ORACLE
FUSION APPLICATIONS

Oracle Cloud Expenses Co-existence and Integration Options

ORACLE WHITE PAPER | MARCH 2016

ORACLE

Contents

1. Introduction

You can implement Oracle Expenses Cloud Service while co-existing with your current financial system. This provides you a low risk phased cloud adoption approach with minimal implementation effort and without disrupting your current financial management processes. You can leverage Expenses Cloud Service for mobile expense entry, corporate card integration, policy enforcement and auditing expense reports and continue to rely on your current financial system for employee payment processing and accounting.

This white paper discusses implementation steps and considerations for implementing Expenses Cloud Service to co-exist with your legacy financial system.

2. Process Flow Summary

- Create, approve and audit expense reports in Expenses Cloud
- Publish expense data in XML format by using Process Expense Reimbursements background process
- Import expense data in your financial application to record invoices and process payments
- Transfer the payment data from your financial application to Expenses Cloud
- Update expense report status as paid and associate payment reference information to expense report

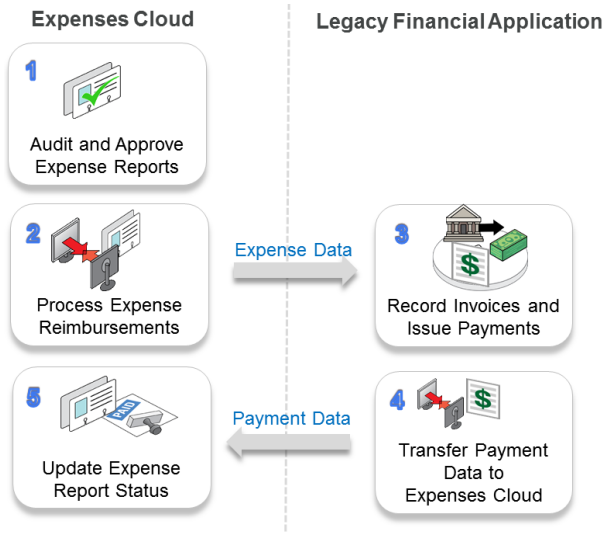

Figure 1- Third Party Payment Processing Flow

3. Configuration Steps

Use the Define Financials Configuration for Rapid Implementation task list to streamline your configuration to focus only on the critical setup steps. The rapid implementation task list minimizes the time needed for you to complete your key setups needed for implementing Expenses Cloud Service. You are not restricted to only the setup configuration in the rapid implementation task list. You can manually add the standard Financials offering task lists and tasks to your rapid implementation project to change and update your setup.

Following steps are needed to configure Expenses Cloud co-existing with the legacy financial application.

3.1. Implementation Users

In order for consultants to gain access and begin the implementation process, you will need to create the primary implementation user for your lead consultant. Once completed, this user will be able to create additional users for the rest of the implementation team. Instructions for how to create this user are included in the post-provisioning notification.

The Define Implementation Users task list includes tasks for creating users, optionally creating data roles, and provisioning users with roles before you have set up enterprise structures.

Note: After you have set up enterprise structures, do not use the Define Implementation Users tasks for creating any users such as expense users that need to be defined as employee in Human Capital Management (HCM).

See Creating the Primary Implementation User (Doc ID 1610683.1)

3.2. Oracle Human Capital Management (HCM) Configuration

Oracle HCM Cloud Service solution includes global human resources, payroll, payroll interface, time and labor, absence, benefits, compensation, and talent management. It also supports coexistence option that lets you use a combination of products in different deployments. For example, you might decide to keep using your current HCM implementation with Oracle E-Business Suite or PeopleSoft, but add Performance Management as a Cloud implementation.

It provides a rich set of integration toolsets such as file based data loader and spreadsheets to exchange data between the Oracle Global Human Resources Cloud and any third-party system. You should use file based data loader to import employees from your legacy system into HCM Cloud. You should also load the setup data such as legal entities, jobs, and departments from your legacy system into HCM Cloud.

For configuring Expenses Cloud, at a minimum employees need to be configured in Oracle HCM Cloud with a default expense account assigned. Jobs and department setup is also needed if your requirement is to have expense report approval based on job level or cost center.

See Get started with your Global HR implementation

3.3. Financials Configuration

Create an implementation project that includes the **Define Financials Configuration for Rapid Implementation** task list from the Manage Implementation Projects page in the Setup and Maintenance work area. This task list focuses only on the critical setup steps that are required in most cases and minimizes the time needed for you to complete the setups.

You can customize the task list and delete the tasklists for the applications you are not implementing or manually add the standard Financials offering task lists and tasks to your rapid implementation project to change or update your setup.

See Getting Started with Oracle Financials Cloud chapter 2 and 3 for more details on the above steps.

3.3.1. Define Common Financials Configuration for Rapid Implementation

i. Manage Geographies

The geography structure and data is shared across many applications. The geography structure is used both for third-party site and address information, and to group geographical regions into territories that share the same requirement, such as a tax zone. You need at least one level under the country level, such as STATE or CITY, to trigger the processes that assign geography records to users.

See Importing Geographies: Explained topic in Oracle Financials Cloud Implementing Common Features guide: Define Enterprise Structures chapter: Define Geographies section.

ii. Create Chart of Accounts, Ledger, Legal Entities, and Business Units in Spreadsheet

- Create a spreadsheet for entering the primary ledger and all the accompanying enterprise structures. The spreadsheet entry includes
  - Chart of accounts with segment values
  - Business units
  - Legal entities
  - Accounts and account hierarchies
  - Sequences for generating sequential IDs for transactions
- Add Manage Legal Entity task to the implementation project to designate the legal entity as legal employer.

iii. Upload Chart of Accounts

Perform the task Upload Chart of Accounts to load the chart of accounts structure, including segments and value sets from spreadsheet.

iv. Deploy Chart of Accounts

Perform the task Deploy Chart of Accounts to run the deployment process for the chart of accounts so that it can be used in ledger and account setup.

v. Upload Ledger, Legal Entities, and Business Units

Perform the task Upload Ledger, Legal Entities, and Business Units to load the key enterprise structures, including the ledger, legal entities, and business units, that depend on the chart of accounts.

See help topic Create Chart of Accounts, Ledger, Legal Entities, and Business Units in Spreadsheets: Explained in Oracle Financials Cloud Implementing Enterprise Structures and General Ledger guide.

3.3.2. Define Invoicing and Payments Configuration for Rapid Implementation

i. Manage Common Options for Payables and Procurement

Perform the task Manage Common Options for Payables and Procurement to set the following mandatory defaults for distributions:

- Liability
- Prepayment

- Conversion rate variance gain
- Conversion rate variance loss
- Realized gain
- Realized loss
- Discount taken

ii. Manage Payment Methods

Predefined payment methods are provided and you do not need to modify unless you wish to associate specific business units instead of all business units and attach data validations.

See user help topics Payment Methods: Explained and Payment Method Defaulting: Explained

3.3.3. Define Expenses Configuration for Rapid Implementation

i. Manage expenses system options

Perform the task Manage Expenses System Options to define setup options for managing expense entry and processing for all business units. Confirm that the default settings are aligned with your business practices. Choose the option to process expense payments in the third-party financial application.

By default, the expense reports are reimbursed via Oracle Payables. To process expense reports in your legacy financial system, change the 'Pay Expense Reports Through' setting to 'Third Party'. When processing expense payments in your third-party application, expense reports would not get interfaced in Oracle Payables for payment processing.

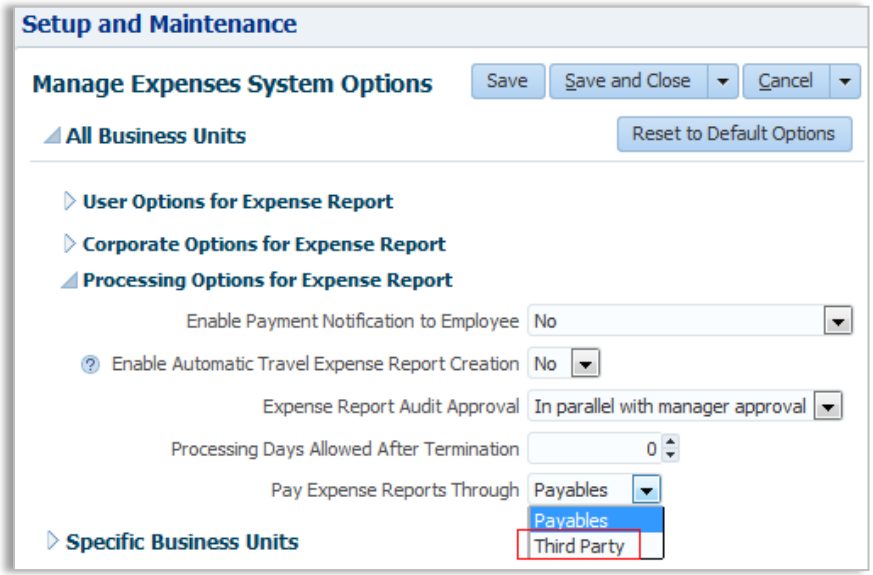

Figure 2 - Expenses System Option for Third party Payment Processing

ii. Manage expense report templates.

Perform the task Manage Expense Report Templates to define expense types applicable to your company and group them into expense templates. Expense templates are defined by business units. The expense templates available in expense report entry is determined by the business unit of the employees. Specify receipt requirements when you define expense types.

See following topics:

- Expense Templates: Points to Consider
- Can expense types be used across expense templates
- Configuring Expense Policies: Points to Consider

iii. **Manage expense approval rules**

Perform the task Manage Expense Approval Rules to define expense report approval rules based on your company's approval policies. Modify the predefined rules as needed. To enable audit of expense reports, you must define audit rules in addition to approval rules.

See following topics:

- Configuring Approval Rules: Explained
- Defining Approval Rules: Explained

iv. **Manage conversion rates and policies.**

Perform the task Manage Conversion Rates and Policies to select the conversion rate type for each business unit.

See Configuring Expense Policies: Points to Consider

4. Third Party Payment Processing

4.1. Audit and Approve Expense Reports

- Employees submit expense reports containing one or more expense lines representing cash or credit card expenses that were incurred.
- Expense reports then go through approval and audit process after which they are eligible for payment.

4.2. Process Expense Reimbursements

- **Background Processes**

  Once expense reports are eligible for payment, following background processes export the expense report data in XML file format. The XML file contains all the information necessary to issue payments to employees or credit card issuers as well as to record accounting entries in the third-party application.

  - Process Expense Reimbursements and Cash Advances
  - Create Card Issuer Invoices

- **XML Output Files**

  The files are also exported to Universal Content Manager (UCM) under **fin/expenses/export** directory from where you can download and parse the file to create the data the in your payables system. Expense auditor job role has file management permissions out of the box for uploading and downloading files from UCM.

  The output files are named as following:

File Name	Expense Report Type
ExmExpenseReport_*RequestId*.xml	Cash expenses and company pay and both pay credit card expense scenarios
ExmCashAdvance_*RequestId*.xml	Cash advance request
ExmCardIssuerInvoice_*RequestId*.xml	Card issuer invoice for company pay scenario

See Section 4.2. of this document for the XML file format and the data elements for the above files.

4.3. Record Invoices and Process Payments

Using the published data that was parsed and imported in your legacy payables application, record an invoice and process payments to employees and corporate card issuers.

For release 12 of E-Business Suite, you can download the XML from UCM and parse the published XML data and insert into E-Business Suite Payables invoice interface tables. Subsequently, Payables Invoice Import can be run to create invoices in E-Business Payables.

4.4. Transfer Payment Data to Expenses Cloud

After processing payments in your legacy payables system, record the payment reference information in a specific XML format and import the file via UCM to Expenses Cloud Service

See Section 4.3 of this document for the XML file format and the data elements.

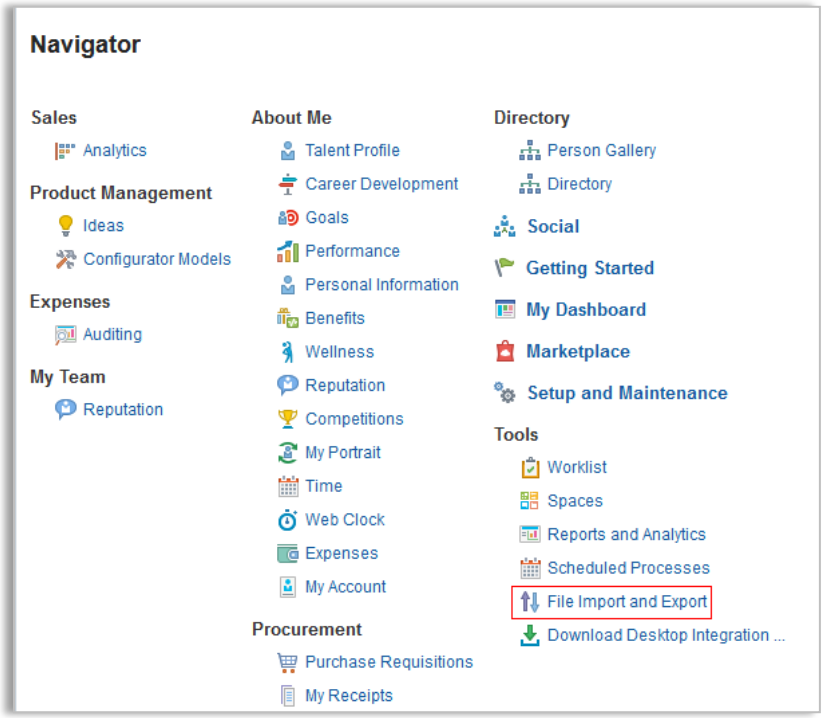

Figure 3 - File Import and Export Navigator menu item

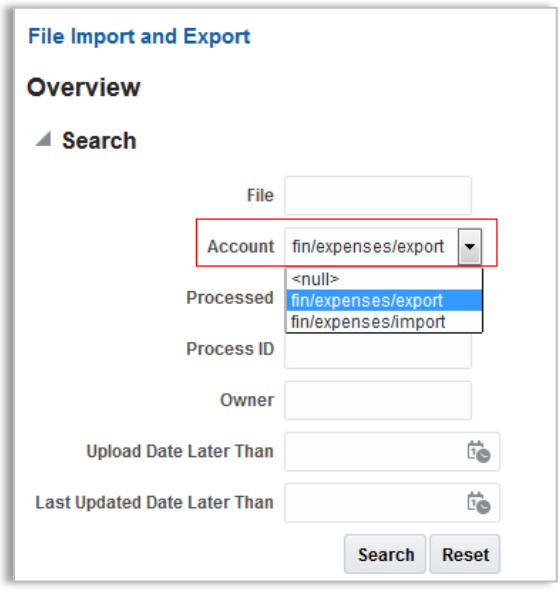

Figure 4 - Expenses folder in UCM

4.5. Update Expense Report Status

The uploaded files are processed by 'Update Expense Report Status' background process to update the expense reports status as paid and attach the payment reference information. Users with Expense Auditor job role can initiate a one-time request or schedule this background process.

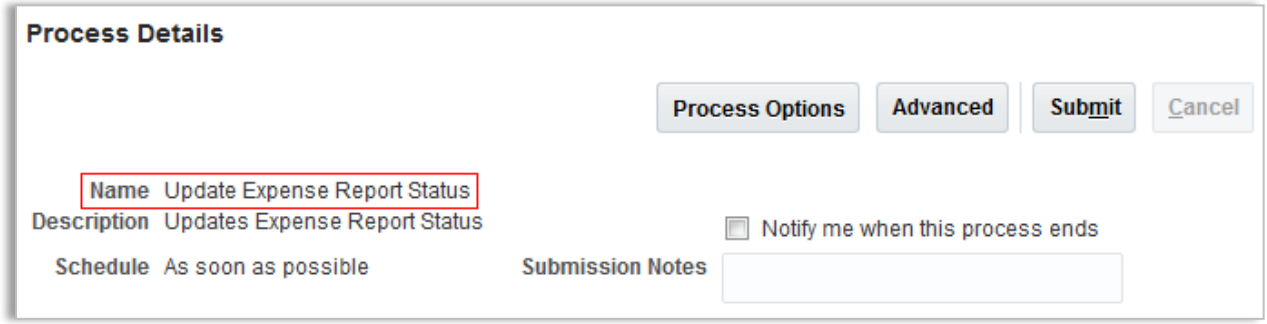

Figure 5 - Update Expense Report Status Background Process

Figure 6 - Expense Report after Payment Reference Association

5. XML Schema Definition (XSD) Files

The XML data files containing the Expenses data differ based on whether cash expenses or credit card expenses are being processed. For credit cards, there is different handling based on both-pay or company pay scenario.

5.1. XML Schema Definition for Expense Report Data (Outbound)

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema elementFormDefault="qualified" sdoJava:package="" targetNamespace="">
<xsd:complexType name=ExpenseReports>
 <xsd:sequence>
  <xsd:element name="RequestId" type="String" 15>
  <xsd:complexType name="ExpenseReport">
  <xsd:sequence>
    <xsd:element name="Number" type="String" 15>
    <xsd:element name="Purpose" type="String" 240>
    <xsd:element name="ReferenceKey" type="String" 15>
    <xsd:element name="ExpenseReportDate" type="Date">
    <xsd:element name="PaymentCurrency" type="String" 30>
    <xsd:element name="PaymentAmount" type="Decimal">
    <xsd:element name="PaymentMethod" type="String" 30>
    <xsd:element name="Source" type="String" values(Employee, CreditCard, Contractor)>
    <xsd:element name="EmployeeName" type="String" 240>
```

```
<xsd:element name="EmployeeNumber" type="String" 30>
<!-- if employee we print employee address code !>
<xsd:element name="EmployeeAddressCode" type="String" 30 values(OFFICE, HOME)>
<xsd:element name="PayeeName" type="String" 240>
<!-- if contingent worker or card issuer we print the payee address!>
        <xsd:simpleType name="PayeeAddress">
          <xsd:element name="AddressLine1" type="String" 240>
          <xsd:element name="AddressLine2" type="String" 240>
          <xsd:element name="AddressLine3" type="String" 240>
          <xsd:element name="AddressLine4" type="String" 240>
          <xsd:element name="City" type="String" 60>
          <xsd:element name="State" type="String" 60>
          <xsd:element name="PostalCode" type="String" 60>
          <xsd:element name="Province" type="String" 60>
          <xsd:element name="County" type="String" 60>
          <xsd:element name="CountryCode" type="String" 2>
        </xsd:simpleType>
<xsd:element name="PayGroup"  type="String">
<xsd:element name="LegalEntity" type="String" 240>
<xsd:element name="LedgerName" type="String" 30>
<xsd:element name="BusinessUnit" type="String" 240>
<xsd:complexType name="LiabilityAccount">
  <xsd:sequence>
      <xsd:simpleType name=?segment name? type="String">
      </xsd:simpleType>
  </xsd:sequence>
</xsd:complexType>
<xsd:element name="ExchangeRate" type="String">
<xsd:complexType name="Lines">
<xsd:sequence>
  <xsd:simpleType name="Line">
    <xsd:element name="ReferenceKey" type="String" 15>
    <xsd:element name="ExpenseDate" type="Date">
    <xsd:element name="ExpenseType" type="String">
    <xsd:element name="Amount" type="Decimal">
    <xsd:element name="Description" type="String" 240>
    <xsd:element name="Location" type="String" >
    <xsd:element name="ReceiptCurrency" type="String" 30>
    <xsd:element name="ReceiptConversionRate" type="Decimal">
    <xsd:element name="ReceiptAmount" type="Decimal">
    <xsd:element name="TaxClassification" type="String" 30>
    <xsd:element name="AttributeCategory" type="String">
    <xsd:element name="Attribute1" type="String" 150>
    <xsd:element name="Attribute2" type="String" 150>
    <xsd:element name="Attribute3" type="String" 150>
    <xsd:element name="Attribute4" type="String" 150>
    <xsd:element name="Attribute5" type="String" 150>
    <xsd:element name="Attribute6" type="String" 150>
    <xsd:element name="Attribute7" type="String" 150>
    <xsd:element name="Attribute8" type="String" 150>
```

```
  <xsd:element name="Attribute9" type="String" 150>
  <xsd:element name="Attribute10" type="String" 150>
  <xsd:element name="Attribute11" type="String" 150>
  <xsd:element name="Attribute12" type="String" 150>
  <xsd:element name="Attribute13" type="String" 150>
  <xsd:element name="Attribute14" type="String" 150>
  <xsd:element name="Attribute15" type="String" 150>
  <xsd:element name="CompanyPrepaidInvoiceId" type="String" 15>
  <xsd:element name="Justification" type="String" 240>
  <xsd:element name="MerchantDocumentNumber" type="String" 80>
  <xsd:element name="MerchantName" type="String" 80>
  <xsd:element name="MerchantReference" type="String" 240>
  <xsd:element name="MerchantTaxpayer Id" type="String" 80>
  <xsd:element name="MerchantTaxRegistrationNumber" type="String" 80>
  <xsd:element name="CountryOfSupply" type="String" 5>
  <xsd:element name="CardReferenceId" type="String" 15>
          <xsd:complexType name="ExpenseAccount>
           <xsd:sequence>
                <!-- Each segment and value of the segment for the expense account will be --!>
                 <!-- printed out below --!>
        <xsd:simpleType name=?segment name? type="String">
        </xsd:simpleType>
    </xsd:sequence>
        </xsd:complexType>
        <xsd:element name="ProjectName" type="String">
        <xsd:element name="TaskName" type="String">
        <xsd:element name="ProjectExpenditureOrganization" type="String">
        <xsd:element name="ProjectExpenditureType" type="String">
</xsd:simpleType>
<xsd:simpleType name="Line">
  <xsd:element name="Type" type="String" value = 'Cash Advance Reversal' >
  <xsd:element name="Description" type="String" value = 'Cash Advance Reversal' >
  <xsd:element name="Amount" type="Decimal">
  <xsd:complexType name="ExpenseClearingAccount>
                  <xsd:sequence>
                          <!-- Each segment and value of the segment for the expense account will be --!>
                           <!-- printed out below --!>
                <xsd:simpleType name=?segment name? type="String">
                </xsd:simpleType>
              </xsd:sequence>
        </xsd:complexType>
</xsd:simpleType>
<xsd:simpleType name="Line">
        <xsd:element name="Type" type="String" value = 'Credit Card Reversal' >
        <xsd:element name="Description" type="String" value = 'Credit Card Reversal' >
        <xsd:element name="CreditCardId" type="String" 15>
        <xsd:element name="Amount" type="Decimal">
        <xsd:complexType name="ExpenseClearingAccount>
                    <xsd:sequence>
                     <!-- Each segment and value of the segment for the expense account will be --!>
```

```
                    <!-- printed out below --!>
                    <xsd:simpleType name=?segment name? type="String">
                  </xsd:simpleType>
                 </xsd:sequence>
               </xsd:complexType>
             </xsd:simpleType>
           </xsd:sequence>
           </xsd:complexType>
          </xsd:sequence>
          </xsd:complexType>
         </xsd:sequence>
        </xsd:complexTypes>
```

5.2. XML Schema Definition for Credit Card Issuer Invoice (Outbound)

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema elementFormDefault="qualified" sdoJava:package="" targetNamespace="">
<xsd:complexType name="Invoice">
 <xsd:sequence>
  <xsd:element name="RequestId" type="String" 15>
  <xsd:element name="ReferenceKey" type="String" 15>
  <xsd:element name="InvoiceDate" type="Date">
  <xsd:element name="PayeeName" type="String" 240>
          <xsd:simpleType name="PayeeAddress">
            <xsd:element name="AddressLine1" type="String" 240>
            <xsd:element name="AddressLine2" type="String" 240>
            <xsd:element name="AddressLine3" type="String" 240>
            <xsd:element name="AddressLine4" type="String" 240>
            <xsd:element name="City" type="String" 60>
            <xsd:element name="State" type="String" 60>
            <xsd:element name="PostalCode" type="String" 60>
            <xsd:element name="Province" type="String" 60>
            <xsd:element name="County" type="String" 60>
            <xsd:element name="CountryCode" type="String" 2>
          </xsd:simpleType>
    <xsd:element name="LegalEntity" type="String" 240>
    <xsd:element name="LedgerName" type="String" 30>
    <xsd:element name="BusinessUnitName" type="String" 240>
    <xsd:complexType name="LiabilityAccount">
      <xsd:sequence>
                    <!-- Each segment and value of the segment for the expense account will be --!>
                    <!-- printed out below --!>
        <xsd:simpleType name=?segment name? type="String">
        </xsd:simpleType>
      </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="Lines">
    <xsd:sequence>
      <xsd:simpleType name="Line">
        <xsd:element name="ReferenceKey" type="String" 15>
                    <!-- card number is masked --!>
```

```
        <xsd:element name="CardNumber" type="String" 30>
        <xsd:element name="EmployeeName" type="String" 240>
        <xsd:element name="EmployeeNumber" type="String" 30>
        <xsd:element name="Currency" type="String" 30>
        <xsd:element name="CardAmount" type="Decimal">
        <xsd:complexType name="ExpenseClearingAccount>
                    <xsd:sequence>
                        <!-- Each segment and value of the segment for the expense account will be --!>
                        <!-- printed out below --!>
                  <xsd:simpleType name=?segment name? type="String">
                  </xsd:simpleType>
                </xsd:sequence>
            </xsd:complexType>
      </xsd:simpleType>
    </xsd:sequence>
    </xsd:complexType>
          <xsd:element name="Currency" type="String" 30>
          <xsd:element name="InvoiceTotal" type="Decimal">
  </xsd:sequence>
</xsd:complexType>
```

5.3. XML Schema Definition for Cash Advance Request (Outbound)

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema elementFormDefault="qualified" sdoJava:package="" targetNamespace="">
<xsd:complexType name="CashAdvances">
 <xsd:sequence>
          <xsd:element name="RequestId" type="String" 15>
          <xsd:simpleType name="CashAdvance">
    <xsd:element name="AdvanceNum" type="String" 60>
    <xsd:element name="ReferenceKey" type="String" 15>
    <xsd:element name="Purpose" type="String" 240>
    <xsd:element name="AdvanceDate" type="Date">
    <xsd:element name="Currency" type="String" 30>
    <xsd:element name="AdvanceAmount" type="Decimal">
          <xsd:element name="ExchangeRateType" type="String">
    <xsd:element name="ExchangeRate" type="String">
    <xsd:element name="EmployeeName" type="String" 240>
    <xsd:element name="EmployeeNumber" type="String" 30>
    <xsd:element name="LegalEntityName" type="String" 240>
    <xsd:element name="LedgerName" type="String" 30>
    <xsd:element name="BusinessUnitname" type="String" 240>
    <xsd:complexType name="LiabilityAccount">
      <xsd:sequence>
          <!-- Each segment and value of the segment for the expense account will be --!>
          <!-- printed out below --!>
        <xsd:simpleType name=?segment name? type="String">
        </xsd:simpleType>
      </xsd:sequence>
    </xsd:complexType>
```

```
<xsd:complexType name="ExpenseClearingAccount>
        <xsd:sequence>
            <!-- Each segment and value of the segment for the expense account will be --!>
            <!-- printed out below --!>
          <xsd:simpleType name=?segment name? type="String">
          </xsd:simpleType>
    </xsd:sequence>
   </xsd:complexType>
  </simpleType>
 </xsd:sequence>
</xsd:complexType>
```

5.4. XML Schema Definition for Payment Update (Inbound)

```
<?xml version="1.0" encoding="UTF-8"?>
  <xsd:schema elementFormDefault="qualified" sdoJava:package="" targetNamespace="">
  <xsd:complexType name=Payments>
   <xsd:sequence>
   <xsd:complexType name="Payer">
    <xsd:sequence>
    <xsd:element name="PayerName" type="String" 240>
    <xsd:element name="PayerBankName" type="String" 80>
     <xsd:simpleType name="PayerAddress">
      <xsd:element name="AddressLine1" type="String" 240>
      <xsd:element name="AddressLine2" type="String" 240>
      <xsd:element name="AddressLine3" type="String" 240>
      <xsd:element name="AddressLine4" type="String" 240>
      <xsd:element name="City" type="String" 80>
      <xsd:element name="State" type="String" 80>
      <xsd:element name="PostalCode" type="String" 80>
      <xsd:element name="Province" type="String" 80>
      <xsd:element name="County" type="String" 80>
      <xsd:element name="CountryCode" type="String" 5>
     </xsd:simpleType>
    <xsd:complexType name="Payment">
     <xsd:sequence>
      <xsd:element name="PaymentReferenceNumber" type="String" 15>
      <xsd:element name="PaymentDate" type="Date">
      <xsd:element name="PaymentCurrency" type="String" 3>
      <xsd:element name="PaymentAmount" type="Decimal">
      <xsd:element name="PayeeName" type="String" 240>
      <xsd:element name="EmployeeNumber" type="String" 30>
      <xsd:element name="PaymentMethod" type="String" 100>
      <xsd:element name="CheckNumber" type="String" 18>
      <xsd:element name="MailingAddressType" type="String" 30>
      <xsd:element name="PayeeBankName" type="String" 80>
      <xsd:element name="PayeeBranchName" type="String" 80>
      <xsd:element name="PayeeMaskedBankAccountNumber" type="String" 100>
       <xsd:complexType name="Documents">
        <xsd:sequence>
```

```
            <xsd:simpleType name="Document">
              <xsd:element name="DocumentReferenceNumber" type="String" 50>
              <xsd:element name="DocumentCurrency" type="String" 15>
              <xsd:element name="DocumentAmount" type="Decimal">
              <xsd:element name="AmountPaid" type="Decimal">
            </xsd:simpleType>
          </xsd:sequence>
        </xsd:complexType>
    </xsd:sequence>
  </xsd:complexType>
  </xsd:sequence>
 </xsd:complexType>
</xsd:sequence>
</xsd:complexType>
<xsd:element name="Payment" type="Payment">
<name>Payment</name>
</xsd:schema>
```

6. Example XML Files

6.1. Expense Report with Cash Expense

» Consider an expense report with two cash expenses - $550 for Taxis and $22 for Meals

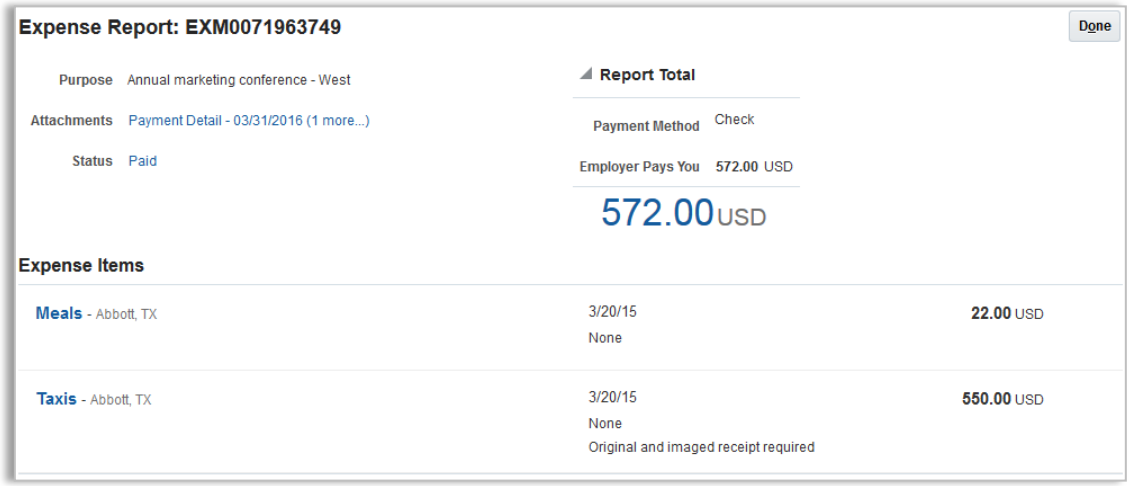

Figure 7- Expense Report with Cash Expenses

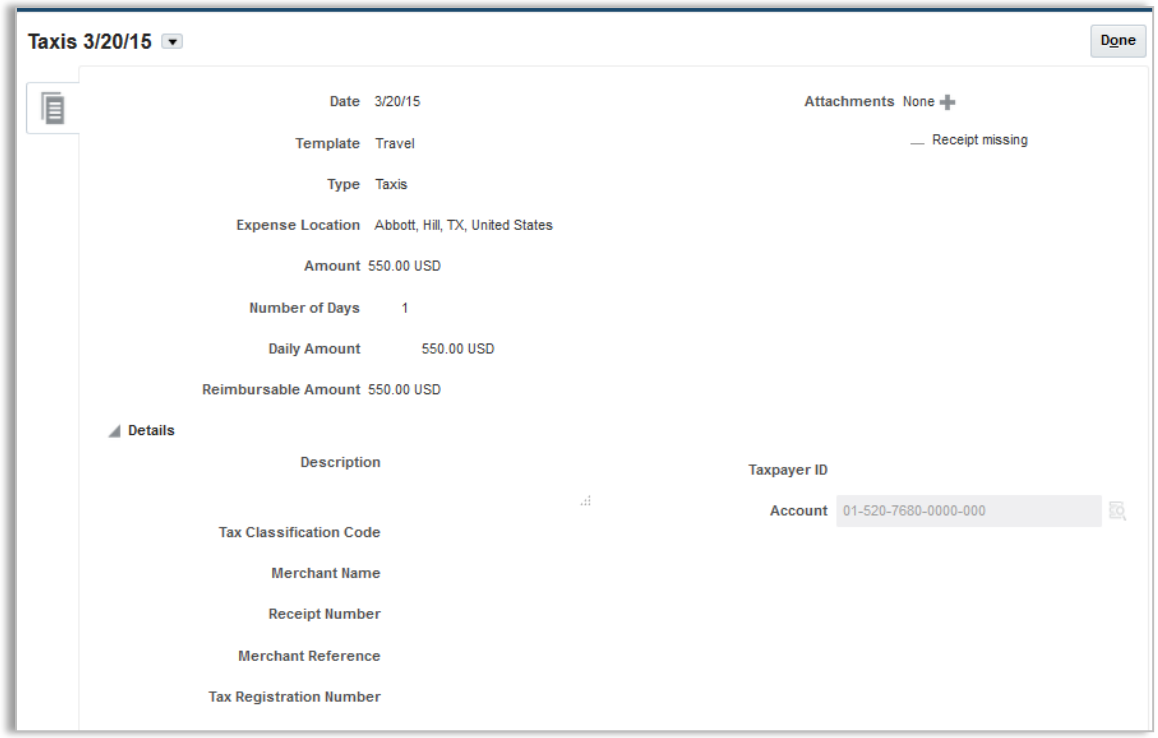

Figure 8 - Taxis Expense Line

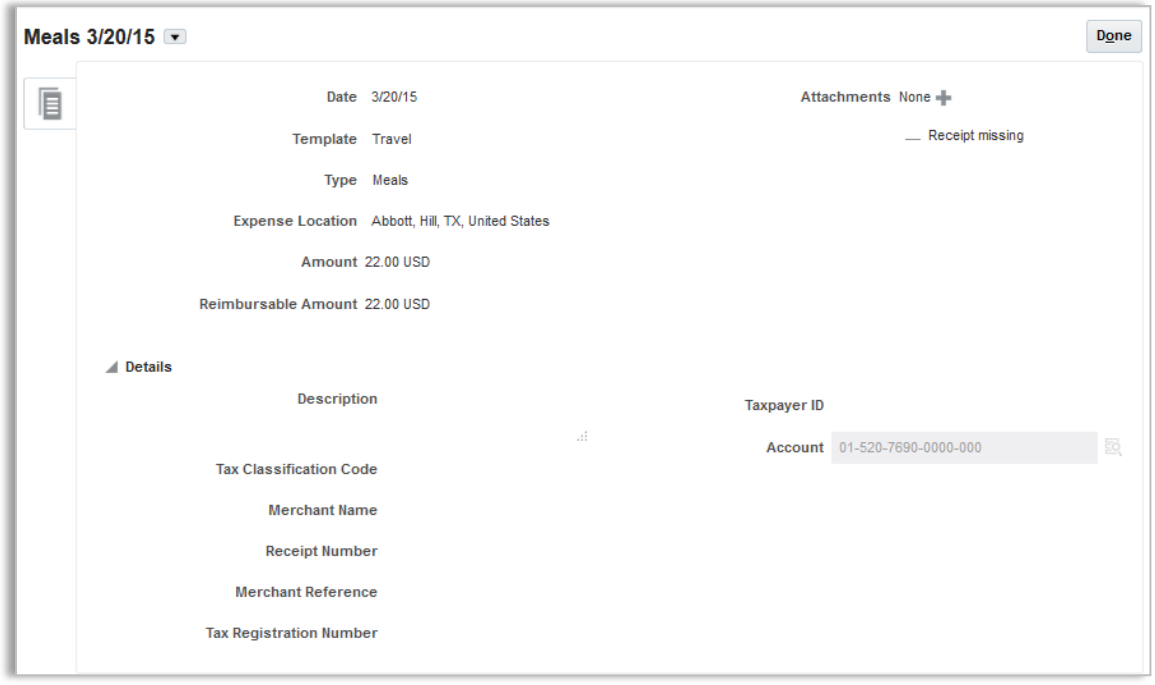

Figure 9 - Meals Expense Line

6.1.1. XML for the Expense Report with Cash Expense

Below is published XML for the above expense report with 2 cash expenses payable to the employee.

```
<ExpenseReports>
  <RequestId>37840</RequestId>
  <ExpenseReport>
    <Number>EXM0071963749</Number>
    <Purpose>Annual marketing conference - West</Purpose>
    <ReferenceKey>300100071963749</ReferenceKey>
    <ExpenseReportDate>2015-03-20</ExpenseReportDate>
    <PaymentCurrency>USD</PaymentCurrency>
    <PaymentAmount>572</PaymentAmount>
    <PaymentMethod>CHECK</PaymentMethod>
    <Source>Employee</Source>
    <EmployeeName>Johnson, Mary</EmployeeName>
    <EmployeeNumber>10026335772</EmployeeNumber>
    <EmployeeAddressCode>OFFICE</EmployeeAddressCode>
    <PayeeName>Mary Johnson</PayeeName>
    <LegalEntity>Vision Corporation</LegalEntity>
    <LedgerName>Vision Operations (USA)</LedgerName>
    <BusinessUnit>Vision Operations</BusinessUnit>
    <LiabilityAccount>
      <Company>01</Company>
      <Department>000</Department>
      <Account>2440</Account>
      <Sub-Account>0000</Sub-Account>
      <Product>000</Product>
    </LiabilityAccount>
    <Lines>
      <Line>
        <ReferenceKey>300100071963743</ReferenceKey>
        <ExpenseDate>2015-03-20</ExpenseDate>
        <ExpenseType>Taxis</ExpenseType>
        <Amount>550</Amount>
        <Description>Miscellaneous-Taxis</Description>
        <Location>Abbott, Hill, TX, United States</Location>
        <ReceiptCurrency>USD</ReceiptCurrency>
        <ReceiptConversionRate>1</ReceiptConversionRate>
        <ReceiptAmount>550</ReceiptAmount>
        <AttributeCategory>Taxis</AttributeCategory>
        <ExpenseAccount>
          <Company>01</Company>
          <Department>520</Department>
          <Account>7680</Account>
          <Sub-Account>0000</Sub-Account>
          <Product>000</Product>
        </ExpenseAccount>
      </Line>
      <Line>
        <ReferenceKey>300100071963747</ReferenceKey>
        <ExpenseDate>2015-03-20</ExpenseDate>
        <ExpenseType>Meals</ExpenseType>
```

```
        <Amount>22</Amount>
        <Description>Meals-Meals</Description>
        <Location>Abbott, Hill, TX, United States</Location>
        <ReceiptCurrency>USD</ReceiptCurrency>
        <ReceiptConversionRate>1</ReceiptConversionRate>
        <ReceiptAmount>22</ReceiptAmount>
        <AttributeCategory>Meals</AttributeCategory>
        <ExpenseAccount>
          <Company>01</Company>
          <Department>520</Department>
          <Account>7690</Account>
          <Sub-Account>0000</Sub-Account>
          <Product>000</Product>
        </ExpenseAccount>
      </Line>
    </Lines>
  </ExpenseReport>
</ExpenseReports>
```

6.1.2. Elements description for Expense Report with Cash Expense

Node	Element	Description
ExpenseReports		
	RequestId	Request ID for the Process Expense Reimbursements and Cash Advances background process
Expense Report		
	Number	Expense Report Number
	Purpose	Business activities that justify the expense report.
	ReferenceKey	Identifier of the expense report.
	ExpenseReportDate	Month, day, and year in which an expense report is saved or submitted.
	PaymentCurrency	Currency of the amount to be reimbursed to the individual for the expenses incurred.
	PaymentAmount	Total amount of the expense report in reimbursement currency. Amount in approver preferred currency is also shown if approver has a different preferred currency.
	PaymentMethod	Method of payment for this expense report such as check, cash, or credit.
	Source	Whether the expense report is for an employee or corporate card issuer *Values: Employee, CreditCard, Contractor*
	EmployeeName	Employee whose expenses are included in this expense report.
	EmployeeNumber	Employee number for the employee whose expenses are included in this expense report.
	EmployeeAddressCode	For employee, whether the reimbursement address for the employee is OFFICE or HOME

		Values: OFFICE, HOME
	PayeeName	The name of the person identified as Payee for the expenses included in this expense report.
	LegalEntity	Name of the legal employer of the employee
	LedgerName	Name of the ledger associated with the legal employer of the employee.
	BusinessUnit	Name of the business unit processing the expense reports.
LiabilityAccount		
	Company	Company segment of the liability account
	Department	Department or cost center segment of the liability account
	Account	Account segment of the liability account
	Sub-Account	Sub-account segment of the liability account
	Product	Product segment of the liability account
Lines		
Line		
	ReferenceKey	Identifier of the expense line.
	ExpenseDate	Date the expense was incurred.
	ExpenseType	Type of Expense as defined in Expense Type setup
	Amount	Amount for the expense.
	Description	Description of the expense.
	Location	Location of the expense.
	ReceiptCurrency	The currency of the expense as recorded in the receipt.
	ReceiptConversionRate	The exchange rate used to convert the receipt currency to employee reimbursement currency.
	ReceiptAmount	The expense amount as recorded on the expense receipt.
	AttributeCategory	The category of expense as per pre-defined expense categories.
	MerchantName	The name of the merchant for the incurred expense.
ExpenseAccount		
	Company	Company segment of the employee expense account
	Department	Department or cost center segment of the employee expense account
	Account	Account segment of the employee expense account
	Sub-Account	Sub-account segment of the employee expense account
	Product	Product segment of the employee expense account

6.2. Expense Report with Company Pay Credit Card Expense

- Consider an expense report with 2 expenses – $129.32 for Airfare and $34.00 for Meals
- The corporate card program has a company account that is setup with payment scenario as company pay. This implies that the employer will pay the card issuer for all credit card transactions but the employee would be reimbursed for cash expense.
- For expense reports containing Both Pay and Company Pay transactions, the reimbursement process creates reversal lines so the employee invoice can derive the net amount due to the employee.

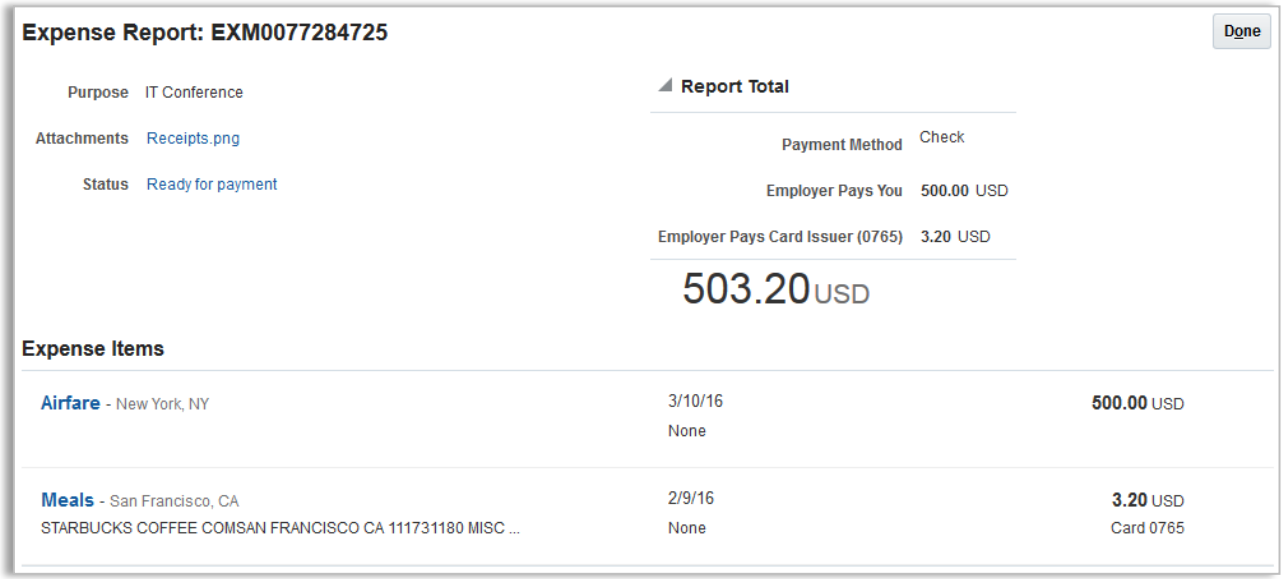

Figure 10 - Expense report with company pay and cash expense

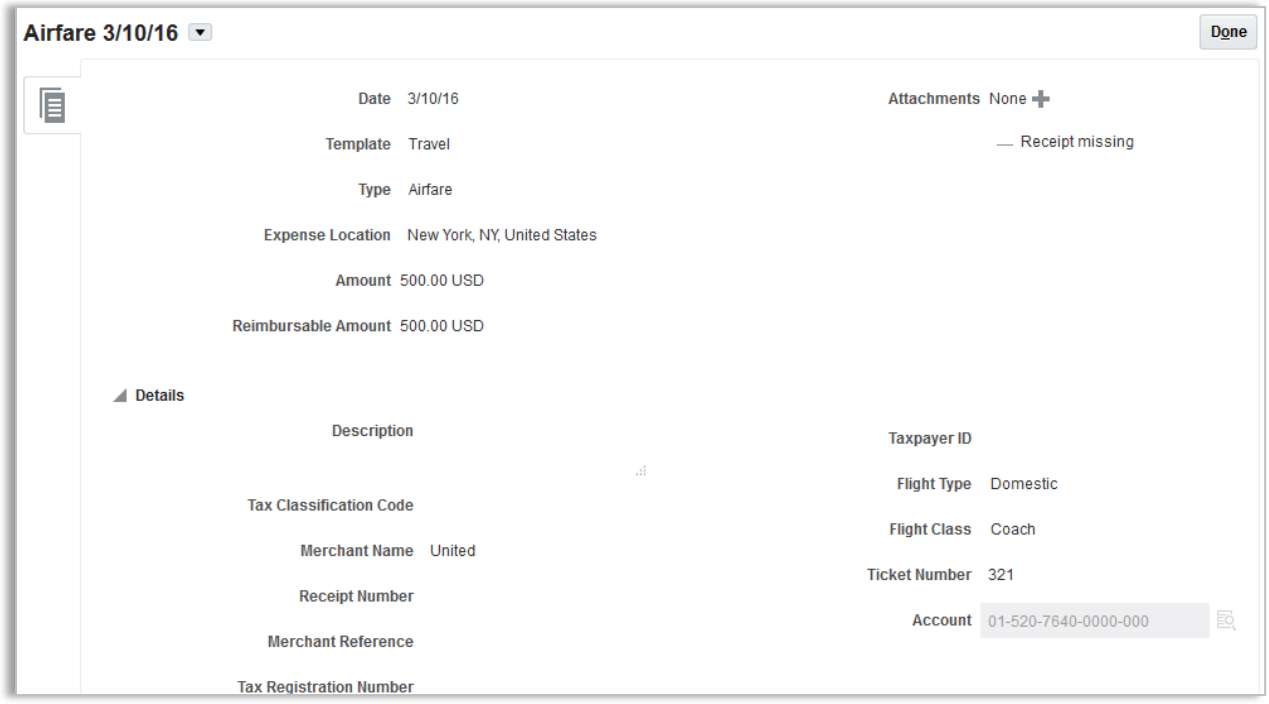

Figure 11 - Corporate Card Airfare Expense

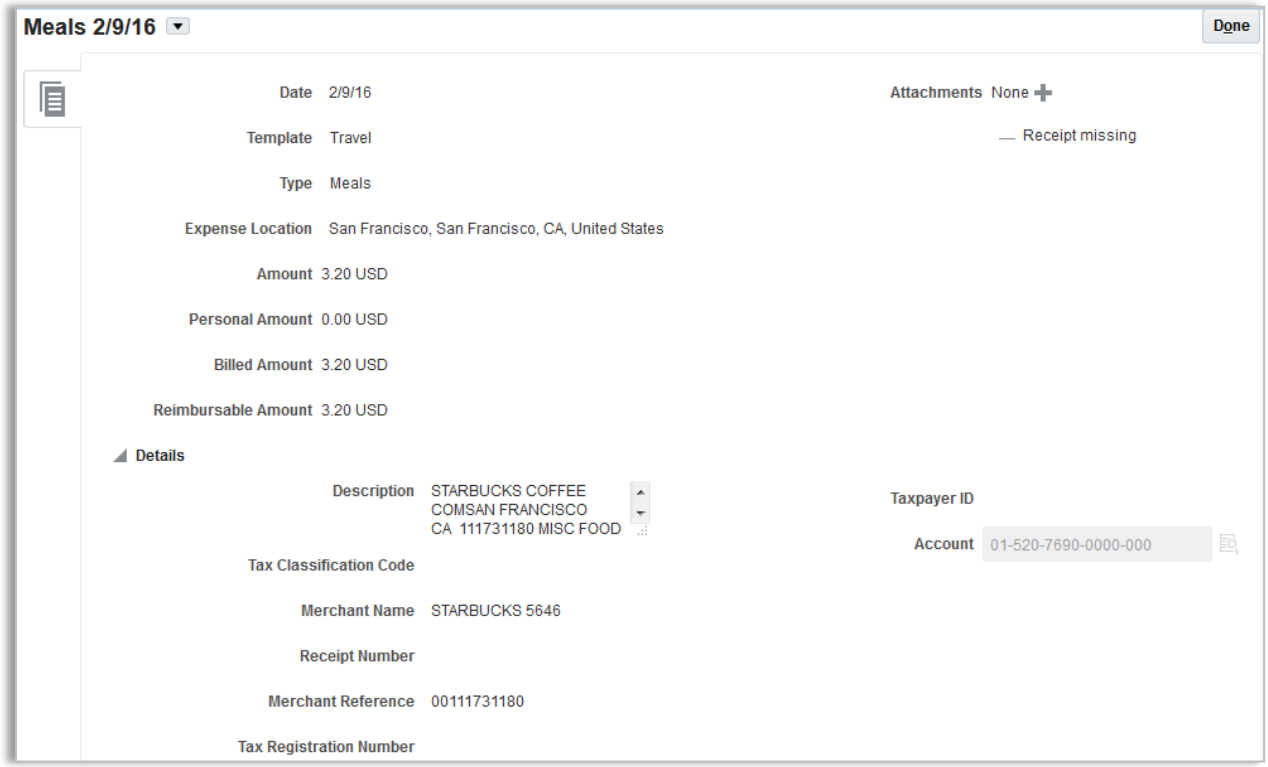

Figure 12 – Meals Cash Expense

6.2.1. XML for Expense Report with Company Pay Card

Below is published XML for the above expense report with one cash and the one corporate card expense. Employee would be reimbursed only for the cash expense and the corporate card issuer would be reimbursed for the credit card expense.

```
<ExpenseReports>
  <RequestId>69551</RequestId>
  <ExpenseReport>
    <Number>EXM0077284725</Number>
    <Purpose>IT Conference</Purpose>
    <ReferenceKey>300100077284725</ReferenceKey>
    <ExpenseReportDate>2016-03-10</ExpenseReportDate>
    <PaymentCurrency>USD</PaymentCurrency>
    <PaymentAmount>500</PaymentAmount>
    <PaymentMethod>CHECK</PaymentMethod>
    <Source>Employee</Source>
    <EmployeeName>Johnson, Mary</EmployeeName>
    <EmployeeNumber>10026335772</EmployeeNumber>
    <EmployeeAddressCode>OFFICE</EmployeeAddressCode>
    <PayeeName>Mary Johnson</PayeeName>
    <LegalEntity>Vision Corporation</LegalEntity>
    <LedgerName>Vision Operations (USA)</LedgerName>
    <BusinessUnit>Vision Operations</BusinessUnit>
    <LiabilityAccount>
      <Company>01</Company>
      <Department>000</Department>
```

```
      <Account>2440</Account>
      <Sub-Account>0000</Sub-Account>
      <Product>000</Product>
    </LiabilityAccount>
    <Lines>
      <Line>
        <ReferenceKey>300100072471693</ReferenceKey>
        <ExpenseDate>2016-03-10</ExpenseDate>
        <ExpenseType>Airfare</ExpenseType>
        <Amount>500</Amount>
        <Description>Airfare-Airfare</Description>
        <Location>New York, NY, United States</Location>
        <ReceiptCurrency>USD</ReceiptCurrency>
        <ReceiptConversionRate>1</ReceiptConversionRate>
        <ReceiptAmount>500</ReceiptAmount>
        <AttributeCategory>Airfare</AttributeCategory>
        <MerchantName>United</MerchantName>
        <ExpenseAccount>
          <Company>01</Company>
          <Department>520</Department>
          <Account>7640</Account>
          <Sub-Account>0000</Sub-Account>
          <Product>000</Product>
        </ExpenseAccount>
      </Line>
      <Line>
        <ReferenceKey>119914</ReferenceKey>
        <ExpenseDate>2016-02-09</ExpenseDate>
        <ExpenseType>Meals</ExpenseType>
        <Amount>3.2</Amount>
        <Description>STARBUCKS COFFEE COMSAN FRANCISCO     CA  111731180 MISC FOOD STORE
02/09/16</Description>
        <Location>San Francisco, San Francisco, CA, United States</Location>
        <ReceiptCurrency>USD</ReceiptCurrency>
        <ReceiptConversionRate>1</ReceiptConversionRate>
        <ReceiptAmount>3.2</ReceiptAmount>
        <AttributeCategory>Meals</AttributeCategory>
        <MerchantName>STARBUCKS 5646</MerchantName>
        <MerchantReference>00111731180</MerchantReference>
        <CardReferenceId>42765</CardReferenceId>
        <ExpenseAccount>
          <Company>01</Company>
          <Department>520</Department>
          <Account>7690</Account>
          <Sub-Account>0000</Sub-Account>
          <Product>000</Product>
        </ExpenseAccount>
      </Line>
      <Line>
        <Type>Credit Card Reversal</Type>
```

```
    <Description>Credit Card Reversal</Description>
    <CreditCardId>300100077521015</CreditCardId>
    <Amount>-3.2</Amount>
    <ExpenseClearingAccount>
      <Company>01</Company>
      <Department>000</Department>
      <Account>7699</Account>
      <Sub-Account>0000</Sub-Account>
      <Product>000</Product>
    </ExpenseClearingAccount>
  </Line>
 </Lines>
</ExpenseReport>
</ExpenseReports>
```

6.2.2. Elements description for Expense Report with Company Pay Card

Node	Element	Description
ExpenseReports		
	RequestId	Request ID for the Process Expense Reimbursements and Cash Advances background process
Expense Report		
	Number	Expense Report Number
	ReferenceKey	Identifier of the expense report.
	ExpenseReportDate	Month, day, and year in which an expense report is saved or submitted.
	PaymentCurrency	Currency of the amount to be reimbursed to the individual for the expenses incurred.
	PaymentAmount	Total amount of the expense report in reimbursement currency. Amount in approver preferred currency is also shown if approver has a different preferred currency.
	PaymentMethod	Method of payment for this expense report such as check, cash, or credit.
	Source	Whether the expense report is for an employee or corporate card issuer
	EmployeeName	Employee whose expenses are included in this expense report.
	EmployeeNumber	Employee number for the employee whose expenses are included in this expense report.
	EmployeeAddressCode	Whether the reimbursement address for the employee is OFFICE or HOME
	PayeeName	The name of the person identified as Payee for the expenses included in this expense report.
	LegalEntity	Name of the legal employer of the employee
	LedgerName	Name of the ledger associated with the legal employer of the employee.
	BusinessUnit	Name of the business unit processing the expense reports.
LiabilityAccount		
	Company	Company segment of the liability account
	Department	Department or cost center segment of the liability account
	Account	Account segment of the liability account
	Sub-Account	Sub-account segment of the liability

		account
	Product	Product segment of the liability account
Lines Line		
	ReferenceKey	Identifier of the expense line.
	ExpenseDate	Date the expense was incurred.
	ExpenseType	Type of Expense as defined in Expense Type setup
	Amount	Amount for the expense.
	Description	Description of the expense.
	Location	Location of the expense.
	ReceiptCurrency	The currency of the expense as recorded in the receipt.
	ReceiptConversionRate	The exchange rate used to convert the receipt currency to employee reimbursement currency.
	ReceiptAmount	The expense amount as recorded on the expense receipt.
	AttributeCategory	The category of expense as per pre-defined expense categories.
	MerchantName	The name of the merchant for the incurred expense.
ExpenseAccount		
	Company	Company segment of the employee expense account
	Department	Department or cost center segment of the employee expense account
	Account	Account segment of the employee expense account
	Sub-Account	Sub-account segment of the employee expense account
	Product	Product segment of the employee expense account
Line		
	ReferenceKey	Identifier of the expense line.
	ExpenseDate	Date the expense was incurred.
	ExpenseType	Type of Expense as defined in Expense Type setup
	Amount	Amount for the expense.
	Description	Description of the expense.
	Location	Location of the expense.
	ReceiptCurrency	The currency of the expense as recorded in the receipt.
	ReceiptConversionRate	The exchange rate used to convert the receipt currency to employee reimbursement currency.
	ReceiptAmount	The expense amount as recorded on the expense receipt.
	AttributeCategory	The category of expense as per pre-defined expense categories.
	MerchantName	The name of the merchant for the incurred expense.
	MerchantReference	Additional supplier reference information that may appear on the expense report, such as contact name and address.
	CardReferenceId	Identifier of the corporate card.
ExpenseAccount		
	Company	Company segment of the employee expense account
	Department	Department or cost center segment of the employee expense account
	Account	Account segment of the employee expense account
	Sub-Account	Sub-account segment of the employee expense account
	Product	Product segment of the employee expense account
Line		
	Type	Expense line indicating a credit card reversal.

	Description	Description of expense line.
	CreditcardId	Identifier of corporate card
	Amount	The amount for the corporate card expense line.
ExpenseClearing Account		
	Company	Company segment of the employee expense clearing account
	Department	Department or cost center segment of the employee expense clearing account
	Account	Account segment of the employee expense clearing account
	Sub-Account	Sub-account segment of the employee expense clearing account
	Product	Product segment of the employee expense clearing account

6.2.3. XML for Card Issuer Invoice - Payable to Card Issuer

Below is published XML for the amount payable to card issuer for the credit card expenses. Create Corporate Card Issuer Payment Request background process creates payment request for the total amount reimbursable to the card issuer for credit card expense for all employees.

```
<Invoice>
  <RequestId>69584</RequestId>
  <ReferenceKey>1459533475000</ReferenceKey>
  <InvoiceDate>2016-04-01</InvoiceDate>
  <PayeeName>EXM Card Issuer</PayeeName>
  <PayeeAddress>
    <AddressLine1>500 Oracle Parkway</AddressLine1>
    <City>Redwood City</City>
    <State>CA</State>
    <CountryCode>US</CountryCode>
  </PayeeAddress>
  <LegalEntity>Vision Operations</LegalEntity>
  <LedgerName>Vision Operations (USA)</LedgerName>
  <BusinessUnitName>Vision Operations</BusinessUnitName>
  <LiabilityAccount>
    <Company>01</Company>
    <Department>000</Department>
    <Account>2210</Account>
    <Sub-Account>0000</Sub-Account>
    <Product>000</Product>
  </LiabilityAccount>
  <Lines>
    <Line>
      <ReferenceKey>300100077521015</ReferenceKey>
      <CardNumber>0765</CardNumber>
      <EmployeeName>Mary Johnson</EmployeeName>
      <EmployeeNumber>10026335772</EmployeeNumber>
      <Currency>USD</Currency>
      <CardAmount>3073.97</CardAmount>
      <ExpenseClearingAccount>
        <Company>01</Company>
        <Department>000</Department>
        <Account>7699</Account>
```

```
        <Sub-Account>0000</Sub-Account>
        <Product>000</Product>
      </ExpenseClearingAccount>
    </Line>
  </Lines>
  <Currency>USD</Currency>
  <InvoiceTotal>3073.97</InvoiceTotal>
</Invoice>
```

6.2.4. Elements Description for Card Issuer Invoice

Below table lists elements and the data details for total credit card expense payable to corporate card issuer for credit card expense for all employees

Node	Element	Description
Invoice		
	RequestId	Request ID for the Create Card Issuer Payment request background process
	ReferenceKey	Identifier of the expense report.
	InvoiceDate	Month, day, and year in which an invoice has to be recorded.
	PayeeName	The name of the person identified as Payee for the credit card expense included in this expense report.
PayeeAddress		
	AddressLine1	The first line of payee remittance address
	City	City for payee remittance address
	State	State for payee remittance address
	CountryCode	CountryCode for the payee remittance address
	LegalEntity	Name of the legal employer of the employee
	LedgerName	Name of the ledger associated with the legal employer of the employee.
	BusinessUnitName	Name of the business unit processing the expense reports.
LiabilityAccount		
	Company	Company segment of the liability account
	Department	Department or cost center segment of the liability account
	Account	Account segment of the liability account
	Sub-Account	Sub-account segment of the liability account
	Product	Product segment of the liability account
Lines Line		
	ReferenceKey	Identifier of the expense line.
	CardNumber	Last 4 digits of the credit card number
	EmployeeName	Name of the employee incurring the credit card expense.
	EmployeeNumber	Employee number identifier for the employee incurring the credit card expense
	Currency	Currency for the credit card expense
	CardAmount	Total corporate card expense by the employee.
ExpenseClearing Account		
	Company	Company segment of the employee expense clearing account
	Department	Department or cost center segment of the employee expense clearing account
	Account	Account segment of the employee

	expense clearing account
Sub-Account	Sub-account segment of the employee expense clearing account
Product	Product segment of the employee expense clearing account

6.3. Cash Advance Request

» Consider a cash advance request for $150.00

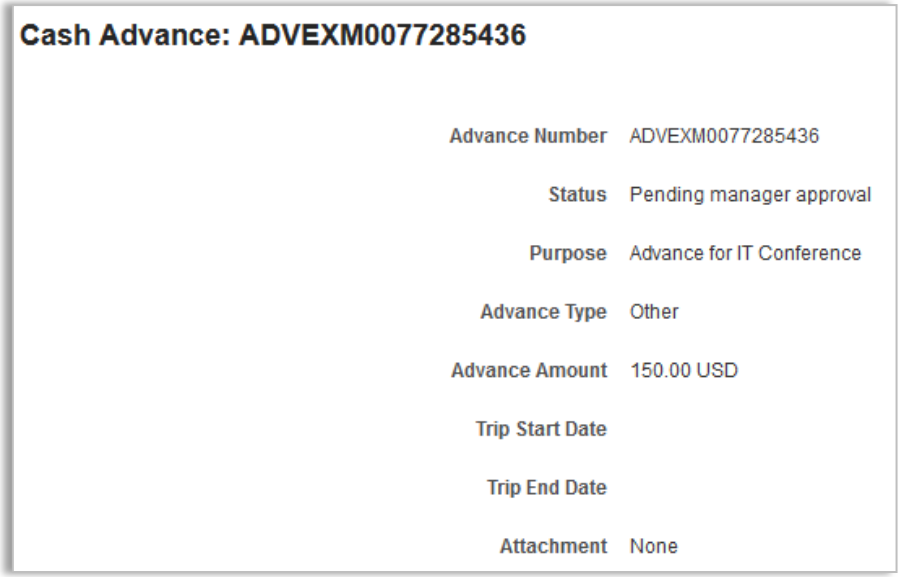

Figure 13 - Cash Advance Request

6.3.1. XML for Cash Advance Request

Below is the XML for the payment request to employee for cash advance.

```
<CashAdvances>
  <RequestId>69606</RequestId>
  <CashAdvance>
    <AdvanceNum>ADVEXM0077285436</AdvanceNum>
    <ReferenceKey>300100077285436</ReferenceKey>
    <Purpose>Advance for IT Conference</Purpose>
    <AdvanceType>OTHER</AdvanceType>
    <AdvanceDate>2016-04-01</AdvanceDate>
    <Currency>USD</Currency>
    <AdvanceAmount>150</AdvanceAmount>
    <EmployeeName>Johnson, Mary</EmployeeName>
    <EmployeeNumber>10026335772</EmployeeNumber>
    <LegalEntityName>Vision Operations</LegalEntityName>
    <LedgerName>Vision Operations (USA)</LedgerName>
    <BusinessUnitName>Vision Operations</BusinessUnitName>
    <LiabilityAccount>
      <Company>01</Company>
      <Department>000</Department>
      <Account>2440</Account>
      <Sub-Account>0000</Sub-Account>
      <Product>000</Product>
```

```
    </LiabilityAccount>
    <ExpenseClearingAccount>
      <Company>01</Company>
      <Department>520</Department>
      <Account>2220</Account>
      <Sub-Account>0000</Sub-Account>
      <Product>000</Product>
    </ExpenseClearingAccount>
  </CashAdvance>
</CashAdvances>
```

6.3.2. Elements Description for Cash Advance Request

Below table lists elements and the data details for cash advance payment request.

Node	Element	Description
CashAdvances		
	RequestId	Request ID for the Process Expense Reimbursements and Cash Advances background process
CashAdvance		
	AdvanceNum	Number identifier for the cash advance
	ReferenceKey	Identifier of the expense report.
	Purpose	Business activities that justify the cash advance.
	AdvanceDate	Month, day, and year on which the advance was requested.
	AdvanceAmount	Amount requested ascash advance.
	EmployeeName	Employee whose expenses are included in this cash advance request.
	EmployeeNumber	Employee number for the employee whose expenses are included in this ash advance request.
	LegalEntity	Name of the legal employer of the employee
	LedgerName	Name of the ledger associated with the legal employer of the employee.
	BusinessUnitName	Name of the business unit processing the cash advance.
LiabilityAccount		
	Company	Company segment of the liability account
	Department	Department or cost center segment of the liability account
	Account	Account segment of the liability account
	Sub-Account	Sub-account segment of the liability account
	Product	Product segment of the liability account
ExpenseClearing Account		
	Company	Company segment of the employee expense clearing account
	Department	Department or cost center segment of the employee expense clearing account
	Account	Account segment of the employee expense clearing account
	Sub-Account	Sub-account segment of the employee expense clearing account
	Product	Product segment of the employee expense clearing account

» **Subsequently, employee submits an expense report for $375.00. Cash advance would be applied to the expense report and the employee would be reimbursed $225.00 after deducting the advance amount for $150.**

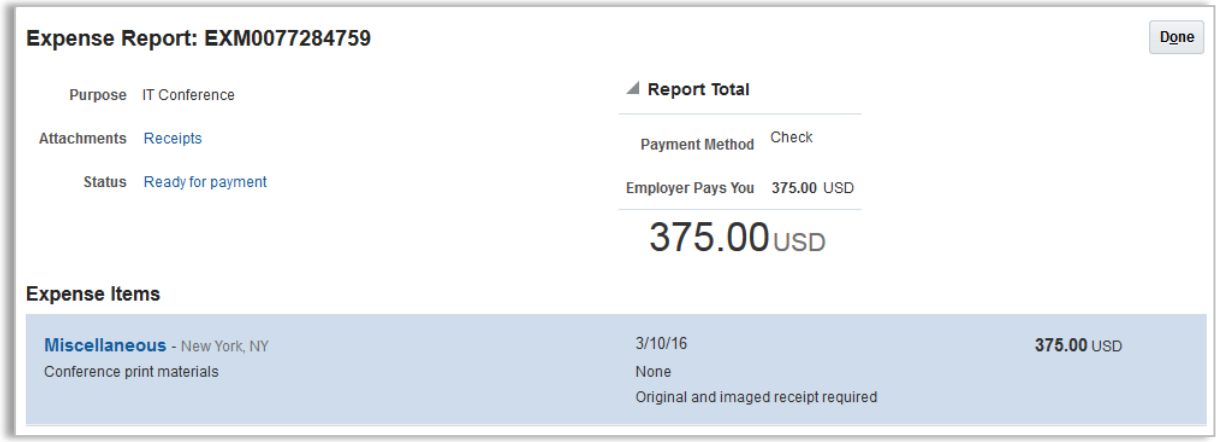

Figure 14 - Expense Report to which Advance is applied

6.3.3. XML for an Expense Report with Cash Advance Application

Below is the XML for the expense report after cash advance application. There would be 1 line for the actual cash expense and another for the cash advance reversal.

```
<ExpenseReports>
  <RequestId>69646</RequestId>
  <ExpenseReport>
    <Number>EXM0077284759</Number>
    <Purpose>IT Conference</Purpose>
    <ReferenceKey>300100077284759</ReferenceKey>
    <ExpenseReportDate>2016-03-10</ExpenseReportDate>
    <PaymentCurrency>USD</PaymentCurrency>
    <PaymentAmount>225</PaymentAmount>
    <PaymentMethod>CHECK</PaymentMethod>
    <Source>Employee</Source>
    <EmployeeName>Johnson, Mary</EmployeeName>
    <EmployeeNumber>10026335772</EmployeeNumber>
    <EmployeeAddressCode>OFFICE</EmployeeAddressCode>
    <PayeeName>Mary Johnson</PayeeName>
    <LegalEntity>Vision Corporation</LegalEntity>
    <LedgerName>Vision Operations (USA)</LedgerName>
    <BusinessUnit>Vision Operations</BusinessUnit>
    <LiabilityAccount>
      <Company>01</Company>
      <Department>000</Department>
      <Account>2440</Account>
      <Sub-Account>0000</Sub-Account>
      <Product>000</Product>
    </LiabilityAccount>
    <Lines>
      <Line>
        <ReferenceKey>300100077284762</ReferenceKey>
        <ExpenseDate>2016-03-10</ExpenseDate>
```

```
      <ExpenseType>Miscellaneous</ExpenseType>
      <Amount>375</Amount>
      <Description>Conference print materials</Description>
      <Location>New York, NY, United States</Location>
      <ReceiptCurrency>USD</ReceiptCurrency>
      <ReceiptConversionRate>1</ReceiptConversionRate>
      <ReceiptAmount>375</ReceiptAmount>
      <AttributeCategory>Miscellaneous</AttributeCategory>
      <MerchantDocumentNumber>R1277654</MerchantDocumentNumber>
      <ExpenseAccount>
        <Company>01</Company>
        <Department>520</Department>
        <Account>7699</Account>
        <Sub-Account>0000</Sub-Account>
        <Product>000</Product>
      </ExpenseAccount>
    </Line>
    <Line>
      <Type>Cash Advance Reversal</Type>
      <Description>Cash Advance Reversal</Description>
      <Amount>-150</Amount>
      <ExpenseClearingAccount>
        <Company>01</Company>
        <Department>000</Department>
        <Account>2220</Account>
        <Sub-Account>0000</Sub-Account>
        <Product>000</Product>
      </ExpenseClearingAccount>
    </Line>
  </Lines>
</ExpenseReport>
```

6.4. XML File for Payment Update

Once you make payments in your financials application, you need to import the payment reference back to Expenses Cloud Service. Below is the XML format to be used for payment reference data upload.

```
<?xml version="1.0" encoding="UTF-8"?>
<Payments>
 <Payer>
  <PayerName>Vision Operations (USA)</PayerName>
  <PayerBankName>Bank of the West</PayerBankName>
  <PayerAddress>
   <AddressLine1>500 Wolfe Road</AddressLine1>
   <AddressLine2 />
   <AddressLine3 />
   <AddressLine4 />
   <City>Santa Clara</City>
   <State>CA</State>
   <PostalCode>95051</PostalCode>
   <Province>CA</Province>
```

```
  <County>Sants Clara</County>
  <CountryCode>US</CountryCode>
 </PayerAddress>
 <Payment>
  <PaymentReferenceNumber>1203141</PaymentReferenceNumber>
  <PaymentDate>2016-04-01</PaymentDate>
  <PaymentCurrency>USD</PaymentCurrency>
  <PaymentAmount>225.00</PaymentAmount>
  <PayeeName>Mary Johnson</PayeeName>
  <EmployeeNumber>10026335772</EmployeeNumber>
  <PaymentMethod>Check</PaymentMethod>
  <CheckNumber>2502114</CheckNumber>
  <CheckMailingAddress>Office</CheckMailingAddress>
  <PayeeBankName>Wells Fargo</PayeeBankName>
  <PayeeBranchName>San Jose Branch</PayeeBranchName>
  <PayeeMaskedBankAccountNumber>XXXXXXX99</PayeeMaskedBankAccountNumber>
  <Documents>
   <Document>
    <DocumentReferenceNumber>EXM0077284759</DocumentReferenceNumber>
    <DocumentCurrency>USD</DocumentCurrency>
    <DocumentAmount>225.00.00</DocumentAmount>
    <AmountPaid>225.00</AmountPaid>
   </Document>
  </Documents>
  </Payment>
</Payer>
</Payments>
```

6.4.1. Element Description for Payment Update

Below table lists elements and the data details for cash advance payment request.

Node	Element	Description
Payments		
Payer		
	PayerName	The name of the entity issuing payment.
	PayerBankName	Name of the bank issuing the payment.
PayerAddress		
	AddressLine1	First line of the payer's address.
	AddressLine2	Second line of the payer's address.
	AddressLine3	Third line of the payer's address.
	AddressLine4	Forth line of the payer's address.
	City	City component of payer's address.
	State	State component of payer's address.
	Province	Province component of payer's address.
	County	Country component of payer's address.

	CountryCode	CountryCode of payer's address.
Payment		
	PaymentReferenceNumber	Reference number or identifier of the payment,
	PaymentDate	Date when the payment was issued.
	PaymentCurrency	Currency in which the payment was issued.
	PaymentAmount	Payment amount.
	PayeeName	Name of the person to whom the payment was issued.
	EmployeeNumber	Number identifier for the employee.
	PaymentMethod	Name of the payment method.
	CheckNumber	Check number identifier.
	MailingAddressType	Type of mailing address such as Home or Office.
	PayeeBankName	Name of the payee's bank
	PayeeBranchName	Branch name of the payee's bank
	PayeeMaskedBankAccountNumber	The masked bank account number for the payee showing the last 4 digits.
Documents Document		
	DocumentReferenceNumber	The reference number identifier for the document such as expense report or cash advance getting paid.
	DocumentCurrency	The currency for the reimbursable amount for the document.
	DocumentAmount	The reimbursable amount for the document.
	AmountPaid	The amount actually reimbursed to the employee.

7. Frequently Asked Questions

i. Do I still have to subscribe to Financials Cloud Service even though it won't be used if I subscribe to Expenses Cloud Service?

- Yes. It is a pre-requisite when subscribing to Expenses Cloud Service. Expenses Cloud Service requires subscription to Financials Cloud Service and Transaction Business Intelligence for Financials Cloud Service as prerequisites (10 users minimum).

ii. Is corporate card processing supported when using third-party payment processing option in Expenses Cloud Service?

- Yes. Expense processing using the corporate cards is supported when issuing payments using the legacy third party financial application. Corporate Card Issuers need to be setup for Expenses utilizing the *Manage Corporate Cards* task in Functional Setup Manager. Based on whether the corporate card setup is company-pay or both-pay, the corporate card issuer, employee or both would be reimbursed for expenses

iii. Can cash advances be processed when utilizing the third-party payment processing option in Expenses Cloud Service?

- Yes. Expenses Cloud Service allows employees to request cash advance and approvers to review and approve the cash advance request. When using the third-party payment processing option, the payments for the cash advance would be issued in the legacy third-party application.

iv. Can HCM, Accounting Hub Reporting Cloud Service (FAHRCS) and Expenses be provisioned on single instance so that integration is not the issue.

» Yes. A global single instance(GSI) with FAHRCS and Expenses can be provisioned.

v. Do I need to create employees in Oracle Cloud for Expenses implementation?

Yes. For expenses implementation, employees need to created in Oracle HCM Cloud but unlike E-Business Suite, employees are not modeled as suppliers in Expenses Cloud Service.

vi. Can there be single sign-on (SSO) between Expenses Cloud service and Oracle Applications Unlimited on-premise instance.

» Yes. The single sign-on between Cloud and legacy on-premise systems can be supported by using federaion which needs to be requested for provisioning.

vii. Can we leverage the synchronization/setup of the Chart of Accounts (COA)/Ledgers and balances when considering Expenses co-existence with EBS?

» Yes, you can leverage synchronization of COA/Ledger provided by Accounting Hub Reporting Center Service (FAHRCS) for E-Business Suite release 12. After the ledger and the supporting setup are created in Oracle Financials Cloud, you can perform the additional setups as specified in the Financials Configuration section.

viii. Since the generated documents are stored in Oracle UCM (Universal Content Manager), is UCM included with the Cloud Expenses subscription?

» Yes.

8. Conclusion

There are many advantages to moving your Expenses system to the Oracle Cloud but it should not increase your cost and complexity of the integration. With Expenses Cloud Service, you can streamline expense entry for on-the-go workforce with mobile expenses solution with travel and corporate card integration. Integrate with your existing financial system to continue to benefit from existing business processes and reduce cost.

ORACLE

Oracle Corporation, World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065, USA

Worldwide Inquiries
Phone: +1.650.506.7000
Fax: +1.650.506.7200

CONNECT WITH US


blogs.oracle.com/oracle


facebook.com/oracle


twitter.com/oracle


oracle.com

Hardware and Software, Engineered to Work Together

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. This document is provided for information purposes only, and the contents hereof are subject to change without notice. This document is not warranted to be error-free, nor subject to any other warranties or conditions, whether expressed orally or implied in law, including implied warranties and conditions of merchantability or fitness for a particular purpose. We specifically disclaim any liability with respect to this document, and no contractual obligations are formed either directly or indirectly by this document. This document may not be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without our prior written permission.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.0115

Oracle Cloud Expenses Co-existence and Integration Options
Mar 2016


Oracle is committed to developing practices and products that help protect the environment