minsait

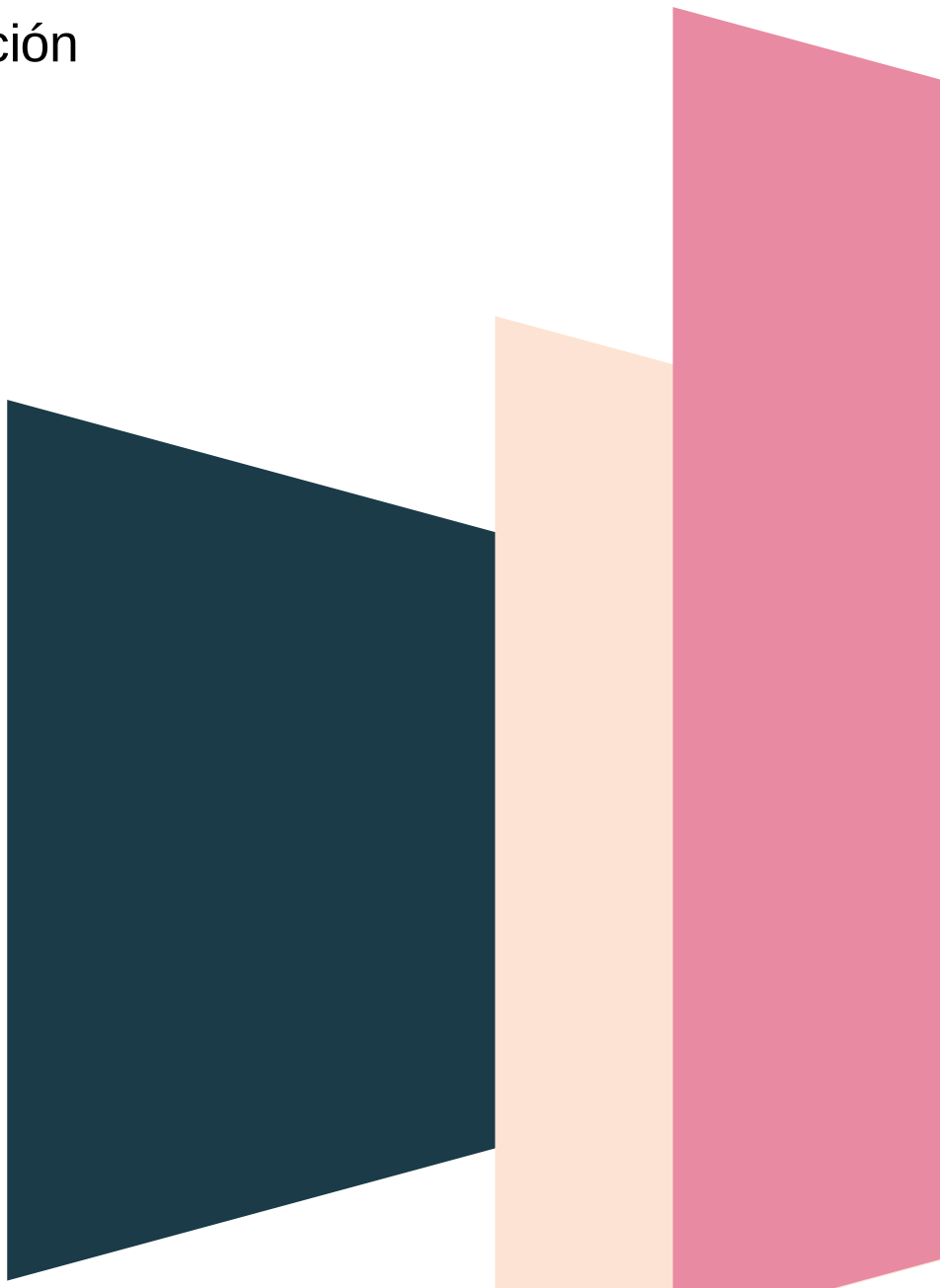
An Indra company

# EDITRAN/XAdES

## Firma y Verificación XAdES
## z/OS
## Manual de Instalación

mayo de 2019

ecosystems
communications



# 1. INTRODUCCIÓN

El objetivo de este documento es explicar la funcionalidad desarrollada por EDITRAN que permite firmar y verificar ficheros firmados en formato XAdES, hasta nivel de protección EPES así como en formato PKCS#7. EDItranSignatureServices es un servidor Java que admite peticiones hechas desde EDITRAN/FF para la firma y verificación de ficheros, tanto 'Datasets' como ficheros del HFS.

Para firmar serán certificados válidos todos aquellos instalados en el sistema USS en cualquier keystore generado por el usuario.

El proceso de verificación incluye la extracción de los datos firmados que después serán tratados por las aplicaciones del cliente.

En este documento se describen las acciones necesarias para:

- Instalar EDItranSignatureServices (en la parte USS del z/OS).
- Indicaciones sobre el manejo de certificados (keystores)
- Modo de funcionamiento

# 2. INSTALACIÓN y REQUISITOS

La verificación y firma que proporciona EDITRAN están desarrolladas como un servidor en Java que será instalado en los servicios Unix del z/OS.

## 2.1. Requisitos de instalación.

Se debe tener instalado, al menos, IBM 31-bit SDK for z/OS, Java Technology Edition, V6 y tener acceso al JZOS para el acceso a ficheros de MVS, los ficheros para ello se encuentran en $JAVA_HOME/lib/ext.

Con el fin de superar las limitaciones en el tamaño de las claves criptográficas que se utilicen en las firmas, se deben reemplazar los archivos de política restringidos US_export_policy.jar and local_policy.jar del directorio $JAVA_HOME/lib/security, con los que IBM entrega el SDK, por otros de versión no restringida que están en $JAVA_HOME/demo/jce/policy-files/unrestricted.

Un servidor TCP java, el cual debe disponer de una dirección y un puerto en el que escuchar las peticiones realizadas desde el EDITRAN en MVS.

Acceso a los servicios UNIX del ZOS.

Versión mínima EDItran V5R2F00.

## 2.2. Instalar en USS.

1. Se recomienda crear un directorio en la partición Unix de z/OS (USS) para instalar el software de verificación de firma, por ejemplo: /u/edixd

2. Enviar, en modo binario, al USS el paquete XAdES-zos.Vn.n-AAAA-MM-DD.tar . Puede usar cualquier utilidad de transferencia de ficheros, como el ftp.

3. Conectarse al USS y descomprimir el fichero dentro del directorio creado (u/edixd>tar -xof XAdES-zos.Vn.n-AAAA-MM-DD.tar). Dejará una estructura como en el ejemplo que sigue:

```
/u/edixd: >ls -l
total 25456
-rw-r-----   1 KI10139  KISNCE   12953600 Apr 28 15:44 XAdES-zos.V2.2-2015-04-28.tar
drwxr-xr-x   2 KI1056E  KISNCE       8192 Feb 11 10:13 bin
drwxr-xr-x   2 KI1056E  KISNCE       8192 Apr 29 11:16 conf
drwxr-xr-x   2 KI1056E  KISNCE       8192 Apr 28 15:37 crl
drwxr-xr-x   3 KI1056E  KISNCE       8192 Apr 29 11:16 lib
drwxr-xr-x   2 KI1056E  KISNCE       8192 Apr 28 15:37 logs
drwxr-xr-x   2 KI1056E  KISNCE       8192 Apr 29 11:16 plantillas
drwxr-xr-x   2 KI1056E  KISNCE       8192 Apr 29 11:16 politicas
drwxr-xr-x   4 KI1056E  KISNCE       8192 Feb 13 09:49 rsc
```

Los scripts de configuración del producto están en el directorio **bin** y debemos de asegurar que tienen permiso de escritura y ejecución. Se han de modificar dichos scripts con el directorio **JAVA_HOME** de la instalación.

También es recomendable que la carpeta de **logs** tenga permiso de escritura, al menos, para el grupo de usuarios.

4. Ejecutando **/u/edixd/bin/ConfiguracionXades.sh** se procederá a la adaptación de la configuración del producto a la instalación propia. El comando nos devuelve:

```
Los valores con contrasena (passProxy y passTrustStore) se guardan codificados
y los demas en claro.
Si se quisiera se pueden editar los valores en claro con un editor de texto
plano, pero no los valores codificados.

Valor del archivo conf/xades.properties
#Parametros de EDITRAN/XAdES (Obligatorio)
ipEditranXades=127.0.0.1
puertoEditranXades=7760
#TrustStore de las CAs (Obligatorio)
pathTrustStore=rsc/truststore/trustStore.pfx
passTrustStore=********
#KeyStore por defecto (Opcional)
pathKeyStore=rsc/keystore/keyStore.pfx
#Conexion con EDITRAN/OCSP remoto (Opcional, se rellena si se usa EDITRAN/OCSP
remoto)
ipEditranOcsp=
puertoEditranOcsp=
#Conexion con proxy para la conexion a Internet. Es necesario en el caso de
querer usar OCSP o CRL para verificar la revocacion
#de certificados o para el caso de querer firmar con TimeStamp para lo que se
necesita la conexion con un servidor
ipProxy=
puertoProxy=
userProxy=
passProxy=
#Sistema Operativo donde se instala EDITRAN/XAdES. En el caso de Sistemas
Operativos Windows, Unix o AS400, su valor debe ser N(No).
#En el caso de Sistemas Operativos z/OS, su valor debe ser S(Si)
zOS=S

Modificar las propiedades del fichero conf/xades.properties? (S/N)S
```

Poniendo S irá pidiendo los valores de los parámetros; se tendrán que introducir sólo aquellos que sea necesario modificar.

Se recomienda utilizar el puerto 7760 para el servidor EDITRAN/XAdES y cambiar la password del TrustStore (véase apartado 3).

En caso de que los certificados que se usen en las aplicaciones necesiten validación OCSP y ésta se haga de forma remota, se necesita el servidor EDITRAN/OCSP, debiendo configurar la dirección y puerto en el que está instalado.

Para poder hacer la verificación OCSP/CRL de los certificados y también para la firma con TimeStamp para conectarse al servidor que selle la hora de firma, normalmente se necesitará un proxy para salir del Host, por lo que en ese caso se tendrá que implementar la dirección y puerto del proxy así como un usuario y password para poder acceder a través de él a Internet.

A continuación se muestra el diálogo de ejemplo en el que se han dejado los ficheros truststore y keystore por defecto y no se ha configurado la validación OCSP/CRL:

```
IP EDITRAN/XAdES: nnn.nnn.nnn.nnn
Puerto EDITRAN/XAdES: 7760
Path TrustStore:
Password TrustStore:
Path KeyStore:
IP EDITRAN/OCSP:
Puerto EDITRAN/OCSP:
IP Proxy:
Puerto Proxy:
Usuario Proxy:
Password Proxy:
Sistema Operativo z/OS (S/N):S
```

Esta configuración quedará guardada en el fichero **xades.properties** del directorio **conf** (en el ejemplo: /u/edixd/conf/xades.properties).

Describimos a continuación todas estas propiedades:

- **Propiedades obligatorias:**
  - **IP de EDITRAN/XAdES:** IP en la que se arranca el Servidor Java EDITRAN/XAdES.
  - **Puerto de EDITRAN/XAdES:** Puerto en el que se arranca el Servidor Java EDITRAN/XAdES.
  - **Path del truststore:** Es la ruta del almacén de certificados donde se guarda el certificado de la TGSS (CA) y de todas las CA en las que debamos confiar. Además de en este almacén, se confía en las CA que estén en el almacén de Windows instaladas.
  - **Password:** Password del truststore.
- **Propiedades opcionales:**
  - **EDITRAN/OCSP remoto:** Sólo es necesario en el caso de querer usar EDITRAN/OCSP de forma remota, por defecto se usa de forma local en el caso de querer hacer verificación de certificados:
    - **IP EDITRAN/OCSP:** Dirección IP de la máquina donde esté instalado y ejecutándose el servidor EDITRAN/OCSP Remoto.
    - **Puerto EDITRAN/OCSP:** Puerto del servidor EDITRAN/OCSP para poder conectarse a él.
  - **Proxy:** Uso de proxy para la conexión a Internet de EDITRAN/XAdES. Es necesario en el caso de verificación de los certificados tanto por medio de CRL u OCSP, y también para la firma con TimeStamp para conectarse al servidor que selle la hora de firma:
    - **IP Proxy:** Puerto del proxy con el que necesita conectarse EDITRAN/XAdES.
    - **Puerto Proxy:** Puerto del proxy para la conexión a Internet.
    - **Usuario Proxy:** En caso de ser necesario, usuario del proxy para la conexión a Internet.
    - **Password Proxy:** Password del usuario del proxy.
- **Sistema z/OS:** Indica si el sistema donde se está ejecutando el programa Java es una máquina z/OS o no.

5. Para arrancar y parar el proceso se utilizarán:

**./start_xades.sh** y **./stop.sh**

# 3. INSTALACIÓN CERTIFICADOS

Los certificados que se usan en EDITRAN/XAdES se almacenan, según su uso, en dos ficheros: TrustStore, para los certificados de las CA en las que debamos confiar, y keyStore, para los certificados de firmantes.

## 3.1. Fichero truststore

En la instalación de EDITRAN/XAdES se suministra un fichero TrustStore, **trustStore.pfx** en la carpeta **/rsc/truststore/** (en el ejemplo /u/edixd/rsc/truststore/trustStore.pfx), con varias CA ya incorporadas. Las CA incorporadas son la del DNI, la de EDITRAN, la de la FNMT y la de la TGSS. Este almacén es de tipo PKCS12 pero no contiene claves, sólo certificados que son públicos.

Para modificar la contraseña de este fichero (por defecto "password"), se transmitirá en binario a Windows, se administrará con una herramienta adecuada – por ejemplo Portecle http://portecle.sourceforge.net/ - y se volverá a enviar al USS.

Además del trustStore, se adjuntan los certificados de algunas de las CA incluidas en dicho truststore. Con estos certificados podremos crear un fichero truststore propio mediante las utilidades que proporciona la instalación Java de IBM (keytool, ikeyman) con las limitaciones para keystores tipo PKCS12 en manejo de passwords. Se aconseja usar la herramienta anteriormente citada para incorporar o borrar CA.

## 3.2. Fichero keystore

El fichero keystore es un fichero que contiene un o más certificados que serán usados para firmar. Los certificados son obtenidos por el usuario de la aplicación y se deberán almacenar en una carpeta del USS o incorporarlos a uno o varios keystore cuyo path deberemos configurar en cada usuario de EDITRAN/FF (consultar manual EFF52USUI.doc – IMS - y EFF52USUC.doc – CICS-) o, en caso de existir uno único, en el path por defecto (ejecutando ConfiguracionXades.sh, situado en directorio bin).

Para configurar este tipo de almacenes debe ejecutar el fichero de procesamiento por lotes **FicheroKeyStores.sh,** situado en el directorio **bin**. Esta configuración se queda guardada en el fichero **FicheroKeyStores.txt** en el directorio **conf** (/u/edixd/conf/FicheroKeyStores.txt).

Al ejecutar el programa, si ya existe el fichero, nos muestra los valores que tenemos ya guardados. En caso contrario estarán en blanco:

```
Script para la gestion del fichero que guarda la informacion sobre el almacen,
el alias y las password necesarias para la firma electronica

Claves guardadas:
KeyStore: rsc/keystore/keyStore.pfx; Alias: Certificado de prueba XAdES

Listar las claves guardadas (L), guardar una clave (G), borrar una clave (B) o
no modificar (N):
```

Las opciones que tenemos son guardar una nueva clave (G), borrar una clave ya existente (B) o no hacer nada (N). Los datos a guardar de cada firmante son:

- **Path keystore:** Ruta del almacén de claves donde está almacenado el par de claves
- del firmante.
- **Password keystore:** Password del almacén de claves.

- **Alias:** Alias del par de claves que queremos usar para firmar.
- **Password de la clave:** Password de la clave. Puede ser la misma que la del keystore, en cuyo caso se deja en blanco.

Para poder probar el producto sin necesidad de contar con certificados propios también se adjunta el fichero **keystore.pfx** que está en el directorio **rsc/keystore** con un certificado de prueba (en el ejemplo: /u/edixd/rsc/keystore/keystore.pfx); el alias del certificado guardado es "Certificado de prueba XAdES" (contraseña: "password").

Este almacén ya está configurado en el fichero FicheroKeyStores.txt y es el que se pone como fichero keystore por defecto en la configuración inicial.

## 3.3. Configuración de la consulta del estado de revocación de los certificados

El uso del fichero **revocados.properties**, situado en la carpeta **conf,** permite ampliar los lugares donde consultar el estado de revocación de los certificados respecto a la información que el propio certificado trae. Esta facilidad se ha implementado para poder hacer uso de aquellos certificados que no aporten dicha información de manera autosuficiente, por ejemplo cuando incluyen el nombre de la lista crl pero no la dirección donde consultarla.

El contenido del fichero puede tener tantos bloques como el siguiente como sean necesarios:

nombre.ca.0=*Nombre CA*

ldap.0=

ocsp.0=

crl.0.0=

crl.0.1=

crl.0.2=

En el registro "nombre.ca.n=" debe escribirse una palabra clave que identifique todos los certificados emitidos por una misma CA y que se encuentre tanto en el DN del emisor del certificado firmante como en todos los DN de los certificados de la ruta de certificación del mismo, por ejemplo "FMNT", "DNIE", "SWIFT", etc.

En el registro "ocsp.0=" se escribirá la dirección URL para conectarse al servidor OCSP, por ejemplo http://ocsp.dnie.es.

En el registro "ldap.0=" se escribirá la dirección LDAP para conectarse al servidor. En el caso de que esa información venga ya incluida en el certificado no es necesario indicarla.

En los registros "crl.0.n=", se referirán tantos como sean necesarios para cada CA, se indicarán direcciones donde consultar la lista de CRL e incluso simplemente los nombres de las CRL, en este caso la lista deberá estar descargada en la subcarpeta **crl.**

La consulta se hace en primer lugar donde indica el propio certificado. Si la información no es autosuficiente y la búsqueda no se puede llegar a realizar entonces se busca la lista descargada en la carpeta crl. Si tampoco está se continúa buscando en los lugares indicados en revocados.properties, en el mismo orden en el que aparecen las entradas de la CA correspondiente (según el ejemplo se buscaría primero en ocsp.0, luego en crl.0.0, crl.0.1, etc).

### 3.4. Autenticación del Servidor de LDAP

En caso de que el Servidor LDAP pida autenticación, se puede ejecutar el script *autenticacionLDAP* que hay en la carpeta *bin*. Esto permitirá a las direcciones de LDAP que hay ya inscritas en el fichero *revocados.properties* añadirlas un usuario y contraseña si es necesario. Este programa sólo edita las direcciones de LDAP que hay en ese fichero por lo que primero se deberá rellenar la dirección de LDAP antes de ejecutarlo.

# 4. ANEXO A

## 4.1. Códigos de Resultado del servidor

**Resultados correctos**

**00-** Certificado correcto

**01-** Warning. No se ha encontrado o no se ha podido conectar a la Url del OCSP o del CRL en el certificado. Este warning ocurre con un certificado al que se le ha pedido verificación OCSP y no ha podido si está revocado o no. Es decisión del cliente tratar este código como un error o no. Por defecto se da como válida aunque salta un warning.

**02-** Warning. Se ha realizado la firma con Warnings. Esto puede ocurrir al firmar un documento en otro formato distinto al que se ha pedido originalmente porque el formato pedido no se puede realizar.

**Errores de verificación de la firma**

**10-** Política de firma incluida. No se ha cumplido la política.
**11-** No se confía en uno de los certificados firmantes, CA no válida
**12-** Servidor OCSP devuelve certificado(s) incorrecto(s)
**13-** Servidor OCSP o lista CRL devuelve certificado(s) revocado(s)
**14-** Certificado(s) caducado(s)
**15-** Documento modificado
**16-** Firmante(s) no autorizados en la cuenta (Esto es una validación posterior de la firma contra los perfiles de EDITRAN/XAdES).

**Lectura de los datos de la conexión con el Servidor Java EDITRAN/XAdES**

**17-** Petición no construida correctamente
**18-** Error al leer del buffer de lectura la longitud de la trama
**19-** Función peticionaria no reconocida
**20-** Tipo de firma no reconocido
**21-** Lenguaje máquina origen no reconocido
**22-** Valor de validación CRLS no reconocido
**23-** Valor de zip no reconocido
**24-** Valor de cifrado no reconocido
**25-** Valor de conversión a base 64 no reconocido
**26-** Identificador fichero de entrada no reconocido
**27-** Identificador fichero detached externo no reconocido
**28-** Identificador fichero de salida no reconocido

**29-** La ruta del fichero de entrada debe ser mayor que 0
**30-** La ruta del fichero detached externo debe ser mayor que 0
**31-** La ruta del fichero de salida debe ser mayor que 0
**32-** La ruta del fichero de salida de DN debe ser mayor que 0
**33-** La ruta del keystore debe ser mayor o igual que 0
**34-** El alias o DN del certificado debe ser mayor que 0
**35-** No se ha informado de ningún certificado firmante
**36-** Identificador de la plantilla no reconocido
**37-** El nombre de la plantilla debe ser mayor que 0
**38-** El lugar geográfico debe ser mayor o igual que 0
**39-** El rol del firmante debe ser mayor que 0
**40-** La acción del firmante debe ser mayor que 0
**41-** Fallo al leer el fichero de configuracion xades.properties

**Lectura del fichero firmado**

**48-** Error en la lectura del fichero origen
**49-** Error al parsear el xml del fichero origen
**50-** No se encuentra el elemento Signature en el documento
**51-** Error en el parseo del elemento Signature
**52-** Error en el proceso de validación de la firma
**53-** Error obteniendo la información de firma no válida
**54-** No se ha encontrado el certificado con el que se firmo dentro del documento
**55-** Error leyendo el truststore

**Escritura de los ficheros de salida**

**64-** Error en la escritura del fichero destino
**65-** Error en la escritura del fichero dn

**Validación OCSP y CRL**

**80-** Error al codificar el certificado
**81-** Error conexión con EDITRAN/OCSP
**82-** Error EDITRAN/OCSP leer Mensaje de Conexión
**83-** Las versiones no corresponden entre si
**84-** Error EDITRAN/OCSP escribir Respuesta Conexión
**85-** Error EDITRAN/XAdES leer Respuesta Conexión
**86-** Error EDITRAN/OCSP leer Mensaje de Petición
**87-** EDITRAN/OCSP no puede conectar Servidor OCSP
**88-** Error al intentar hallar la lista CRL del certificado

**Firma de ficheros**

**96-** Error leyendo el keystore
**97-** No se ha podido extraer la clave del keystore
**98-** La plantilla es incorrecta
**99-** No se ha podido crear el elemento Signed Info
**100-** No se ha podido crear el elemento KeyInfo
**101-** No se ha podido crear el elemento QualifyingProperties
**102-** URL incorrecta para el TimeStamp de la firma
**103-** No se ha encontrado el firmante para la contrafirma
**104-** Error al firmar

**Error de memoria**

**112-** OutOfMemoryError (Cuando se produzca este error, se debe aumentar los parámetros de asignación de memoria que están en el fichero **start_xades.bat** de la carpeta **bin**)

**Error inesperado**

**113-** Error inesperado en el servidor Java EDITRAN/XAdES

**Error sin definir**

**114-** Valor sin definir, no debería ser alcanzable (Este valor nunca se debería producir, es el valor que se asigna a la entrada del programa, pero no debería dar como salida

**Contacto**
editran@indra.es
T +34 91 480 80 80

Avda. de Bruselas 35
28108 Alcobendas,
Madrid, España
T +34 91 480 50 00
F +34 91 480 50 80

www.minsait.com