ORACLE
INSURANCE

Oracle® Insurance Policy Administration

DI Client Utility

Version 10.1.1.0

Document Part Number: E55503-01
August, 2014

License Restrictions

Table of Contents

INTRODUCTION

DATA INTAKE CLIENT UTILITY TOOL HELPS THE CUSTOMER TO SEND JMS MESSAGES IN THE FORMAT EXPECTED BY THE OIPA SERVER FOR DATA INTAKE PROCESSING.

CUSTOMER SUPPORT

If you have any questions about the installation or use of our products, please visit the My Oracle Support website: https://support.oracle.com, or call (800) 223-1711.
Oracle customers have access to electronic support through My Oracle Support. For information, visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info or visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs if you are hearing impaired.

OVERVIEW

Data Intake Client Utility tool is required by the customer to send JMS messages in the format expected by the OIPA server for Data Intake processing. OIPA server expects following four JMS Messages for record processing.

Input data is expected to be in XML format based on the schema defined in Intake profile Definition for a Customer which is configurable using Rules Palette in OIPA.

These messages are expected to be sent in sequential order for every file. Each message have considerable impact on changing file status during loading and processing. Each of these messages are explained below.

JMS MESSAGES

- **Begin Loading**
  1. This message submits file data to the server.
  2. Submitted once for each file in a data intake scenario which carries file related fields like expected count of records etc.
  3. Message content is File Field XML in XML structure as mentioned below (content in bold). This is Optional and can be empty.
     1.
        1.
           1. <dataIntakeScenario>

              <!-- Global elements like Group Customer Number, Profile Name will be here-->

              <testFile>

                <!--File Fields -->

                <fields>

                <field NAME="FieldName1" TYPE="TEXT">FieldValue1</field>

           2. **<field NAME="FieldName2" TYPE="TEXT">FieldValue2</field>**

              </fields>

              <testRecord>

                <!-- Record Data -->

              </testRecord>

              </testFile>

              </dataIntakeScenario>

  4. Below properties should be attached to process this message on server side.

PropertyName	Property Type	PropertyValue	Description
operation	String	"beginLoading"(hard coded string)	Name of the operation
fileId	String	FileId (Generated Value)	Generated GUID vaue for each file
groupCustomerNumber	String	GroupCustomerNumber (captured from Intake File)	Customer Number
profileName	String	ProfileName (Captured from Intake	Intake Profile Name

		File)	
expectedRecordCount	Integer	expectedRecordCount (Calculated number of records for a file)	Count of Records for each file. This can be calculated using XPATH on the input file

- ## Add Record
    1. This message submits record data to the server
    2. Submitted for every record in order of its presence within DI file.
    3. It sets the process order[1] for every record based on its precedence within a record group (Uniquely identified by RecordGroupId[2]) of a DI File.
    4. Message content is the record XML data as shown below.
        1.
            1. <dataIntakeScenario>

               <pathToMemberId></pathToMemberId>

               <pathToRecordGroupId></pathToRecordGroupId>

               <groupCustomerNumber></groupCustomerNumber>

               <profileName></profileName>

               <testFile>

                  <!--File Fields -->

               <testRecord>

            2. **<oipa:GroupMemberIntakeRecord FILEID="string" xmlns:oipa="http://xmlns.oracle.com/insurance/oipa/v1">**

                  <!-- Record Data -->

                  <Fields>

                  <Field1></Field1>

                  <Field2></Field2>

                </Fields>

            3. **</oipa:GroupMemberIntakeRecord>**

                  </testRecord>

               </testFile>

            </dataIntakeScenario>

5. Below properties should be attached to process this message on server side.

PropertyName	Property Type	PropertyValue	Description
operation	String	"addRecord"(hard coded string)	Name of the operation
fileId	String	FileId (Generated Value)	Generated GUID vaue for each file
groupCustomerNumber	String	GroupCustomerNumber (captured from Intake File)	Customer Number
memberId	String	memberId (Captured from Intake Record)	Unique representation of a Record
recordGroupId	String	recordGroupId (Captured from Intake Record)	Unique representation of a record group.
processOrder	Integer	Process Order (Assigned for each record)	Records under each record Group will be assigned an incremental process order

- ## Complete Loading

1. This message updates the file status to Complete Loading on Server side.
2. Below properties should be attached to process this message on server side.

PropertyName	Property Type	PropertyValue	Description
operation	String	"completeLoading"(hard coded string)	Name of the operation
fileId	String	FileId (Generated Value)	Generated GUID vaue for each file
groupCustomerNumber	String	GroupCustomerNumber (captured from Intake File)	Customer Number

- ## Process

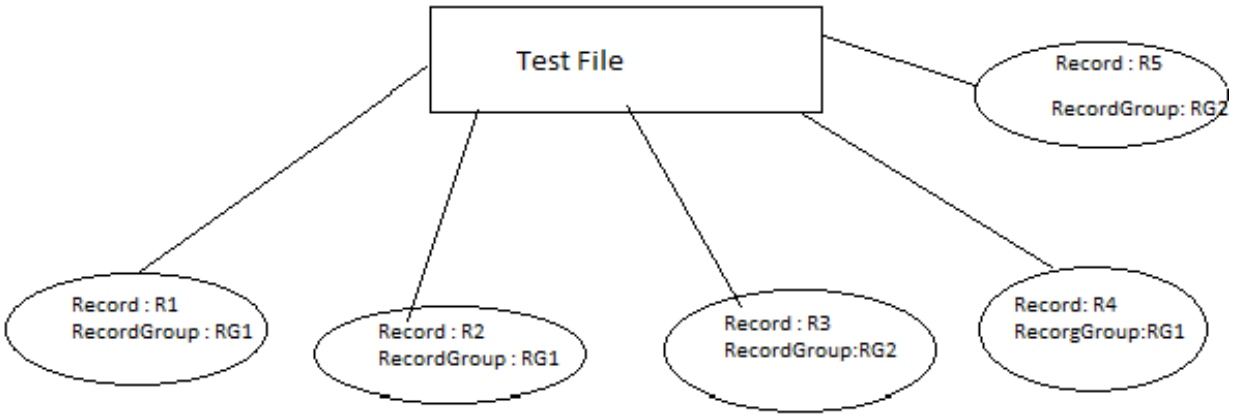
1. This message updates the file status to "Pending" on the Server side.
2. Below properties should be attached to process this message on server side.

PropertyName	Property Type	PropertyValue	Description
operation	String	"process"(hard coded string)	Name of the operation
fileId	String	FileId (Generated Value)	Generated GUID vaue for each file
groupCustomerNumber	String	GroupCustomerNumber (captured from Intake File)	Customer Number

RECORD GROUP ID AND PROCESS ORDER

1. Multiple Records in the file can belong to a record group which is uniquely identified by a RecordGroupID (Example Sponsor Id)
2. Process Order will be assigned to every record on basis of which the records will be processed on the server side.

Consider the below scenario.



In the above example

Records R1,R2 and R4 belong to a RecordGroup RG1

Records R3 and R5 belong to Record Group RG2

So the Processing order for the first Record Group will be

Record	Record Group	Processing Order
R1	RG1	0
R2	RG1	1
R4	RG1	2
R3	RG2	0
R5	RG2	1

Observe that processing order starts from 0 whenever a new record group is found while reading the input file.

JMS CONNECTION

JMS Connection Properties can be either defined in a property file or XML file which need to be loaded on client startup. These are required for message communication between DI Client and OIPA Server.
Below properties are required for DI Client to connect to the OIPA Server through JMS

Property	Description	Value
contextFactory	Property that determines which Context Factory to be used to connect JMS provider	Weblogic weblogic.jndi.WLInitialContextFactory Websphere com.ibm.websphere.naming.WsnInitialContextFactory
provider.url	Property that determines JMS provider URL	Weblogic t3://server:port Websphere iiop://server:BootStrapAddress
jndi.connectionFactoryName	Property that determines connection factory JNDI name	**IntakeConnectionFactory**
jndi.queueName	Property that determines Queue JNDI name	**DIQueue**

jndi.connectionFactoryName : **IntakeConnectionFactory**
jndi.queueName : **DIQueue**
The above property values should match with the Connection Factory name and Queue name configured on Application Server.

STRUCTURE OF A DATA INTAKE FILE

A sample data intake input file is constructed as shown:

```
<dataIntakeScenario>
   <!--
      SET TO THE XPATH IN EACH RECORD WHERE WE FIND THE MEMBER ID AND
      SPONSOR ID TO BE SENT
   -->
   <pathToMemberId>Person/Fields/TaxId</pathToMemberId>
   <!--
      SET TO THE XPATH IN EACH RECORD WHERE WE FIND THE GROUP ID FOR A
      RECORD
   -->
   <pathToRecordGroupId>Person/Fields/SponsorId</pathToRecordGroupId>
   <!--
      SET TO WHATEVER GROUP CUSTOMER IS BEING USED FOR TEST, THIS IS THE
      CUSTOMER NUMBER
   -->
   <groupCustomerNumber></groupCustomerNumber>
   <!-- SET TO THE NAME OF THE PROFILE USED FOR THIS TEST -->
   <profileName></profileName>
   <testFile>
      <testRecord>
         <oipa:GroupMemberIntakeRecord FILEID="string"
            xmlns:oipa="http://xmlns.oracle.com/insurance/oipa/v1">
            <Fields>
               <MyField>SomeString</MyField>
            </Fields>
            <Entity>
               <Fields>
                  <TaxId></TaxId>
                  <SponsorId></SponsorId>
                  <Fields>
            </Entity>
         </oipa:GroupMemberIntakeRecord>
      </testRecord>
   </testFile>
</dataIntakeScenario>
```

- As shown in the above example, each input file represents a data intake scenario which can have multiple files.
- Each file can have multiple records.
- Content under each and every Record will be loaded as record data into AsIntakeRecord table on the server side under respective AsIntakeFile w.r.t IntakeFileGuid