



Intro XUL

La plate-forme Mozilla



Firefox

Thunderbird

XULRunner

SeaMonkey

Camino

Sunbird and
Lightning

Embedding

...

The Mozilla Platform

Toolkit

Extension Manager, Update, Moz Storage, Spell Checking, Brakepad Crash Reporting, ...

Content

Layout

XUL

XML User Interface Language

XBL

XML Binding Language

SVG

Scalable Vector Graphics

DOM

Document Object Model

CSS

Cascading Style Sheets

HTML and XML Parser

NSS / PSM

Network Security Services, Personal Security Manager

XPCOM

Cross Platform Component Object Model

XPCoconnect

Bridges JavaScript and XPCOM

JavaScript

NSPR

Netscape Portable Runtime: Cross Platform API for System Level Functions

Necko
Network
Library

Widget
Event
Handling and
Windowing

GFX /
Thebes
Graphics

Cairo
Graphics

SQLite
Storage



Extension vs. Application

- XulRunner ?
 - Moteur Gecko + API Mozilla
- Firefox, Thunderbird, Camino, Sunbird...basés sur XulRunner depuis Gecko 1.9 (~ Firefox 3.0).
- XulRunner est de plus en plus présent sur les postes de travail.
- Deux possibilités avec XUL :
 - Applications standalone dans XulRunner
 - Utiliser les overlays pour développer des extensions sur les clients existants (Firefox, Thunderbird...).



URL chrome

- Organisation des répertoires d'une « application » chrome :

```
monappli/  
  chrome/  
    content/monappli.xul  
    locale/  
      fr-FR/monappli.dtd  
      en-US/monappli.dtd  
    skin/global/monappli.css  
  defaults/  
    preferences/prefs.js  
  chrome.manifest
```

- Tous les fichiers du répertoire chrome sont accessibles par une URL

```
chrome://monappli/content/monappli.xul  
chrome://monappli/locale/monappli.dtd  
chrome://monappli/skin/monappli.css
```

```
content tp1helloworld file:content/  
skin tp1helloworld global file:skin/global/  
skin tp1helloworld neofuture file:skin/neofuture/  
locale tp1helloworld fr-FR file:locale/fr-FR/  
locale tp1helloworld en-EN file:locale/en-EN/
```



URL chrome

- Exemple de fichier chrome.manifest

```
content exo file:content/
```

```
# main.xul dans file:content/ => chrome://exo/content/main.xul
```

```
locale exo fr-FR file:locale/fr-FR/
```

```
locale exo en-EN file:locale/en-EN/
```

```
#locale.dtd dans file:locale/fr-FR/ => chrome://exo/locale/locale.dtd
```

```
#locale.dtd dans file:locale/en-EN/ => chrome://exo/locale/locale.dtd
```



XUL

- **XML User interface Language**
 - Langage de description de GUI
 - Exclusivement XML
 - Large éventail de composants
 - Gestionnaire d'évènements intégré
 - Accessible comme un DOM HTML
- Puissance du moteur Gecko
 - Rendu graphique de nombreux formats avec le moteur Cairo
 - Avec les namespaces : combinaison de plusieurs langages en un fichier
 - Par exemple : XUL + HTML + MathML + SVG



XUL : Hello World !

- Les bases :

```
<?xml version="1.0"?>
<?xml-stylesheet href="chrome://global/skin/" type="text/css"?>
<window id="main"
  xmlns="http://www.mozilla.org/keymaster/gatekeeper/there.is.only.xul"
  title="helloworld"
  width="300"
  height="300">
<caption label="Hello World !!"/>
</window>
```



XUL : La localisation

- Localiser son application :

```
<?xml version="1.0"?>
<!DOCTYPE overlay SYSTEM "chrome://monappli/locale/monappli.dtd">
<?xml-stylesheet href="chrome://global/skin/" type="text/css"?>
<window id="main"
  xmlns="["...]"
  title="helloworld"
  width="300"
  height="300">
<caption label="&monappli.hellotext;"/>
<caption label="&monappli.worldtext;"/>
</window>
```

- Contenu du fichier monappli.dtd

```
<!ENTITY monappli.hellotext "Hello...">
<!ENTITY monappli.worldtext "...World !!">
```




XUL : Le skin

- Les applications XUL ont le skin système par défaut

```
<?xml-stylesheet href="chrome://global/skin/" type="text/css"?>
```

- Possibilité de skinner tous les composants XUL avec CSS !

```
<?xml version="1.0"?>
<?xml-stylesheet href="chrome://monappli/skin/monappli.css" type="text/css"?>
<window id="main"
  xmlns="[]"
  title="helloworld"
  width="300"
  height="300">
  <caption id="mytext" label="Hello World !!"/>
  <button class="okbtn" label="OK"/>
</window>
```

- Fichier **monappli.css**

```
#mytext{
  color:red;
  font-weight:bold;
}
.okbtn{
  border:1px solid green;
}
```



XUL : Les overlays

- Les overlays dans XUL permettent de modifier dynamiquement un DOM XUL au chargement.
 - Utilisé pour les extensions Firefox/Thunderbird.
 - Nécessite de connaître l'ID de l'élément XUL à côté duquel on veut s'insérer (facile avec le DOM Inspector).
- Exemple d'overlay :

```
<?xml version="1.0"?>
<overlay xmlns="[...]">
<toolbox id="navigator-toolbox">
  <toolbar id="monextension-toolbar"
    toolbarname="Toolbar Name"
    class="chrome-class-toolbar"
    hidden="false"
    persist="hidden">
    <toolbarbutton label="Label 1" />
    <toolbarbutton label="Label 2" />
    <toolbarbutton label="Label 3" />
  </toolbar>
</toolbox>
</overlay>
```



XUL : Les datasources et les templates (1/2)

- Possibilité de « peupler » ses composants XUL à partir de sources de données externes : XML, RDF, BDD SQLite.
- *Fichier XML à mapper*

```
<people>
  <person name="Napoleon Bonaparte" gender="male"/>
  <person name="Cleopatra" gender="female"/>
  <person name="Julius Caesar" gender="male"/>
  <person name="Ferdinand Magellan" gender="male"/>
  <person name="Laura Secord" gender="female"/>
</people>
```

- *Code XUL pour générer le composant*

```
<listbox datasources="people.xml" ref="*" querytype="xml">
<template>
  <query expr="person"/>
  <action>
    <listitem uri="?" label="?name"/>
  </action>
</template>
</listbox>
```



XUL : Les datasources et les templates (2/2)

- Mapping RDF natif

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#" xmlns:dc="http://purl.org/dc/elements/1.1/">
<rdf:Seq rdf:about="http://www.xulplanet.com/rdf/myphotos">
  <rdf:li rdf:resource="http://www.xulplanet.com/ndeakin/images/t/palace.jpg"/>
  <rdf:li rdf:resource="http://www.xulplanet.com/ndeakin/images/t/canal.jpg"/>
  <rdf:li rdf:resource="http://www.xulplanet.com/ndeakin/images/t/obelisk.jpg"/>
</rdf:Seq>
<rdf:Description rdf:about="http://www.xulplanet.com/ndeakin/images/t/palace.jpg" dc:title="Palace from Above"/>
<rdf:Description rdf:about="http://www.xulplanet.com/ndeakin/images/t/canal.jpg" dc:title="Canal"/>
<rdf:Description rdf:about="http://www.xulplanet.com/ndeakin/images/t/obelisk.jpg" dc:title="Obelisk"/>
</rdf:RDF>
```

- *Code XUL pour générer le composant*

```
<vbox datasources="template-guide-ex2.rdf" ref="http://www.xulplanet.com/rdf/myphotos">
<template>
  <query>
    <content uri="?start"/>
    <member container="?start" child="?photo"/>
    <triple subject="?photo" predicate="http://purl.org/dc/elements/1.1/title" object="?title"/>
  </query>
  <action>
    <vbox uri="?photo" align="start">
      <image src="?photo"/>
      <label value="?title"/>
    </vbox>
  </action>
</template>
</vbox>
```



XPCOM

- Les XPCOM sont les objets compilés de l'API Mozilla.
- Bibliothèque d'objets très fournie qui peut être enrichie.
- Tous les objets du framework mozilla décrivent leurs interfaces en IDL.
- Ils sont accessibles en JS/Python/Perl.

Core

[Base Types](#)
[Component Management](#)
[Files](#)

[Data Structures](#)
[XPCOM](#)
[Streams](#)

[Enumerations](#)

User Interface

[Auto Complete](#)
[Selection](#)

[Clipboard](#)
[Transactions](#)

[Drag and Drop](#)
[XUL](#)

Application Development

[Application Info](#)
[Command Lines](#)
[Extensions and Update](#)
[Window Management](#)
[Web Browser](#)

[Places](#)
[Profiles](#)
[Preferences](#)
[History](#)
[Spell Checking](#)

[Web Feeds](#)
[Storage](#)
[RDF](#)
[XML](#)
[XML Schema](#)

Network

[Base Network](#)

[Channels](#)

[Protocol Handlers](#)
[Proxies](#)
[ZIP and JAR Files](#)

[About Handlers](#)

[Cookies](#)

[Sockets](#)
[URL Classification](#)

[Cache](#)
[Download and Content Handling](#)
[Stream Converters](#)
[Web Services](#)

DOM

[W3C DOM](#)
[DOM SVG](#)

[DOM Level 3](#)
[DOM XUL](#)

[DOM HTML](#)
[Mozilla Specific DOM](#)

Mail

[General Mail](#)

[IMAP](#)

[Searching Messages](#)
[News](#)

[Mail Storage](#)

[MailUI](#)

[Mail Import](#)
[LDAP](#)

[POP3](#)
[Message Composition and Sending](#)
[Address Book](#)

Mozilla Applications

[Browser](#)
[Editor](#)

[Form fill-in](#)
[JavaScript Debugger](#)

[DOM Inspector](#)

Other Components

[Accessibility](#)

[Content and Layout](#)
[JavaScript](#)
[XPInstall](#)

[Security and Privacy](#)

[Graphics and Widgets](#)
[Localization](#)

[Docshell and Content Viewers](#)
[Images](#)
[Plugins](#)



XBL (1/2)

- Langage de description d'objets XUL
- Permet d'enrichir la librairie de composants existante.

- XUL (example.xul):

```
<?xml version="1.0"?>
<?xml-stylesheet href="chrome://global/skin/" type="text/css"?>
<?xml-stylesheet href="chrome://example/skin/example.css" type="text/css"?>
<window xmlns="http://www.mozilla.org/keymaster/gatekeeper/there.is.only.xul">
  <okcancelbutton label="mon bouton OK CANCEL"/>
</window>
```

- CSS (example.css):

```
okcancelbuttons { -moz-binding: url('chrome://example/skin/example.xml#okcancel'); }
```

-



XBL (2/2)

```
<?xml version="1.0"?>
<bindings xmlns="http://www.mozilla.org/xbl" xmlns:xul="[...] ">
<binding id="okcancel">
  <content>
    <xul:button xbl:inherits="label"/>
    <xul:button label="Annuler"/>
  </content>
  <implementation>
  <constructor>
    this.setAttribute('loaded','true');
  </constructor>
  <method name="init">
  <param name="loop"/>
  <body>
    for(var i=0; i<loop; i++){
      window.dump(loop);
    }
  </body>
  </method>
  </implementation>
  <handlers>
    <handler event="click" action="alert('OK');"/>
  </handlers>
</binding>
</bindings>
```