<!ENTITY dol     "$">
<!ENTITY ti      "~">
<!ENTITY pc      "%">
<!ENTITY bs      "\">
<!ENTITY aco     "{">
<!ENTITY acf     "}">
<!ENTITY circ    "^">
<!ENTITY et      "&">
```



```

<!-- Elements -->
<!ELEMENT formation (info,(xi:include|include|theme)+)>
<!ELEMENT theme (info,(xi:include|include|module)+)>
<!ELEMENT module (info,page+)>
<!ELEMENT slideshow ((info|shortinfo),slide+)>

<!ELEMENT include EMPTY>
<!ATTLIST include href CDATA #REQUIRED>

<!-- Text attributes -->
<!ENTITY % text "#PCDATA">

<!ELEMENT em (%text;)>
<!ELEMENT code (%text;)>
<!ATTLIST code
  visible (true|false) "false">

<!ELEMENT menu      (%text;)>
<!ELEMENT cmd       (%text;)>
<!ELEMENT file      (%text;)>

<!ELEMENT url       (%text;)>
<!ATTLIST url
  href CDATA #REQUIRED>

<!ENTITY % text_attribute "em|code|cmd|menu|file|url">
<!ENTITY % container "%text;|%text_attribute;|list|table|image|math|g

<!-- The info header -->

<!ELEMENT info (title,ref,description,objectives,ratio,duration,prere
<!ELEMENT shortinfo (title,description,version+)>
<!ELEMENT title (%text;|%text_attribute;)*>
<!ELEMENT ref (%text;)>
<!ELEMENT description (para+)>
<!ELEMENT objectives (item+)>
<!ELEMENT ratio EMPTY>
<!ATTLIST ratio
  value CDATA #REQUIRED>
<!ELEMENT duration EMPTY>
<!ATTLIST duration
  value CDATA #REQUIRED
  unit CDATA #REQUIRED>
<!ELEMENT prerequisite (para*)>
<!ELEMENT dependency (ref*)>
<!ELEMENT suggestion (ref*)>
<!ELEMENT version (author,email?,comment,date?)>
<!ATTLIST version
  number CDATA #REQUIRED>

```



```

<!ELEMENT author (%text;)>
<!ELEMENT comment (%text;)>
<!ELEMENT email (%text;)>
<!ELEMENT date (%text;)>
<!ELEMENT level EMPTY>
<!ATTLIST level
  value CDATA #REQUIRED>
<!ELEMENT state EMPTY>
<!ATTLIST state
  finished (true|false) #REQUIRED
  proofread (true|false) #REQUIRED
>
<!ELEMENT proofreaders (item*)>

<!-- The page -->

<!ELEMENT page (title,(slide|para|note)*,(slide|section)*,exercise*)>
<!ATTLIST page
  restriction CDATA "all" >

<!-- The optionnal slide -->

<!ELEMENT slide (title,(subtitle?,para)+)>
<!ATTLIST slide
  background CDATA #IMPLIED>

<!ELEMENT subtitle (%text;|%text_attribute;)*>

<!-- The content of the page (associated with the slide) -->

<!ELEMENT section (title,(section|para|note|exercise|slide)+)>
<!ATTLIST section
  restriction CDATA "all" >

<!ELEMENT para (%container;|note)*>
<!ATTLIST para
  icon CDATA #IMPLIED
  restriction CDATA "all" >

<!ELEMENT legend (%container;)*>

<!ELEMENT glossary (%container;)*>
<!ATTLIST glossary
  name CDATA #REQUIRED >

<!ELEMENT list (item|list)+>
<!ELEMENT item (%container;|para)*>

<!ELEMENT note (%container;|para)*>

```



```

<!ATTLIST note
  trainer      (true|false) "false"
  icon         CDATA #IMPLIED
  restriction  CDATA "all" >

<!ELEMENT image (legend?)>
<!ATTLIST image
  src         CDATA #REQUIRED
  visible    (true|false) "false"
  scale      CDATA "0" >

<!-- Exercises -->

<!ELEMENT exercise (title?,question,answer?)>
<!ATTLIST exercise
  icon         CDATA #IMPLIED
  restriction  CDATA "all" >
<!ELEMENT question (para+)>
<!ELEMENT answer  (para+)>

<!-- Tables -->

<!ELEMENT table (row+)>
<!ATTLIST table
  border (0|1) "1">
<!ELEMENT row   (col+)>
<!ATTLIST row
  border (0|1)  "1">
<!ELEMENT col (%container;)*>
<!ATTLIST col
  border (0|1)  #IMPLIED
  align  (c|l|r) #IMPLIED
  head   (0|1)  "0">

<!-- Math notation -->

<!ELEMENT math (#PCDATA)>
<!ATTLIST math
  notation (tex|html) #REQUIRED>

```



1.2. Les outils logidee-tools

- Vue d'ensemble
- Changements
 - Version 1.2
 - Version 1.0

Vue d'ensemble

Logidee-tools est un ensemble de fichiers (feuilles de style, classe TeX, Makefile, ...) qui sont regroupés dans un répertoire comme `/usr/share/logidee-tools`. Ces makefiles permettent de transformer les fichiers XML supports de formation en fichiers exploitables, c'est-à-dire en PDF directement imprimables et consultables, en HTML prêt à être mis sur le web.

Les documents peuvent être soit intégrés dans l'arborescence de logidee-tools soit dans une arborescence à part. Dans ce second cas, il est nécessaire de préparer l'arborescence des documents afin de pouvoir exploiter logidee-tools. La section "Installer les makefiles" décrira cela en détail.

Changements

Version 1.2

Cette version est bien plus "propre" que la précédente.

- Il n'y a plus besoin de faire attention à utiliser les entités `&ud;` et autres pour s'assurer que le fichier TeX généré soit correct. La feuille de style TeX fait désormais ce travail.
- Toutes les balises ont des noms anglais. La feuille de style `xml/upgrade-1.2.xml` permet de générer des documents conformes à la nouvelle DTD à partir des anciens. Cf la page sur les conversions pour les détails.
- Il est possible de ne rédiger qu'un diaporama avec la balise `slideshow`.
- On peut récupérer des documents docbook-xml dans les outils Logidée avec la feuille de style `xml/db2logidee.xml`. Cf la page sur les conversions.
- On peut générer des documents dans plusieurs langues. Les textes introduits par les outils Logidée (type "Sommaire", "Exercice") sont disponibles en plusieurs langues. Les outils sont internationalisés et prêts à recevoir d'autres traductions.

Version 1.0

Première version publique



- Dépendances
- Pour les utilisateurs de Debian

Dépendances

Pour exploiter la totalité des fonctionnalités des outils Logidée vous devez disposer de :

- xsltproc, le transformateur XSLT de Gnome
- TeTeX (et tetex-extra) pour le support LaTeX
- package euro pour TeTeX (cf tetex-eurosym)
- make
- imagemagick pour convertir des images .eps en .png

Pour les utilisateurs de Debian

Les heureux utilisateurs de Debian disposent de paquets prêts à l'emploi. Il suffit de configurer apt pour pointer vers la source fournie par Logidée et d'utiliser `apt-get install logidee-tools`. Pour rajouter la source Logidée à apt il faut ajouter ces deux lignes au fichier `/etc/apt/sources.list` :

```
deb http://www.logidee.com/tools/ packages/  
deb-src http://www.logidee.com/tools/ packages/
```



- Installer les makefiles
 - Dans l'arborescence logidee-tools
 - Dans une arborescence à part
- Utiliser les makefiles

Installer les makefiles

Dans l'arborescence logidee-tools

Si vous avez une copie de l'arborescence logidee-tools obtenue soit depuis les sources soit simplement en copiant `/usr/share/logidee-tools`, alors vous pouvez placer vos documents XML dans le sous-répertoire `xml` et sans rien modifier, vous n'avez qu'à lancer `make makefiles` depuis la racine de l'arborescence logidee-tools. Vous devriez voir ceci :

```
$ make makefiles
Creating Makefile in doc
Creating Makefile in xml
$
```

Dans une arborescence à part

Si vous avez choisi (et c'est un choix très raisonnable) d'avoir vos documents dans une arborescence à part, alors vous devez préparer cette arborescence afin de pouvoir utiliser logidee-tools sur vos documents XML. Pour effectuer cette tâche vous allez utiliser `setup-logidee-tools` (correspondant à `bin/setup` dans l'archive source). Ce petit script doit être lancé depuis la racine de l'arborescence de documents. Il accepte en paramètre une liste des sous-répertoires contenant des documents XML. Vous pouvez également préciser à l'aide du paramètre « `-r` » le répertoire où sont présents les fichiers de logidee-tools que vous souhaitez utiliser. Si vous ne précisez pas ces paramètres, le script va vous interroger de manière interactive. Deux exemples d'utilisation :

```
$ setup-logidee-tools -r /usr/share/logidee-tools \
  programmation outils reseau
Creating Makefile in programmation
Creating Makefile in outils
Creating Makefile in reseau
$ setup-logidee-tools
Where is the main directory of logidee-tools ?
```



Main directory of logidee-tools: /usr/share/logidee-tools

Which directories may contain xml documents ?

Give a space delimited list of directories : programmation \
outils reseau

Creating Makefile in programmation

Creating Makefile in outils

Creating Makefile in reseau

\$

Note



Dans tous les cas, il est souhaitable que les documents XML soient dans des sous-répertoires de premier niveau de l'arborescence des documents. Ceci est souhaitable pour harmoniser les inclusions de documents et des images. Le détail de ces problèmes est donné dans la partie traitant de la rédaction des documents.

Utiliser les makefiles

Générer les PDF et autres pages HTML n'est qu'une question d'appeler les bonnes cibles des fichiers Makefile que l'on vient d'installer. Voici les cibles génériques qui vous seront les plus utiles :

- make pdf** : génère tous les documents PDF possibles, y compris les diaporamas ;
- make ps** : génère tous les documents PostScript possibles ;
- make 2pp** : génère tous les documents PostScript possibles en mode 2 pages par feuille ;
- make tex** : génère tous les documents LaTeX possibles ;
- make html** : génère tous les documents HTML possibles ;
- make slideshow** : génère tous les diaporamas possibles ;
- make check** : vérifie la validité de l'ensemble des fichiers « *.xml » vis à vis de la DTD ;
- make clean** : supprime tous les fichiers intermédiaires créés mais conserve les .ps, .pdf et .html générés ;
- make realclean** : supprime tous les fichiers intermédiaires et tous les fichiers générés.

Voici une revue d'ensemble des autres cibles disponibles :

- make formations** : génère tous les supports de cours « formation*.pdf » à partir des documents « formation*.xml » ;
- make formations_show** : génère tous les diaporamas « formation*_slideshow.pdf » à partir des documents « formation*.xml » ;
- make formations_2pp** : génère tous les supports de cours « formation*_2pp.ps » en version 2 pages par feuille à partir des documents « formation*.xml » ;
- make themes** : génère tous les supports de cours « theme*.pdf » à partir des documents « theme*.xml » ;
- make themes_show** : génère tous les diaporamas « theme*_slideshow.pdf » à partir des documents « theme*.xml » ;



- ❑ `make themes_2pp` : génère tous les supports de cours « `theme*_2pp.ps` » en version 2 pages par feuille à partir des documents « `theme*.xml` » ;
- ❑ `make fichier` : génère le support de cours « `fichier.pdf` » et le diaporama « `fichier_slideshow.pdf` » à partir de « `fichier.xml` » ;
- ❑ `make fichier.pdf` : génère le PDF adéquat à partir de « `fichier.xml` » ;
- ❑ `make fichier_2pp.ps` : génère le fichier « `fichier_2pp.ps` » avec 2 pages par feuille à partir de « `fichier.xml` » ;
- ❑ `make module.html` : génère les pages HTML « `module.html/*_html` » à partir de « `module.xml` ». Le répertoire `module.html` est prêt à l'emploi, vous pouvez le copier sur votre site web sans rien changer ;
- ❑ `make module.txt` : génère une version texte « `module.txt` » à partir de « `module.xml` » ;
- ❑ `make fichier.xml-check` : vérifie la validité de « `fichier.xml` ».

Il est possible de définir des variables d'environnement pour altérer la génération des documents.

- ❑ `CHARTER` : valeur par défaut « `default` ». Cette variable permet de choisir la charte graphique utilisée pour générer les documents.
- ❑ `TRAINER` : valeur par défaut « `false` ». En positionnant cette variable à « `true` » vous incluez les notes réservées aux formateurs.
- ❑ `SELECTION` : valeur par défaut « `all` ». En positionnant cette variable à un mot-clé vous n'incluez que les données spécifiques à ce mot-clé (cf attribut "restriction") et laisserez de côté les données spécifiques aux autres mots-clés.
- ❑ `CYCLE` : valeur par défaut « `false` ». En positionnant cette variable à « `true` », vous indiquez que la présentation doit utiliser en cycle une série de fonds d'écran indiquées dans la charte graphique.
- ❑ `LANG` : valeur par défaut « `en` » (langue anglaise). En positionnant cette variable à « `fr` » vous pouvez générer des documents totalement en langue française.

Exemple :

```
$ make formations SELECTION=debian TRAINER=true LANG=fr
```



- Avec un éditeur de texte
- Ce qu'il faut savoir en rédigeant
 - Les chemins d'accès aux fichiers
 - Le format des images

Avec un éditeur de texte

Les documents XML ne sont que de simples documents textes. Il suffit donc d'un éditeur de texte pour les rédiger. Toutefois il est quelque peu rébarbatif de saisir régulièrement les mêmes balises. C'est pourquoi il est intéressant de disposer de quelques raccourcis clavier lors de la saisie.

Pour les utilisateurs de vim 6 et supérieur, vous trouverez quelques fichiers dans le répertoire `vim/`. Installez ces fichiers de la manière suivante :

- `xml.vim => ~/.vim/ftplugin/xmllogidee.vim`
- `syntax.vim => ~/.vim/syntax/xmllogidee.vim`
- `filetype.vim => ~/.vim/filetype.vim`

Ces fichiers fournissent une coloration syntaxique adaptée pour le XML et un ensemble de raccourcis clavier pour le mode insertion. Tous les raccourcis commencent par une virgule. Ainsi « `,xm` » permet d'insérer la ligne d'en-tête XML et « `,in` » insère un bloc d'en-tête info. Consultez le fichier `xml.vim` pour découvrir les autres raccourcis.

Ce qu'il faut savoir en rédigeant

Les chemins d'accès aux fichiers

Il faut être particulièrement vigilant avec les chemins d'accès aux fichiers aussi bien lorsque vous incluez des images (avec la balise `image`) que des documents (avec la balise `include`). En effet, le jeu des inclusions peut changer le "répertoire courant" et fausser les données.

Imaginez un répertoire `module` contenant tous les modules et un répertoire `theme` contenant l'ensemble des thèmes. Supposez que `theme/theme_un.xml` inclue un module avec une balise `include` similaire à celle ci :

```
<xi:include href="../../../module/sujet.xml"/>
```

Supposez également que le module `sujet.xml` inclue une image `schema.eps` (présente dans le répertoire `module`) avec un élément :



```
<image src="schema.eps"/>
```

Que va t'il se passer lorsque l'on va essayer de transformer `theme_un.xml` ? Et bien `theme_un.xml` va contenir une balise image listant `schema.eps` qui n'est pas disponible dans le répertoire `theme` et il va y avoir une erreur. D'où la nécessité de mettre des chemins d'accès aux fichiers plus explicites même si l'image est dans le même répertoire que le module :

```
<image src="../../module/schema.eps"/>
```

Le format des images

Les images qui sont incluses dans les documents XML doivent être au format « .eps ». Même si ce format ne convient pas pour toutes les sorties (HTML notamment), c'est le format de référence choisi. L'image sera automatiquement convertie en PNG pour les besoins de la version HTML.



- Passage de la version 1.0 à 1.2
- Récupération de documents docbook-xml

Passage de la version 1.0 à 1.2

Entre la version 1.0 et 1.2, la DTD des outils Logidée a été traduite en anglais. Afin de continuer à exploiter les documents rédigés dans l'ancienne DTD, il est fourni une feuille de style pour générer des documents valides selon la nouvelle DTD à partir des anciens fichiers. Il s'agit de `xsl/upgrade-1.2.xsl`. Pour générer un nouveau document à partir d'un ancien, il suffit de se placer dans le répertoire racine de vos documents et de lancer :

```
$ xsltproc xsl/upgrade-1.2.xsl xml/ancien.xml > xml/nouveau.xml
```

Récupération de documents docbook-xml

Si vous avez des documents existants en docbook-xml et que vous souhaitez migrer vers le format exploité par les outils Logidée, vous pouvez utiliser la feuille de style `xsl/db2logidee.xsl`. Une fois placé dans le répertoire des outils Logidée vous pouvez utiliser une commande similaire à celle ci :

```
$ xsltproc xsl/db2logidee.xsl xml/docbook.xml > xml/logidee.xml
```

Docbook-xml étant une DTD très vaste, il est possible (voire probable) que la feuille de style ne convertisse pas parfaitement le document. Il vous faudra reprendre manuellement le document généré sachant que l'essentiel du travail aura déjà été fait.



- Les chartes graphiques
- Localisation d'une charte

Les chartes graphiques

Une charte graphique de nom "<charte>" est constitué :

- d'une classe TeX avec le fichier `charte/<charte>/tex/logidoc.cls`. Vous pouvez personnaliser ce fichier pour adapter à vos besoins la sortie .pdf des supports de cours et des présentations ;
- d'une feuille de styles CSS (`charte/<charte>/default.css`) utilisée pour la sortie HTML ;
- de plusieurs fichiers HTML, `charte/<charte>/footer.html.<lang>` pour les pieds de page HTML, et `charte/<charte>/html/<lang>/*.html` pour des fichiers additionnels (au contenu fixe) ;
- d'un jeu d'icônes dans le répertoire `charte/<charte>/icones/` ;
- d'un document XML `charte/<charte>/charte.xml` donnant les correspondances entre les noms symboliques des icônes et le nom réel des icônes présentes dans le répertoire précédemment cité ;
- et d'un autre document XML `charte/<charte>/messages.xml` listant les traductions du vocabulaire utilisé par les outils.

Par défaut, c'est la charte graphique nommée « default » qui est utilisée. Si la mise en page des sorties n'est pas satisfaisante pour vos besoins, il est possible de l'adapter en créant une nouvelle charte graphique dérivée de celle par défaut. Il suffit de recopier les fichiers ci-dessus et de les modifier selon les besoins.

Deux chartes sont fournies avec logidee-tools. Voici leurs caractéristiques :

- default* est une charte "restrictive" qui inclue des mentions "Reproduction interdite" et qui d'une manière générale réserve tous les droits aux auteurs. Le logo utilisé est le célèbre pingouin Tux.
- gfdl* est une charte "libre" qui inclue des mentions signalant que le document est disponible sous la licence *Gnu Free Documentation License*. Le logo utilisé est le gnou, symbole du projet GNU.



Localisation d'une charte

Si vous souhaitez localiser une charte dans une nouvelle langue, c'est possible. Il suffit de :

- ❑ rajouter les traductions dans `charte/<charte>/messages.xml` ;
- ❑ fournir des fichiers `charte/<charte>/html/<lang>/*.html` et `charte/<charte>/footer.html.<lang>` ;
- ❑ modifier `charte/<charte>/tex/messages.inc` pour rajouter un bloc `\DeclareOption{<lang>}{ ... }` similaire à ceux déjà présents ;
- ❑ modifier `charte/<charte>/tex/logidoc.cls` pour inclure le paquet TeX correspondant à la nouvelle langue pour Babel dans la ligne `\usepackage[french, english]{babel}` ;
- ❑ modifier `Makefile.generic` et `xsl/charte.xsl` afin d'autoriser une nouvelle valeur pour les paramètres indiquant la langue.

N'hésitez pas à nous envoyer les traductions que vous faites afin de les inclure dans la version officielle des outils Logidée.

