

# Diffusion de données depuis une base XML - Projet OiDB

Retour d'expérience

Patrick Bernaud (JMMC)

# Plan

- eXist-db - base de données XML
- OiDB - présentation

# eXist-db - Introduction

Base de données orientée document développée depuis 2001 sous LGPL 2.1 (<http://www.exist-db.org/>).

Version 2.2 sortie la semaine dernière.

- ☛ amélioration performances
- ☛ mise à jour modèle de sécurité
- ☛ nouveau indexes

# eXist-db - Caractéristiques générales

Écrit en Java.

Jeu de servlets nécessitant un conteneur (Jetty).

Stocke des documents XML ou binaires organisés en collections dans un file system virtuel.

# eXist-db - Caractéristiques générales (2)

- ☛ solution NoSQL / sans schéma
- ☛ XML
- ☛ rapid prototyping
- ☛ application web “embarquée” en XQuery

# eXist-db - Interfaces

- 🐛 WebDAV
- 🐛 XML-RPC
- 🐛 XML:DB
- 🐛 SOAP
- 🐛 REST

# eXist-db - Index

Une recherche du genre `//foo[bar="xyz"]` nécessite scan complet de la collection → définition d'index adaptés à l'application.

Plusieurs types d'index : Range, Full Text, Spatial, xml:id, Structural...

Configuration par des documents XML dans chaque collection.

# eXist-db - Index (example)

```
<collection xmlns="http://exist-db.org/collection-config/1.0">
  <index>
    <lucene>
      <text qname="SPEECH">
        <ignore qname="SPEAKER"/>
      </text>
      <text qname="TITLE"/>
    </lucene>

    <range>
      <create qname="title" type="xs:string"/>
      <create qname="author" type="xs:string"/>
      <create qname="year" type="xs:integer"/>
    </range>
  </index>
</collection>
```

# eXist-db - XQuery

Langage programmation fonctionnelle  
orienté recherche/transformation/création  
documents structurés ou non structurés  
(extensions).

Recommandation W3C depuis 2007,  
version 3.0 - Avril 2014.

Similaire à XSLT sur l'aspect  
transformation, en moins verbeux.

Utile pour manipuler de larges collection  
de documents.

# eXist-db - XQuery (2)

- High Order Function
- Pure function
- XPath
- FLWOR expression: FOR, LET, WHERE, ORDER BY, RETURN

# eXist-db - XQuery (example)

```
for $book in doc("catalog.xml")/books/book
let $title := $book/title/text()
let $price := $book/price/text()
where xs:decimal($price) gt 50.00
order by $title
return
  <book>
    <title>{$title}</title>
    <price>{$price}</price>
  </book>
```

# eXist-db - Services

- XUpdate (non standard)
- XSLT
- Security Manager
- Scheduler
- templating
- trigger
- RESTXQ
- XForms
- eXide

# eXist-db - Security Manager

File system-like, permissions sur le modèle UNIX.

ACL (Access Control List) pour gestion plus fines des droits.

Authentication par mot de passe, LDAP, OpenID, OAuth.

# eXist-db - Scheduler

Permet d'exécuter du code régulièrement, en tâche de fond à la manière d'un cron.

Scripts en XQuery ou en Java, avec droits administrateur.

# eXist-db - Templating

Module permettant de scripter la vue :  
insertion, tests simples, substitution  
depuis le modèle.

⊖ Limité, verbeux.

# eXist-db - Triggers

Exécution de code déclenchée sur des événements sur les documents de la base.

Déclaration en XML, script en XQuery ou en Java.

# eXist-db - RESTXQ

RESTXQ 1.0: RESTful Annotations for XQuery 3.0.

Jeu d'annotations à ajouter à une fonction XQuery (resource function).

“Routage” automatique, serialization automatique.

# eXist-db - RESTXQ (example)

```
declare
  %rest:GET
  %rest:path("/oidb/keyword")
  %rest:query-param("q", "{$q}")
  %output:media-type("application/json")
  %output:method("json")
function kw:list($q as xs:string*) {
  let $all := collection($kw:keywords-uri)//keyword
  let $keywords := if (exists($q)) then
    $all[contains(upper-case(.), upper-case($q))]
  else
    $all
  return <keywords> {
    for $kw in $keywords
    return <json:value>{ $kw/text() }</json:value>
  } </keywords> };
```

# eXist-db - eXide

IDE dans le navigateur.

Fourni avec eXist-db.

Coloration syntaxique (XQuery, HTML, JavaScript), validation, exécution de code.

# eXist-db - eXide (2)

The screenshot displays the eXide web interface within a browser window. The browser's address bar shows the URL `localhost:8080/exist/apps/eXide/index.html`. The interface includes a top navigation bar with the eXide logo and a menu of links. Below this is a toolbar with options like 'New', 'New XQuery', 'Open', 'Save', 'Close', 'Run', and 'Check'. The main area is divided into two panes. The left pane is a file explorer showing a tree view of application modules, including `$app:base-upload`, `$app:collections-query`, `$app:data-pis-query`, `$app:facilities-query`, `$app:instruments-query`, `$app:latest-query`, `$app:main-metadata`, `$app:oifits-query`, `$app:UCD_URL`, `app:add-attribute`, `app:adsbib-url`, `app:collection`, `app:collection-granules`, `app:collection-stats`, `app:collections`, `app:collections-options`, `app:column-sort`, `app:data-pis`, `app:data-stats`, `app:deserialize-query-string`, `app:doc`, `app:each-granule`, `app:each-row`, `app:ellipsisize`, `app:facilities`, `app:format-access-url`, `app:format-collection-url`, `app:format-mjd`, `app:format-wavelengths`, `app:get-data`, `app:get-obfuscated-email`, `app:granules`, `app:homepage-header`, `app:input-each-band`, `app:input-user-email`, `app:input-user-name`, `app:instruments`, `app:latest`, `app:public-status`, `app:row-cells`, `app:row-data`, `app:search`, and `app:serialize-query-`. The right pane is an XQuery editor showing a file named `app.xml` with the following code:

```
386 templates:process($node/$node(), $map:map($model, 'map:entry' data, '3'))
387 }
388 }
389
390 (:~
391 ::Return an element with counts of the number of observations, all OIFits
392 ::files and private OIFits files in the database.~
393 ::
394 ::@param $params a sequence of parameters
395 ::@return a <stats> element with attributes for counts.~
396 ::)
397 declare %private function app:data-stats($params as xs:string*) as node() {
398   let $base-query :=
399     adql:clear-pagination(-
400       adql:clear-select-list(-
401         adql:clear-order(-
402           adql:clear-filter($params, 'public')))-
403       let $count := function($q) { tap:execute($q)//*/TD/text() }
404       (: FIXME 3 requests... nasty, nasty :)
405       return <stats> {
406         attribute {"noobservations"} { $count(adql:build-query(($base-query, 'count=*')))-
407         attribute {"nprivatefiles"} { $count(adql:build-query(($base-query, 'count=*', 'public=no')))-
408         (: FIXME even worse... can you believe it? :)
409         attribute {"noifitsfiles"} { $count('SELECT COUNT(*) FROM (| | adql:build-query(($base-query, 'distinct', 'col=access_url') | |) AS urls') }-
410       }
411     }
412 }
413 (:~
414 ::Display the result of the query in a paginated table.~
415 ::
416 ::The query is passed to a TAP service and the returned VOTable
417 ::content is put in the model for further template processing.~
418 ::
419 ::@param $node
420 ::@param $model
421 ::@param $page offset into query result (page * perpage)
422 ::@param $perpage number of results displayed per page
423 ::@param $all display all columns or only a subset
424 ::@return a new model with search results for presentation
425 ::)
426 declare
427   %templates:default("page", 1)
428   %templates:default("perpage", 25)
429 function app:search($node as node(), $model as map(),
430   $page as xs:integer, $perpage as xs:integer, $all as xs:string?) as map(*) {
431   try {
432     (: Search database, use request parameters :)
433     (: clean up pagination stuff, recovered later from function parameters :)
434     let $params := adql:clear-pagination(adql:split-query-string())
435     let $votable := tap:execute(
436       adql:build-query(
```

# eXist-db - Web Applications

Packaging de modules XQuery avec format prédéfini (MVC).

Serialization en HTML5

# eXist-db - XForms

Solution générique d'écriture de formulaires.

Pas de support natif d'XForms dans les navigateurs → Betterform (server side) ou XSLForms (client side).

Échange de fragments XML avec eXist.

⊗ Intéressant mais mise en route difficile (eXist-db 2.1). À réévaluer.

# OiDB

Portail de diffusion de metadonnées observationnelles en interférométrie optique avec interface Observatoire Virtuel.

Projet JMMC.

Début: décembre 2013 à partir prototype.

Prochainement version 1.0 en accès libre.

# OiDB (2)

- ✎ extraction de metadonnées /  
indexation
- ✎ interface Web de consultation
- ✎ interface Web de soumission
- ✎ interface Web d'administration
- ✎ interopérabilité VO

# OiDB - Data Model

ObsCore DM: modèle de données adapté aux observations en astronomie.

Ajout de colonnes spécifiques au domaine (extensions).

Définition d'une granule: une target, un instrument, une nuit.

# OiDB - Interopérabilité

- 🐛 ObsCore DM
- 🐛 TAP = Table Access Protocol. Standard IVOA.
- 🐛 Langage de requête (ADQL) proche du SQL.
- 🐛 SAMP.
- 🐛 Registry: TODO.

# OiDB - Extraction

Formats multiples: OIFITS, observation logs.

Sources multiples: fichiers locaux, fichiers distants, base de données externes...

Pour les OIFITS, utilisation outils existants  
JMMC : écriture module interface/wrapper  
Java → XQuery.

# OiDB - Data storage

Dual storage:

- SQL pour la table dérivée ObsCore
- XML pour les données textuelles.

Seules les données SQL sont requêtables en TAP.

# OiDB - Authentication

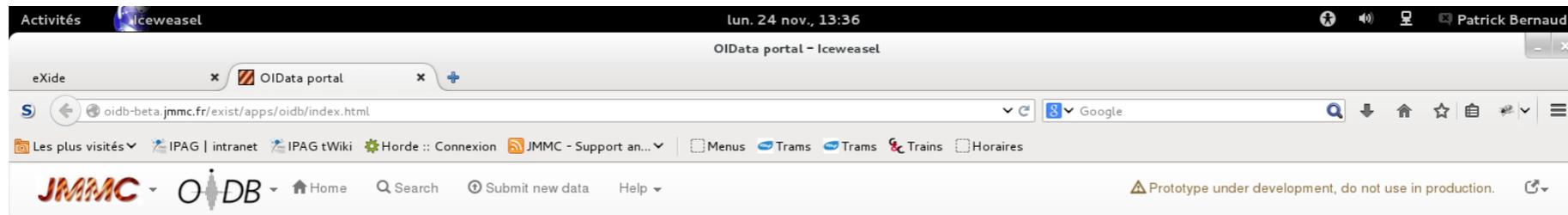
En attendant LDAP, écriture module interface avec la base utilisateur JMMC existante.

Utilisation du Security Manager intégré → Persistent login pour l'interface Web, HTTP Basic Auth pour le REST headless.

# OiDB - Tools

- 🐙 RESTXQ
- 🐙 Bootstrap
- 🐙 jQuery

# OiDb - Screenshots



## Optical Interferometry DataBase



Enter target name or [visit the advanced form](#)

### Beta release under preparation !

A Beta release instance will replace the [current public version](#) in the coming weeks. It's database content is not shared so the working group can test without restriction. In any case, please wait our announcement before using and referencing any data from the oidb portal.

# OiDb - Screenshots (2)

The screenshot shows the OiDb search interface. At the top, there is a navigation bar with the JMMC logo and 'O-IDB' text. Below this is a search bar and a 'Submit new data' button. The main content area is titled 'Filters' and contains several search criteria:

- Position:** A text input field with 'Name or coordinates', a dropdown menu set to 'J2000', and a 'Radius' input field set to '2' with a unit dropdown set to 'arcmin'.
- Date of observation:** A 'between' range selector with two 'YYYY-MM-DD' input fields and a calendar icon.
- Instrument:** A dropdown menu set to 'Any Instrument'.
- Wavelength range:** A series of checkboxes for spectral bands: U, B, V, R, I, J, H, K, L, M, N, Q. Below these are checkboxes for 'Visible', 'Near infrared', and 'Mid infrared'.
- Collection:** A dropdown menu set to 'Any Collection'.
- DataPI name:** A dropdown menu set to 'Any DataPI'.
- Data reduction level:** Checkboxes for 'L0', 'L1', 'L2', and 'L3'. 'L1', 'L2', and 'L3' are checked.
- Availability:** Radio buttons for 'Public', 'Restricted', and 'All'. 'All' is selected.

Below the filters, there is a 'Sort by' dropdown set to 'Instrument' with a 'descending' checkbox checked. To the right, there is a 'Max rows per page' dropdown set to '25'. At the bottom of the filter section, there are 'Search' and 'Reset' buttons.

**Results**

Meta-data will try to follow VO4OI proposal and Ivoa:ObsCore document (get metadata description in the associated doc)  
5804 observations from 5434 oifits files (1268 private)

Page 1 / 233 [Next](#) [Last](#)

Results for `SELECT ALL * FROM oidb AS t WHERE ( t.calib_level=1 OR t.calib_level=2 OR t.calib_level=3 )`  
( [Edit query](#) )

target_name	access_url	t_min	instrument_name	em_min	em_max
-------------	------------	-------	-----------------	--------	--------

# OiDb - Screenshots (3)

Activités Iceweasel lun. 24 nov., 13:45 Patrick Bernaud

OIData portal - Iceweasel

oidb-beta.jmmc.fr/exist/apps/oidb/collection.html?id=J%2FA%2BA%2F536%2FA55

Les plus visités IPAG | intranet IPAG tWiki Horde :: Connexion JMMC - Support an... Menus Trams Trams Trains Horaires

JMMC O-IDB Home Search Submit new data Help Prototype under development, do not use in production. Logged in.

## Milli-arcsecond imaging of SS Lep (Blind+, 2011)

### Description

**Name:** J/A+A/536/A55  
**Title:** Milli-arcsecond imaging of SS Lep (Blind+, 2011)  
**Description:**

The observations made use of the VLTI instruments AMBER and PIONIER. The data are reduced. AMBER ones were obtained from archive data of ESO (program IDs: 082.D-0019(A); 082.D-0019(B); 082.D-0019(D); 082.D-0062(A); 082.D-0019(E); 083.D-0028(A)). PIONIER data were obtained during the commissioning of the instrument. Data cover eight different epochs. They allowed to compute the orbit of the system and the size of the giant star.

### Articles:

**Title:** An incisive look at the symbiotic star SS Leporis. Milli-arcsecond imaging with PIONIER/VLTI  
**Authors:** Blind, N. Boffin, H. M. J. Berger, J.-P. Le Bouquin, J.-B. Mérand, A. Lazareff, B. Zins, G.  
**Publication date:** 2011-12-01  
**Keywords:** stars: AGB and post-AGB accretion accretion disks binaries: spectroscopic stars: fundamental parameters techniques: interferometric binaries: symbiotic

### Granules

Page 1 / 1

Target	Instrument	Instrument mode	Night
<a href="#">AMBER_070409.fits</a>			
ss-lep - 06:04:59.150 -16:29:03.984	VLTI - AMBER		2009-04-06 23:31:12 - 2009-04-06 23:45:35
ss-lep - 06:04:59.150 -16:29:03.984	VLTI - AMBER		2009-04-10 00:23:02 - 2009-04-10 00:23:02
<a href="#">AMBER_111108.fits</a>			
SS-Lep - 06:04:59.143 -16:29:03.984	VLTI - AMBER		2008-11-11 06:11:31 - 2008-11-11 07:01:55
SS-Lep - 06:04:59.143 -16:29:03.984	VLTI - AMBER		2008-11-13 06:33:07 - 2008-11-13 07:58:04
SS-Lep - 06:04:59.143 -16:29:03.984	VLTI - AMBER		2008-11-11 07:40:47 - 2008-11-11 07:40:47
<a href="#">AMBER_210209.fits</a>			

# Questions?