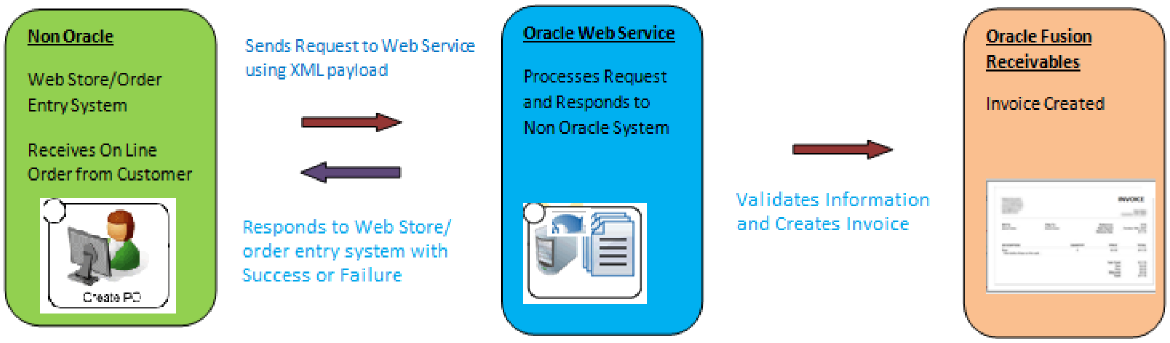
Usage Notes for Receivables Invoice Service

Table of Contents

1 Overview

An invoice or bill is a Receivables document issued by a seller to a buyer, indicating the products, quantities, and agreed prices for products or services the seller has provided the buyer. The **Receivables Invoice Service** integrates the deploying company's non-Oracle order entry/online web-store systems to Oracle Fusion Receivables. This service receives the order information from the online user and creates the corresponding invoice or interfaces the order details to Receivables.


Non Oracle
Web Store/Order Entry System
Receives On Line Order from Customer
Create PO
Sends Request to Web Service using XML payload
Responds to Web Store/ order entry system with Success or Failure
Oracle Web Service
Processes Request and Responds to Non Oracle System
Validates Information and Creates Invoice
Oracle Fusion Receivables
Invoice Created
INVOICE

.

Web Services Description Language (WSDL) of the **Receivables Invoice Service** can be accessed from the SOAP Web Services for Oracle Financials Cloud guide and is available to any user who has access to Oracle Fusion Receivables. This service supports both synchronous and asynchronous processing.

The **Receivables Invoice Service** is a SOAP encoded Web service which uses HTTPS transport as defined in the WSDL. A SOAP client can be manually created to invoke this service (example described in appendix 3.1). Another alternative is to automatically invoke the service via WSDL invocation tools, such as the Web Services Invocation Framework (WSIF) for Java clients or SOAP::Lite for Perl.

2 Service Details

2.1 Supported Operations

The following operations are available in the Receivables Invoice Service.

createSimpleInvoice

This operation is used to create a simple invoice document in synchronous mode. It accepts an object of Receivables Invoice Header SDO as the input parameter and returns a response of Receivables Invoice Result SDO object.

createSimpleInvoiceAsync.

This operation is used to create a simple invoice document in asynchronous mode. It accepts an object of Receivables Invoice Header SDO as the input parameter and returns a response of Receivables Invoice Result SDO object.

createSimpleInvoiceAsyncResponse

This operation is used to get the response of the asynchronous invocation.

processInterfaceLine

This operation is used to populate the Lines Interface (ra_interface_lines_all) table with bulk order lines from the external system in synchronous mode. It accepts an object of Receivables Interface Line SDO as the input parameter and returns a response of Receivables Interface Line SDO object.

processInterfaceLineAsync

This operation is used to populate the Lines Interface (ra_interface_lines_all) table with bulk order lines from the external system in asynchronous mode. It accepts an object of Receivables Interface Line SDO as the input parameter and returns a response of Receivables Interface Line SDO object.

processInterfaceDistribution

This operation is used to populate the Distributions Interface (ra_interface_distributions_all) table with bulk distribution lines from the external system in synchronous mode. It accepts an object of Receivables Interface Distribution SDO as the input parameter and returns a response of Receivables Interface Distribution SDO object.

processInterfaceDistributionAsync

This operation is used to populate the Distributions Interface (ra_interface_distributions_all) table with bulk distribution lines from the external system in asynchronous mode. It accepts an object of Receivables Interface Distribution SDO as the input parameter and returns a response of Receivables Interface Distribution SDO object.

processInterfaceSalesCredit

This operation is used to populate the Sales Credit Interface (ra_interface_salescredits_all) table with bulk sales credit lines from the external system in synchronous mode. It accepts an object of Receivables Interface Sales Credit SDO as the input parameter and returns a response of Receivables Interface Sales Credit SDO object.

processInterfaceSalesCreditAsync

This operation is used to populate the Sales Credit Interface (ra_interface_salescredits_all) table with bulk sales credit lines from the external system in asynchronous mode. It accepts an object of Receivables Interface Sales Credit SDO as the input parameter and returns a response of Receivables Interface Sales Credit SDO object.

processInterfaceContingency

This operation is used to populate the Contingencies Interface (ar_interface_conts_all) table with bulk contingency lines from the external system in synchronous mode. It accepts an object of Receivables Interface Contingencies SDO as the input parameter and returns a response of Receivables Interface Contingencies SDO object.

processInterfaceContingencyAsync

This operation is used to populate the Contingencies Interface (ar_interface_conts_all) table with bulk contingency lines from the external system in asynchronous mode. It accepts an object of Receivables Interface Contingencies SDO as the input parameter and returns a response of Receivables Interface Contingencies SDO object.

createInterfaceLine

This operation is used to populate the Lines Interface (ra_interface_lines_all) table with one order line from the external system in synchronous mode. It accepts an object of Receivables Interface Line SDO as the input parameter and returns a response of Receivables Interface Line SDO object.

createInterfaceLineAsync

This operation is used to populate the Lines Interface (ra_interface_lines_all) table with one order line from the external system in asynchronous mode. It accepts an object of Receivables Interface Line SDO as the input parameter and returns a response of Receivables Interface Line SDO object.

createInterfaceDistribution

This operation is used to populate the Distributions Interface (ra_interface_distributions_all) table with one distribution line from the external system in synchronous mode. It accepts an object of Receivables Interface

Distribution SDO as the input parameter and returns a response of Receivables Interface Distribution SDO object.

createInterfaceDistributionAsync

This operation is used to populate the Distributions Interface (ra_interface_distributions_all) table with one distribution line from the external system in asynchronous mode. It accepts an object of Receivables Interface Distribution SDO as the input parameter and returns a response of Receivables Interface Distribution SDO object.

createInterfaceSalesCredit

This operation is used to populate the Sales Credit Interface (ra_interface_salescredits_all) table with one sales credit line from the external system in synchronous mode. It accepts an object of Receivables Interface Sales Credit SDO as the input parameter and returns a response of Receivables Interface Sales Credit SDO object.

createInterfaceSalesCreditAsync

This operation is used to populate the Sales Credit Interface (ra_interface_salescredits_all) table with one sales credit line from the external system in asynchronous mode. It accepts an object of Receivables Interface Sales Credit SDO as the input parameter and returns a response of Receivables Interface Sales Credit SDO object.

createInterfaceContingency

This operation is used to populate the Contingencies Interface (ar_interface_conts_all) table with one contingency line from the external system in synchronous mode. It accepts an object of Receivables Interface Contingencies SDO as the input parameter and returns a response of Receivables Interface Contingencies SDO object.

createInterfaceContingencyAsync

This operation is used to populate the Contingencies Interface (ar_interface_conts_all) table with one contingency line from the external system in asynchronous mode. It accepts an object of Receivables Interface Contingencies SDO as the input parameter and returns a response of Receivables Interface Contingencies SDO object.

The asynchronous requests to Oracle web service are placed in a processing queue and handled asynchronously with other requests. The client application does not wait for a response. Once a job is submitted, a job ID is returned in the Web services response. The client application can then check on the status and result of the request by referencing the job ID.

2.2 Setups and Security

2.2.1 Prerequisite Setups

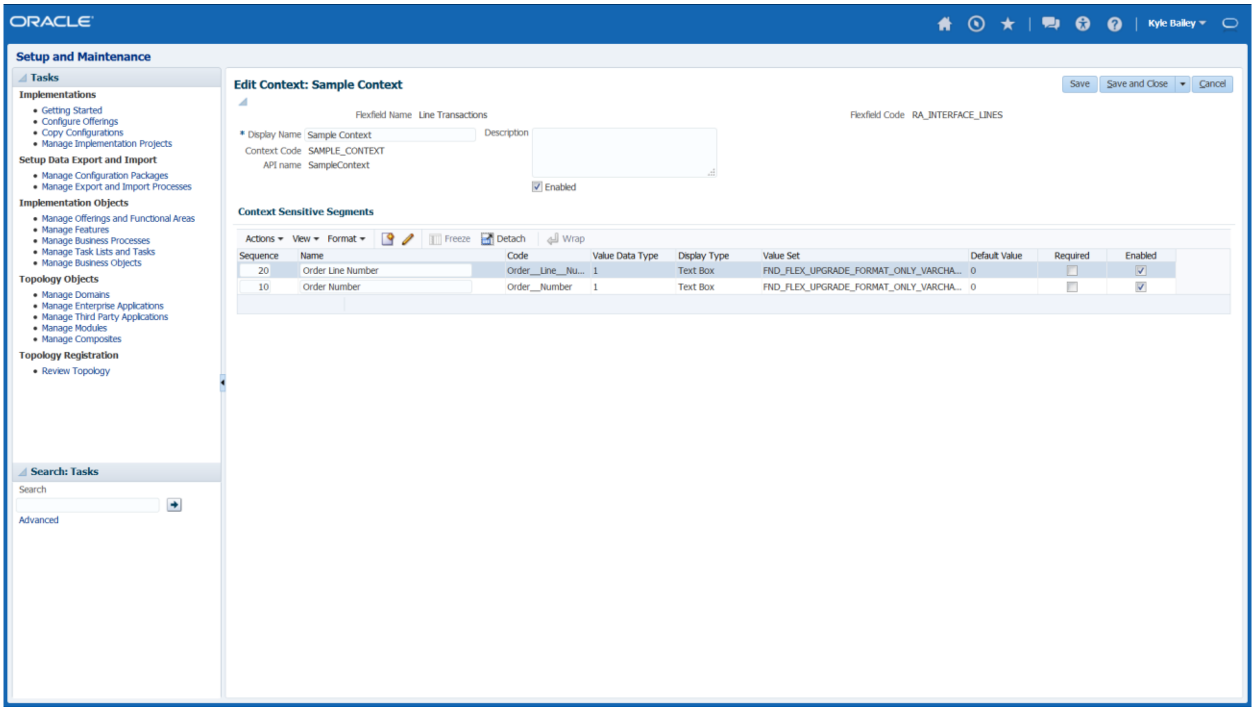
The prerequisite setups needed to invoke the Receivables invoice service are the same as creating an invoice. These can be set up in the Fusion instance through the Functional Setup Manager.

- Customer
- Receivables System Options
- Receivables Payment Terms
- Transaction Type
- Transaction Source
- AutoAccounting Rules
- Remit-to Address

2.2.2 Line Transaction Flexfield Setup for Interface Service Operations

Query the RA_INTERFACE_LINES Flexfield and create the desired context according to the business requirements. A sample context definition is shown below which will be referred in the sample payloads.

a) Descriptive Flexfield Context Details



Edit Context: Sample Context

Flexfield Name Line Transactions

Flexfield Code RA_INTERFACE_LINES

* Display Name Sample Context

Description

Context Code SAMPLE_CONTEXT

API name SampleContext

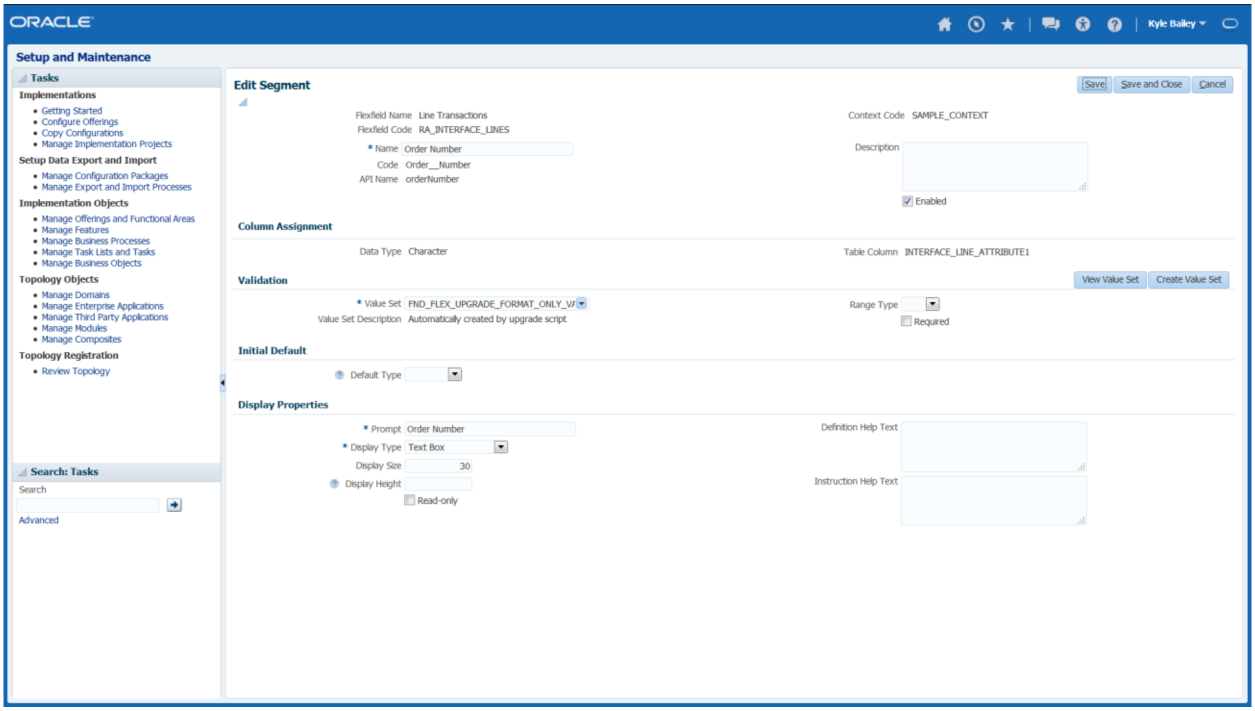
☑ Enabled

Context Sensitive Segments

Sequence	Name	Code	Value Data Type	Display Type	Value Set	Default Value	Required	Enabled
20	Order Line Number	Order__Line__Nu...	1	Text Box	FND_FLEX_UPGRADE_FORMAT_ONLY_VARCHA...	0	□	☑
10	Order Number	Order__Number	1	Text Box	FND_FLEX_UPGRADE_FORMAT_ONLY_VARCHA...	0	□	☑

Please note down the *Context Code* and the *API name* which needs to be used in the web service invocation.

b) Context Segment Details



Edit Segment

Flexfield Name: Line Transactions
Flexfield Code: RA_INTERFACE_LINES
* Name: Order Number
Code: Order__Number
API Name: orderNumber
Context Code: SAMPLE_CONTEXT
Description:
☑ Enabled

Column Assignment

Data Type: Character
Table Column: INTERFACE_LINE_ATTRIBUTE1

Validation

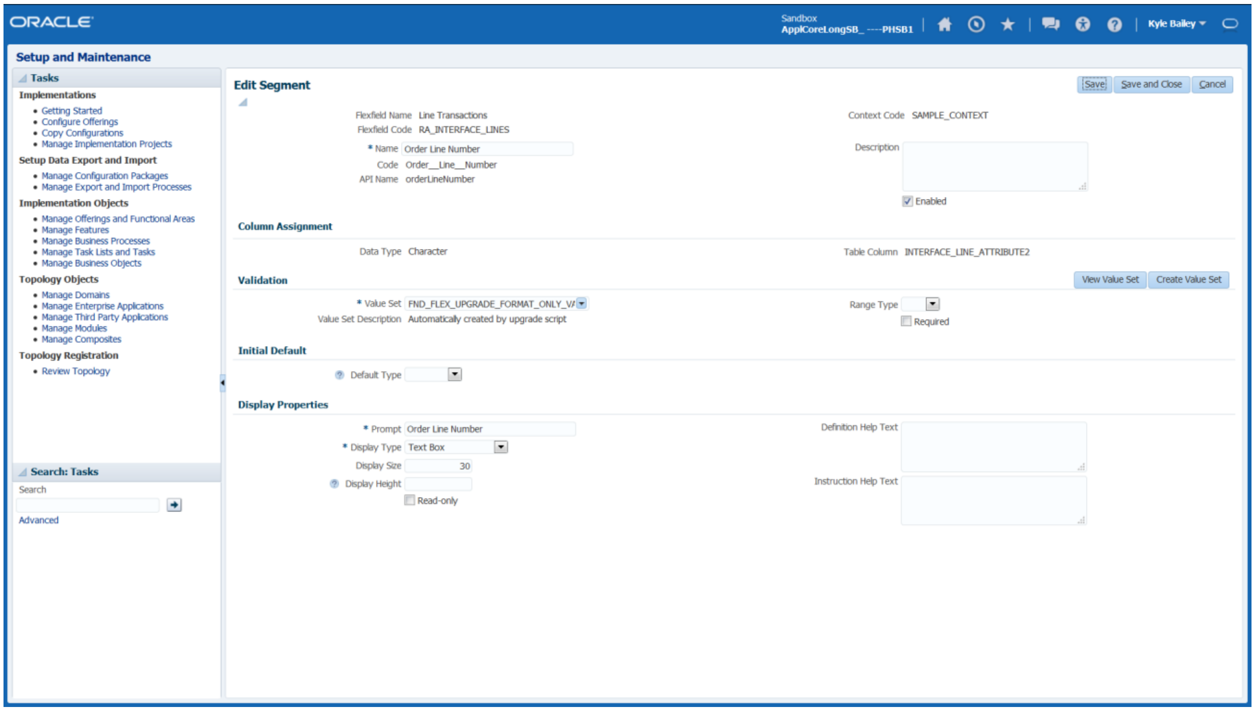
* Value Set: FND_FLEX_UPGRADE_FORMAT_ONLY_V/
Value Set Description: Automatically created by upgrade script
Range Type:
☐ Required

Initial Default

Default Type:

Display Properties

* Prompt: Order Number
* Display Type: Text Box
Display Size: 30
Display Height:
☐ Read-only
Definition Help Text:
Instruction Help Text:



Edit Segment

Flexfield Name: Line Transactions
Flexfield Code: RA_INTERFACE_LINES
* Name: Order Line Number
Code: Order__Line__Number
API Name: orderLineNumber
Context Code: SAMPLE_CONTEXT
Description:
☑ Enabled

Column Assignment

Data Type: Character
Table Column: INTERFACE_LINE_ATTRIBUTE2

Validation

* Value Set: FND_FLEX_UPGRADE_FORMAT_ONLY_V/
Value Set Description: Automatically created by upgrade script
Range Type:
☐ Required

Initial Default

Default Type:

Display Properties

* Prompt: Order Line Number
* Display Type: Text Box
Display Size: 30
Display Height:
☐ Read-only
Definition Help Text:
Instruction Help Text:

Please note down the *API name* which needs to be used in the web service invocation.

Also make sure the Descriptive Flex field is deployed properly after the changes and the deployment status show as SUCCESS.

2.2.3 Security Details

The users and their credentials in the source system should be synchronized with the Oracle Fusion instance so that the user invoking the web service from the source system will get authorized at the Oracle Fusion instance. Once the service is invoked, the XML request payload is submitted to the web service which processes the request and creates an invoice in Oracle Fusion Receivables. The user who has the Billing Specialist or Billing Manager Job Role will be able view the invoice successfully created in Oracle Fusion Receivables.

2.3 Invoking Receivables Invoice Service using Web Service Proxy Client

Step 1: Get the service description from the InvoiceService WSDL URL.
https://<host>:<port>/finArTrxnsInvoices/InvoiceService?WSDL

Step 2: Use WSDL to write a Java class (for example InvoiceServiceClient.java) to invoke
the createSimpleInvoice operation of InvoiceService web service and return the response.
Complete Details of a sample Service Client Java code are found in Section 3.
The Java client class (InvoiceServiceClient.java) should have the following two main methods:

a) constructPayload() : Create the Java method to construct the XML payload based on the functional parameter values. This XML format should follow the standard SOAP (Simple Object Access Protocol) specifications. The XML request payload will be passed while invoking the service. The XML Tag names for the input parameters (Service Data Object attributes) can be obtained from the SDO Section [2.4]. Example of a sample request payload can be found in Section [2.6].

b) createInvoice() : Create the Java method to invoke the service via http connection and get the response. This method will accept the following parameters:

- Service URL
- Username/Password
- Keystore details
- Input Payload

Step 3: Prepare the Keystore information for certificate authentication.

- Launch the WSDL URL in the browser, click on the Lock icon appearing on the starting of the URL and export the certificate to the local system.
- Get the keystore file from the Financial Domain of the Fusion instance.

- Use the key tool command to import the certificate into the keystore
  `keytool -import -file <sslcertfile> -keystore <keystorefile>.`
- Use this keystore file in the InvoiceServiceClient. Java class.

Step 4: Integrate the InvoiceServiceClient.java class to the source system to create the invoice. There are four major areas used in the source system to integrate the client Java class.

- **Set the input parameters**
  The input parameters are the functional parameter values that are required for invoice creation. The values entered here will be formulated as XML payload in constructPayload() method of InvoiceServiceClient.java class. The XML tags for the functional parameters are shown in the SDO attribute names mentioned in SDO Section (2.4).

- **Set the Security information**
  The combination of keystore information (obtained from step 3) and the user credentials are required for authentication of the service call at the Oracle Fusion instance.

- **Call the Client code for service invocation and creation of invoice**
  This steps calls the createInvoice() method of InvoiceServiceClient.java class with the input parameters, host information and the security information.

- **Parse the service response and handling error messages**
  The invocation of the service returns the response upon completion. The response is in a XML format and it can be parsed using any standard Java class (like XPathFactory) to read the values of the tags and analyze them to determine the status of the service call. If the service call reports an error status, then the service error message can be sent to the user to review the cause for the service invocation failure.

Sample code for invocation of the InvoiceServiceClient is available in Appendix 3.2

Step 5: Test the InvoiceService web service.

- **Install Java and set PATH system**
  Intall the Java Development Kit (JDK) and set the system environment path and variables.

Details of the testing are available in Appendix 3.3

2.4 Service Data Objects

The **Receivables Invoice Service** Web service supports the following Service Data Objects (SDO).

- Receivables Invoice Header SDO
- Receivables Invoice Line SDO
- Receivables Transaction Header Flex SDO
- Receivables Transaction Interface Header Flex SDO
- Receivables Transaction Line Flex SDO
- Receivables Transaction Global Flex SDO
- Receivables Invoice Result SDO
- Receivables Lines Interface SDO
- Receivables Distributions Interface SDO
- Receivables Sales Credits Interface SDO
- Receivables Contingencies Interface SDO

Relationships

- Receivables Invoice Header SDO is the parent SDO.
- Receivables Invoice Line SDO is the child of Receivables Invoice Header SDO.
- Receivables Transaction Header Flex SDO is the child of Receivables Invoice Header SDO.
- Receivables Transaction Interface Header Flex SDO is the child of Receivables Invoice Header SDO.
- Receivables Transaction Global Flex SDO is the child of Receivables Invoice Header SDO.
- Receivables Transaction Line Flex SDO is the child of Receivables Invoice Line SDO.
- Receivables Transaction Line Flex SDO is the child of Receivables Lines Interface SDO, Receivables Distributions Interface SDO, Receivables Sales Credits Interface SDO, Receivables Contingencies Interface SDO.

Receivables Invoice Header SDO

This is the Service SDO for the header level of the Receivables Invoice that contains information pertaining to the debit document. The SDO Attributes are:

Attribute Name	Attribute Description	Data Type	Required
BusinessUnit	A unit of an enterprise that performs one or more business functions that can be rolled up in a management hierarchy.	java.lang.String	No. Value will be derived from the Profile Option “Default Business Unit”.

TransactionSource	Attribute that assigns transactions the standard transaction type and determines whether to automatically number transactions.	java.lang.String	No. Value will be derived from Profile Option “Transaction Batch Source”
TransactionType	Invoice control feature that specifies default attributes.	java.lang.String	No. Value will be derived from the transaction source.
TrxNumber	Number that identifies a transaction.	java.lang.String	Yes, if Transaction Source with Manual numbering is specified.
TrxDate	Date when the transaction is created.	java.sql.Date	No. System date will be used.
GlDate	The date, referenced from Oracle Fusion General Ledger, used to determine the accounting period for the invoice.	java.sql.Date	No. Value will be derived from the transaction date. If the transaction date is in a closed period, the first date in the next open period will be used.
BillToCustomerName	Name that identifies a bill-to customer account.	java.lang.String	Yes
BillToCustomerNumber	Number that identifies a bill-to customer account.	java.lang.String	Yes
PaymentTermsName	Specifies the due date and discount date on the invoice.	java.lang.String	No. Value will be derived from Transaction type or from the customer site/account profile setup
InvoiceCurrenyCode	Currency of the Receivables Invoice.	java.lang.String	Yes
ConversionDate	Date used to derive a currency conversion rate.	java.sql.Date	No
ConversionRateType	Classification of currency conversion rates.	java.lang.String	No
ConversionRate	A fixed charge per currency unit used to perform foreign currency conversions.	java.math.BigDecimal	No

Receivables Invoice Line SDO

This is the Service SDO for the line level of the Receivables Invoice that contains detail information pertaining to the debit document. The SDO Attributes are:

Attribute Name	Attribute Description	Data Type	Required
LineNumber	Number that identifies a transaction line.	java.lang.Integer	Yes
ItemNumber	An inventory item in a transaction line item.	java.lang.String	No
MemoLineName	Line assigned to a transaction when the item is not an inventory item.	java.lang.String	No

Description	Description of a good or service on a transaction line.	java.lang.String	Yes
Quantity	The number of products or services referred to on a single transaction line.	java.math.BigDecimal	Yes
UnitSellingPrice	Price of an individual item on a transaction line.	java.math.BigDecimal	Yes
TaxClassificationCode	Tax classification code for each transaction line.	java.lang.String	No. Value will be derived from Tax Application System Options or product level (item or memo line) or customer level

Receivables Transaction Header Flex SDO

This is the Service SDO for the transaction flexfield of the Receivables transaction document. The SDO Attributes are:

Attribute Name	**Attribute Description**	Data Type	**Required**
_Flex_Context	Context for the Receivables Transaction descriptive flexfield.	java.lang.String	No. This value is entered by the user if the user chooses to use a flexfield.
_Num_Of_Segments	Number of Segments for the Receivables Transaction descriptive flexfield.	java.lang.Integer	No. This value is entered by the user if the user chooses to use a flexfield.

Receivables Transaction Interface Header Flex SDO

This is the Service SDO for the interface flexfield of the Receivables transaction document. The SDO Attributes are:

Attribute Name	**Attribute Description**	**Data Type**	**Required**
_Flex_Context	Context for the Receivables Transaction Interface descriptive flexfield.	java.lang.String	No. This value is entered by the user if the user chooses to use a flexfield.
_Num_Of_Segments	Number of Segments for the Receivables Transaction Interface descriptive flexfield.	java.lang.Integer	No. This value is entered by the

			user if the user chooses to use a flexfield.

Receivables Transaction Global Flexfield SDO

This is the Service SDO for the global flexfield of the Receivables transaction document. The SDO Attributes are:

Attribute Name	Attribute Description	Data Type	Required
_Flex_Context	Context for the Receivables Transaction Global descriptive flexfield.	java.lang.String	No. This value is entered by the user if the user chooses to use a flexfield.
_Num_Of_Segments	Number of Segments for the Receivables Transaction Global descriptive flexfield.	java.lang.Integer	No. This value is entered by the user if the user chooses to use a flexfield.

Receivables Transaction Line Flexfield SDO

This is the Service SDO for the transaction line flexfield of the Receivables transaction document. The SDO Attributes are:

Attribute Name	Attribute Description	Data Type	Required
_Flex_Context	Context for the Receivables Transaction Line descriptive flexfield.	java.lang.String	No. This value is entered by the user if the user chooses to use a flexfield.
_Num_Of_Segments	Number of Segments for the Receivables Transaction Line descriptive flexfield.	java.lang.Integer	No. This value is entered by the user if the user chooses to use a flexfield.

Receivables Invoice Result SDO

This is the Service SDO for the results returned by the invoice service. The SDO Attributes are:

AttributeName	Attribute Description	User Data Type
ServiceStatus	Status of the Receivables Invoice Service invocation.	String
TransactionNumber	Transaction Number that is created by the Receivables Invoice Service.	String
CustomerTrxId	Unique Identifier of the transaction.	Number

Receivables Lines Interface SDO

This is the Service SDO that contains the Attributes level of the Receivables Lines Interface that contains information pertaining to the debit document. The SDO Attributes are:

Attribute Name	Attribute Description	User Data Type
OrgId	A unit of an enterprise that performs one or more business functions that can be rolled up in a management hierarchy.	java.lang.Long
AccountingRuleDuration	Revenue scheduling rule duration.	java.lang.Long
AccountingRuleId	ID for revenue scheduling rule.	java.lang.Long
AccountingRuleName	Name of revenue scheduling rule.	java.lang.String
AddressVerificationCode	Credit card address verification code from Oracle Fusion Payment Server.	java.lang.String
Amount	Amount of this transaction line.	java.math.BigDecimal
AmountIncludesTax	Code that indicates whether the line amount is inclusive of tax. NULL: Use the value on the tax rate code.	java.lang.String
ApplicationId	Source application importing transactions into Receivables.	java.lang.Long
AssessableValue	The deemed price at which a product is valued, by a tax authority for a given tax, for tax calculation purposes.	java.math.BigDecimal
ApprovalCode	Payment approval code from the credit card issuer.	java.lang.String
BatchSourceName	Indicates the transaction source associated with the row.	java.lang.String
CustomerTrxTypeName	Indicates the transaction type associated with the row.	java.lang.String
BillContactPartyNumber	Value used to uniquely identify the Bill-to contact party for the transaction.	java.lang.String
BillCustomerAccountNumber	Value used to uniquely identify the Bill-to customer account number of the transaction.	java.lang.String
BillCustomerSiteNumber	Value used to uniquely identify the Bill-to customer site for the transaction.	java.lang.String

BillingDate	Billing date to use for invoice creation. Provide a value in the format: YYYY/MM/DD.	java.sql.Date
Comments	Comments for the transaction line.	java.lang.String
ConsBillingNumber	Used for custom billing and print formatting. Not used for balance forward bills.	java.lang.String
ContractId	Contract Identifier	java.lang.Long
ContractLineId	Contract Line Identifier	java.lang.Long
SalesOrder	Sales order number.	java.lang.String
SalesOrderDate	Sales order date/RMA Date. Provide a value in the format: YYYY-MM-DD.	java.sql.Date
SalesOrderLine	Sales order line number for this transaction.	java.lang.String
SalesOrderRevision	Sales order revision number.	java.math.BigDecimal
SalesOrderSource	Source of Sales Order.	java.lang.String
TrxDate	Identifies the transaction date. Provide a value in the format: YYYY/MM/DD.	java.sql.Date
CurrencyCode	Currency code of the currency on the transaction line. Use the three character ISO currency code. Example: US Dollars is USD.	java.lang.String
ConversionType	Conversion type for the currency on the transaction line.	java.lang.String
ConversionDate	Date used to calculate the currency conversion rate, if the currency on the transaction line is not the ledger currency. Provide a value in the format: YYYY-MM-DD.	java.sql.Date
ConversionRate	Conversion rate for the currency on the transaction line, if it is not the ledger currency.	java.math.BigDecimal
CreditMethodForAccountRule	The credit method for crediting a transaction that uses a revenue scheduling rule. Valid values: PRORATE, UNIT, LIFO. Last In first out (LIFO) backs out revenue starting with the last general ledger period and reverses all prior periods until it has used up the credit memo. Prorate credits an equal percentage to all account assignments for this invoice. Unit reverses the revenue for the number of units you specify from an original line of the invoice. Do not use if LineType is TAX or CHARGES, or for header freight charges.	java.lang.String
CreditMethodForInstallments	The credit method for crediting a transaction that uses split payment terms. Valid values: PRORATE, LIFO, or FIFO or NULL. First in first out (FIFO) credits the first installment first. Last in first out (LIFO) credits the last installment first. Prorate credits the installments of the credited transaction and prorates them based on the amount remaining for each installment. Do not	java.lang.String

	use if LineType is TAX or CHARGES, or when passing header freight charges.	
CustomerTrxTypeSequenceId	Transaction type identifier.	java.lang.Long
CustomerBankAccountName	Name of the bill-to customer bank account.	java.lang.String
DefaultTaxationCountry	Default taxation country for tax reporting purposes.	java.lang.String
DeferralExclusion	If the value is Y, AutoInvoice excludes the transaction line from automated revenue deferral.	java.lang.String
Description	Description of the transaction line.	java.lang.String
DocumentNumber	Document number of the transaction.	java.lang.Long
DocumentSubType	In certain countries, a tax or governmental authority defines and classifies document types for reporting purposes.	java.lang.String
ExceptionId	Tax exception identifier.	java.lang.Long
ExemptionId	Tax exemption identifier.	java.lang.Long
FinalDischargeLocationCode	Final destination location code or final customer location where shipment will be sent	java.lang.String
FinalDischargeLocationId	Final destination location ID or final customer location where shipment will be sent	java.lang.Long
FirstPtyRegId	First party tax registration ID of the transaction. Derived from the Legal Entity of the transaction.	java.lang.Long
FirstPtyRegNumber	First party tax registration number of the transaction. Derived from the Legal Entity of the transaction.	java.lang.String
FOBPoint	Free on board point, which is the location at which the ownership title of the goods is transferred from the seller to the buyer.	java.lang.String
GlDate	Identifies the accounting date of the transaction. Must be in an open or future enterable accounting period.	java.sql.Date
IntendedUseClassifId	Intended use identifier of the transaction line. It identifies the purpose for which the product is used.	java.lang.Long
InternalNotes	Additional comments.	java.lang.String
ItemNumber	Inventory item number of the line item.	java.lang.String
InventoryItemId	Inventory item ID for this transaction.	java.lang.Long
InvoicedLineAcctgLevel	Identifies the accounting level for lines eligible for invoice creation in the original system.	java.lang.String
InvoicingRuleId	ID for Invoicing rule for this transaction.	java.lang.Long
InvoicingRuleName	Invoicing rule name for this transaction. Valid values are 'Advance Invoice' and 'Arrears Invoice'. This column is optional. For invoice lines with rules, you must enter a value. Use either InvoicingRuleId or InvoicingRuleName, but not both.	java.lang.String
LastPeriodToCredit	Last period number for crediting. Credit	java.lang.Integer

	memo use only. Must be between 0 and the invoice's revenue scheduling rule duration (inclusive). Positive integers only.	
LastTrxDebitAuth	Specifies whether a transaction is the last one of a recurrent series of ISO20022 direct debit collections for a given debit authorization. Valid values are Y and N.	java.lang.String
LegalEntityId	Identifies the legal entity responsible for issuing the transaction line.	java.lang.Long
LineIntendedUse	Code used to classify items when the intended use of the item is a factor in tax determination.	java.lang.String
LineType	Enter 'LINE', 'TAX', 'FREIGHT' or 'CHARGES' to specify the line type for this transaction. (CHARGES refers to finance charges.) You must enter a value in this column. For credit memos enter the type of line you are crediting.	java.lang.String
MemoLineName	Enter the name of the standard memo line for this transaction.	java.lang.String
MemoLineSequenceId	Enter the standard memo line Sequence ID for this transaction.	java.lang.Long
OrigSystemShipPartyId	ID value used to uniquely identify the ship-to customer from the feeder system.	java.lang.Long
OrigSystemShipPartyReference	Reference value used to uniquely identify the ship-to customer from the feeder system.	java.lang.String
OrigSystemShipPartySiteId	ID value used to uniquely identify the customer ship-to address from the feeder system.	java.lang.Long
OrigSystemShipPartySiteReference	Reference value used to uniquely identify the customer ship-to address from the feeder system.	java.lang.String
OrigSystemShipPtyContactId	ID value used to uniquely identify the customer ship-to contact from the feeder system.	java.lang.Long
OrigSystemShipPtyContactReference	Reference value used to uniquely identify the customer ship-to contact from the feeder system.	java.lang.String
OrigSystemSoldPartyId	ID value used to Identify the sold-to customer from the feeder system.	java.lang.Long
OrigSystemSoldPartyReference	Reference value used to Identify the sold-to customer from the feeder system.	java.lang.String
OrigSystemBatchName	Original batch identifier from the source system.	java.lang.String
OrigSystemBillAddressId	ID value used to uniquely identify the bill-to customer address from the feeder system.	java.lang.Long
OrigSystemBillAddressReference	Reference value used to uniquely identify the bill-to customer address from the feeder system.	java.lang.String
OrigSystemBillContactId	ID value used to uniquely identify the bill-to customer contact from the feeder system.	java.lang.Long
OrigSystemBillContactReference	Reference value used to uniquely identify the bill-to customer contact from the feeder system.	java.lang.String

OrigSystemBillCustomerId	ID value used to uniquely identify the bill-to customer from the feeder system.	java.lang.Long
OrigSystemBillCustomerReference	Reference value used to uniquely identify the bill-to customer from the feeder system.	java.lang.String
OrigSystemShipAddressId	ID value used to uniquely identify the customer account ship-to address from the feeder system.	java.lang.Long
OrigSystemShipAddressReference	Reference value used to uniquely identify the customer account ship-to address from the feeder system.	java.lang.String
OrigSystemShipContactId	ID value used to uniquely identify the customer account ship-to contact from the feeder system.	java.lang.Long
OrigSystemShipContactReference	Reference value used to uniquely identify the customer account ship-to contact from the feeder system.	java.lang.String
OrigSystemShipCustomerId	ID value used to uniquely identify the ship-to customer account from the feeder system.	java.lang.Long
OrigSystemShipCustomerReference	Reference value used to uniquely identify the ship-to customer account from the feeder system.	java.lang.String
OrigSystemSoldCustomerId	ID value used to Identify the sold-to customer account from the feeder system.	java.lang.Long
OrigSystemSoldCustomerReference	Reference value used to Identify the sold-to customer account from the feeder system.	java.lang.String
OverrideAutoAccounting	If the value is Y, AutoInvoice uses the deferred revenue account provided through the interface table for revenue accounting. If the value is N, AutoInvoice uses the accounts generated by AutoAccounting.	java.lang.String
PaymentAttributes	Payment attributes used for grouping transaction lines.	java.lang.String
PaymentServerOrderNumber	Number that indicates whether the credit card payment was authorized by Oracle Fusion Payment Server.	java.lang.String
PaymentSetId	Identifies the prepayment application payment set ID.	java.lang.Long
PaymentTrxnExtensionId	Oracle Fusion Payments transaction identifier for processing a credit card or bank account payment.	java.lang.Long
PrimarySalesrepNumber	Number that identifies the primary salesperson.	java.lang.String
PrintingOption	Indicates whether the transaction can be printed.	java.lang.String
ProdFcCategId	Product fiscal classification identifier of the inventory item.	java.lang.Long
ProductCategory	Code used to classify non-inventory items and other special product classification needs for tax determination or tax reporting purposes.	java.lang.String
ProductFiscClassification	Code used to classify products that have a tax requirement for tax determination or tax reporting purposes.	java.lang.String

ProductType	Identifies a transaction line item for tax purposes. Valid values are Goods and Services.	java.lang.String
PurchaseOrder	Purchase order number for this transaction.	java.lang.String
PurchaseOrderDate	Date of purchase order.	java.sql.Date
PurchaseOrderRevision	Purchase order revision number.	java.lang.String
Quantity	Either the number of units shipped on this transaction line, or the number of units on this credit memo line.	java.math.BigDecimal
QuantityOrdered	Original number of units customer sales ordered for this transaction.	java.math.BigDecimal
ReasonCode	Enter the reason code for this transaction.	java.lang.String
ReasonCodeMeaning	Credit memo reason code meaning.	java.lang.String
ReceiptMethodId	ID of the receipt method for the transaction.	java.lang.Long
ReceiptMethodName	Name of the receipt method for the transaction.	java.lang.String
ReferenceLineId	Reference line identifier.	java.lang.Long
RelatedBatchSourceName	Enter the name of the batch source of the document to which this transaction is related.	java.lang.String
RelatedTrxNumber	Enter the document number to which this transaction is related.	java.lang.String
ResetTrxDate	Indicates whether AutoInvoice resets the transaction date to the accounting date when the transaction date is not passed.	java.lang.String
ResourceSalesrepId	Enter the primary salesperson ID for this transaction.	java.lang.Long
ContractStartDate	The start date of the contract, if the transaction line is associated with a contract. For recurring invoices, this is the billing period start date on the first invoice.	java.sql.Date
ContractEndDate	The end date of the contract, if the transaction line is associated with a contract.	java.sql.Date
RuleEndDate	Date that you want to end the revenue scheduling rule for this transaction.	java.sql.Date
RuleStartDate	Date that you want to start the revenue scheduling rule for this transaction.	java.sql.Date
SalesTaxId	Sales tax identifier.	java.lang.Long
SetOfBooksId	Ledger identifier.	java.lang.Long
ShipContactPartyNumber	Value used to uniquely identify the Ship-to contact party for the transaction.	java.lang.String
ShipCustomerAccountNumber	Value used to uniquely identify the Ship-to customer account for the transaction.	java.lang.String
ShipCustomerSiteNumber	Value used to uniquely identify the Ship-to customer site for the transaction.	java.lang.String
ShipDateActual	Shipment date.	java.sql.Date
ShipVia	Code that identifies the shipping method.	java.lang.String
SoldCustomerAccountNumber	Sold-to customer account for the transaction.	java.lang.String
SourceApplicationId	Source application identifier. Used for	java.lang.Long

	tax purposes.	
SourceEntityCode	Source entity code. Used for tax purposes.	java.lang.String
SourceEventClassCode	Source event class code. Used for tax purposes.	java.lang.String
SourceTrxDetailTaxLineId	Source transaction detail tax line identifier. Used for tax purposes.	java.lang.Long
SourceTrxId	Source transaction identifier. Used for tax purposes.	java.lang.Long
SourceTrxLineId	Source transaction line identifier. Used for tax purposes.	java.lang.Long
SourceTrxLineType	Source transaction line type. Used for tax purposes.	java.lang.String
Tax	Code that represents a charge imposed by a fiscal or tax authority within a tax regime.	java.lang.String
TaxCode	Tax classification code assigned to the transaction line. Mandatory if LineType is TAX. Do not use if LineType is CHARGES or FREIGHT.	java.lang.String
TaxExempt	Indicates the applicable tax exemption value for the transaction line.	java.lang.String
TaxExemptNumber	Number of the tax exemption certificate for line items that have the TaxExemptFlag set to E (Exempt). Use if LineType is LINE and TaxExemptFlag is E. Otherwise, leave blank.	java.lang.String
TaxExemptReasonCode	Code that provides the reason for a tax exemption for a line item.	java.lang.String
TaxExemptReasonCodeMeaning	Text description of the code entered in the TaxExemptReasonCode column.	java.lang.String
TaxInvoiceDate	Date on the fiscal document generated during the shipment of goods.	java.sql.Date
TaxInvoiceNumber	Number on a fiscal document generated during the shipment of goods.	java.lang.String
TaxJurisdictionCode	Code that represents a geographic area where taxes are levied. Defined within the scope of a tax and tax regime.	java.lang.String
TaxPrecedence	Number that indicates the order in which a tax is calculated in the compounding process, when more than one tax applies to an invoice line. Taxes are calculated in ascending order of compounding precedence.	java.math.BigDecimal
TaxRate	Tax rate for this tax line. If LineType is TAX you must enter a value for either this or the AMOUNT column. Do not use if LineType is LINE, CHARGES, or FREIGHT.	java.math.BigDecimal
TaxRateCode	Code that represents the numeric value for calculating the amount of a certain tax. Defined within the scope of a tax status, tax, and tax regime.	java.lang.String
TaxRegimeCode	Code that represents a high-level entity for defining a common set of tax rules for one or more taxes with the same tax requirement.	java.lang.String
TaxStatusCode	Code that represents the taxable nature	java.lang.String

	of a product in the context of a transaction and a specific tax on the transaction. Defined within the scope of a tax and tax regime.	
TaxableAmount	Calculated taxable basis amount. The tax amount is derived by applying the tax rate to the taxable amount.	java.math.BigDecimal
Taxable	Determines whether the current line is taxable. Valid values are Y, N, and NULL.	java.lang.String
PaymentTermsId	Indicates the ID of the payment terms associated with the row.	java.lang.Long
PaymentTermsName	Indicates the name of the payment terms associated with the row.	java.lang.String
ThirdPtyRegId	Third party tax registration ID of the transaction. Derived from the Bill-to Customer of the Transaction.	java.lang.Long
ThirdPtyRegNumber	Third party tax registration number of the transaction. Derived from the Bill-to Customer of the Transaction.	java.lang.String
TranslatedDescription	Translated description of the transaction line.	java.lang.String
TrxBusinessCategory	Code used to classify business transactions for tax purposes.	java.lang.String
TrxNumber	Identifier of the transaction, which is either system generated or provided by the user.	java.lang.String
UnitSellingPrice	Selling price per unit.	java.math.BigDecimal
UnitStandardPrice	Sales item standard price per unit in inventory.	java.math.BigDecimal
UOMCode	Unit of measure code.	java.lang.String
UOMName	Unit of measure name.	java.lang.String
UserDefinedFiscClass	Code used to classify any tax requirements that you cannot define using the existing fiscal classification types.	java.lang.String
VATTaxId	VAT Tax Identifier	java.lang.Long
WarehouseCode	Inventory organization code indicating ship-from location	java.lang.String
WarehouseId	Inventory organization ID indicating ship-from location	java.lang.Long
WaybillNumber	Waybill number.	java.lang.String
RecurringBill	Option that indicates whether the transaction line is for a recurring invoice. Valid values are Y, N, null. Null is the same as N.	java.lang.String
Periodicity	Period of the recurring invoice. Valid values are DAY, WEEK, MONTH, QUARTER, YEAR. A value is required if RecurringBill is Y.	java.lang.String
SecondInvoiceDate	Transaction date of the second invoice for recurring invoices, if the first invoice was for a partial period.	java.sql.Date
ContractedPeriods	Number of recurring invoices to issue.	java.lang.Integer
FirstOverridePeriod	Number indicating the period that has an override amount. For example, the first invoice is for period 1, the second invoice	java.lang.Long

	is for period 2, and so on.	
FirstOverrideAmount	The override amount for the period captured in FirstOverridePeriod.	java.math.BigDecimal
SecondOverridePeriod	Number indicating the period that has an override amount. For example, the first invoice is for period 1, the second invoice is for period 2, and so on.	java.lang.Long
SecondOverrideAmount	The override amount for the period captured in FifthOverridePeriod.	java.math.BigDecimal
ThirdOverridePeriod	Number indicating the period that has an override amount. For example, the first invoice is for period 1, the second invoice is for period 2, and so on.	java.lang.Long
ThirdOverrideAmount	The override amount for the period captured in FifthOverridePeriod.	java.math.BigDecimal
FourthOverridePeriod	Number indicating the period that has an override amount. For example, the first invoice is for period 1, the second invoice is for period 2, and so on.	java.lang.Long
FourthOverrideAmount	The override amount for the period captured in FifthOverridePeriod.	java.math.BigDecimal
FifthOverridePeriod	Number indicating the period that has an override amount. For example, the first invoice is for period 1, the second invoice is for period 2, and so on.	java.lang.Long
FifthOverrideAmount	The override amount for the period captured in FifthOverridePeriod.	java.math.BigDecimal
SecondBillingPeriodStartDate	For recurring invoices, the start date of the billing period on the second invoice, if the first invoice did not cover a full billing period.	java.sql.Date
TransactionInterfaceGdf	Receivables Transaction Region Information global descriptive flexfield belonging to the lines interface table.	commonj.sdo.DataObject
TransactionLineInterfaceGdf	Receivables Transaction Line Region Information global descriptive flexfield belonging to the lines interface table.	commonj.sdo.DataObject
TransactionInterfaceLineDff	Line Transactions descriptive flexfield belonging to the lines interface table.	commonj.sdo.DataObject
TransactionInterfaceLinkToDff	Link-to Transactions descriptive flexfield belonging to the lines interface table.	commonj.sdo.DataObject
TransactionInterfaceReferenceDff	Reference Transactions descriptive flexfield belonging to the lines interface table.	commonj.sdo.DataObject
TransactionLineDff	Invoice Lines descriptive flexfield belonging to the lines interface table.	commonj.sdo.DataObject

Receivables Distributions Interface SDO

This is the Service SDO for the Distributions interface table that stores transaction distribution records. The SDO Attributes are:

Attribute Name	Attribute Description	Data Type
AccountClass	Account class for this accounting distribution.	java.lang.String
AcctdAmount	This column is optional. If you enter 'AMOUNT' for your batch source option 'Revenue Account Allocation', then AutoInvoice will accept whatever is passed in this column without validation. If this column is null, then AutoInvoice will compute the accounted amount for this distribution line. For imported amounts in the ledger currency, AutoInvoice will reject the line if you enter a value in the ACCTD_AMOUNT column that does not equal the line amount.	java.math.BigDecimal
Amount	The amount for this accounting distribution. You may provide a value for this attribute or Percent based on the transaction source option	java.math.BigDecimal
CodeCombinationId	Enter the code combination ID of the Accounting Flexfield for this accounting distribution.	java.lang.Long
InterimTaxCcid	Account Code Combination identifier for tax account to be used for deferred tax amounts.	java.lang.Long
OrgId	Indicates the identifier of the business unit associated with the row.	java.lang.Long
Percent	The percent for this accounting distribution.	java.math.BigDecimal
DistributionInterfacLineDff	Line Transactions descriptive flexfield belonging to the distributions interface table.	commonj.sdo.DataObject
TransactionDistributionDff	Interface Distributions descriptive flexfield belonging to the distributions interface table.	commonj.sdo.DataObject

Receivables Sales Credits Interface SDO

This is the Service SDO for the Sales credits interface table that stores sales credit records**.** The SDO Attributes are

Attribute Name	Attribute Description	Data Type
OrgId	Indicates the identifier of the business unit associated with the row.	java.lang.Long
ResourceSalesrepId	Enter the primary salesperson ID for this transaction.	java.lang.Long
SalesCreditAmountSplit	Sales credit amount for this salesperson. You may provide a value for this attribute or Sales	java.math.BigDecimal

	Credit Percentage Split based on the transaction source option.	
SalesCreditPercentSplit	The sales credit percent assignment for this salesperson.	java.math.BigDecimal
SalesCreditTypeId	ID of the sales credit type for this sales credit assignment. Depending on the value you entered for your Transaction source you must enter either a value in this column or in SalesCreditTypeName. If you specify the sales credit type name in your transaction source, AutoInvoice defaults a value in this column.	java.lang.Long
SalesCreditTypeName	The name of the sales credit type.	java.lang.String
SalesgroupId	Enter the sales group ID for this sales credit assignment. This column is optional.	java.lang.Long
SalesrepNumber	The salesperson number for this sales credit assignment.	java.lang.String
SalesCreditInterfaceLineDff	Line Transactions descriptive flexfield belonging to the sales credits interface table.	commonj.sdo.DataObject
TransactionSalesCreditDff	Interface Sales Credits descriptive flexfield belonging to the sales credits interface table.	commonj.sdo.DataObject

Receivables Contingencies Interface SDO

This is the Service SDO for the Contingencies interface table that stores contingency records. The SDO Attributes are

Attribute Name	Attribute Description	User Data Type
ContingencyCode	Contingency Code. The values should be from the Lookup Code of AR_REVENUE_CONTINGENCIES.	java.lang.String
ExpirationDate	Contingency Expiration Date.	java.sql.Date
ExpirationDays	Contingency Expiration in Days.	java.math.BigDecimal
ExpirationEventDate	Expiration event date.	java.sql.Date
ContingencyId	Contingency identifier.	java.lang.Long
OrgId	Indicates the identifier of the business unit associated with the row.	java.lang.Long
Completed	Completed flag indicates if contingency has been completed.	java.lang.String
ContingencyInterfaceLineDff	Line Transactions descriptive flexfield belonging to the contingencies interface table.	commonj.sdo.DataObject

2.5 Error Handling

When a Web service request is being processed and an error is encountered, the nature of the error will be communicated to the client. The SOAP specification defines a standard, platform-independent way of describing the error within the SOAP message using a SOAP fault.

SOAP faults can be one of the following types:

- **Modeled**
  This refers to an exception that is thrown explicitly from the business logic and mapped to wsdl:fault definitions in the WSDL file.
- **Unmodeled**
  This refers to an exception that is generated at run-time when no business logic fault is defined in the WSDL. In this case, Java exceptions are represented as generic SOAP fault exceptions, javax.xml.ws.soap.SOAPFaultException.

The faults are returned to the sender only in case of synchronous mode. If a Web service invocation is asynchronous, the SOAP fault is not returned to the sender but stored for further processing. For more information about exception handling refer to the Exception Handling SOAP documentation.

The table below shows the error message names and accompanying text that describes each error.

Receivables Invoice Service Error Messages Table

Message Name	Message Text
AR_INVAL_BTH_SRC_SEQ_ID	The transaction source that you selected is invalid.
AR_TW_BAD_DATE_SOURCE	Invalid transaction source for this transaction date.
AR_INVAL_CUST_TRX_TYPE_ID	The transaction type that you selected is invalid.
AR_TW_BAD_DATE_TRX_TYPE	Invalid transaction type for this transaction date.
AR_INVAL_BILL_TO_CUST_ID	The bill-to customer that you selected is invalid.
AR_INVAL_BILL_TO_SITE_USE_ID	The bill-to customer site that you selected is invalid.
AR_INVAL_TERM_ID	The payment terms that you selected are invalid.

AR_NO_ROW_IN_SYSTEM_PARAMETERS	The business unit defined in the Business Unit profile option is invalid.
AR_TW_BAD_DATE_TERM	Invalid payment terms for this transaction date.
AR_INVAL_EXCHG_RATE_TYPE	You entered an invalid conversion type on the transaction.
AR_EXCHG_RATE_NOT_NULL	Enter a conversion rate for the customer profile.
AR_INVAL_INV_CURR_CODE	The currency that you selected is invalid.
AR_TW_BAD_DATE_CURRENCY	Invalid currency for this transaction date.
AR_EXCHANGE_RATE_NEGATIVE	Enter a positive conversion rate.
AR_DAPI_REMIT_ADDRESS_DFT_ERR	The default remit-to address was not populated. Enter a valid remit-to address.
AR_TRX_NUMBER_REQUIRED	Enter the transaction number.
AR_TW_INVALID_TRX_NUMBER	This transaction number already exists. Enter a unique transaction number.
AR_INVALID_ACCOUNTING_DATE	The accounting date {GL_DATE} is not in an open or future-enterable period.
AR_INVALID_INVENTORY_ITEM	You entered an invalid item.
AR_INVALID_MEMO_LINE	The memo line for line {LINE_NUMBER} with this transaction date is invalid.
AR_INAPI_LINE_NUM_NOT_NULL	The transaction line must have a line number.
AR_TRX_ENTER_VALID_LINE_INFO	Enter the required line information for line number {LINENUMBER}.
AR_RAXTRX-1665A	Each line must have a unique combination of INTERFACE_LINE_ATTRIBUTE1-15 and INTERFACE_LINE_CONTEXT values for the line transaction flexfield.
AR_INVALID_UOM_CODE	The unit of measure that you selected is invalid.
AR_BILL_CUST_ACC_NUMBER	The bill-to customer account number {BILL_CUSTOMER_ACCOUNT_NUMBER} is invalid.
AR_BILL_CUST_SITE_NUMBER	The bill-to customer site number {BILL_CUSTOMER_SITE_NUMBER} is invalid.
AR_INVALID_REASON	The credit reason that you selected is invalid.
AR_CONTRACT_END_DATE_INV_ERR	The contract end date must be after the contract start date.

3 Appendix

3.1 Sample of the Service Client Java code.

```
InvoiceServiceClient.Java code:

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.io.OutputStream;
import java.io.OutputStreamWriter;
import java.net.HttpURLConnection;
import java.net.URL;
import java.util.HashMap;

public class InvoiceServiceClient {
    public InvoiceServiceClient() {
        super();
    }

    private static String serviceURL;
    private static String inputPayload;
    private static String username;
    private static String password;
    private static String userToken;
    private static String outputPayload;


    public static String httpPost(String destUrl, String postData,
                                  String authStr, String keyStoreLocation,
                                  String keyStorePassword) throws Exception {

        System.out.println();
        System.out.println("Invoking the Service");

      // Setting the KeyStore Properties

        System.setProperty("javax.net.ssl.trustStore", keyStoreLocation);
        System.setProperty("javax.net.ssl.trustStorePassword",
                           keyStorePassword);

      // Open the HTTP connection and set the connection properties

        URL url = new URL(destUrl);
        HttpURLConnection conn = (HttpURLConnection)url.openConnection();
        if (conn == null) {
            return null;
        }

        conn.setRequestProperty("Content-Type", "text/xml;charset=UTF-8");
        conn.setDoOutput(true);
        conn.setDoInput(true);
        conn.setUseCaches(false);
        conn.setFollowRedirects(true);
        conn.setAllowUserInteraction(false);
        conn.setRequestMethod("POST");
```

```
   // Set the Authorization property for the HTTP connection using the
   username and password

    byte[] authBytes = authStr.getBytes("UTF-8");
    String auth = Base64.byteArrayToBase64(authBytes);
    conn.setRequestProperty("Authorization", "Basic " + auth);

   // Post the http request. This will invoke the Invoice Web Service for
   creating the invoice

    OutputStream out = conn.getOutputStream();
    OutputStreamWriter writer = new OutputStreamWriter(out, "UTF-8");
    writer.write(postData);
    writer.close();
    out.close();

    try {
        InputStream errIs = conn.getErrorStream();
        if (errIs != null) {
            String err = getString(errIs);
            if (err != null && !err.isEmpty()) {
                System.out.println(err);
            }
            errIs.close();
        }
    } catch (Exception e) {
        e.printStackTrace();
    }



   // Read the response and return it to the calling Java API

    String response = null;
    try {
        InputStream in = conn.getInputStream();
        if (in != null) {
            response = getString(in);
            in.close();
        }
    } catch (Exception e) {
        e.printStackTrace();
    }

    conn.disconnect();
    return response;
}

public static String getString(InputStream errIs) {

    BufferedReader br = null;
    StringBuilder sb = new StringBuilder();

    String line;
    try {

        br = new BufferedReader(new InputStreamReader(errIs));
        while ((line = br.readLine()) != null) {
            sb.append(line);
        }

    } catch (IOException e) {
        e.printStackTrace();
    } finally {
        if (br != null) {
            try {
                br.close();
            } catch (IOException e) {
                e.printStackTrace();
            }
        }
    }
    return sb.toString();

}

public void setInputPayload(String inputPayload) {
    this.inputPayload = inputPayload;
}

public String getInputPayload() {
    return this.inputPayload;
}

public void setWebService(String webService) {
    this.serviceURL = webService;
}

public static String getWebService() {
    return serviceURL;
}

public void setUsername(String username) {
    this.username = username;
}

public static String getUsername() {
    return username;
}

public void setPassword(String password) {
    this.password = password;
}

public static String getPassword() {
    return password;
}

public void setUserToken(String userToken) {
    this.userToken = userToken;
}

public String getUserToken() {
    return userToken;
}

public void setOutputPayload(String outputPayload) {
    this.outputPayload = outputPayload;
}

public String getOutputPayload() {
    return outputPayload;
}
```

```
public String constructPayload(HashMap invoiceHeader,
                               HashMap invoiceLines) {
    String payload =
                     "<soap:Envelope
         xmlns:soap=\"http://schemas.xmlsoap.org/soap/envelope/\">" +
        "<soap:Body>" +
                     "<ns1:createSimpleInvoiceAsync
         xmlns:ns1=\"http://xmlns.oracle.com/apps/financials/receivables/tr
         ansactions/invoices/invoiceService/types/\">" +
        "<ns1:invoiceHeaderInformation
        xmlns:ns2=\"http://xmlns.oracle.com/apps/financials/receivables/tran
        sactions/invoices/invoiceService/\">" +
        "<ns2:InvoiceCurrencyCode>" +
        invoiceHeader.get("InvoiceCurrencyCode") +
        "</ns2:InvoiceCurrencyCode>" + "<ns2:BillToCustomerName>" +
        invoiceHeader.get("BillToCustomerName") +
        "</ns2:BillToCustomerName>" + "<ns2:BillToAccountNumber>" +
        invoiceHeader.get("BillToAccountNumber") +
        "</ns2:BillToAccountNumber>" + "<ns2:InvoiceLine>" +
        "<ns2:LineNumber>" + invoiceLines.get("LineNumber") +
        "</ns2:LineNumber>" + "<ns2:Description>" +
        invoiceLines.get("Description") + "</ns2:Description>" +
        "<ns2:Quantity>" + invoiceLines.get("Quantity") +
        "</ns2:Quantity>" + "<ns2:UnitSellingPrice>" +
        invoiceLines.get("UnitSellingPrice") + "</ns2:UnitSellingPrice>" +
        "</ns2:InvoiceLine>" + "</ns1:invoiceHeaderInformation>" +
        "</ns1:createSimpleInvoiceAsync>" + "</soap:Body>" +
        "</soap:Envelope>";
    return payload;
}

public String createInvoice(String hostName, int port, String username,
                            String password, HashMap invoiceHeader,
                            HashMap invoiceLines, String keyStoreLocation,
                            String keyStorePassword) throws Exception {

    if (port < 0)
        this.setWebService("https://" + hostName +
                           "/finArTrxnsInvoices/InvoiceService");
    else
        this.setWebService("https://" + hostName + ":" + port +
                           "/finArTrxnsInvoices/InvoiceService");
    this.setUsername(username);
    this.setPassword(password);

   // Construct the XML input payload

    String reqPayload = this.constructPayload(invoiceHeader, invoiceLines);
    this.setInputPayload(reqPayload);

   // Invoke the service via a http secure connection

    String response =
                     httpPost(getWebService() + "?invoke=",
         getInputPayload(), getUsername() +
                 ":" + getPassword(), keyStoreLocation, keyStorePassword);
    return response;
   }}
```

Base64.Java code:

```
public class Base64 {
    public Base64() {
        super();
    }

    public static String byteArrayToBase64(byte[] a) {
        return byteArrayToBase64(a, a.length);
    }

    public static String byteArrayToBase64(byte[] a, int aLen) {
        int numFullGroups = aLen / 3;
        int numBytesInPartialGroup = aLen - 3 * numFullGroups;
        int resultLen = 4 * ((aLen + 2) / 3);
        StringBuffer result = new StringBuffer(resultLen);
        char intToAlpha[] = intToBase64;

        int inCursor = 0;
        for (int i = 0; i < numFullGroups; i++) {
            int byte0 = a[inCursor++] & 0xff;
            int byte1 = a[inCursor++] & 0xff;
            int byte2 = a[inCursor++] & 0xff;
            result.append(intToAlpha[byte0 >> 2]);
            result.append(intToAlpha[byte0 << 4 & 0x3f | byte1 >> 4]);
            result.append(intToAlpha[byte1 << 2 & 0x3f | byte2 >> 6]);
            result.append(intToAlpha[byte2 & 0x3f]);
        }

        if (numBytesInPartialGroup != 0) {
            int byte0 = a[inCursor++] & 0xff;
            result.append(intToAlpha[byte0 >> 2]);
            if (numBytesInPartialGroup == 1) {
                result.append(intToAlpha[byte0 << 4 & 0x3f]);
                result.append("==");
            } else {
                int byte1 = a[inCursor++] & 0xff;
                result.append(intToAlpha[byte0 << 4 & 0x3f |
                byte1 >> 4]);
                result.append(intToAlpha[byte1 << 2 & 0x3f]);
                result.append('=');
            }
        }
        return result.toString();
    }


    private static final char intToBase64[] =
    { 'A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J', 'K', 'L',
       'M', 'N','O', 'P', 'Q', 'R', 'S', 'T', 'U', 'V', 'W', 'X', 'Y',
       'Z', 'a', 'b','c', 'd', 'e', 'f', 'g', 'h', 'i', 'j', 'k', 'l',
        'm', 'n', 'o', 'p', 'q', 'r', 's', 't', 'u', 'v', 'w', 'x',
        'y', 'z', '0', '1', '2', '3','4', '5', '6', '7', '8', '9',
        '+', '/' };

       }
```

3.2 Sample of the InvoiceProcessInvoke.java code

```
import java.io.StringReader;
import java.util.HashMap;
import javax.xml.parsers.DocumentBuilder;
import javax.xml.parsers.DocumentBuilderFactory;
import javax.xml.xpath.XPath;
import javax.xml.xpath.XPathFactory;
import org.w3c.dom.Document;
import org.xml.sax.InputSource;


public class InvoiceProcessInvoke {
    public InvoiceProcessInvoke() {
        super();
    }

    public static void main(String[] args) throws Exception {

        HashMap invoiceHeader = new HashMap();
        HashMap invoiceLines = new HashMap();

       // Setting necessary input parameter values utilizing the Invoice
       Header and Lines SDOs Attributes. See Section 2.2 for list of SDOs
       and Attributes

        invoiceHeader.put("InvoiceCurrencyCode", "USD");
        invoiceHeader.put("BillToCustomerName", "Customer 1");
        invoiceHeader.put("BillToAccountNumber", "1");
        invoiceLines.put("LineNumber", "1");
        invoiceLines.put("Description", "Item 1");
        invoiceLines.put("Quantity", "4");
        invoiceLines.put("UnitSellingPrice", "1400.00");

       // Setting keyStore Location

        String keyStoreLocation = "C:\\default-keystore.jks";
        String keyStorePass = "welcome1";
        String username = “guest”;
        String password = “guest”;

       // Calling the invoice service invocation
       // The createInvoice method calls the Http Post to invoke the
       service

        InvoiceServiceClient invoice = new InvoiceServiceClient();
        String response =
                   invoice.createInvoice("efops-rel8-cdrmdit-external-
       fin.us.oracle.com",-1, username, password, invoiceHeader,
       invoiceLines,keyStoreLocation, keyStorePass);

       // Parse the response to read the service output details and
       handle errors if any.

        if (response != null && !response.isEmpty()) {
            System.out.println();
             InputSource source = new InputSource(new
             StringReader(response));
```

```
                DocumentBuilderFactory dbf =
                DocumentBuilderFactory.newInstance();
                DocumentBuilder db = dbf.newDocumentBuilder();
                Document document = db.parse(source);
                XPathFactory xpathFactory = XPathFactory.newInstance();
                XPath xpath = xpathFactory.newXPath();
                String servStatus =

                 xpath.evaluate("Envelope/Body/createSimpleInvoiceAsyncRespo
                 nse/result/ServiceStatus",  document);
                String trxNumber =
                xpath.evaluate("Envelope/Body/createSimpleInvoiceAsyncRespons
                e/result/TransactionNumber",   document);
                if ("".equals(servStatus) && "".equals(trxNumber)) {
                                 System.out.println("Service Errored. Parse
                 the Response to review the Error Message ");
                } else {
                    System.out.println("Service Status = " + servStatus);
                    System.out.println("Transaction Number = " + trxNumber);
                 }
           }
       }
   }
```

3.3 Testing the InvoiceService web service.

One-time step required for running any Java executable on a Windows system is given below.
Download and install Java and set PATH system environment variable.

- Get the latest JDK version for appropriate operating system if Java is not installed already.
- Install the Java from the executable.
- Note the path of the bin in the Java installation directory.
  The default path is `C:\Program Files\Java\jdkx.x.x\bin`.

Set the System Environment variables.

- Right-click on 'My Computer' and select 'Properties'.
- Go to Advance System Setting.
- Click Environment Variable.
- Set the PATH variable.
  If present double click it and append the bin path like
  ";`C:\Program Files\Java\jdkx.x.x\bin;`"
  Else create a New Variable by clicking NEW.
  Provide variable name as "Path" and Variable value as path of bin directory
  "`C:\Program Files\Java\jdkx.x.x\bin;`"

- Click OK.

Javac InvoiceProcessInvoke.java – This generates the Java class file.
Java InvoiceProcessInvoke – This invokes the service and prints the output.

Output :

```
Invoking Invoice Web Service
Response received.
        Service Status       =  Success
        Transaction Number =  1566
```

The invoice with transaction number = 1566 can be queried on the Fusion instance to review the invoice.

3.4 Examples of Sample Payloads

3.4.1 Request Payload for CreateSimpleInvoice operation

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
<soap:Body>
        <ns1:createSimpleInvoice
xmlns:ns1="http://xmlns.oracle.com/apps/financials/receivables/transactio
ns/invoices/invoiceService/types/">
            <ns1:invoiceHeaderInformation
xmlns:ns2="http://xmlns.oracle.com/apps/financials/receivables/transactio
ns/invoices/invoiceService/">
<ns2:BusinessUnit>Vision Operations</ns2:BusinessUnit>
<ns2:TransactionSource>AR_Transource_01</ns2:TransactionSource>
<ns2:TransactionType>AR_TransType_01</ns2:TransactionType>
<ns2:InvoiceCurrencyCode>USD</ns2:InvoiceCurrencyCode>
<ns2:TrxNumber>ServiceInv_901</ns2:TrxNumber>
<ns2:BillToCustomerName>Ar_Customer</ns2:BillToCustomerName>
<ns2:BillToAccountNumber>1004</ns2:BillToAccountNumber>
<ns2:PaymentTerm>30 Net</ns2:PaymentTerm>
<ns2:InvoiceLine>
      <ns2:LineNumber>1</ns2:LineNumber>
      <ns2:Description>InvServItem1</ns2:Description>
      <ns2:Quantity>1</ns2:Quantity>
      <ns2:UnitSellingPrice>120</ns2:UnitSellingPrice>
       <ns2:TaxClassificationCode>VAT20</ns2:TaxClassificationCode>
</ns2:InvoiceLine>
<ns2:InvoiceLine>
      <ns2:LineNumber>2</ns2:LineNumber>
      <ns2:Description>InvServItem2</ns2:Description>
      <ns2:Quantity>1</ns2:Quantity>
      <ns2:UnitSellingPrice>160</ns2:UnitSellingPrice>
       <ns2:TaxClassificationCode>VAT20</ns2:TaxClassificationCode>
</ns2:InvoiceLine>
<ns2:InvoiceLine>
      <ns2:LineNumber>3</ns2:LineNumber>
      <ns2:Description>InvServItem3</ns2:Description>
      <ns2:Quantity>1</ns2:Quantity>
      <ns2:UnitSellingPrice>180</ns2:UnitSellingPrice>
       <ns2:ExtendedAmount>180</ns2:ExtendedAmount>
       <ns2:TaxClassificationCode>VAT20</ns2:TaxClassificationCode>
</ns2:InvoiceLine>
</ns1:invoiceHeaderInformation>
</ns1:createSimpleInvoice>
</soap:Body>
</soap:Envelope>
```

Response Payload for CreateSimpleInvoice operation

The Response payload will contain the transaction number in case of successful invocation.

```
<ns0:createSimpleInvoiceResponse xmlns=""
xmlns:env="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ns0="http://xmlns.oracle.com/apps/financials/receivables/transactio
ns/invoices/invoiceService/types/"
xmlns:wsa="http://www.w3.org/2005/08/addressing"
xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-utility-1.0.xsd">
   <ns0:result
   xmlns:ns2="http://xmlns.oracle.com/apps/financials/receivables/transac
   tions/invoices/invoiceService/"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:type="ns2:InvoiceResult">
      <ns2:ServiceStatus>S</ns2:ServiceStatus>
      <ns2:TransactionNumber>ServiceInv</ns2:TransactionNumber>
      <ns2:CustomerTrxId>300100030350867</ns2:CustomerTrxId>
   </ns0:result>
</ns0:createSimpleInvoiceResponse>
```

3.4.2 Request Payload for processInterfaceLine operation

The below payload contains four order lines of a particular order.

- Order line 1 = Item line
- Order line 2 = Description Line
- Order Line 3 = Memo Line
- Order Line 4 = Freight Line (Header level freight for the order)

Important points to note in the payload are

- changeOperation value has to be "Create".
- The xsi namespace has to be declared as shown in the payload.
- While passing the values for the Descriptive Flexfield, the value for xsi:Type has to be passed. This value is nothing but the API name of the Context that is obtained during Line Transaction Flexfield Setup.
- The Xml Tag Names for the DFF segment are the API names of the segment obtained in Step 1.

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
   <soap:Body>
      <ns1:processInterfaceLine
xmlns:ns1="http://xmlns.oracle.com/apps/financials/receivables/transactions/invoices/invoiceService/types/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
         <ns1:changeOperation>Create</ns1:changeOperation>
         <ns1:interfaceLine
xmlns:ns2="http://xmlns.oracle.com/apps/financials/receivables/transactions/invoices/invoiceService/">
              <ns2:OrgId>204</ns2:OrgId>
              <ns2:BillCustomerAccountNumber>1009</ns2:BillCustomerAccountNumber>
             <ns2:BillCustomerSiteNumber>1004</ns2:BillCustomerSiteNumber>
              <ns2:BatchSourceName>Sample Imported Batch Source</ns2:BatchSourceName>
               <ns2:CurrencyCode>USD</ns2:CurrencyCode>
             <ns2:CustomerTrxTypeName>Invoice</ns2:CustomerTrxTypeName>
             <ns2:ItemNumber>AS10000</ns2:ItemNumber>
             <ns2:Description>Envoy Standard Laptop</ns2:Description>
               <ns2:LineType>LINE</ns2:LineType>
              <ns2:Quantity>113</ns2:Quantity>
               <ns2:PaymentTermsName>30 Net</ns2:PaymentTermsName>
             <ns2:SalesOrder>1001</ns2:SalesOrder>
             <ns2:SalesOrderDate>2015-04-03</ns2:SalesOrderDate>
             <ns2:UnitSellingPrice>1510.3</ns2:UnitSellingPrice>
             <ns2:UOMName>Each</ns2:UOMName>
              <ns2:ContractStartDate>2015-04-03</ns2:ContractStartDate>
              <ns2:ContractEndDate>2015-04-03</ns2:ContractEndDate>
              <ns2:TransactionInterfaceLineDff
xmlns:ns3="http://xmlns.oracle.com/apps/flex/financials/receivables/transactions/autoInvoices/TransactionLineInter
faceLineDff/" xsi:type="ns3:SampleContext">
                                          <ns3:__FLEX_Context>SAMPLE_CONTEXT</ns3:__FLEX_Context>
                          <ns3:orderNumber>1001</ns3:orderNumber>
                          <ns3:orderLineNumber>1</ns3:orderLineNumber>
                          </ns2:TransactionInterfaceLineDff>
         </ns1:interfaceLine>
         <ns1:interfaceLine
xmlns:ns2="http://xmlns.oracle.com/apps/financials/receivables/transactions/invoices/invoiceService/">
             <ns2:OrgId>204</ns2:OrgId>
             <ns2:BillCustomerAccountNumber>1009</ns2:BillCustomerAccountNumber>
             <ns2:BillCustomerSiteNumber>1004</ns2:BillCustomerSiteNumber>
              <ns2:BatchSourceName>Sample Imported Batch Source</ns2:BatchSourceName>
              <ns2:CurrencyCode>USD</ns2:CurrencyCode>
             <ns2:CustomerTrxTypeName>Invoice</ns2:CustomerTrxTypeName>
             <ns2:Description>Envoy Standard Laptop</ns2:Description>
               <ns2:LineType>LINE</ns2:LineType>
             <ns2:Quantity>13</ns2:Quantity>
               <ns2:PaymentTermsName>30 Net</ns2:PaymentTermsName>
             <ns2:SalesOrder>1001</ns2:SalesOrder>
             <ns2:SalesOrderDate>2015-04-03</ns2:SalesOrderDate>
             <ns2:UnitSellingPrice>15330.33</ns2:UnitSellingPrice>
             <ns2:UOMName>Each</ns2:UOMName>
              <ns2:ContractStartDate>2015-04-03</ns2:ContractStartDate>
              <ns2:ContractEndDate>2015-04-03</ns2:ContractEndDate>
             <ns2:TransactionInterfaceLineDff
xmlns:ns3="http://xmlns.oracle.com/apps/flex/financials/receivables/transactions/autoInvoices/TransactionLineInter
faceLineDff/" xsi:type="ns3:SampleContext">
                                         <ns3:__FLEX_Context>SAMPLE_CONTEXT</ns3:__FLEX_Context>
                          <ns3:orderNumber>1001</ns3:orderNumber>
```

```
                    <ns3:orderLineNumber>2</ns3:orderLineNumber>
           </ns2:TransactionInterfaceLineDff>
       </ns1:interfaceLine>
      <ns1:interfaceLine
xmlns:ns2="http://xmlns.oracle.com/apps/financials/receivables/transactions/invoices/invoiceService/">
          <ns2:OrgId>204</ns2:OrgId>
          <ns2:BillCustomerAccountNumber>1009</ns2:BillCustomerAccountNumber>
          <ns2:BillCustomerSiteNumber>1004</ns2:BillCustomerSiteNumber>
            <ns2:BatchSourceName>Sample Imported Batch Source</ns2:BatchSourceName>
            <ns2:CurrencyCode>USD</ns2:CurrencyCode>
           <ns2:CustomerTrxTypeName>Invoice</ns2:CustomerTrxTypeName>
           <ns2:MemoLineName>AR_Memo_Line_04</ns2:MemoLineName>
          <ns2:Description>Memo Line Line</ns2:Description>
           <ns2:LineType>LINE</ns2:LineType>
          <ns2:Quantity>46</ns2:Quantity>
           <ns2:PaymentTermsName>30 Net</ns2:PaymentTermsName>
          <ns2:SalesOrder>1001</ns2:SalesOrder>
          <ns2:SalesOrderDate>2015-04-03</ns2:SalesOrderDate>
          <ns2:UnitSellingPrice>2600.50</ns2:UnitSellingPrice>
          <ns2:UOMName>Each</ns2:UOMName>
           <ns2:ContractStartDate>2015-04-03</ns2:ContractStartDate>
           <ns2:ContractEndDate>2015-04-03</ns2:ContractEndDate>
          <ns2:TransactionInterfaceLineDff
xmlns:ns3="http://xmlns.oracle.com/apps/flex/financials/receivables/transactions/autoInvoices/TransactionLineInter
faceLineDff/" xsi:type="ns3:SampleContext">
                      <ns3:__FLEX_Context>SAMPLE_CONTEXT</ns3:__FLEX_Context>
                    <ns3:orderNumber>1001</ns3:orderNumber>
                    <ns3:orderLineNumber>3</ns3:orderLineNumber>
          </ns2:TransactionInterfaceLineDff>
       </ns1:interfaceLine>
       <ns1:interfaceLine
xmlns:ns2="http://xmlns.oracle.com/apps/financials/receivables/transactions/invoices/invoiceService/">
          <ns2:OrgId>204</ns2:OrgId>
          <ns2:BillCustomerAccountNumber>1009</ns2:BillCustomerAccountNumber>
          <ns2:BillCustomerSiteNumber>1004</ns2:BillCustomerSiteNumber>
            <ns2:BatchSourceName>Sample Imported Batch Source</ns2:BatchSourceName>
            <ns2:CurrencyCode>USD</ns2:CurrencyCode>
           <ns2:CustomerTrxTypeName>Invoice</ns2:CustomerTrxTypeName>
            <ns2:LineType>FREIGHT</ns2:LineType>
          <ns2:Description>Freight Line</ns2:Description>
          <ns2:Quantity>1</ns2:Quantity>
            <ns2:PaymentTermsName>30 Net</ns2:PaymentTermsName>
          <ns2:SalesOrder>1001</ns2:SalesOrder>
          <ns2:SalesOrderDate>2015-04-03</ns2:SalesOrderDate>
          <ns2:UnitSellingPrice>4500.50</ns2:UnitSellingPrice>
          <ns2:UOMName>Each</ns2:UOMName>
         <ns2:ContractStartDate>2015-04-03</ns2:ContractStartDate>
         <ns2:ContractEndDate>2015-04-03</ns2:ContractEndDate>
         <ns2:TransactionInterfaceLineDff
xmlns:ns3="http://xmlns.oracle.com/apps/flex/financials/receivables/transactions/autoInvoices/TransactionLineInter
faceLineDff/" xsi:type="ns3:SampleContext">
                    <ns3:__FLEX_Context>SAMPLE_CONTEXT</ns3:__FLEX_Context>
                    <ns3:orderNumber>1001</ns3:orderNumber>
                    <ns3:orderLineNumber>4</ns3:orderLineNumber>
          </ns2:TransactionInterfaceLineDff>
       </ns1:interfaceLine>
    </ns1:processInterfaceLine>
```

```
   </soap:Body>
</soap:Envelope>
```

Response Payload for processInterfaceLine operation

The Response payload will be similar to the request payload in case of success state. If there is any error, then the response payload will contain the error details.

3.4.3 Request Payload for processInterfaceDistribution operation

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
   <soap:Body>
      <ns1:processInterfaceDistribution
xmlns:ns1="http://xmlns.oracle.com/apps/financials/receivables/transactions/invoices/invoiceService/types/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
         <ns1:changeOperation>Create</ns1:changeOperation>
         <ns1:interfaceDistribution
xmlns:ns2="http://xmlns.oracle.com/apps/financials/receivables/transactions/invoices/invoiceService/">
            <ns2:AccountClass>REC</ns2:AccountClass>
            <ns2:CodeCombinationId>21445</ns2:CodeCombinationId>
            <ns2:OrgId>911</ns2:OrgId>
             <ns2:Percent>100</ns2:Percent>
            <ns2:DistributionInterfacLineDff
xmlns:ns3="http://xmlns.oracle.com/apps/flex/financials/receivables/transactions/autoInvoices/TransactionDistribut
ionInterfaceLineDff  xsi:type="ns3:SampleContext">
                           <ns3:__FLEX_Context>SAMPLE_CONTEXT</ns3:__FLEX_Context>
                           <ns3:orderNumber>1001</ns3:orderNumber>
                           <ns3:orderLineNumber>4</ns3:orderLineNumber>
            </ns2:DistributionInterfacLineDff>
         </ns1:interfaceDistribution>
         <ns1:interfaceDistribution
xmlns:ns2="http://xmlns.oracle.com/apps/financials/receivables/transactions/invoices/invoiceService/">
            <ns2:AccountClass>REV</ns2:AccountClass>
            <ns2:CodeCombinationId>21445</ns2:CodeCombinationId>
            <ns2:OrgId>911</ns2:OrgId>
           <ns2:Percent>100</ns2:Percent>
            <ns2:DistributionInterfacLineDff
xmlns:ns3="http://xmlns.oracle.com/apps/flex/financials/receivables/transactions/autoInvoices/TransactionDistribut
ionInterfaceLineDff/" xsi:type="ns3:SampleContext">
                           <ns3:__FLEX_Context>SAMPLE_CONTEXT</ns3:__FLEX_Context>
                           <ns3:orderNumber>1001</ns3:orderNumber>
                           <ns3:orderLineNumber>4</ns3:orderLineNumber>
            </ns2:DistributionInterfacLineDff>
         </ns1:interfaceDistribution>
      </ns1:processInterfaceDistribution>
   </soap:Body>
</soap:Envelope>
```

Response Payload for processInterfaceDistribution operation

The Response payload will be similar to the request payload in case of success state. If there is any error, then the response payload will contain the error details.

3.4.4 Request Payload for processInterfaceSalesCredit operation

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
   <soap:Body>
      <ns1:processInterfaceSalesCredit
xmlns:ns1="http://xmlns.oracle.com/apps/financials/receivables/transactions/invoices/invoiceService/types/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
         <ns1:changeOperation>Create</ns1:changeOperation>
         <ns1:interfaceSalesCredit
xmlns:ns2="http://xmlns.oracle.com/apps/financials/receivables/transactions/invoices/invoiceService/">
            <ns2:OrgId>911</ns2:OrgId>
            <ns2:ResourceSalesrepId>100000000177813</ns2:ResourceSalesrepId>
            <ns2:SalesrepNumber>11750</ns2:SalesrepNumber>
            <ns2:SalesCreditInterfaceLineDff
xmlns:ns3="http://xmlns.oracle.com/apps/flex/financials/receivables/transactions/autoInvoices/TransactionSalesCre
ditInterfaceLineDff/" xsi:type="ns3:SampleContext">
                           <ns3:__FLEX_Context>SAMPLE_CONTEXT</ns3:__FLEX_Context>
                           <ns3:orderNumber>1001</ns3:orderNumber>
                           <ns3:orderLineNumber>4</ns3:orderLineNumber>
            </ns2:SalesCreditInterfaceLineDff>
         </ns1:interfaceSalesCredit>
         <ns1:interfaceSalesCredit
xmlns:ns2="http://xmlns.oracle.com/apps/financials/receivables/transactions/invoices/invoiceService/">
            <ns2:OrgId>911</ns2:OrgId>
            <ns2:ResourceSalesrepId>100000008143368</ns2:ResourceSalesrepId>
            <ns2:SalesrepNumber>999990000955450</ns2:SalesrepNumber>
            <ns2:SalesCreditInterfaceLineDff
xmlns:ns3="http://xmlns.oracle.com/apps/flex/financials/receivables/transactions/autoInvoices/TransactionSalesCre
ditInterfaceLineDff/" xsi:type="ns3:SampleContext">
                           <ns3:__FLEX_Context>SAMPLE_CONTEXT</ns3:__FLEX_Context>
                           <ns3:orderNumber>1001</ns3:orderNumber>
                           <ns3:orderLineNumber>4</ns3:orderLineNumber>
            </ns2:SalesCreditInterfaceLineDff>
         </ns1:interfaceSalesCredit>
      </ns1:processInterfaceSalesCredit>
   </soap:Body>
</soap:Envelope>
```

Response Payload for processInterfaceSalesCredit operation

The Response payload will be similar to the request payload in case of success state. If there is any error, then the response payload will contain the error details.

3.4.5 Request Payload for processInterfaceContingency operation

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
   <soap:Body>
      <ns1:processInterfaceContingency
xmlns:ns1="http://xmlns.oracle.com/apps/financials/receivables/transactions/invoices/invoiceService/types/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
                                        <ns1:changeOperation>Create</ns1:changeOperation>
         <ns1:interfaceContingency
xmlns:ns2="http://xmlns.oracle.com/apps/financials/receivables/transactions/invoices/invoiceService/">
            <ns2:ContingencyCode>Extended Payment Term</ns2:ContingencyCode>
             <ns2:ContingencyId>5</ns2:ContingencyId>
            <ns2:ExpirationDays>2</ns2:ExpirationDays>
            <ns2:OrgId>911</ns2:OrgId>
            <ns2:ContingencyInterfaceLineDff
xmlns:ns3="http://xmlns.oracle.com/apps/flex/financials/receivables/transactions/autoInvoices/TransactionContingencyInterfaceLineDff/" xsi:type="ns3:SampleContext">
                          <ns3:__FLEX_Context>SAMPLE_CONTEXT</ns3:__FLEX_Context>
                          <ns3:orderNumber>1001</ns3:orderNumber>
                          <ns3:orderLineNumber>4</ns3:orderLineNumber>
              </ns2:ContingencyInterfaceLineDff>
         </ns1:interfaceContingency>
             <ns1:interfaceContingency
xmlns:ns2="http://xmlns.oracle.com/apps/financials/receivables/transactions/invoices/invoiceService/">
            <ns2:ContingencyCode>Customer Creditworthiness</ns2:ContingencyCode>
             <ns2:ContingencyId>3</ns2:ContingencyId>
            <ns2:ExpirationDays>2</ns2:ExpirationDays>
            <ns2:OrgId>911</ns2:OrgId>
            <ns2:ContingencyInterfaceLineDff
xmlns:ns3="http://xmlns.oracle.com/apps/flex/financials/receivables/transactions/autoInvoices/TransactionContingencyInterfaceLineDff  xsi:type="ns3:SampleContext">
                          <ns3:__FLEX_Context>SAMPLE_CONTEXT</ns3:__FLEX_Context>
                          <ns3:orderNumber>1001</ns3:orderNumber>
                          <ns3:orderLineNumber>4</ns3:orderLineNumber>
            </ns2:ContingencyInterfaceLineDff>
         </ns1:interfaceContingency>
      </ns1:processInterfaceContingency>
   </soap:Body>
</soap:Envelope>
```

Response Payload for processInterfaceContingency operation

The Response payload will be similar to the request payload in case of success state. If there is any error, then the response payload will contain the error details.