

**Bank Payment**

**Technical Design Specification**

|  |  |  |
| --- | --- | --- |
| **Author** | : | Petchimuthu M |
| **Creation Date** | : | September 27, 2021 |
| **Last Updated** | : |  |
| **Document Ref** | : |  |
| **Version** | : | Draft |

**Table of Contents**

**1 Payment Automation……………………………………………………………………………………….1**

**1.1 Supplier Goods…………………………………………………….……………………………………1**

**1.2 Automate the Payment……………………………………………………………………1**

**1.3 Transaction.…………………………………………………………………………………………………1**

**1.4 Transaction Types…………………………………………………………………………………..2**

**1.5 Fund Capture………………………………………………………………………………………………2**

**1.6 Fund Disbursement……………………………………………………………………………….2**

**1.7 Payment Types………………………………………………………………………………………….2**

**1.8 Outbound Payment………………………………………………………………………………2**

**1.9 Inbound Payment…………………………………………………………………………………..2**

**2 Technical Design……………………………………………………………………………………………………3**

**2.1 FSD………………………………………………………………………………………………………..……………3**

**2.2 Mapping Sheet…………………………………………………………………………………………3**

**2.3 XML……………………………………………………………………………………………………………………4**

**2.4 X-Path……………………………………………………………………………………………………………..4**

**2.5 Querying in Data Model…………………………………………………………………..4**

**2.6 Template……………………………………………………………………………………………………….5**

**2.7 E-Text Template………………………………………………………………………………………5**

**2.7.1 Structure of E-Text Template……………………………………6**

**2.7.2 EFT………………………………………………………………………………………………….6**

**2.7.3 Creation of EFT Templates………………………………………….6**

**2.7.4 EDI…………………………………………………………………………………………………..8**

**2.7.5 Creation of EDI Templates………………………………………….8**

**2.8 XSL Templates…………………………………………………………………………………………10**

**2.8.1 Creation of XSL Templates………………………………………..11**

**3 Payment Process Profile………………………………………………………………………………12**

**3.1 Creation of PPP………………………………………………………………………………………12**

**3.2 Transmission Configuration Setup…………………………………………13**

**3.3 Creation of Transmission Configuration……………………………13**

**3.4 Security of Payment Data……………………………………………………………..14**

**3.4.1 Channel Security…………………………………………………………….14**

**3.4.2 Payload Security…………………………………………………………….14**

**3.5 PGP Keys……………………………………………………………………………………………………..14**

**3.6 Naming Convention and Extension……………………………………….16**

**3.7 Signing Keys………………………………………………………………………………………………17**

**3.8 Uploading of Keys………………………………………………………………………………..17**

**3.9 Algorithm……………………………………………………………………………………………………17**

**4 Outbound Flow……………………………………………………………………………………………………19**

**5 Inbound Flow………………………………………………………………………………………………………..19**

**1 Payment Automation**

* 1. **Supplier Goods**
* In an Organization, they must want products or software for their business process. Some of them made their own products by procure raw materials from the supplier called manufacturing company and some of them procure the final products from the suppliers called trading company.
* These products and goods are not buy from a single supplier. There must be more than several suppliers are needed.
* For these products or goods they want to make payment corresponding to the invoices. For one supplier, It’s easy to make the payment. But for several supplier and several invoices, It’s quite difficult to make the payment through manual or through Bank.
* For this difficulties, the concept of Payment Automation is introduced to make the payment.
  1. **Automate the Payment**
* This method is used to reduce the time and errors due to manual payment or Bank payment. In this method, software plays a major role achieve the payment automation. Payment is automated by using a software by the respected bank where the company holds their account.
* The Company want to send the Payment file which is a Flat File(Text File) which is used in the Bank software to make the payment automatically. In that file, all the details are present like Supplier details, Supplier Account details and the Invoice details etc.,.
  1. **Transaction**
* Transaction is anything that is done between two people or two group or two organization. It can be a transaction of raw materials or finished products or funds. In Fund transaction, there are two types.
  1. **Transaction Types**
* Inbound Transaction or Fund Capture
* Outbound Transaction or Fund Disbursement
  1. **Fund Capture**
* Money Inward from other party.
* From - Sales revenue / Refunds from suppliers / Financing transactions / Amounts awarded as a result of legal proceedings.
  1. **Fund Disbursement**
* Money Outward to other party.
* To - Supplier / Partners / Employers.
  1. **Payment Types**

In Bank payment automation the payment is mainly classified into two types based on the file which is transferred and the sender and receiver of the file.

* Outbound Payment
* Inbound Payment
  1. **Outbound Payment (Organization --> Bank)**
* A Flat-File contains information about current negotiable and voided payments.
* It is electronically transmits to the bank by the organization. It is also called as Payment File.
  1. **Inbound Payment (Bank --> Organization)**
* A Flat-File contains information about the recent transaction details like fund inward and outward in the Bank account.
* It is electronically transmits to the organization by the bank. It is also called as Bank Statement.

**2 Technical Design**

Technical design of the Flat-File consists of various stages. It consists of various documents from the receiver side and the files generated with various concepts. They are listed below.

* Functional Specification Document (FSD)
* Mapping Sheet
* Payment XML File
* Template

Note : Now let us see the technical design of the Outbound Payment. For Outbound Payment,

**Sender** ----> Organization

**Receiver** ----> Bank

**2.1 FSD**

* Functional Specification Document is the document which is generated and send by the receiver of the Payment File (Flat-File).
* In our scenario, Bank send the FSD to the Organization. FSD is one of the formal document used to describe the product’s (Flat-File) need, fields used, fields validation, condition, method of payment and the type of template used for processing.
* The expected sample output also mentioned in the FSD.

**2.2 Mapping Sheet**

* Mapping sheet is one of the formal document generated by the Functional Team from the sender side (Bank).
* It is the most important document used to generate the Template for the processing.
* It is used to match the fields from the XML source file by the use of X-Path. It consists of fields details, XML path (X-Path) and the conditions for the fields.
* Only after receiving the both the FSD and Mapping sheet, the Technical team is proceed the create the template design.

Note : In some scenario, the sender only sends the FSD. In such cases, we Technical need to collaborate with our Function team to generate the Mapping sheet for the corresponding FSD.

**2.3 XML**

* XML source file is the file which is used to match the data or fields with the template to generate the Flat-File.
* It is the Digital format of the text. This XML data is map with the template by using the X-Path provided in the mapping sheet.
* This XML is automatically generated by the oracle application whenever a Payment is done in the Oracle application by the Functional.
* By using the payment number or any other payment attribute we can able to get the XML source file by querying in the data model.

**2.4 X-Path**

* X-Path is defined as the XML path which is used to navigate through elements and attribute in an XML file.
* The syntax of X-Path is defined as the XML field separated by the (/) symbol. For example : Payer/Address/Country.

**2.5 Querying in Data Model**

* By using the payment attribute, the technical is able to query the XML source file from the IBY\_TRXN\_DOCUMENTS table is Oracle fusion.
* Query > SELECT \* FROM IBY\_TRXN\_DOCUMENTS ORDER BY LAST\_UPDATE\_DATE DESC

* After querying in the data model, we are able to search the corresponding XML under the Document field by searching the payment attribute provided by the Functional.

Note : Source XML file only present in the Document Type field 100.

* We want to extract the XML data by copying the Document field and save in (.xml) extension. Then the XML file is generated which is used to verify the template.

**2.6 Template**

* In Bank Payment process, It contains several hundred character record length. So, It is difficult to layout in Standard size paper.
* In order to accommodate, Templates are designed using tables. It is a formal document which is defined as the format model of the Flat-File. By using Mapping Sheet, Template is designed and the output Payment File (Flat-File) is generated with the XML file.
* This format requires special handling for XML file input (Global Level). Mainly, In Bank Payment there are two types of templates are used. They are,
  + E-Text Template
  + XSL Template

Note : Flat-File is only generated from E-Text template. An XML file with XSL tags is generated in the XSL Template with the generated XML source file. It is also used for Payment Automation. Based on Customer’s (Bank) requirement mentioned in FSD, template is generated.

**2.7 E-Text Template**

* This template is a RTF based template which generate text output (Flat-File) for Electronic Fund Transfer (EFT) and Electronic Data Interchange (EDI).
* The X-Path mentioned in the template is mapped with the XML file will generate the Flat-File as output.
  + Template ----> (.rtf)
  + Output ----> (.txt)

**2.7.1 Structure of E-Text Template**

* This template composed of series of tables, attributes and elements. The table defines the layout, setup commands and the data field definition.
* There are two types of E-Text templates. They are,
  + Fixed-Position Based Template (EFT Template)
  + Delimiter-Based Template (EDI Template)

Note : Required data description columns vary for these templates. But commands and functions are similar.

**2.7.2 EFT**

* The Electronic Fund Transfer (EFT) is the transmission of Financial data and payments to Bank in a specific Fixed-Position Based Format text File (Flat-File).

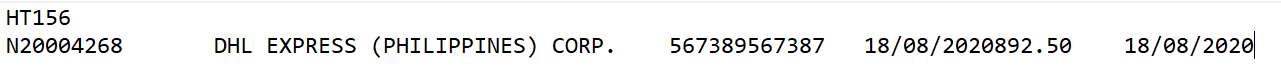
**2.7.3 Creation of EFT Templates**

* The creation of EFT or Fixed-Position Based template include the below documents.

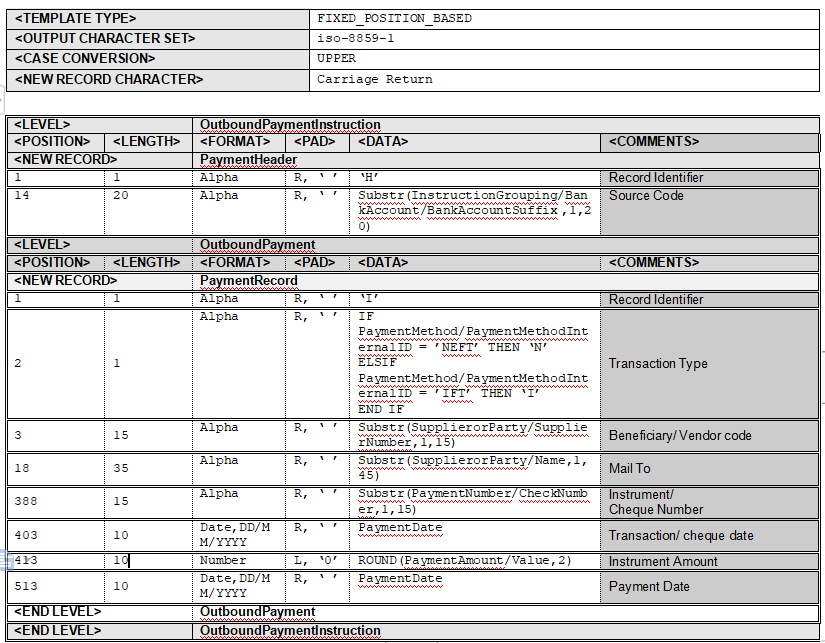
**Input XML Source File (EFT)**



**Expected Output (EFT)**



**E-Text Template (EFT)**



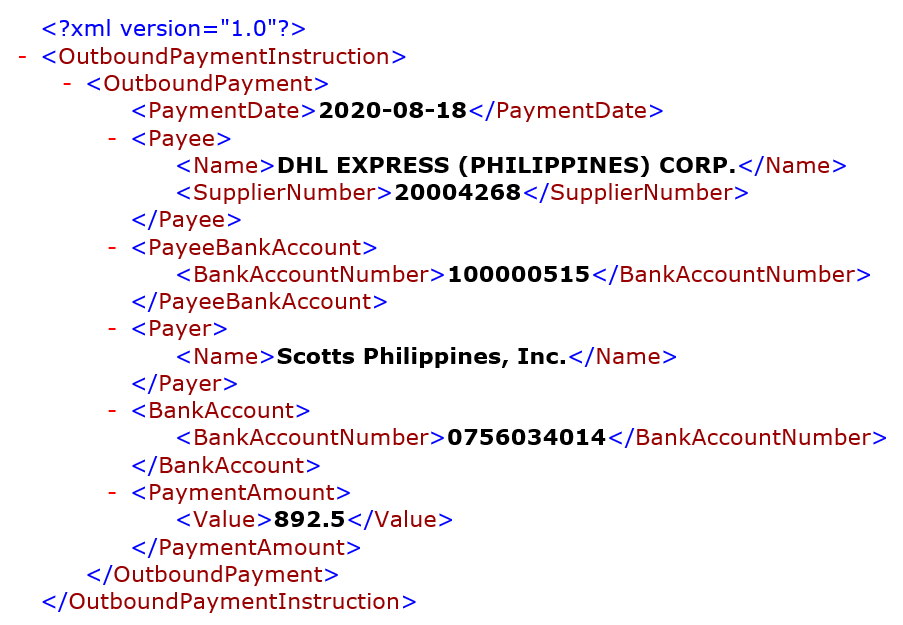
**2.7.4 EDI**

* The Electronic Data Interchange (EDI) is similar to EFT, But it also involves in the exchanging of Business Documents (such as Invoices, Purchase Orders etc.,) between Companies in a specific Delimiter-based format text file (Flat-File).

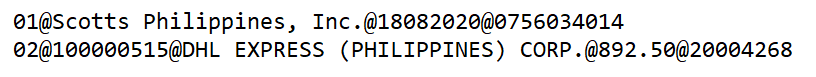
**2.7.5 Creation of EDI Templates**

* The creation of EDI or Delimiter-Based template include the below documents.

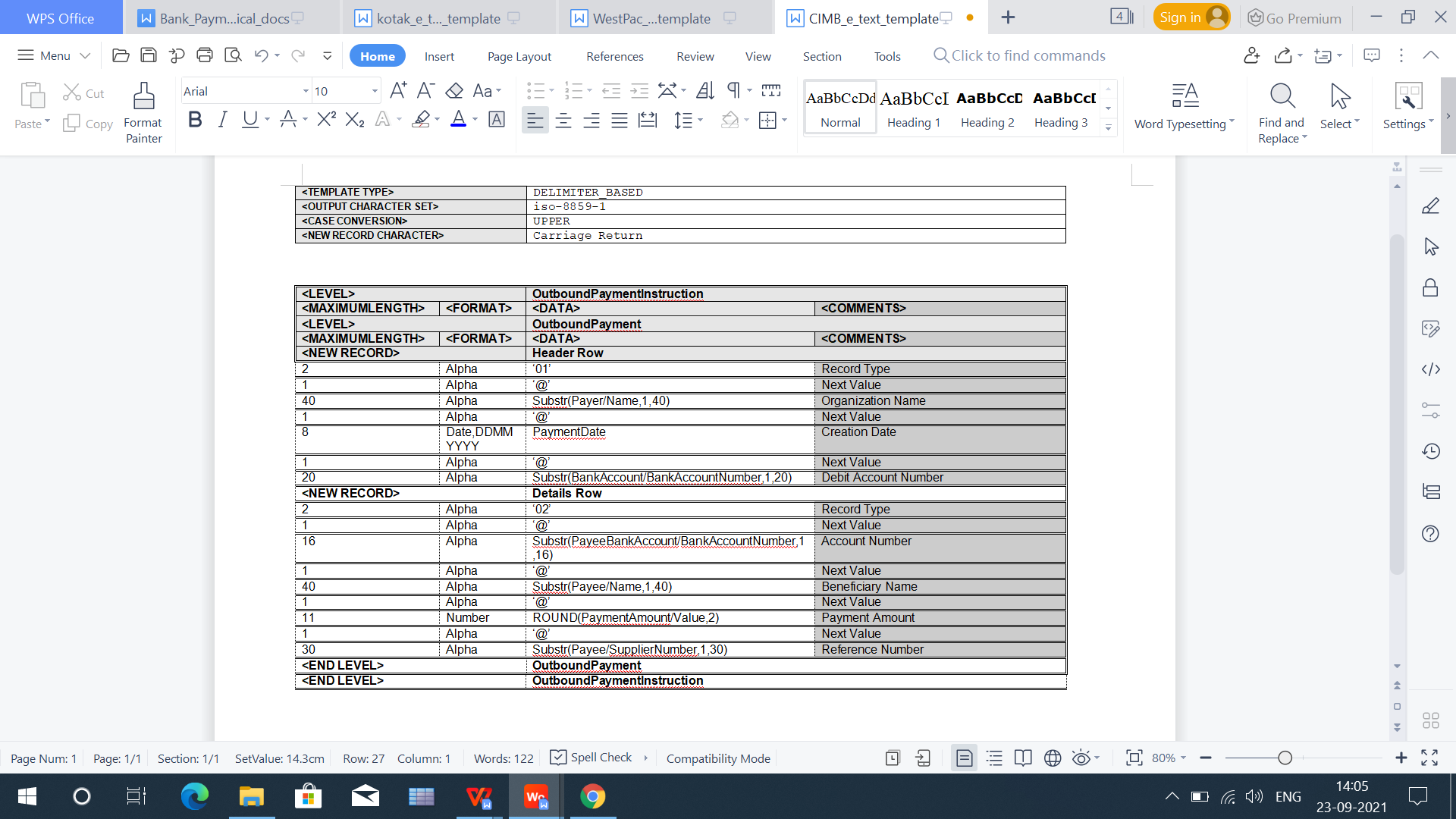
**Input XML Source File (EDI)**



**Expected Output (EDI)**



1. **Text Template (EDI)**



**2.8 XSL Template**

* This template is a XSL based template which generate an XML output with XSL tags. X-Path mentioned inside the XSL tags are used to extract the data from the XML file and generate the XSL output file.
  + Template ----> (.xsl)
  + Output ----> (.xml)

**2.8.1 Creation of XSL Templates**

* The creation of XSL template include the below documents.

**Input XML Source File (XSL)**

****

**Expected output**

****

**Sample Template**



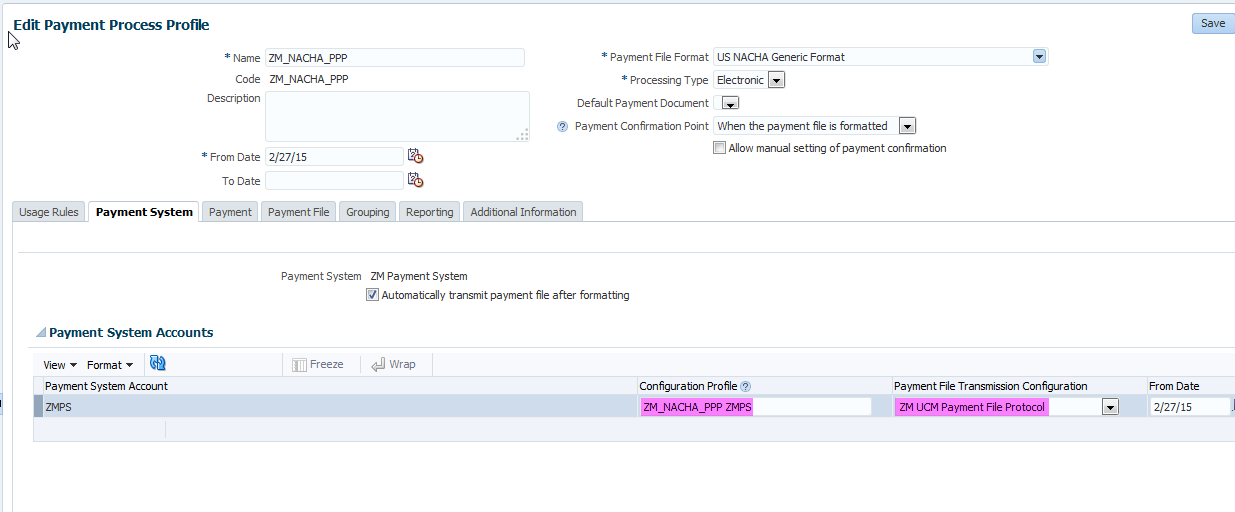
**3 Payment Process Profile**

* A Payment Process Profile (PPP) 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.

**3.1 Creation of PPP**

**Navigation:** Setup and Maintenance--->Manage Payment Process Profile

* Create a new PPP or update an existing one to add payment system and transmission details. Perform the following setups related to transmission:
* Under Payment System Tab, select and add the payment system created earlier. This will add a record for the payment system account associated with this payment system.
* Under Payment File Transmission Configuration, select the transmission configuration we created in previous step.
* If you wish to automatically trigger the payment file transmission after formatting phase, check the option "Automatically transmit payment file after formatting"



* At this point, we can submit the payment process profile to select installments, process payments and create formatted payment file. This payment file is then transmitted to destination specified in the transmission configuration.
* Payment files saved under UCM server can be accessed using Fusion Applications File Import and Export functionality.

**3.2 Transmission Configuration Setup**

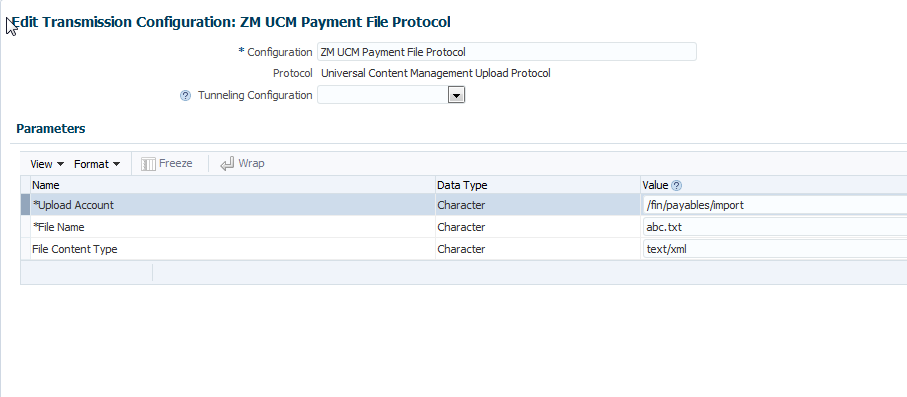
* To make a transaction between Oracle HCM Cloud and your own servers through Oracle WebCenter Content server , the Transmission Configuration must be setup.
* After that, a connection made between the two for transmission.

**3.3 Creation of Transmission Configuration**

**Navigation:** Setup and Maintenance--->Manage Transmission Configurations

Create a new transmission configuration by selecting Universal Content Management Upload Protocol in select protocol LOV and click create. Enter following details:

* Upload Account: /fin/payables/import
* File Name:  Specify a static file name (abc.txt)
* File Content Type: text/xml



**3.4 Security of Payment Data**

**3.4.1 Channel Security**

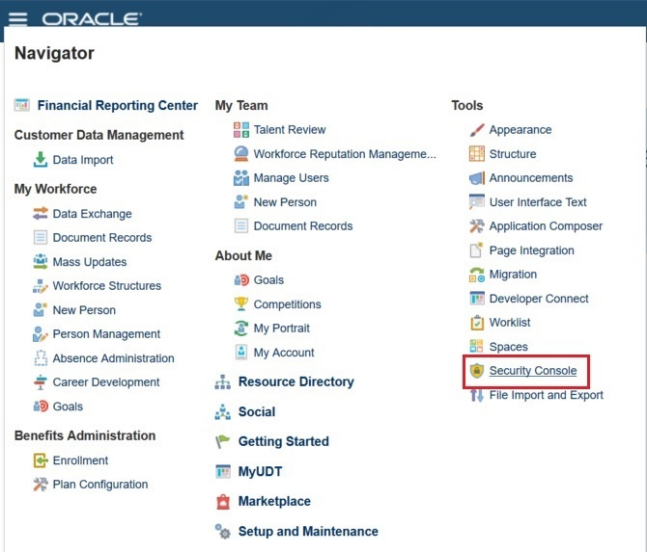
* By using secured transmission protocol such as SFTP, HTTPS etc.

**3.4.2 Payload Security**

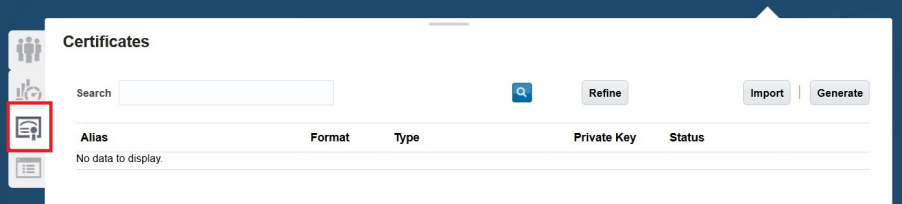
* By securing the payment file via payment file encryption and digital signature.

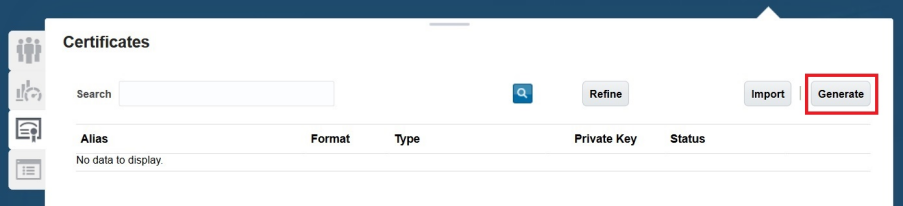
**3.5 PGP Keys**

* For secure transmission PGP keys are used.This PGP-based encryption support is available for secure file transfer using HCM Data Loader and HCM Extracts.
* These keys are generated to encrypt and decrypt the files in the specific format.
* In Oracle go to Security Console.

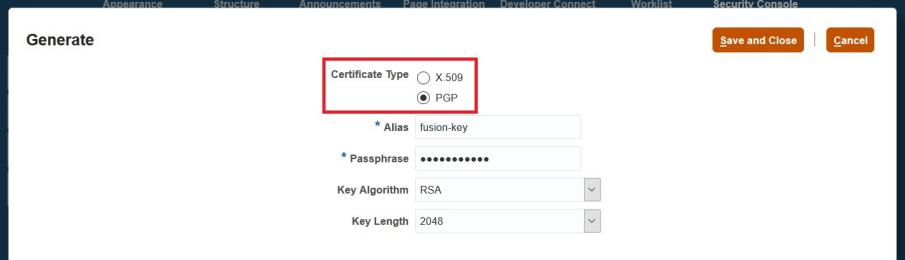


* Click on the Certificates icon and click Generate.

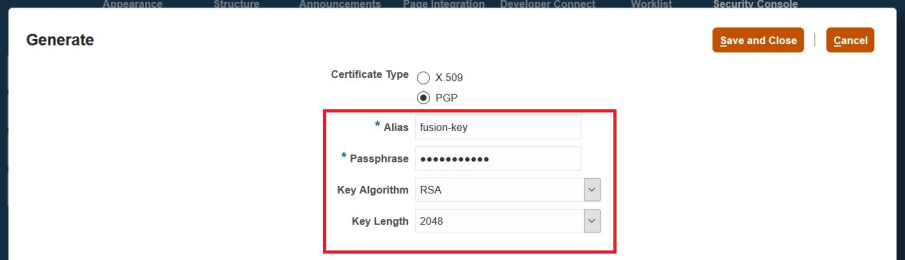




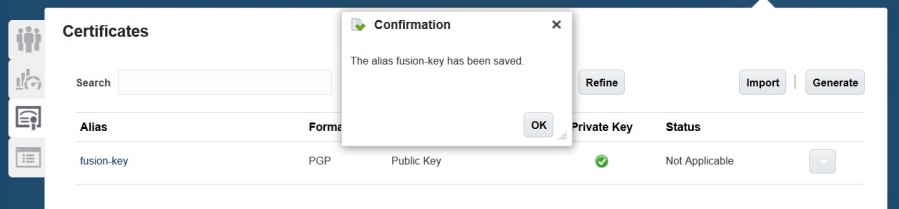
* Select the type as PGP.



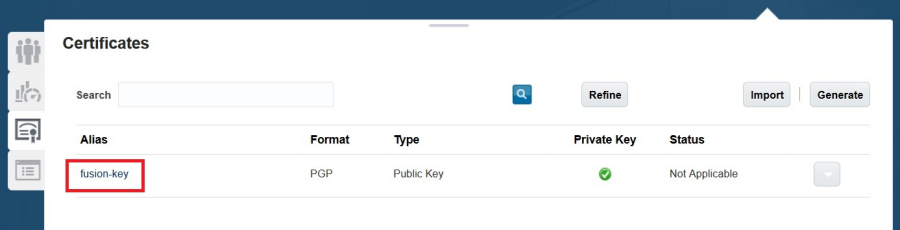
* Type the default Alias Name as fusion-key.
* Give the security Passphrase.
* Select the required Algorithm and Key length.

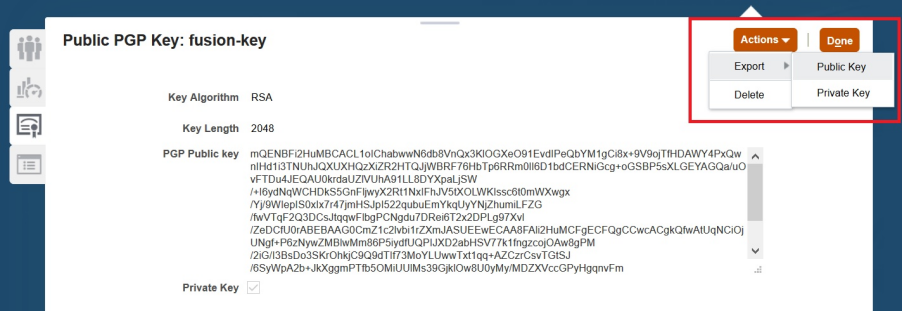


* Click on save and close.



* Click on the generated key and click Action --> Export --> Public Key.





**3.6 Naming Convention and Extension**

**Extension When Download:**

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

**Extension When Upload to UCM:**

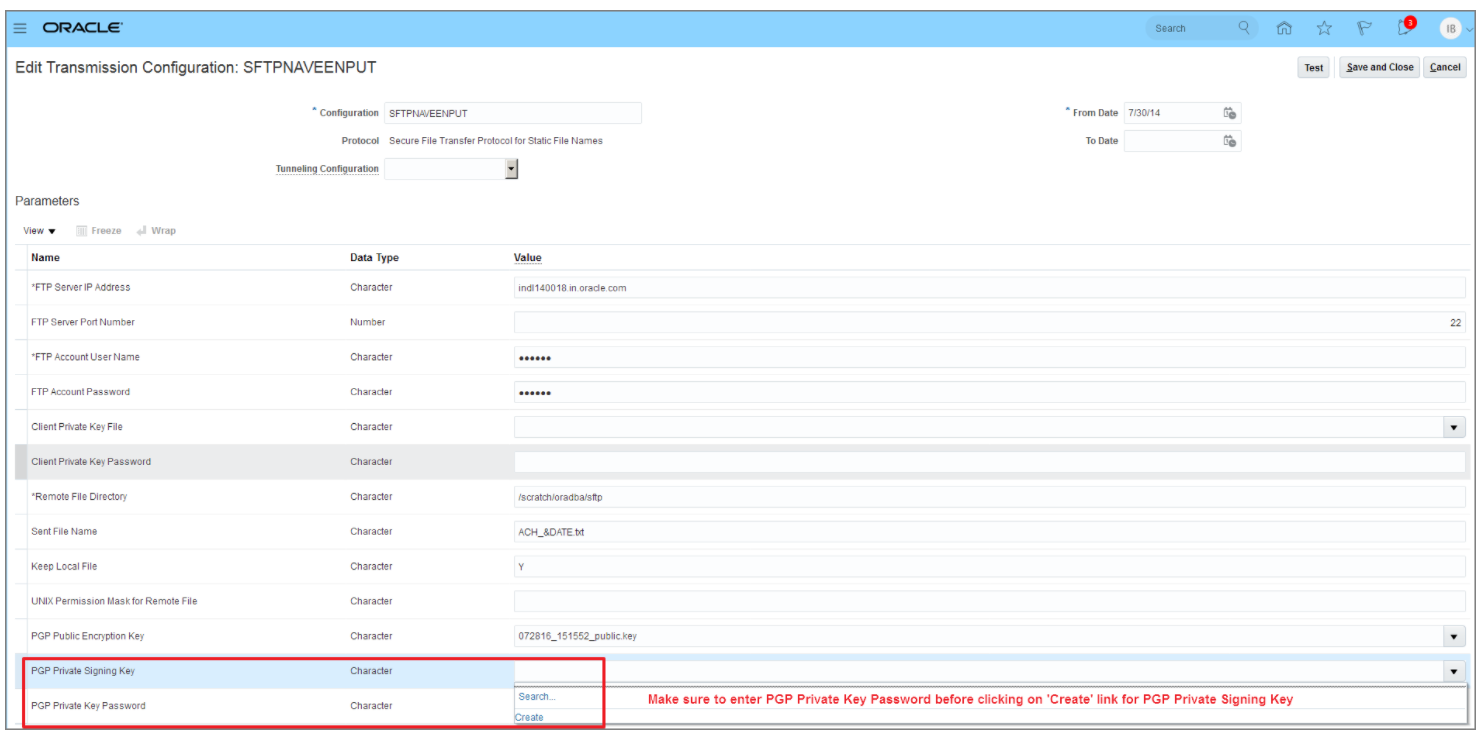
* fusion-key\_public.key
* fusion-key\_secret.key

**3.7 Signing Keys**

* Outbound files are signed using the HCM Cloud private key. You verify these files using the HCM Cloud public key.
* Inbound files are signed using your private key. The data-loading process verifies inbound files using your public key.

**3.8 Uploading of Keys**

* PGP and Signing keys are uploaded in the Transmission Configuration by edit the Transmission Configuration Setup.



**3.9 Algorithm**

**RSA (Rivest-Shamir-Adleman)**

Public key cryptosystem widely used for secure data transmission.

* Cipher: AES-128, Blowfish, CAST5, 3DES
* Compression: bzip2, zlib, .zip, uncompressed
* Hash: SHA-1, SHA-256, SHA-224, SHA-512, MD5, SHA-384, RIPEMD-160

**CIPHER**

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

**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. Use BLOWFISH only for backward compatibility with earlier Oracle GoldenGate versions.

**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.

**SHA - Secure Hashing Algorithm**

* Cryptographic hash function used to produce a hash value from the input file or message. Family - SHA-1, SHA-256, SHA-224, SHA-512 etc.,.

**4 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.*

**5 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.*

=================**END OF DOCUMENT**=================