





`cvs update -dP`

## APIs disponibles

Las especificaciones de BioMOBY sólo regulan el protocolo de comunicaciones que deben seguir MOBY-Centra y los clientes y servicios BioMOBY, y cómo interactúan entre sí. Eso implica que la única restricción para emplear un lenguaje de programación a la hora de codificar los servicios y clientes BioMOBY es la existencia de una API que haga de interfaz con el medio de comunicación y el protocolo. Actualmente hay implementadas en BioMOBY APIs para 3 lenguajes de programación: Perl (clientes y servicios), Java (sólo para clientes) y Python (clientes y servicios).

## Instalación de la API de Perl

Dentro del directorio `moby-live` hay un subdirectorio llamado `Perl`, que contiene la API de BioMOBY para Perl. Para instalar esta API, primero hay que instalar los módulos de Perl `RDF::Core`, `SOAP::Lite` y `XML::DOM`. Una forma sencilla de realizarlo es mediante CPAN, como administrador de la máquina. En el caso del paquete `SOAP::Lite`, si se va a emplear alguna de sus características opcionales, como conexiones encriptadas o usarlo en colaboración con `mod_perl` de Apache, hay que configurarlo explícitamente:

```
root# perl -MCPAN -e shell
cpan> install RDF::Core
cpan> install XML::DOM
cpan> install SOAP::Lite
```

We are about to install SOAP::Lite and for your convenience will provide you with list of modules and prerequisites, so you'll be able to choose only modules you need for your configuration.

XMLRPC::Lite, UDDI::Lite, and XML::Parser::Lite are included by default. Installed transports can be used for both SOAP::Lite and XMLRPC::Lite.

Press <enter> to see the detailed list.

Feature	Prerequisites	Install?
Core Package	[*] URI [*] MIME::Base64 [*] XML::Parser (v2.23)	always
Client HTTP support	[*] LWP::UserAgent	[ yes ]
Client HTTPS support	[ ] Crypt::SSLeay	[ no ]
Client SMTP/sendmail support	[ ] MIME::Lite	[ no ]
Client FTP support	[*] IO::File [*] Net::FTP	[ yes ]
Standalone HTTP server	[*] HTTP::Daemon	[ 062 324.2104 Tm(Ne)Tj 79 0 08811e79 220.12328



Do you plan to use SSL Support for TCP Transport? [no]  
 Do you plan to use Compression support for HTTP? [yes]  
 Do you plan to use MIME interoperability w/ Axis? [no] **yes**

Feature	Prerequisites	Install?
Core Package	[*] URI [*] MIME::Base64 [*] XML::Parser (v2.23)	always
Client HTTP support	[*] LWP::UserAgent	[ yes ]
Client HTTPS support	[ ] Crypt::SSLeay	[ no ]
Client SMTP/sendmail support	[ ] MIME::Lite	[ no ]
Client FTP support	[*] IO::File [*] Net::FTP	[ yes ]
Standalone HTTP server	[*] HTTP::Daemon	[ yes ]
Apache/mod_perl server	[ ] Apache	[ no ]
FastCGI server	[ ] FCGI	[ no ]
POP3 server	[ ] MIME::Parser [*] Net::POP3	[ no ]
IO server		

## Instalación de la API de Python

Dentro del directorio `moby-live` hay un subdirectorio llamado `Python`, que contiene el código fuente correspondiente a las APIs de BioMOBY para Python. Para compilar las APIs es necesario tener instalados Python 2.3 o superior (con sus correspondientes librerías de desarrollo), el compilador C (normalmente gcc), el paquete PyXML (las librerías XML para Python) (<http://pyxml.sourceforge.net/>), el paquete PyIPC (<http://www.theory.org/~mac4/software/>), el paquete `fpconst` (antes en <http://www.analytics.washington.edu/Zope/projects/fpconst>), el paquete `SOAPpy` y el paquete `ZSI 1.5` o superior (disponibles en <http://pywebsvcs.sourceforge.net/>). Este último paquete tiene un problema en la versión 1.5.0 (que es la versión actual) con la decodificación de secciones `CDATA` dentro de XML, por lo que el autor de la API de Python para BioMOBY ha incluido en `moby-live/Python/ZSI-1.5.0-patched` una versión parcheada de la misma.

```
root# mkdir /tmp/pypack && cd /tmp/pypack (instalando PyIPC)
root# wget http://www.theory.org/~mac4/PyIPC.tar.gz
root# gunzip -c PyIPC.tar.gz | tar xf -
root# cd PyIPC
root# python setup.py build
root# python setup.py install
root# cd .. (instalando fpconst)
root# wget ftp://ftp.rediris.es/mlrro/gentoo/distfiles/fpconst-0.6.0.tar.gz
root# gunzip -c fpconst-0.6.0.tar.gz | tar xf -
root# cd fpconst-0.6.0
root# python setup.py build
root# python setup.py install
root# cd .. (instalando SOAPpy)
root# wget http://heanet.dl.sourceforge.net/sourceforge/pywebsvcs/SOAPpy-0.11.4.tar.gz
root# gunzip -c SOAPpy-0.11.4.tar.gz | tar xf -
root# cd SOAPpy-0.11.4
root# python setup.py build
root# python setup.py install
root# cd ../../moby-live/Python (instalando ZSI parcheado)
root# cd ZSI-1.5.0-patched
root# python setup.py build
root# python setup.py install
root# cd .. (instalando APIs de BioMOBY para Python)
root# python setup.py build
root# python setup.py install
```

Una vez ejecutado el último comando ya están disponibles las librerías MOBY para Python.

## Instalación de Taverna

Taverna es una herramienta tipo workbench producida para el proyecto myGrid, que emplea dentro de su infraestructura BioMOBY. Esta herramienta permite construir y comprobar el funcionamiento de flujos de ejecución con servicios MOBY, y muy básicamente se puede emplear para depurar un servicio MOBY. La herramienta se encuentra disponible en <http://taverna.sourceforge.net/>. Para usarla es necesario tener instalado Java2 JRE o Java2 SDK 1.4 o superior. Aunque es fácil de instalar, hay que transformar el script de arranque de la misma para que funcione todo bien:

```
root# mkdir /tmp/download && cd /tmp/download
root# wget http://belnet.dl.sourceforge.net/sourceforge/taverna/workbench-
win32+linux-0.1beta10.zip
root# mkdir /usr/local/taverna && cd /usr/local/taverna
```



```
root# unzip /tmp/download/workbench-win32+linux-0.1beta10.zip
root# chmod +rx runme.sh
root# dos2unix runme.sh (evita problemas con algunos shells)
root# cd /usr/local/bin
root# ln -s /usr/local/taverna/runme.sh taverna
```