**eText(Electronic Text)**

**Introduction**

* An eText template is an RTF - based template.
* It is used to generate text output for Electronic Funds Transfer (EFT) and Electronic Data Interchange (EDI).
* That output text file can be transmitted to a bank or other customer.

**Payment Types:**

* Inbound Transaction
* Outbound Transaction

**Outbound Transaction(FUNDS DISBURSEMENTS)**

Payment Details

**Inbound Transaction (FUNDS CAPTURE)**

Bank Statement

**Types of Etext Template**

* Fixed-position based (EFT templates)
* Delimiter-based (EDI templates)

**Fixed-position based Attributes:**

Position,Length,Format,Pad,Data,Comments

**Delimiter-based Attributes:**

Maximum Length,Format,Data,Comments

**Electronic Fund Transfer(EFT)**

* Transmission of financial data and payments to bank in specific fixed position format flat file(text).

**Electronic Data Interchange(EDI)**

* Exchanging business documents(such as PO,invoice)/business informations between companies.
* Similar to EFT but it involves only in transmission of payment to banks and Transmitted as flat text file.

**Structure of EFT & EDT**

* Files in these formats are transmitted as flat files, rather than printed on paper.
* The length of a record is often several hundred characters and therefore difficult to layout on standard size paper.To accommodate the record length, the EFT and EDI templates are designed using tables.
* Each record is represented by a table.
* Each row in a table corresponds to a field in a record.
* The columns of the table specify the position, length, and value of the field.

**X-path**

* Xpath stands for XML path language.
* Xpath can be used to navigate through elements and attributes in an XML document.
* Xpath uses "path like" syntax to identify and navigate nodes in an XML document.
* Xpath contains over 200 built-in functions.
* Xpath is a major element in the XSLT standard.

**Transmission Configuration Setup**

* To make a transmission between Oracle HCM Cloud and your own servers through Oracle WebCenter Content server , the Transmission Configuration must be setup.
* After that, a connection made between the two for transmission.
* For secure transmission : PGP(Pretty Good Privacy) keys are used.
* PGP keys are generated to encrypt and decrypt the files in the specific format.

**Payment Process Profile**

* A payment process profile is a payment attribute assigned to documents payable, which specifies handling of the documents payable, payments, and payment files by Oracle Fusion Payments.
* Payment process profiles include several types of information, such as specifications for payment file formatting and transmission.

**Signing Process :**

* **Integrity :** Detects whether that file or message has been modified.
* **Authentication :** Verify cryptographically the person who signed a given message.
* **Non - Repudiation :** Prevent re-claim

**Outbound Flow**

Bank creates PGP keys

Bank sends public key to Company

Company creates signing keys

Company sends the public signing key to Bank

Company sign the file using private signing key

Company encrypts file using the public key

Company sends the encrypted file to Bank

Bank decrypts the file using private key

Bank verify the file using public signing key

**Inbound Flow**

Company creates PGP keys

Company sends public key to Bank

Bank creates signing keys

Bank sends the public signing key to Company

Bank sign the file using private signing key

Bank encrypts file using the public key

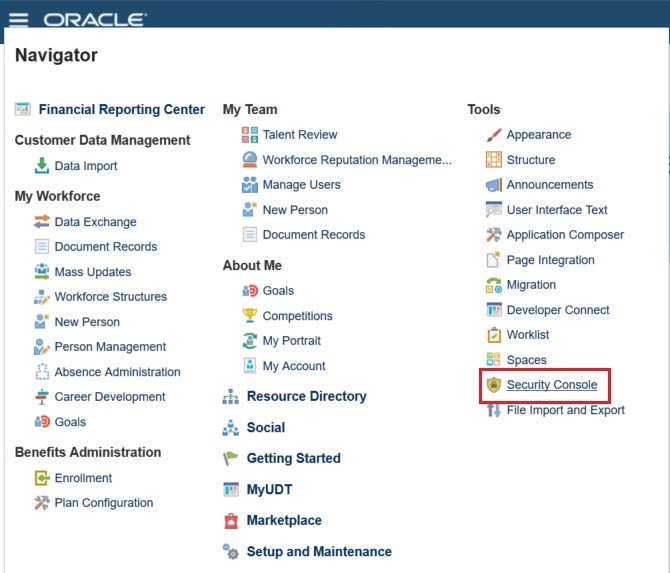
Bank sends the encrypted file to Company

Company decrypts the file using private key

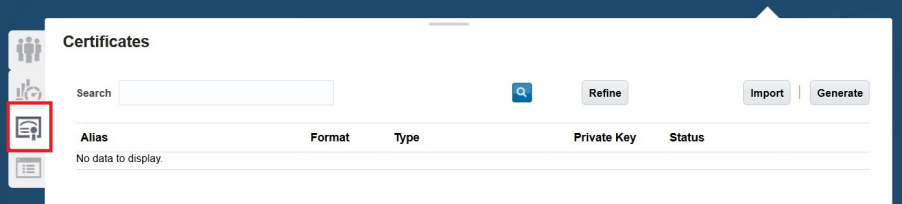
Company verify the file using public signing key.

**Key Creation:**

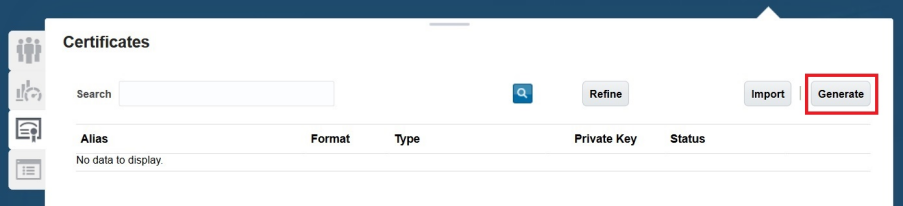
**Step1:**



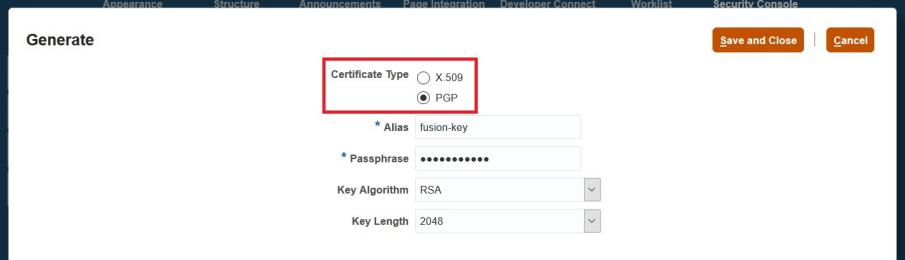
**Step2:**



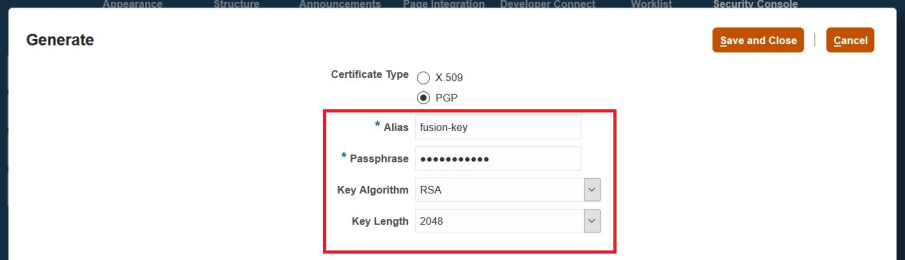
**Step3:**



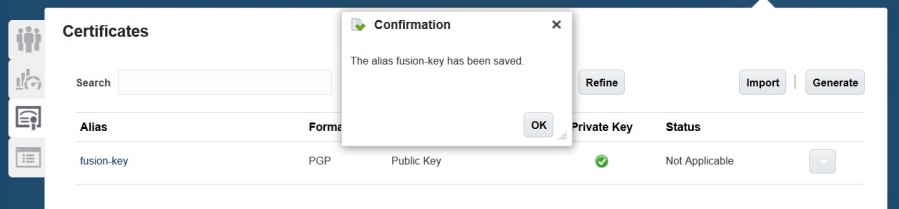
**Step4:**



**Step5:**

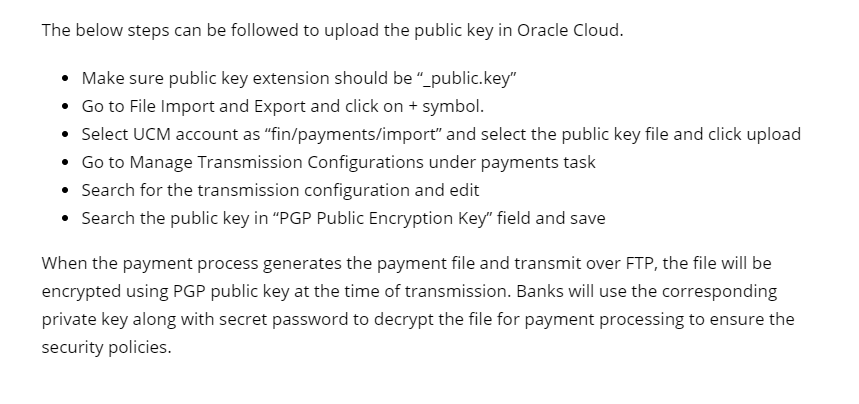


**Step6:**



* fusion-key\_pub.asc
* fusion-key\_priv.asc

**Upload:**



* fusion-key\_public.key
* fusion-key\_secret.key

**Key length** - 1024 or 2048

**Algorithm**

**RSA**

* Rivest-Shamir-Adleman
* Public key cryptosystem widely used for secure data transmission.

**CIPHER**

* A cipher is the algorithm used to encrypt and decrypt data.
* Generally, more bits that a cipher uses during encryption, is stronger or more secure the encryption is.

**Cipher Family:** AES-128, Blowfish

**AES-128**

* AES128 uses the AES-128 cipher, which has a key size of 128 bits.

**BLOWFISH**

* Blowfish encryption with a 64-bit block size and a variable-length key size from 32 bits to 128 bits.

**HASHING**

* It is a process to convert information to a shorter fixed value known as the key that is used to represent the original information.

**Hash:** SHA-1, SHA-256, SHA-224, SHA-512, MD5, SHA-384

**SHA - Secure Hashing Algorithm**

* Cryptographic hash function used to produce a hash value from the input file or message.

**Compression:**

Compression algorithms are normally used to reduce the size of a file without removing information.

Family - bzip2, zlib, .zip