ORACLE
INSURANCE

Oracle Insurance Data Gateway

Integration Guide

Version 1.0

Oracle Insurance Data Gateway Integration Guide

Release .1.0

Part # E93380-01

Library # E93054-01

Contributing Authors: Mark Taylor, Devika Ravindran, Michael Schwitzgebel

CONTENTS

PREFACE

Welcome to the Oracle Insurance Data Gateway (OIDG) Integration Guide. This document describes the integration options for this application.

Audience

This guide is intended for connectivity analysts and integration developers who are tasked with developing and testing the data feeds to be generated and sent to OIDG. Readers of this document should be familiar with XML, Web Services protocols and the ACORD Messaging Library Standards.

Related Documents

For more information, refer to the following resources:

- The ACORD Messaging Library standards

Conventions

The following text conventions are used in this document:

Convention	Description
bold	Boldface type indicates key information regarding the subject
Italic	Italic type indicates book titles, reference materials, emphasis or placeholders
`Monospace`	Monospace type indicates commands within a paragraph, URLs, code in examples

Documentation Accessibility

This documentation may contain links to websites of other companies or organizations that Oracle does not own or control. Oracle neither evaluates nor makes any representations regarding the accessibility of these Websites.

Customer Support

If you have any questions about the installation or use of our products, please call +1.800.223.1711 or visit the My Oracle Support website:

http://www.oracle.com/us/support/index.html

Go to My Oracle Support to find answers in the Oracle Support Knowledge Base, submit, update or review your Service Requests, engage the My Oracle Support Community, download software updates, and tap into Oracle proactive support tools and best practices.

Hearing impaired customers in the U.S. who need to speak with an Oracle Support representative may use a Telecommunications Relay Service (TRS). Information about TRS is available at http://www.fcc.gov/cgb/consumerfacts/trs.html

A list of phone numbers is available at http://www.fcc.gov/cgb/dro/trsphonebk.html. International hearing impaired customers should use the TRS at 1.605.224.1837.

Contact

USA: +1.800.223.1711
Canada: 1.800.668.8921 or +1.905.890.6690
Latin America: 877.767.2253

For other regions including Latin America, Europe, Middle East, Africa, and Asia Pacific regions:
Visit: http://www.oracle.com/us/support/contact/index.html

Follow Us

 https://blogs.oracle.com/insurance

 https://www.facebook.com/oracleinsurance

 https://twitter.com/oracleinsurance

 https://www.linkedin.com/groups?gid=2271161

Chapter – 1

INTRODUCTION

Purpose

This document describes the integration options for the Oracle Insurance Data Gateway application. It is intended for developers and connectivity analysts intending to design and build client feeds into the system.

A Business Process Flow Example

OIDG allows client and host companies to participate in various insurance transaction types while simplifying the technical integration needs on both sides. Client companies will have a standard set of integration APIs and procedures to develop against while the host company is shielded from client specific requirements. OIDG provides the platform to support the varied integration needs including security, transport type, transaction type, job frequency, etc., of each client using the system.

Here is an example set of insurance processes OIDG supports.

- A client company creates a pending policy record and sends it to OIDG to be forwarded on to a receiving company.
- As changes occur to the status of the pending policy by the client, the client sends an updated pending policy through OIDG to the receiving company.
- Once a policy is issued any changes in the policy including change in policy status, change in coverage, change in client details, etc. will be sent through OIDG to the receiving company as a policy update.
- As periodic premium payments are processed by the client company, an updated policy record with the premium payment info included will be sent through OIDG to the receiving company.
- Commission statements generated by the client company are sent through OIDG to the receiving company and will be used to reconcile commission payments.
- Finally a full policy extract will be conducted by the client company and sent the receiving company for policy reconciliation.

Insurance Processes

The following insurance processes are supported by OIDG. The ACORD Messaging Library (AML) message type and proposed modes of operation are given for each.

Note: A Policy Synchronization Message contains the complete* policy and customer information

Business Process	Message Details	Proposed Frequency and Mode

Proposal Notification and New Business(NB) Status Updates	• PolicySynchronizationProcess Message • Transmitted when a proposal is received and processed by new business • Subsequent NB underwriting status changes are also transmitted to host	One policy per message via web-services, or multiple policies in batch via sFTP.
New Policy Notification	• PolicySynchronizationProcess Message • Transmitted when a new policy is issued at the time of issuance	Web service, one message per policy record
Policy Changes	• PolicySynchronizationProcess Message • Transmitted when a policy transaction occurs that changes an active policy • Must include change activity type and date	Web service, one message per policy record
Initial Historical Policy Migration	• PolicySynchronizationProcess Message • Initial migration of historical policy data to host company based on predetermined set of policy statuses to be discussed with ISP. • Includes complete policy detail and client detail	One time scheduled batch process via sFTP
Policy Reconciliation	• Abbreviated PolicySynchronization Process Message • Monthly full policy listing of host company policy records for reconciliation purposes. • Subset of complete policy detail and all client detail	Monthly scheduled batch process via sFTP

Note: A Premium Bordereaux Message contains policy references and premium payment data		
Business Process	**Message Details**	**Proposed Frequency and Mode**
Publish Premium Payments	• PremiumBorderauxProcess Message • When a premium payment is made on a policy • Includes policy reference, amount of premium paid and limited payment detail – payment amount, payment date, payment method	Web service, one message per policy record
Publish Commission Statements	• PremiumBorderauxProcess Message • Commission Statement details • Includes policy reference and commission due details on multiple policies per commission statement	Monthly scheduled batch process via sFTP.

ACORD Messaging Library (AML) Standards

OIDG messaging is based on industry standards developed by the Association for Cooperative Operations Research and Development (ACORD). The ACORD Messaging Library standard will be referenced throughout this document as the AML standard. The following is a brief description from the ACORD Messaging Library standards documentation.

The ACORD Messaging Library (AML) Standard contains business messages that support business functions relevant to all lines of business, varied geographies, business-to-business (B2B), business-to-consumer (B2C), and internal systems. The message inventory is developed on a common set of design rules.

The business message functions vary as needed, either as unique or common. For example, one business message may be specific to one portion of the industry (i.e., obtain life insurance policy quote OR issue commercial auto policy) or have cross-industry relevance (i.e., change name). The resulting business messages provide for consistency regardless of process, product, or geography. Each message is designed at a level of granularity that is appropriate for the business function it supports to alleviate inconsistent interpretation. The technical architecture is vendor neutral and supports cross-industry standards such as HTTP, SOAP, and XML.

Details on the specific messages to be used by the client company to communicate information with the host company via OIDG are defined in this document. However, this is not meant to be a replacement of the ACORD published specifications which should be used as supplementary reference documents for additional details.

Additional Documentation

In addition to this specification, you may need to reference the following documents:

ACORD Messaging Library (AML) Standards

ACORD published the AML specification for Asia Pacific on September, 2016. This new standard is based on the Australia/New Zealand approved standards that are already published in addition to new additions added specifically for inclusion of Asia Pacific requirements. The new standard is available for download at www.acord.org.

ACORD XML Naming and Design Rules

The ACORD Naming and Design Rules Specification documents the common XML architectural functionality, naming conventions, design rules, schema implementation and data types that are used in ACORD XML specifications. This specification provides the information and rules necessary to define XML structures that will be reused across all the domains at ACORD.

ACORD Messaging Library Messages

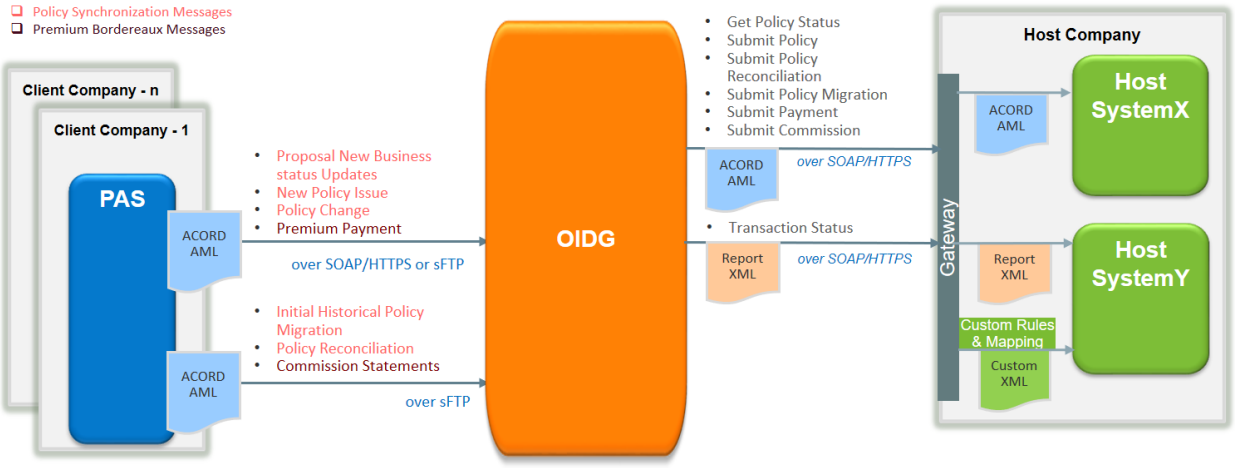
Two AML messages are used as the standard formats for communicating the following required policy information updates between client and host companies.

ACORD AML Process Message	Description
Policy Synchronization Message	A PolicySynchronization message contains policy and customer data and is utilized to communicate policy details between client company and host company. This includes information on the current status and contents of policies from the inception of a new business application at the client company, updates to policies currently being underwritten, and updates to active policies.
Premium Bordereaux Message	A PremiumBordeaux message contains policy data and payment data and is used to communicate premium payments and commission payments between client and host companies.

COMMUNICATION PROTOCOLS

This section discusses the available communication protocols available for data transmission between sending and receiving systems. Two methods will be made available for all processes.

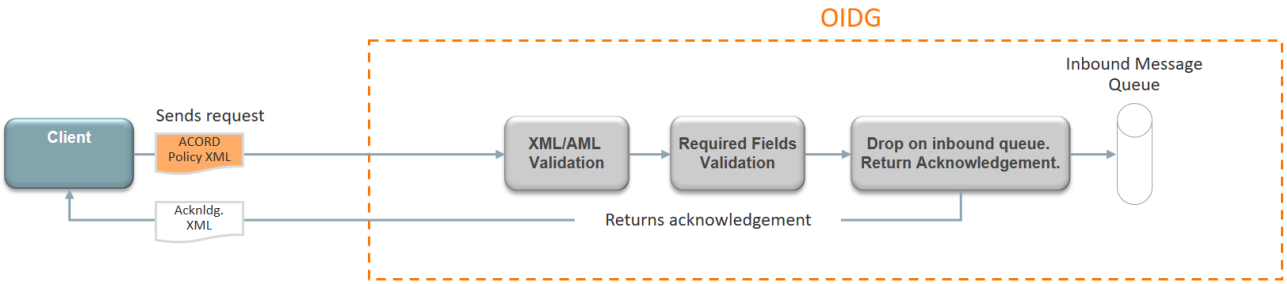
- Web services over HTTPS
- Secure FTP



Web Services

OIDG web-services have the following characteristics:

- Client services are synchronous, one-way, with an acknowledgment response
- Outbound processing resulting from the client request happens at a later time and results are logged.



- Outbound business service calls to host company systems are also synchronous and one-way.
- All web-services follow WS-Security standards for the authentication of SOAP messages where user and password are presented in the SOAP header and are authenticated for access to the service.

OIDG exposed web-services

Name	WSDL	Type
IDXClientWebServices	 IDXClientServices-effective.wsdl	• Synchronous Message • "String" payload type for the WSDL

Creating SOAP Messages

SOAP Overview

SOAP (Simple Object Access Protocol) is an XML-based language used for the transport of structured information from a requester to a provider. A SOAP message is sent from the requesting application to an ISP Interface Web Service. A SOAP response message including the outcome is then returned to the requester. Proper authentication information must be included in the security portion of the header. The body, explained in detail later in this document, simply consists of the message, as defined by the service's WSDL.

Sample SOAP Request:

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
  xmlns:oics="http://www.oracle.com/oidx/services">
  <soap:Header>
    <wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-
200401-wss-wssecurity-secext-1.0.xsd" xmlns:wsu="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">
      <wsse:UsernameToken wsu:Id="UsernameToken-2">
        <wsse:Username>cleartext-user</wsse:Username>
        <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-
200401-wss-username-token-profile-1.0#PasswordText">cleartext-
password</wsse:Password>
        <wsse:Nonce EncodingType="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-soap-message-security-
1.0#Base64Binary">c4WScQsfQhbGwRgr4iixKA==</wsse:Nonce>
        <wsu:Created>2016-10-11T13:06:58.178Z</wsu:Created>
      </wsse:UsernameToken>
      <wsu:Timestamp wsu:Id="TS-1">
        <wsu:Created>2016-10-11T13:06:58.172Z</wsu:Created>
        <wsu:Expires>2016-10-11T21:26:58.172Z</wsu:Expires>
      </wsu:Timestamp>
    </wsse:Security>
    <oics:DXHeader>
      <oics:SenderKey>CompanyUniqueKey</oics:SenderKey>
    </oics:DXHeader>
  </soap:Header>
  <soap:Body>
    <oics:PolicySynchronizationProcessString>
    <![CDATA[<!-- PolicySynchronizationProcess message XML goes here -->]]>
    </oics:PolicySynchronizationProcessString>
```

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


soap12-envelope.xsd


IDXClientServices.xsd


ClientNotifications.xsd

WSDL

WSDL (Web Service Definition Language) is an XML-based language used to describe Web Services. In the case of the OIDG, the WSDL for each available Web Service defines the message format, data type, and transport protocols that should be used between the requester and the provider.

Web Service Security

OIDG adheres to the WS-Security standards for the authentication of SOAP messages. The standards, as developed by the OASIS Open committee, can be referenced here Webservice security standards. As per this approach, consuming applications need to send the User name token as part of the SOAP security header along with the SOAP request.

The <wsse:UsernameToken> element is used to include the authentication information. The username and password are specified inside of the <wsse:Username>, and <wsse:Password> elements, respectively. The WSS specification for the UsernameToken node can be found here WSS <wsse:UsernameToken> Specification.

Here is a sample SOAP header showing the <wsse:UsernameToken> being used.

```
<soap:Header xmlns:soap="http://www.w3.org/2003/05/soap-envelope">
  <wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd" xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">
    <wsse:UsernameToken wsu:Id="UsernameToken-15E9903557CDC8897B151623016683822">
      <wsse:Username>ispuser</wsse:Username>
      <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-profile-1.0#PasswordText">example.password</wsse:Password>
      <wsse:Nonce EncodingType="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-message-security-1.0#Base64Binary">rjMDKGRjMpCmjCV6HLo/KQ==</wsse:Nonce>
      <wsu:Created>2018-01-17T23:02:46.838Z</wsu:Created>
    </wsse:UsernameToken>
    <wsu:Timestamp wsu:Id="TS-15E9903557CDC8897B151623016683821">
      <wsu:Created>2018-01-17T23:02:46.837Z</wsu:Created>
      <wsu:Expires>2018-01-18T07:22:46.837Z</wsu:Expires>
    </wsu:Timestamp>
  </wsse:Security>
</soap:Header>
```

TLS 1.2 is required as the method of encryption for all OIDG SOAP messages.

SOAP fault usage for reporting errors

The SOAP specification defines a standard fault mechanism for reporting errors related to both the SOAP Header and Body. The SOAP specification defines standard high-level SOAP fault codes that can be refined to report specific errors.

In this implementation SOAP faults are considered for exception reporting and are used when it is not possible nor appropriate to send a normal service response or because the service request could not be processed for various reasons (e.g. syntax error in the service messages, congestion, failure in the authentication process).

Sample Soap Fault messages:

Success Acknowledgement

```
<soap:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope"
xmlns:soap="http://www.w3.org/2003/05/soap-envelope">
  <env:Header />
  <env:Body>
    <oics:Acknowledgement xmlns:oics="http://www.oracle.com/oidx/services">
      <TransactionSummary xmlns="http://www.oracle.com/oidx/results">
        <StatusCode>received</StatusCode>
        <TransmissionId>9bf69749-5991-4866-ad54-410bb5f28645</TransmissionId>
        <TransmissionDateTime>2017-11-01T12:27:12.675-7:00
</TransmissionDateTime>
        <FileName />
        <MessageId>85c4907a-171f-421b-8aae-96600510f880</MessageId>
        <CorrelationId>85c4907a-171f-421b-8aae-96600510f880</CorrelationId>
        <MessageDateTime>2017-11-01T12:27:13.232-07:00</MessageDateTime>
        <TransactionDetail>
          <DocumentId>A3C18A9E-71F5-410E-81D3-D456E98D655D</DocumentId>
          <AssignedIds>
            <AssignedId>
              <TypeCode>PolicyNumber</TypeCode>
              <Id>Pol_20171101_89093</Id>
            </AssignedId>
            <AssignedId>
              <TypeCode>ApplicationNumber</TypeCode>
              <Id>APP29385</Id>
            </AssignedId>
          </AssignedIds>
        </TransactionDetail>
      </TransactionSummary>
    </oics:Acknowledgement>
  </env:Body>
</soap:Envelope>
```

Schema Validation Failure

```
<soap:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope"
xmlns:soap="http://www.w3.org/2003/05/soap-envelope">
  <env:Header/>
```

```
  <env:Body>
    <oics:Acknowledgement xmlns:oics="http://www.oracle.com/oidx/services">
      <TransactionSummary xmlns="http://www.oracle.com/oidx/results">
        <StatusCode>rejected</StatusCode>
        <TransmissionId>afacc621-62c9-45c9-8f6b-30e877903371</TransmissionId>
        <TransmissionDateTime>2017-11-01T12:19:41.83-
07:00</TransmissionDateTime>
        <FileName />
        <MessageId>85c4907a-171f-421b-8aae-96600510f880</MessageId>
        <CorrelationId>85c4907a-171f-421b-8aae-96600510f880</CorrelationId>
        <MessageDateTime>2017-11-01T12:19:41.952-07:00</MessageDateTime>
        <TransactionDetail>
          <DocumentId>22170F2F-7D1B-4273-B785-2183C1E60284</DocumentId>
          <AssignedIds>
            <AssignedId>
              <TypeCode>PolicyNumber</TypeCode>
              <Id>Pol_20171101_49808</Id>
            </AssignedId>
            <AssignedId>
              <TypeCode>ApplicationNumber</TypeCode>
              <Id>APP39026</Id>
            </AssignedId>
          </AssignedIds>
          <Messages>
            <Message type="error">
              <TypeCode>XMLSchemaError</TypeCode>
              <Description>QName value
'{http://www.acord.org/schema/data/draft/ReusableDataComponents/1}Annual' is
not a valid enumeration value for ContractAmountItemTypeCode_Type in
namespace
http://www.acord.org/schema/data/draft/ReusableDataComponents/1</Description>
            </Message>
          </Messages>
        </TransactionDetail>
      </TransactionSummary>
    </oics:Acknowledgement>
  </env:Body>
</soap:Envelope>
```

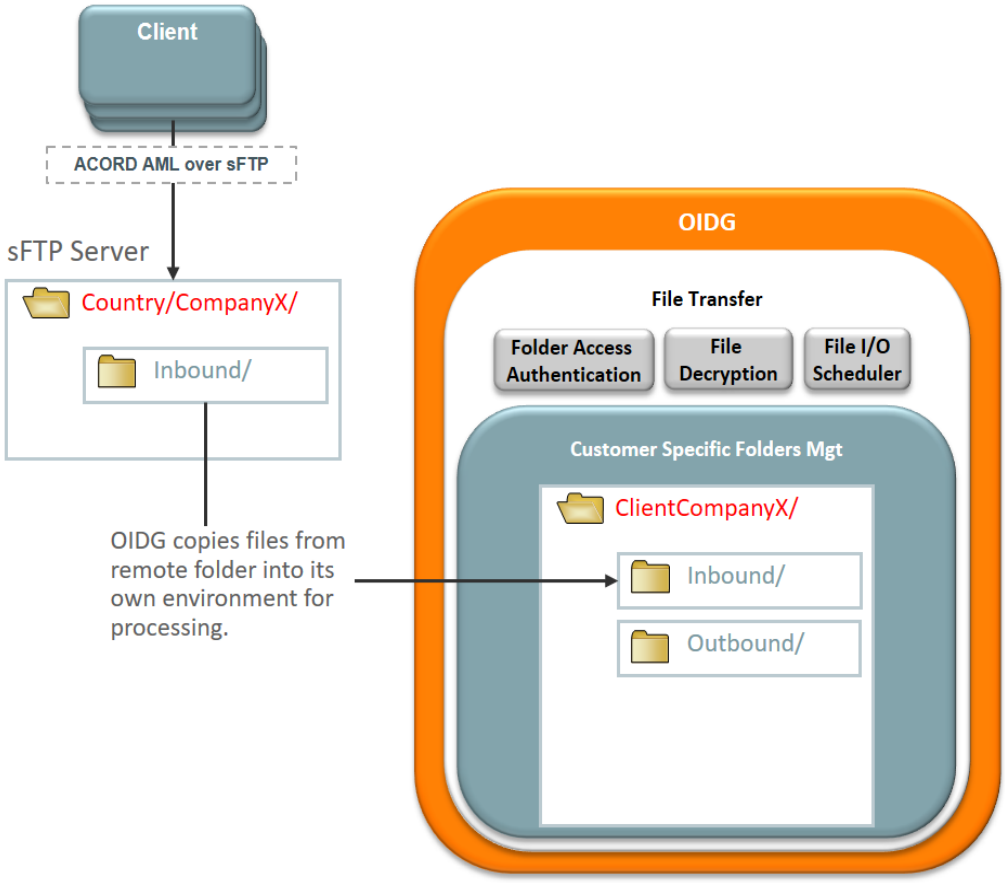
Business Rules Validation Failure

```
<soap:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope"
xmlns:soap="http://www.w3.org/2003/05/soap-envelope">
  <env:Header />
  <env:Body xmlns:oics="http://www.oracle.com/oidx/services">
    <oics:Acknowledgement>
      <TransactionSummary xmlns="http://www.oracle.com/oidx/results">
        <StatusCode>rejected</StatusCode>
        <TransmissionId>6f9bab56-0882-4a0d-ab79-7196d12ee016</TransmissionId>
```

```
        <TransmissionDateTime>2017-11-01T12:07:07.068-
07:00</TransmissionDateTime>
        <FileName />
        <MessageId>85c4907a-171f-421b-8aae-96600510f880</MessageId>
        <CorrelationId>85c4907a-171f-421b-8aae-96600510f880</CorrelationId>
        <MessageDateTime>2017-11-01T12:07:07.607-07:00</MessageDateTime>
        <TransactionDetail>
          <DocumentId>76a88380-d6db-11dd-ad8b-0800200c9a66</DocumentId>
          <AssignedIds>
            <AssignedId>
              <TypeCode>PolicyNumber</TypeCode>
              <Id>Pol_20171101_28171</Id>
            </AssignedId>
          </AssignedIds>
          <Messages>
            <Message type="error">
              <TypeCode>BusinessRuleValidationError</TypeCode>
              <Description>ApplicationSignedDate is required</Description>
            </Message>
            <Message type="error">
              <TypeCode>BusinessRuleValidationError</TypeCode>
              <Description>ISP Name/Organization full name is
required</Description>
            </Message>
          </Messages>
        </TransactionDetail>
      </TransactionSummary>
    </oics:Acknowledgement>
  </env:Body>
</soap:Envelope>
```

File Transfer

- Each client using sFTP should be assigned its own FTP folder.
- Folder permissions are provisioned by the hosting company.
- Files placed in FTP folders should be PGP encrypted and digitally signed by the client company to protect the file contents and to guarantee its integrity. A digital signature serves as a checksum and a non-repudiation device.
- The schedule for file transfers is to be determined between the client and host companies.



A successful integration between the client company and the company hosting OIDG will need to follow the following guidelines:

- Each client company should have their own FTP and local folders.
- Each client company folder structure should include Inbound and Outbound top-level folders to more easily separate files coming from the client and those coming from host.
- All transactions in a batch request need to be of the same transaction type. In other words, a batch request should not mix Policy Migration with Policy Synchronization or Premium Payment transaction types.

Request processing order cannot be guaranteed as OIDG processes files concurrently for better performance.

Error Handling

There are two ways to send (or transmit) requests to the OIDG, web services and FTP. Errors can occur in either transmission type or are reported back to the user, but via different mechanisms. OIDG does not provide any UI to clients for monitoring errors so they are either reported real-time as SOAP faults or written to files. Two levels of validation are done. The first is a check against the ACORD AML schema. The second is a check against the input data rules for the particular transaction being executed. If either of these fail, the request is rejected.

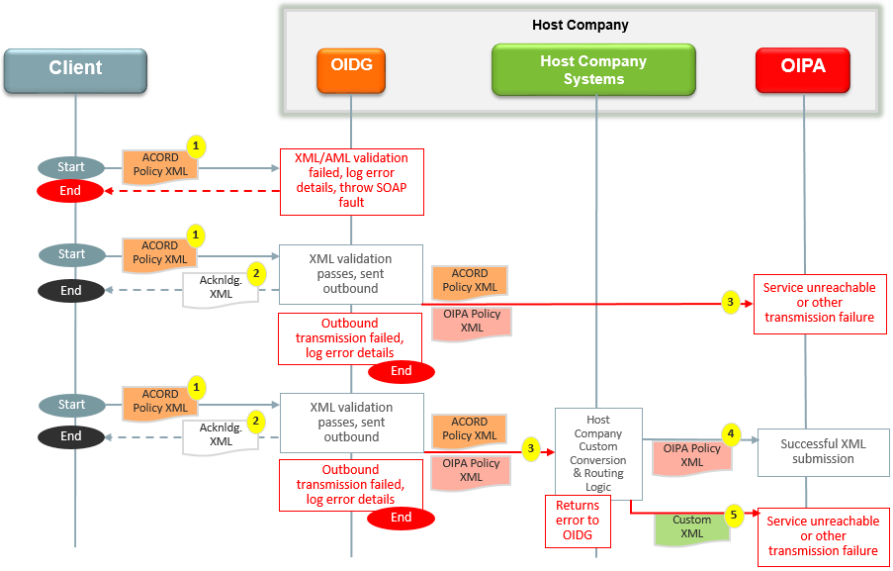
Web service transmissions are processed real-time and the corresponding request validations are processed real-time. When a validation fails during web service processing a SOAP fault is thrown. The web service caller will have to catch and process the SOAP fault. The errors are also logged in OIDG's database for later system-level error reporting. If no failures occur during validation, an acknowledgement is sent back to the caller.

Upon receiving a web-service request, OIDG will:

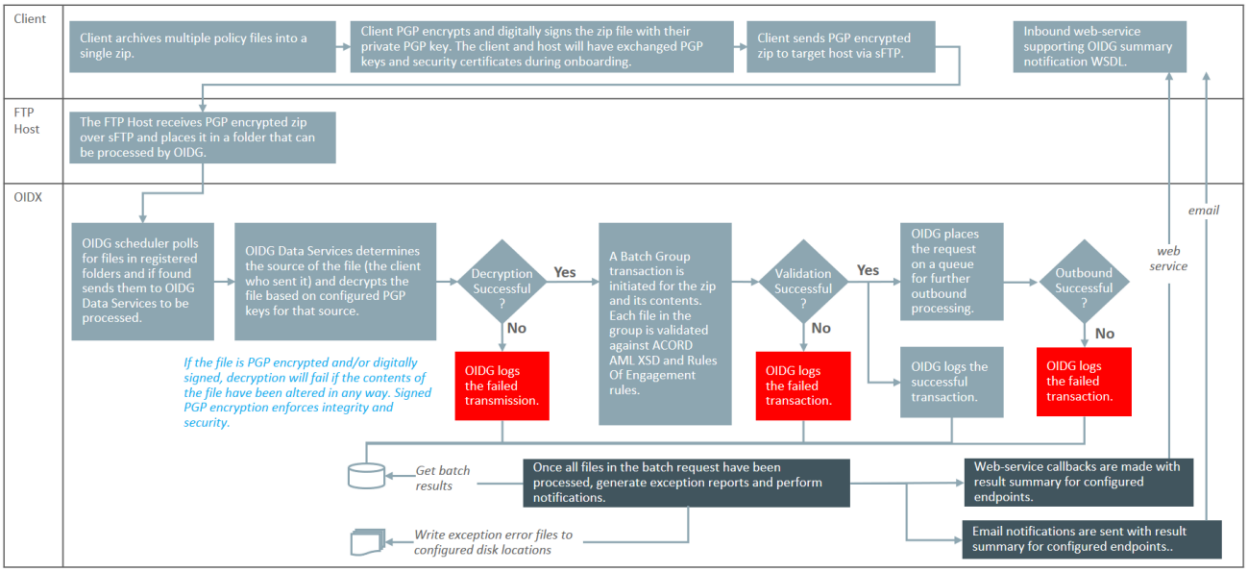
- Send a successful receipt acknowledgement if no XML validation errors are found
- Throw a SOAP fault if XML errors are found

After acknowledgement OIDG will log errors found during outbound processing.

Note: A transmission failure during outbound processing will not roll back the submission. Manual resubmission of some failed outbound transmissions is possible using OIDG QuickView.



FTP transmissions are file-based and are processed asynchronously as batch requests. When errors occur during FTP processing, they are logged. Once the FTP request processing has completed, a report file can be generated for that request and placed into the configured Error folder for the client company that initiated the request.



System administrators will be able to see all errors for all client companies via OIDG's Quick View UI. They will also be able to configure the system to send them error file reports similar to those sent to the client's error folders. Error files are generated per batch. Errors that have

occurred during the batch are put into a report XML file and written to disk in a configurable location.

Result Notification

At the end of batch file processing, OIDG can call web services and/or send emails to report the batch status information. There are two kinds of notifications: results notification, typically for the host, and validation notification, typically for the client. Detail error reports can be generated on the local file system or be sent to an FTP destination. Notification endpoints (web service URLs, emails addresses, report file location, FTP path) are configured using the AdminView web application.

Chapter – 3

POLICY SYNCHRONIZATION PROCESS MESSAGE

XML Message	<PolicySynchronizationProcess>, <PolicySynchronizationProcessResult>
Definition	This message is the means by which policy information is shared between brokers and insurers. This message will contain all policy information from the client company as described in this specification. As the receiving party, the host company can choose to either overlay the existing data with the entire policy record received with each update or compare the records and make the appropriate updates. A PolicySynchronization message is sent from the client to communicate the following policy updates.
Schedule:	When an application is first received and processed as a new business pending policy a PolicySynchronization message is sent at the time a policy number is registered
	When a pending policy's status is updated as it undergoes the underwriting process. Policy status changes on pending policies like Pending, Not Taken, Issued, etc. will trigger a PolicySynchronization message.
	When a policy change occurs on an issued policy such as change in coverage, beneficiary, coverage amount, policy status, etc. a PolicySynchronization message should be triggered

Please refer to the ACORD AML schema, the XPath mapping document for Required Fields (Appendix A) and the corresponding sample AML files (Appendix B) for modeling and mapping purposes.

Chapter – 4

PREMIUM BORDEREAUX PROCESS

XML Message	<PremiumBordereauxProcess>, <PremiumBordereauxProcessResult>
Definition	These messages pass financial report information for a specified time period. The financial report could be related to a premium paid on a policy or commissions paid on a policy.
Schedule	A client company can send a PremiumBordereauxProcess message to the host when a premium payment is made on a policy. This message could be real time or batch mode.
	On a planned schedule the client may send a PremiumBordereauxProcess message in batch mode per commission statement reporting on commissions paid on active policies.

Please refer to the ACORD AML schema, the XPath mapping document for Required Fields and the corresponding sample AML files (Appendix B) for modeling and mapping purposes.

CODE TYPES LOOKUP

The ACORD AML schema contains a number of type codes. Some of these type codes are closed enumerated type code fields mapped in the Xpath mapping document (see Appendix B).

Supported Data Fields with XML Path Mapping to AML PolicySynchronizationProcess and PremiumBordereauxProcess Messages

The embedded document in Appendix B provides the list of data fields required for OIDG submission with the various messages identified in the Chapter – 1. This document is version controlled to manage updates that will be provided in the future and contains the data fields required for the messages and the XPATH mapping for those data fields in the AML messages. Use this in conjunction with the sample files provided in the Appendix B and the AML schema.

Appendix A

RESULT NOTIFICATION MESSAGES AND TRANSACTION ERROR REPORT FILE

Below are the sample notification messages sent by OIDG when a batch group process finishes.

- Web Service
- Email

If OIDG is configured to generate or send transaction error report files, a zip file is created for each batch group in the specified location if the batch group is completed with some error. For each transaction that has an error, an XML file with detail error messages is inserted into the zip file.

```
<TransactionSummary xmlns="http://www.oracle.com/oidx/results">
  <StatusCode>rejected</StatusCode>
  <TransmissionId>038dec36-7ad8-4a02-8979-614a8b021704</TransmissionId>
  <MessageId />
  <CorrelationId>85c4907a-171f-421b-8aae-96600510f880</CorrelationId>
  <MessageDateTime>2017-10-10 19:38:04.447</MessageDateTime>
  <TransactionDetail>
    <DocumentId>cd6db595-8b0c-4a72-ae7e-8597cf8ca966</DocumentId>
    <AssignedIds>
      <AssignedId>
        <TypeCode>PolicyNumber</TypeCode>
        <Id>Pol_20170725_36956</Id>
      </AssignedId>
      <AssignedId>
        <TypeCode>ApplicationNumber</TypeCode>
        <Id />
      </AssignedId>
    </AssignedIds>
    <Messages>
      <Message>
        <TypeCode>BusinessRuleValidationError</TypeCode>
        <Description>OICS-00008: Request payload failed business rules
validation</Description>
      </Message>
      <Message>
        <TypeCode>error</TypeCode>
        <Description>MaritalStaus Code is required</Description>
      </Message>
    </Messages>
  </TransactionDetail>
</TransactionSummary>
```

Appendix B

OIDG REQUIRED FIELDS

There are a number of input fields that OIDG requires beyond the ACORD AML specification. These are documented here.

OIDG Required Fields V1.0.xlsx

AML SAMPLE FILES

[NOTE: REFERENCE ONLY]

1.	PolicySynchronization Real Time Policy Update File – Life Product New Policy, Pending Policy updates and Inforce Policy Changes. A batched version of this sample is used to create the one time full policy migration file.	 PolicySynchronizationMessageLifeOnly 1.4
2.	Monthly Policy Reconciliation – PolicySynchronizationProcess Message Sample batch File – 3 Policies, abbreviated policy data	 PolicySynchronizationMessageLife_PolicyF
3.	Policy Synchronization Real Time Policy Update – Investment Link Product New Policy, Pending Policy updates and Inforce Policy Changes	 PolicySynchronizationMessageInvestment
4.	Policy Synchronization Real Time Policy Update – Personal Auto Product	 PolicySynchronizationMessageGI-Persona
5.	Policy Synchronization Real Time Policy Update – Personal Property Product	 PolicySynchronizationMessageGI-Persona
6.	Premium Bordereaux Real Time Premium Payment Update	 PremiumBordereaux_Premium Payment1.4

7.	Premium Bordereaux Batch Commission Statement	 PremiumBordereaux_ Commission Statemer  PremiumBordereaux_ Commission Statemer