Open-VA 1.1

Tabla de contenidos

Capítulo 1. Introducción

Bienvenido al manual de usuario de Open-VA 1.1.

Este documento explica cómo instalar, configurar y usar Open-VA.

Características principales

Open-VA es un código abierto que demuestra cómo implementar un sistema para validar certificados digitales y documentos firmados.

Puede usarlo y extenderlo como desee.

Aviso

Si usa Open-VA, o la distribuye, le agradeceríamos que incluyera el logo *powered by Open-VA* (ver más abajo). Puede encontrar este logo (`open-VA.gif`) en su directorio de instalación de Open-VA. (Ver *documento licencia* para las condiciones de uso)

Open-VA usa un *WebService* como frontal y un protocolo muy simple como transporte.

Puede usar Open-VA para obtener el archivo `WSDL` y construir sus propios clientes Open-VA.

Acerca de este manual de usuario

Este manual le da una visión completa de Open-VA. Explica cómo instalar, configurar y usar Open-VA y asume que está familiarizado con su sistema operativo y conceptos relacionados con servidores de aplicaciones, servidores de bases de datos y servidores ldap.

El manual de usuario de Open-VA está formado por:

- *Capítulo 1 - Introducción*: Lo está leyendo
- *Capítulo 2 - Instalación*: Define requerimientos de plataforma y cómo preparar los componentes requeridos antes de construir e instalar Open-VA.
- *Capítulo 3 - Usando Open-VA*: Describe el protocolo Open-VA y cómo usarlo.

Capítulo 2. Instalación

Este capítulo explilca requerimientos de sistema y plataforma y procedimientos de instalación. También provee instrucciones para obtener los componentes requeridos.

Requerimientos de instalación

Requerimientos de plataforma

Open-VA es una aplicación *multiplataforma* que *no requiere* nada en especial de su plataforma.

Componentes requeridos

Los componentes requeridos por Open-VA (no suministrados) son:

- Java Software Development Kit (*JSDK*)
- Java Enterprise Edition Application Server (*servidor JEE*)
- Lighweight Directory Access Protocol server (*servidor LDAP*)
- Database server

Requerimiento JSDK

Se necesita una JSDK versión 1.4. Los archivos de política de seguridad sin restricción deben estar instalados y BouncyCastle debe estar registrado como un proveedor de seguridad.

Debe asegurar que su sistema tiene una variable de entorno llamada `JAVA_HOME` que ha sido definido con el directorio de instalación de su JSDK.

Aviso

Esta versión no funciona con otra versión diferente a la 1.4.

Instalando la versión JSDK 1.4 de SUN

Puede descargarla desde SUN download website [http://java.sun.com/j2se/1.4.2/download.html].

Después de la instalación recuerde definir la variable de entorno `JAVA_HOME` tal y como ya se ha explicado en la sección anterior.

Instalando los Archivos de Política sin Restricción

Por defecto la descarga de JSDK incluye archivos de política restringidos, los cuales no permiten el uso de claves de gran longitud. Para permitir claves de cualquier longitud debe instalar los archivos de política sin restricción.

Descárguelos desde JCE download website [http://java.sun.com/j2se/1.4.2/download.html#docs] y siga las instrucciones.

Registrando BouncyCastle

Open-VA requiere BouncyCastle como proveedor de seguridad. Para poder hacerlo debe seguir estos pasos:

1. Descargar e instalar las librerías BouncyCastle

2. Registrar BouncyCastle com un proveedor de seguridad

Puede descargar las librerías BouncyCastle (`bcprov.jar` and `bcmail.jar`) desde BouncyCastle download website [http://www.bouncycastle.org/latest_releases.html]. Tras ello copie ambos archivos en la carpeta `jre/lib/ext` bajo el directorio de instalación de su JSDK.

Tras la descarga e instalación debe registrar BouncyCastle como un proveedor de seguridad editando el archivo de seguridad de su JSDK. Este archivo es `java.security` y lo puede encontrar en la carpeta `jre/lib/security` bajo el directorio de instalación de JSDK.

Aviso

SUN debe ser el primer proveedor y BouncyCastle el segundo. Busque las entradas security.provider y edítelas como sigue:

```
... otras entradas ...

security.provider.1=sun.security.provider.Sun
security.provider.2=org.bouncycastle.jce.provider.BouncyCastleProvider
security.provider.3=com.sun.net.ssl.internal.ssl.Provider
security.provider.4=com.sun.rsajca.Provider
security.provider.5=com.sun.crypto.provider.SunJCE
security.provider.6=sun.security.jgss.SunProvider

... otras entradas ...
```

Requerimiento servidor JEE

Es requerido un servidor compatible con JEE 1.4.

Con esta entrega de Open-VA se usa un servidor de aplicaciones JBoss. Si usted desea usar otro, deberá suministrar los descriptores usted mismo.

Instalando el servidor JBoss

Puede descargar un servidor JBoss (4.02 o mayor recomendado) desde JBoss download website [http://labs.jboss.com/portal/jbossas/download/index.html].

Requerimiento servidor LDAP

No se requiere ningún servidor LDAP en especial.

Instalando OpenLDAP

Si no dispone de ningún servidor LDAP puede usar OpenLDAP. Este servidor puede ser descargado desdeOpenLDAP download website [http://www.openldap.org/software/download/]

Tras la descarga e instalación debe configurar el dominio y las credenciales. Esto puede realizarse editando su archivo de configuración (`slapd.conf`) el cual se encuentra en el directorio de instalación de OpenLDAP.

Editar `suffix` para los datos del dominio y `rootdn` y `rootpw` para los datos de usuario.

Ejemplo de configuración estableciendo el dominio *Open-VA.org* y *admin* como nombre de usuario administrador y *adminpwd* como su constraseña:

```
... otras entradas ...

suffix          "dc=Open-VA,dc=org"
rootdn          "cn=admin,dc=Open-VA,dc=org"
rootpw          adminpwd

... otras entradas ...
```

Importante

Recuerde usar *nombres distinguidos* para describir objetos LDAP.

Requerimiento servidor base de datos

No se requiere ningún servidor de base de datos en especial.

Instalando el servidor MySQL

Si no dispone de ningún servidor de base de datos, puede usar MySQL.. Este servidor puede ser descargado desde MySQL download website [http://dev.mysql.com/downloads/mysql/4.1.html].

Para acceder al servidor MySQL desde una aplicación Java debe descargar el conector para Java. Este archivo puede ser descargado desde MySQL connector website [http://dev.mysql.com/downloads/connector/j/3.1.html]. Este conector tiene que estar en el classpath de la aplicación como se describe más abajo (*Preparando JBoss*).

Aviso

En el momento de este documento el conector Java para MySQL se encuentra todavía en desarrollo. Debido a esto no intente instalar la útilma entrega del servidor MySQL e instale la versión 4.

Instalando Open-VA

Antes de instalar Open-VA debe construir y preparar las herramientas requeridas anteriormente.

Preparando herramientas

Nota

Algunos archivos para las herramientas se encuentran en la carpeta `resources` en el directorio de instalación de Open-VA.

Preparando LDAP

Añadir esquemas *java* y *user* a la instalación por defecto.

Para hacerlo debe copiar `user.schema`, que se encuentra en `resources`, en la carpeta `schemas` de OpenLDAP.

Tras ello registre los esquemas en OpenLDAP editando su archivo de configuración `slapd.conf`

```
... otras entradas ...

include         ./schema/java.schema
include         ./schema/user.schema

... otras entradas  ...
```

LDAP se usa para almacenar certificados aceptados para que la aplicación puede conocer la ruta de validación y el estado de revocación.

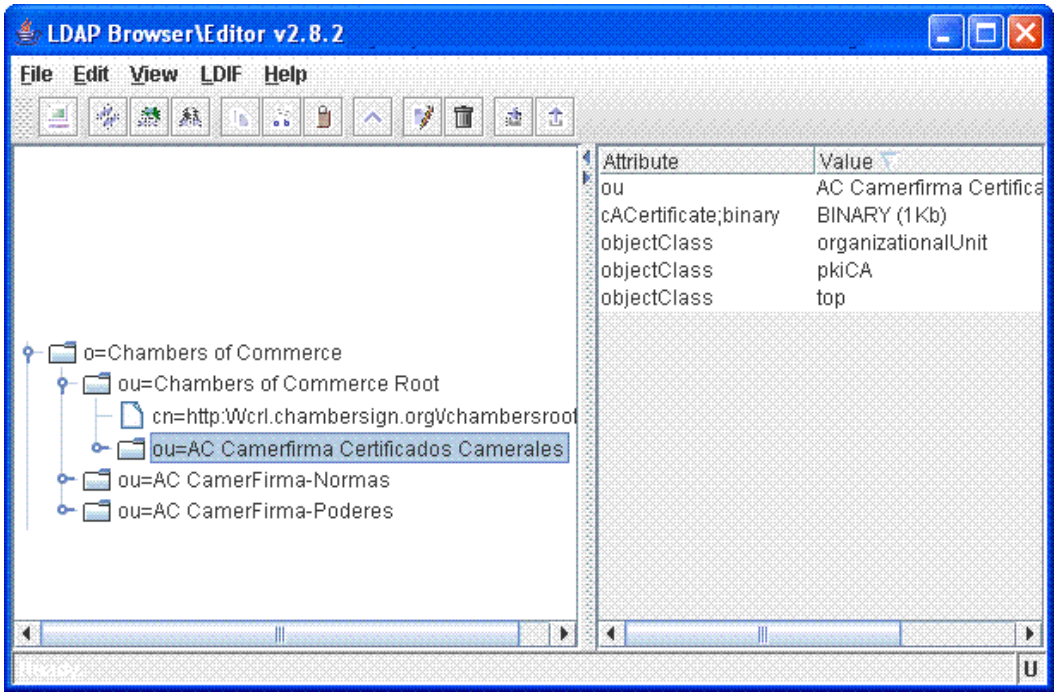
Debe almacenarlo como sigue:

- Crear una *organization (entityClass = o)* sólo para agrupar certificados.
- Almacene sus certificados en *organizational units (entityClass = ou)* del tipo *pkiCA* y con un atributo binario llamado *caCertificate* cuyo valor sea el certificado.

Nota

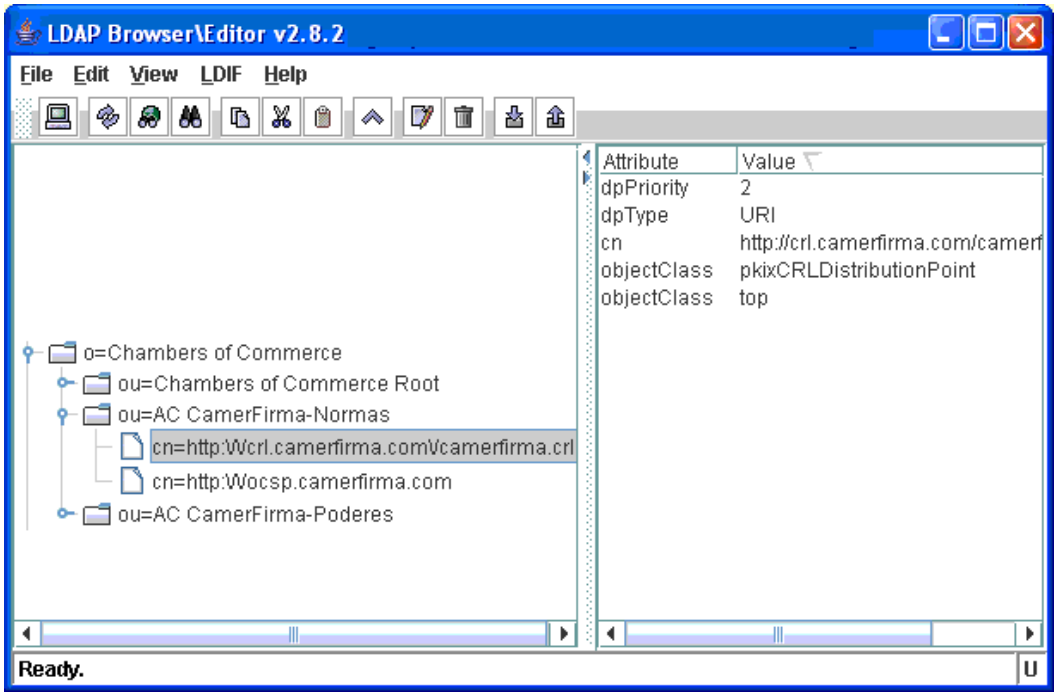
Se recomienda, aunque no es obligado, respetar la jerarquía de certificados en una manera legible.

Un ejemplo de entrada CA:

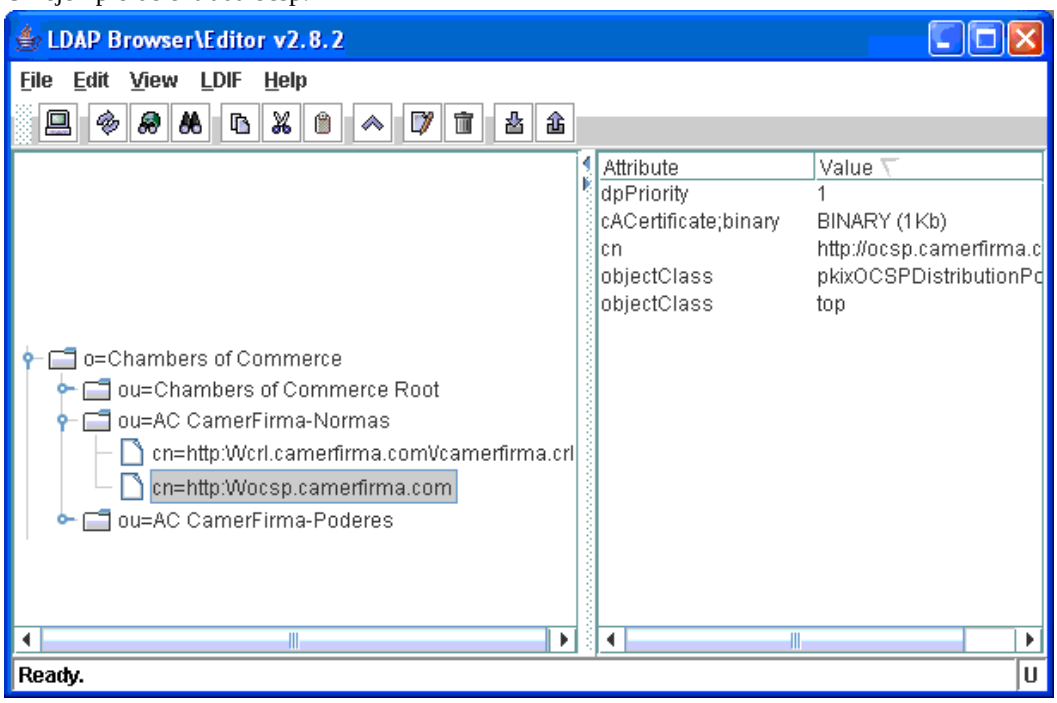


- Cada CA almacena información de revocación. Dicha información incluye métodos *CRL* y *OCSP* de revocación. Para cada CA varios métodos se pueden configurar dando a cada uno una prioridad. Para crl sólo la url completa es necesaria. En el caso de ocsp además el certificado debe ser incluido para poder validar la respuesta del servidor.

  Un ejemplo de entrada crl:



Un ejemplo de entrada ocsp:



Preparando el servidor de base de datos

Debe crear un esquema de base de datos para usar con Open-VA. En `resources` puede encontrar un guión `sql` con instrucciones para crear y llenar las tablas.

Hemos creado para usted dos guiones para los más populares servidores de base de datos: MySQL y Oracle.

En este guión se asume un esquema con nombre *EPSILON*. Puede cambiarlo pero no olvide actualizar el guión.

Preparando JBoss

Nota

- Debería definir la variable de entorno `JBOSS_HOME` con el valor del directorio de instalación de JBoss si desea desplegar Open-VA automáticamente.
- Asumimos que usaremos el servidor default.
- Copie los controladores o conectores para su servidor de base de datos (no suministrados) en la carpeta `lib`

Atención

Necesita un archivo de certificado en formato `.pfx` o `.p12`, el cual no se incluye porque depende de su organización. Este archivo debe copiarse a la carpeta `conf` del servidor default de JBoss.

En `resources` tiene varios arhivos que debe copiar en las carpetas de JBoss. Antes de copiarlos debe editar algunas propiedades con sus propios valores:

- Archivos que deben ser copiados en la carpeta `conf`
  - `KeyStoreConfiguration.properties`

    No se preocupe por él
  - `ValidationConfiguration.properties`

    No se preocupe por él
  - `LDAPLoginConfiguration.properties`

Nombre de propiedad	Descripción de la propiedad
providerURL	La dirección de su servidor LDAP
principal	Un nombre de usuario con derechos de consulta en su servidor LDAP
credentials	La contraseña

  - `pkiva.webservices.properties`

Nombre de propiedad	Descripción de propiedad
signature.ks	Su archivo `.pfx` o `.p12`
signature.ks.password	Constraseña de su archivo de certificados
signature.alias.name	Nombre de usuario del certificado de firma de respuestas
signature.alias.password	Constraseña del usuario del certificado de firma de respuestas

- Archivos que deben copiarse en la carpeta `deploy`

  - `open-VA-XXX-db-ds.xml`

    Nota

    Escoja XXX entre mysql u oracle, en función de su servidor de base de datos. Si tiene otro servidor deberá crear su propio conector.

    Este archivo configura una *fuente de datos (datasource)* para la base de datos a usar en Open-VA. Debe respetar la entrada *JNDI name* (*EPSILON*) y editar la propiedad *JDBC* con el valor correcto para su sistema.

  - `open-VA-crl_ra-ds.xml`

    No se preocupe por él

  - `open-VA-ldap_ra-ds.xml`

    No se preocupe por él

  - `open-VA-ocsp_ra-ds.xml`

    No se preocupe por él

Construyendo Open-VA

Encontrará un guión para construir Open-VA. Su nombre es `build.cmd` para sistemas Windows y `build.sh` para sistemas UNIX.

Tras ejecutar este guión Open-VA estará construida como una *aplicación JEE* con nombre open-VA.ear.

Desplegando Open-VA

Si el servidor JBoss está detenido, arránquelo.

Si ha definido la variable de entorno `JBOSS_HOME` puede ejecutar **build deploy** y Open-VA será automáticamente desplegada. Si no ha definido esta variable de entorno deberá copiar manualmente el archivo `open-VA.ear`. que ha sido creado en la carpeta `dist` de su instalación Open-VA, a la carpeta `deploy` de JBoss. Si está usando *default* será: `server/default/deploy`.

Tras el despliegue puede comprobar en el log del servidor (`log/server.log`) que el archivo para la aplicación Open-VA (`open-VA.ear`) ha sido desplegado y arrancado.

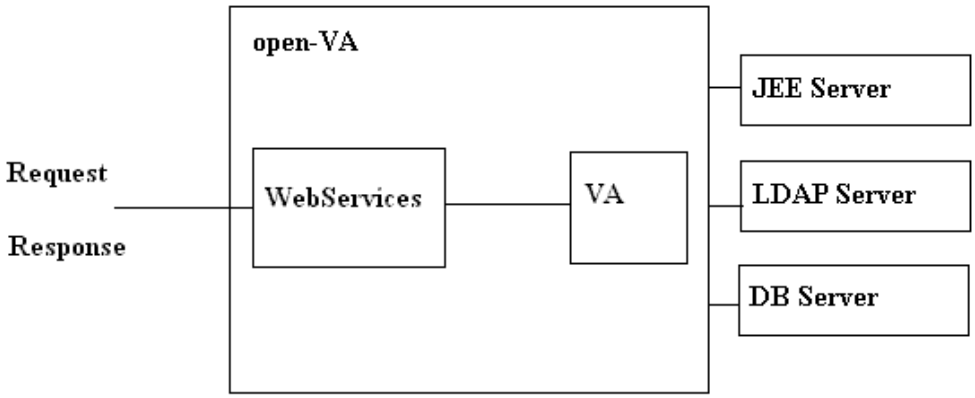
Verificando Open-VA

Después de desplegar y arrancar Open-VA puede verificar que la apliación está funcionando invocando su página principal tal y como se muestra en la figura:


open-VA - Mozilla Firefox
File Edit View Go Bookmarks Tools Help
http://localhost:8080/open-VA/
Go
Getting Started
Latest Headlines
Welcome to open-VA
Powered by
open-VA.org
Done

Capítulo 3. Usando Open-VA

Open-VA es un código abierto que demuestra cómo validar certificados digitales y documentos firmados.



Como puede ver en el diagrama anterior Open-VA tiene un *WebService* como frontal. Éste acepta peticiones, las procesa y devuelve una respuesta.

Comenzando

Debe construir un cliente *webservice* para poder operar contra Open-VA. Para hacerlo necesita el descriptor de servicio (*WSDL*). No tiene porqué escribirlo manualmente. Puede ser obtenido invocando **Open-VA/ValidateWS?wsdl** en el servidor.

En esta entrega de Open-VA sólo un método está disponible: **validate**.

Esta operación acepta un *documento XML* conteniendo la petición y devuleve otro `xml` con la respuesta.

Nota

Open-VA devuelve la respuesta firmada. Primero calcula la firma para la respuesta y luego añade a la respuesta un nuevo campo con nombre `signature`.

Los `dtd` para estos `xml` son:

- petición

```
<!DOCTYPE request [<!ELEMENT request (elementId,elementContent,?)>
<!ELEMENT elementId (#PCDATA)>
<!ELEMENT elementContent (certificate|signedDoc|(signature,doc))>
<!ELEMENT certificate (#PCDATA)>
<!ELEMENT signedDoc (#PCDATA)>
<!ELEMENT signature (#PCDATA)>
<!ELEMENT doc (#PCDATA)>]>
```

- respuesta

```
<!DOCTYPE response [
<!ELEMENT response (value, codeError?, status, statusReason?, signature)>
<!ELEMENT value (#PCDATA)>
<!ELEMENT codeError (#PCDATA)>
<!ELEMENT status (#PCDATA)>
<!ELEMENT statusReason (#PCDATA) ?>
<!ELEMENT signature (#PCDATA)>]>
```

Protocolo de operaciones Open-VA

Debe crear un `xml` en la petición y parsear la respuesta tras invocar la operación **validate**, según el `dtd` en la sección anterior.

Nota

Los datos binarios como certificados, firmas o documentos se pasan en formato *base 64*.

Se soportan tres tipos de peticiones:

- Validación de certificados

  Debe informar el campo **certficate**

- Documentos firmados con firma no adjunta

  Debe informar los campos **signature** y **doc**

- Documtos firmados con firma adjunta

  Debe informar el campo **signedDoc**

La respuesta puede ser un poco confusa porque hay dos códigos de respuesta.

- `value`

  Indica que el documento está bien formado y puede ser procesado.

  Sus valores pueden ser: *Refused*, *Failure* o *Success*

  Debe prestar atención porque *Success* no significa que la validación es correcta. Sólo significa que la petición ha sido aceptada para ser procesada.

  En el caso de *Refused* o *Failure* el campo *codeError* da mayor información.

- `status`

  Sus valores pueden ser: *Valid* o *Invalid*

  Un valor de *Valid* indica que la validación ha sido correcta.

  En el caso de una validación incorrecta el campo *statusReason* da mayor información.