EXPath
A practical introduction

XQuery Meetup, December 3d, 2009
Paris

Florent Georges

fgeorges.org

EXPath
A practical introduction

- **Introduction** ←
- The project
- HTTP Client
- ZIP facility
- Packaging
- Putting it all together
- Sibling projects

Introduction

- XPath is a textual language to navigate XDM trees
- It is used standalone or embedded in XSLT, XQuery, XProc, XForms, and other languages
- A recommendation defines a standard library of functions
- An implementation or host language can provide additional functions: the *extension functions*

Introduction

- More and more demand for extensions for XSLT 2.0 and XQuery for one year
- Extension functions are the easiest way
- They are at the XPath level
- Acting at the XPath level allows them to be used in another languages
- XProc is a good example of such another language (EXProc defines also extensions for a few months)

EXPath
A practical introduction

- Introduction
- **The project** ←
- HTTP Client
- ZIP facility
- Packaging
- Putting it all together
- Sibling projects

The project

- Collaborative
- The base delivery unit is the *module*
- The main deliverable is the specification
- Each module has its own maintainer
- Implementations as external extensions are encouraged during specification
- Independent on any particular vendor
- ...though they are welcome, of course ;-)

The project

- Extension functions libraries
- But also:
  - Testing framework (based on Jeni's XSpec?)
  - Documentation system (based on Ken's XSLStyle?)
  - General-purpose packaging system
  - Help identifying best practices
  - Servlet-like container
  - ...

EXPath
A practical introduction

- Introduction
- The project
- **HTTP Client** ←
- ZIP facility
- Packaging
- Putting it all together
- Sibling projects

HTTP Client

- Send HTTP requests and handle responses
- Based on the XProc step p:http-request
- Enable one to:
  - Retrieve plain resources on the web
  - Query REST-based Web sevices
  - Query SOAP-based Web services
  - Query Google services

HTTP Client

```
http:send-request($request as element(http:request)) as item()+
```

HTTP Client

```
http:send-request($request as element(http:request)) as item()+


<http:request href="http://www.example.com/..." method="post">
   <http:header name="X-Header" value="some value"/>
   <http:body content-type="application/xml">
      <hello>World!</hello>
   </http:body>
</http:request>
```

HTTP Client

```
http:send-request($request as element(http:request)) as item()+
```

```
<http:request href="http://www.example.com/..." method="post">
  <http:header name="X-Header" value="some value"/>
  <http:body content-type="application/xml">
    <hello>World!</hello>
  </http:body>
</http:request>
```

```
<http:response status="200" message="Ok">
  <http:header name="..." value="..."/>
  ...
  <http:body content-type="application/xml"/>
</http:response>
```

HTTP Client

```
http:send-request(
   <http:request href="http://www.balisage.net/" method="get"/>)

↘

(

   ?
)
```

HTTP Client

```
http:send-request(
   <http:request href="http://www.balisage.net/" method="get"/>)

↘

(
 <http:response status="200" message="OK">
    <http:header name="Server" value="Apache/1.3.41 (Unix) ..."/>
    ...
    <http:body content-type="text/html"/>
 </http:response>

’
   ?

)
```

HTTP Client

```
http:send-request(
   <http:request href="http://www.balisage.net/" method="get"/>)

↘

(
 <http:response status="200" message="OK">
    <http:header name="Server" value="Apache/1.3.41 (Unix) ..."/>
    ...
    <http:body content-type="text/html"/>
 </http:response>

,
 <html xmlns="http://www.w3.org/1999/xhtml">
    <head>
       <title>Balisage: The Markup Conference</title>
       ...
)
```

HTTP Client

- Live samples:
  - XQuery with Saxon, MarkLogic and eXist
  - Google's GData API
  - WSDL Compiler

EXPath
A practical introduction

- Introduction
- The project
- HTTP Client
- **ZIP facility** ←
- Packaging
- Putting it all together
- Sibling projects

ZIP facility

- Read and write ZIP files
  - List all entries
  - Extract specific entries
  - Update existing entries
  - Create brand-new ZIP files
- Well suited for XML + ZIP documents (OpenDocument, Open XML, EPUB, etc.)

ZIP facility

- List entries:
  - zip:entries($href) as element(zip:file)
- Extract entries:
  - zip:xml-entry($href, $path) as document-node()
  - zip:html-entry($href, $path) as document-node()
  - zip:text-entry($href, $path) as xs:string
  - zip:binary-entry($href, $path) as xs:base64Binary
- Create new ZIP files:
  - zip:zip-file($zip) as empty()
  - zip:update-entries($zip, $output) as empty()

ZIP facility

```
<zip:file href="some.zip" xmlns:zip="http://www.expath.org/mod/zip">
   <zip:entry name="file.xml" output="xml">
      <hello>World!</hello>
   </zip:entry>
   <zip:entry name="index.html" output="html" href="/some/file.html"/>
   <zip:dir name="dir">
      <zip:entry name="file.txt" output="text">
         Hello, world!
      </zip:entry>
   </zip:dir>
</zip:file>
```

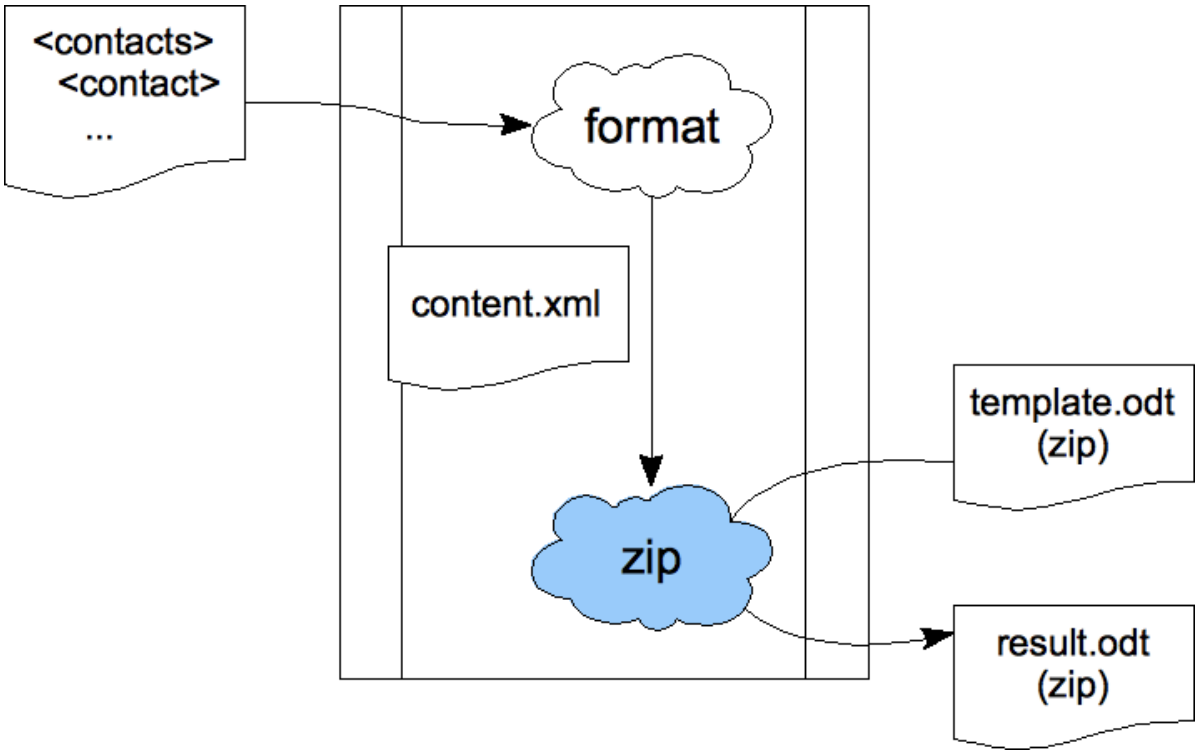
ZIP facility

- Live samples:
  - List entries in a ZIP file
  - Extract an XML entry
  - Create a new file

ZIP facility

- “Compound Document Template” pattern
- Create a new ZIP file by copying an existing one, updating and adding some entries
- For instance, you can edit a text document directly within OpenOffice, and use it as a template to format an input data document

ZIP facility



EXPath
A practical introduction

- Introduction
- The project
- HTTP Client
- ZIP facility
- **Packaging** ←
- Putting it all together
- Sibling projects

Packaging

- Support XSLT, XQuery and XProc
- Can be extended for other X* technologies
- Independent on the processor for std X* files
- Allow processor-dependent features (i.e. for Java extension functions)
- Support only deploying libraries
- Can be used as a building block for more complex frameworks, like XRX

Packaging

- Deployment descriptor:

```
<package xmlns="http://expath.org/mod/expath-pkg">
  <module version="0.1" name="google-xslt">
    <title>Simple XQuery package for tests</title>
    <xsl>
      <import-uri>http://www.fgeorges.org/google/gdata.xsl</import-uri>
      <file>xsl/gdata.xsl</file>
    </xsl>
    ...
  </module>
</package>
```

Packaging

- Example of use:

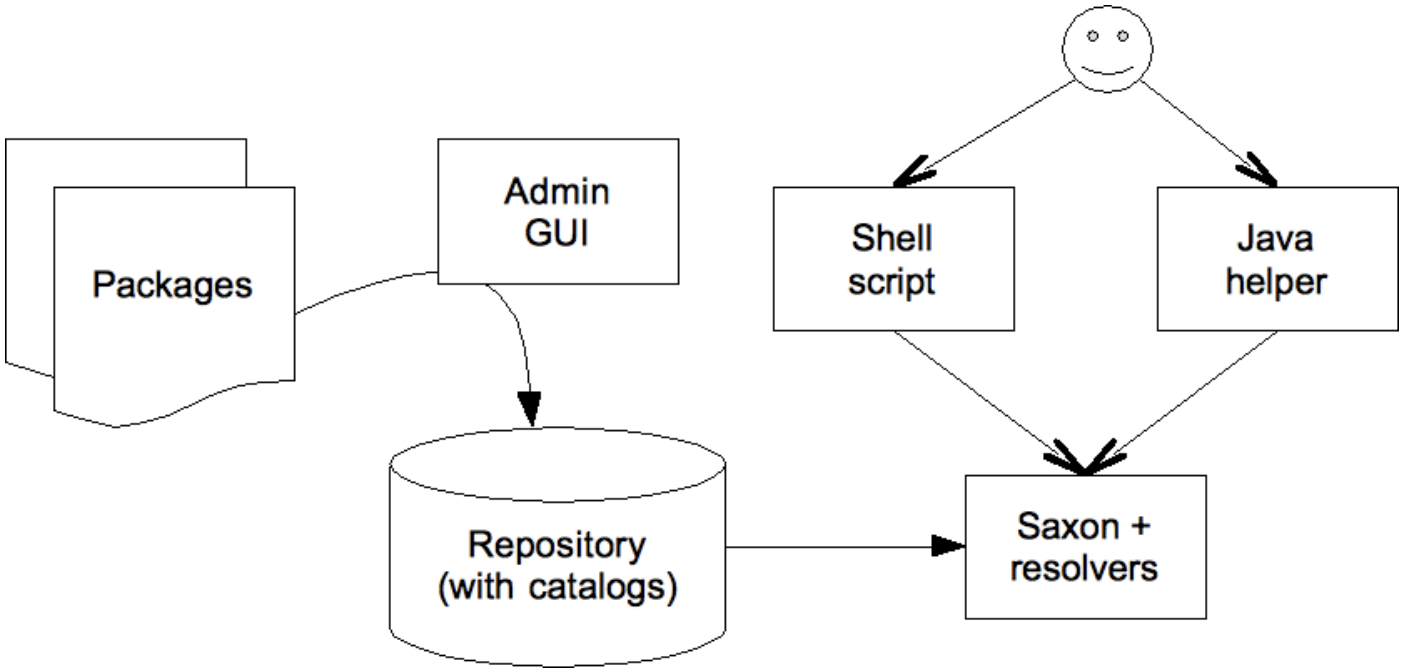
```
<xsl:stylesheet xmlns:f="http://fxsl.sf.net/" version="2.0">

  <xsl:import href="http://fxsl.sf.net/f/func-Fibonacci.xsl"/>

  <xsl:template match="/" name="main">
    <fibo>
      <xsl:value-of select="f:fibo(10)"/>
    </fibo>
  </xsl:template>

</xsl:stylesheet>
```

Packaging


Packages
Admin GUI
Shell script
Java helper
Repository (with catalogs)
Saxon + resolvers

Packaging

- Examples:
  - Deploy XSLT and XQuery for Saxon
  - Deploy Java extensions for Saxon
  - Deploy XQuery for eXist
  - Deploy Java extensions for eXist

Packaging

- For now, use an external application to deploy
- Limited to the processor's mechanism to resolve URIs
- For some processors, not possible to deploy without changing importing stylesheets/queries
- Ideal situation: supported natively by a broad number of processors
- Is a support for other frameworks, and CXAN

Packaging

- In XSLT

```
<!-- the public and absolute import URI -->
<xsl:import href="http://www.expath.org/mod/http-client.xsl"/>
```

- In XQuery? There is no convention.

```
import module namespace
   http = "http://www.expath.org/mod/http-client"
   at "http://www.expath.org/mod/http-client.xq";
```

- versus -

```
import module namespace
   http = "http://www.expath.org/mod/http-client";
```

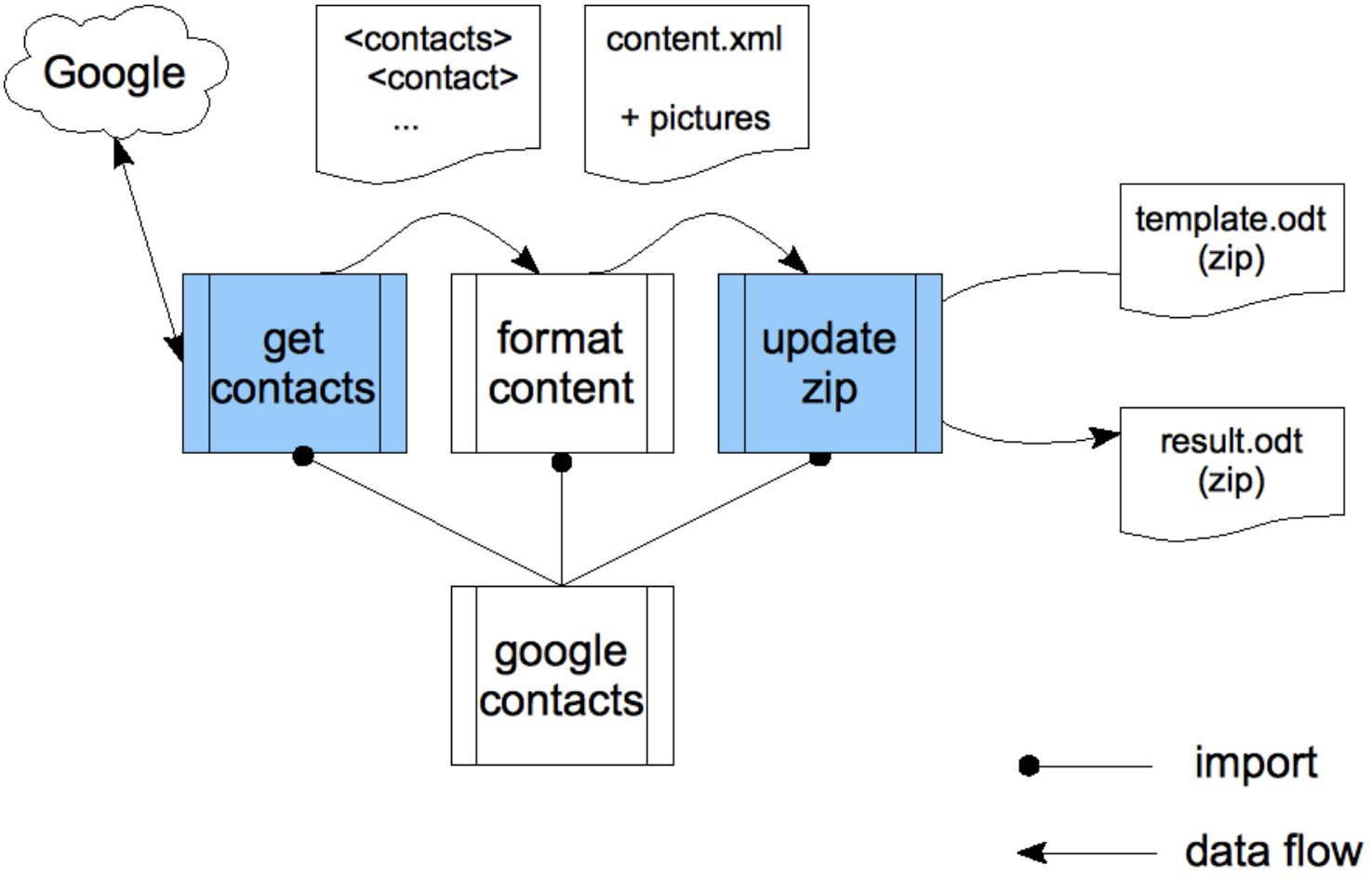
EXPath
A practical introduction

- Introduction
- The project
- HTTP Client
- ZIP facility
- Packaging
- **Putting it all together** ←
- Sibling projects

Putting it all together

- Use the HTTP Client to access various Google APIs (REST Web services): contacts, maps, ...
- Use ZIP facility and the Compound Document Template pattern
- All those libraries are accessed through the Packaging System

Putting it all together



EXPath
A practical introduction

- Introduction
- The project
- HTTP Client
- ZIP facility
- Packaging
- Putting it all together
- **Sibling projects**

←

Sibling projects

- EXQuery, EXSLT 2.0, and EXProc
- Where's the border?
  - EXPath Packaging system
  - Servlet-like container definition
  - Full XRX container definition

That's all Folks!

- Plenty of other potential extensions
- More low-level and general-purpose: nested sequences and first-class function items
- Join the mailing list and browse the website:

http://www.expath.org/