OUM

DS.140 Design Specification

RAYA

OFM-SOA-Implementation-

Integration Design –DROP 1.5

Author: Karunanidhi Natarajan

Creation Date: March 04, 2015

Last Updated: March 04, 2015

Version: 1.0 - Draft

1. **Title, Subject, Last Updated Date, Reference Number**, **and** **Version** are marked by a Word Bookmark so that they can be easily reproduced in the header and footer of documents. When you change any of these values, be careful not to accidentally delete the bookmark. **You can make bookmarks visible by selecting Tools->Options…View and checking the Bookmarks option in the Show region.**

**Approvals:**

|  |  |
| --- | --- |
| Mohammad Abumalloh |  |
| Osama AlDameri |  |

1. To add additional approval lines, press [Tab] from the last cell in the table above.
2. You can delete any elements of this cover page that you do not need for your document.

# Document Control

## Change Record

4

| Date | Author | Version | Change Reference |
| --- | --- | --- | --- |
| 04-Mar-2015 | Karunanidhi N | 1.0 | Initial or Draft Version |
| 26-Mar-15 | Karunanidhi N | 1.1 | Added stmt the Sequence of XML data elements in payload as described in WSDL |
|  |  |  |  |
|  |  |  |  |

## Reviewers

| Name | Position |
| --- | --- |
| Mohammad Abumalloh | SBM Project Manager |
| Osama Al Dameri | EMAD Project Manager |
|  |  |
|  |  |

Contents

1 Document Control 2

1.1 Change Record 2

1.2 Reviewers 2

2 Introduction 5

2.1 Assumptions 5

2.2 Building Blocks 5

2.3 Block Relationship Diagram 6

3 Integration Design for Insurance Creation 7

3.1 Business Rule Design 7

3.2 Integration Design - Quick Reference 7

3.3 Data Sources 7

3.4 Technical Design Information 10

3.5 Error Handling Considerations 11

3.6 Web Security Considerations 11

3.7 Performance Considerations 11

4 Integration Design for Insurance Status Update 12

4.1 Business Rule Design 12

4.2 Integration Design - Quick Reference 12

4.3 Data Sources 12

4.4 Technical Design Information 13

4.5 Error Handling Considerations 14

4.6 Web Security Considerations 14

4.7 Performance Considerations 14

5 Integration Design for Claim Request 15

5.1 Business Rule Design 15

5.2 Integration Design - Quick Reference 15

5.3 Data Sources 15

5.4 Technical Design Information 17

5.5 Error Handling Considerations 18

5.6 Web Security Considerations 18

5.7 Performance Considerations 18

6 Integration Design for Claim Status Update 19

6.1 Business Rule Design 19

6.2 Integration Design - Quick Reference 19

6.3 Data Sources 19

6.4 Technical Design Information 20

6.5 Error Handling Considerations 21

6.6 Web Security Considerations 21

6.7 Performance Considerations 21

7 Quality of Service Design Considerations 22

7.1 Restart Strategy 22

7.2 Crash Recovery 22

7.3 Security 22

7.4 Performance 22

8 Installation Considerations 23

9 Open and Closed Issues 24

9.1 Open Issues 24

9.2 Closed Issues 24

10 Appendix 25

1. To update the table of contents, put the cursor anywhere in the table and press [F9]. To change the number of levels displayed, select the menu option Insert‑>Index and Tables, make sure the Table of Contents tab is active, and change the Number of Levels to a new value.

# Introduction

The purpose of this task is to assemble all the information that is required to describe the design of a software component into a complete Design Specification. This task is not a substitute for executing the individual design tasks. This specification work product can serve as a structure for completing the design for each component by providing pointers back into the Design Tasks:   
- DS.040 Develop Design Architecture Description  
- DS.080 Design Software Components  
- DS.090 Design Data  
- DS.100 Design Behavior  
- DS.130 Design User Interface

EMAD is implementing Oracle SOA to integrate the data flow between Siebel, OLFM and Thrid party. This design document covers the technical specifications required to implement the same. This Application Technical Design document complements the Siebel, OLFM and Third party Functional and Technical Specification and the same should be considered for complete detailed design

.

## Assumptions

The design has below assumtions:

1. Oracle SOA layer will act as Integration layer betweem OLFM and SIEBEL and will be a prerequisite for data flows.
2. OLFM and SIEBEL system should exists and the same is configured in Oracle Fusion Middleware-SOA to connect with each other via SOA integration adapters such as DB, APPS, FTP, File adapters and etc.
3. Fault Handler Service should exist.
4. Siebel inbound/outbound dependency services should exist and available to communicate.
5. EBS Staging tables and Concurrent Programs available to Invoke and get the response as per the design.

## Building Blocks

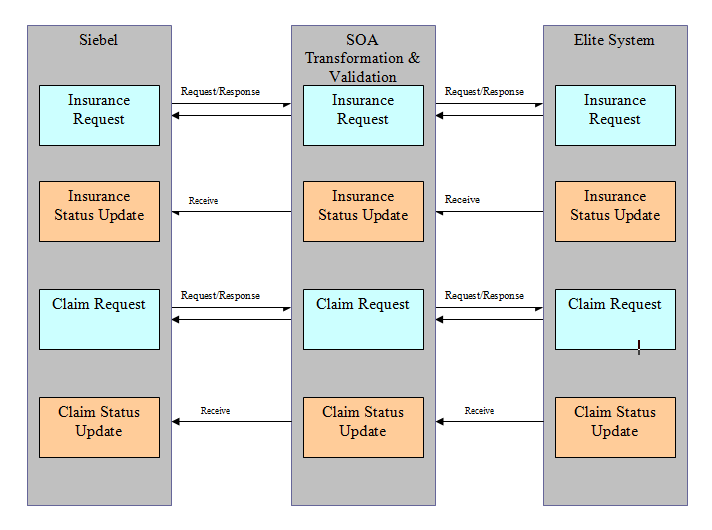
The Component design includes several Integration as listed below:

1. InsuranceRequestProcess
2. InsuranceStatusUpdate
3. ClaimRequestProcess
4. ClaimStatusUpdate

## Block Relationship Diagram

1. The intent of this section is to graphically depict how the component under consideration interfaces to related components, external systems, and other actors that interact with the use-case package. Reference the Conceptual View and Module View of the Architecture Description (RD.130) and the class diagram prepared in the Software Component Design (DS.080) and Component Behavior Design (DS.100).

The process flow diagram explains the blocks and their relationships at the processes (Building blocks) level for Insurance Integration Components.



# Integration Design for Insurance Creation

1. The intent of this section is to design the services between the components and the interfaces with external systems for each Use Case. Refer to DS.080 Software Component Design and focus on calling arguments (i.e., service signature) and logic definition.

The Integration service provides facility to create Insurance with the third party. Here Elite system will act as third party where in the insurance request gets created by Siebel. The Elite Service will be called by using Username & password authentication. The business functionality and validation rules are covered in Elite Technical Specification.

## Business Rule Design

Siebel will invoke the Insurance Integration Service to create Insurance in third party system (Elite). The XML Data elements should in the same order how it is described in the WSDL for any kind of operation such as Insert, Modify, Update, Cancel and Renewl

1. Receive the Input payload of Insurance Request

2. Validate and Transform the input to Elite Service based on the operation types.

3. Get the response from Elite Service and pass it to Siebel.

## Integration Design - Quick Reference

|  |  |
| --- | --- |
| SOA Project Name | **InsuranceRequestProcess** |
| Service Type | Synchronous |
| Source - Target System | Siebel - Elite |
| **Inbound SOA Service Design** | SOA WSDL Interface(Request and Response XSD) is derived from Elite Service wsdl (attached in 4.3.1) |
| Outbound SOA Reference Design | Elite Web service, Fault Handler |
| Frequency | On Demand |
| Volume | Single/Batch |

## Data Sources

### Request/Response Information

|  |  |
| --- | --- |
| **Type** | **Details** |
|
| Request |  |
| Response |

### Domain Value Mapping (DVM)

This DVM will be used to store Elite Web Service access credentials. This needs to be created and stored in MDS.

The same can be used in ClaimRequest integration service as well.

|  |  |
| --- | --- |
| **Name** | **Description** |
|
| InsuranceAccessCredentials | Elite WS access credentials to create Insurance/Claim |

Domain Map Table for InsuranceAccessCredentials.dvm.

(The values are used for DEV instance only)

|  |  |
| --- | --- |
| **Field** | **Value** |
| userName | RAYA |
| passWord | RAYA |
| groupID | ZAHRAN |

### Mapping Information

**Request (** Mandatory Fields varies based on Operation Type. Please refer 3.4 Technical Design for more details**)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Source Data Entity (Siebel)** | **Required** | **Target Data Entity(Elite WS)** | **Data Type & Length** | **Default Value** | **Comments** |
| MCAR\_OPERATION |  | MCAR\_OPERATION | VARCHAR(1) |  |  |
| MCAR\_REQSEQNO |  | MCAR\_REQSEQNO | NUMBER(5) |  |  |
| MCAR\_POLNUM |  | MCAR\_POLNUM | VARCHAR(14) |  |  |
| MCAR\_APPNUM |  | MCAR\_APPNUM | NUMBER(18) |  |  |
| MCAR\_APPLIN |  | MCAR\_APPLIN | NUMBER(5) |  |  |
| MCAR\_EFFDATE |  | MCAR\_EFFDATE | VARCHAR(10) DD/MM/YYYY |  |  |
| MCAR\_EXPDATE |  | MCAR\_EXPDATE | VARCHAR(10) DD/MM/YYYY |  |  |
| MCAR\_FIRSTNAME |  | MCAR\_FIRSTNAME | VARCHAR(100) |  |  |
| MCAR\_LASTNAME |  | MCAR\_LASTNAME | VARCHAR(100) |  |  |
| MCAR\_FATHERNAME |  | MCAR\_FATHERNAME | VARCHAR(100) |  |  |
| MCAR\_FIRSTNAME\_AR |  | MCAR\_FIRSTNAME\_AR | VARCHAR(400) |  |  |
| MCAR\_LASTNAME\_AR |  | MCAR\_LASTNAME\_AR | VARCHAR(400) |  |  |
| MCAR\_FATHERNAME\_AR |  | MCAR\_FATHERNAME\_AR | VARCHAR(400) |  |  |
| MCAR\_IQAMAID |  | MCAR\_IQAMAID | VARCHAR(30) |  |  |
| MCAR\_GOVSEQNBR |  | MCAR\_GOVSEQNBR | VARCHAR(50) |  |  |
| MCAR\_MAKE |  | MCAR\_MAKE | VARCHAR(5) |  |  |
| MCAR\_MODEL |  | MCAR\_MODEL | VARCHAR(5) |  |  |
| MCAR\_BODYTYPE |  | MCAR\_BODYTYPE | VARCHAR(3) |  |  |
| MCAR\_USAGE |  | MCAR\_USAGE | VARCHAR(6) |  |  |
| MCAR\_HSPOWER |  | MCAR\_HSPOWER | NUMBER(3) |  |  |
| MCAR\_YEAR |  | MCAR\_YEAR | NUMBER(4) |  |  |
| MCAR\_NBPLACE |  | MCAR\_NBPLACE | NUMBER(5) |  |  |
| MCAR\_ENGINENo |  | MCAR\_ENGINENo | VARCHAR(25) |  |  |
| MCAR\_CHASSNo |  | MCAR\_CHASSNo | VARCHAR(25) |  |  |
| MCAR\_PLATENo |  | MCAR\_PLATENo | VARCHAR(50) |  |  |
| MCAR\_BENEFICIARY |  | MCAR\_BENEFICIARY | VARCHAR(250) |  |  |
| MCAR\_AREA |  | MCAR\_AREA | VARCHAR(5) |  |  |
| MCAR\_DRIVER\_NAME\_1 |  | MCAR\_DRIVER\_NAME\_1 | VARCHAR(150) |  |  |
| MCAR\_DRIVER\_DOB\_1 |  | MCAR\_DRIVER\_DOB\_1 | VARCHAR(10) DD/MM/YYYY |  |  |
| MCAR\_DRIVER\_NAME\_2 |  | MCAR\_DRIVER\_NAME\_2 | VARCHAR(150) |  |  |
| MCAR\_DRIVER\_DOB\_2 |  | MCAR\_DRIVER\_DOB\_2 | VARCHAR(10) DD/MM/YYYY |  |  |
| MCAR\_DRIVER\_NAME\_3 |  | MCAR\_DRIVER\_NAME\_3 | VARCHAR(150) |  |  |
| MCAR\_DRIVER\_DOB\_3 |  | MCAR\_DRIVER\_DOB\_3 | VARCHAR(10) DD/MM/YYYY |  |  |
| MCAR\_CV1\_DESC |  | MCAR\_CV1\_DESC | VARCHAR(25) |  |  |
| MCAR\_CV2\_DESC |  | MCAR\_CV2\_DESC | VARCHAR(25) |  |  |
| MCAR\_CV3\_DESC |  | MCAR\_CV3\_DESC | VARCHAR(25) |  |  |
| MCAR\_CV4\_DESC |  | MCAR\_CV4\_DESC | VARCHAR(25) |  |  |
| MCAR\_CV5\_DESC |  | MCAR\_CV5\_DESC | VARCHAR(25) |  |  |
| MCAR\_CV6\_DESC |  | MCAR\_CV6\_DESC | VARCHAR(25) |  |  |
| MCAR\_REPAIR |  | MCAR\_REPAIR | VARCHAR(1) |  |  |
| MCAR\_DEDUCT\_PERC |  | MCAR\_DEDUCT\_PERC | NUMBER(3) |  |  |
| MCAR\_DEDUCT\_MIN |  | MCAR\_DEDUCT\_MIN | NUMBER(18,2) |  |  |
| MCAR\_DEDUCT\_MAX |  | MCAR\_DEDUCT\_MAX | NUMBER(18,2) |  |  |
| MCAR\_SUMINS |  | MCAR\_SUMINS | NUMBER(18,2) |  |  |
| MCAR\_IKAMA |  | MCAR\_IKAMA | VARCHAR(250) |  |  |
| MCAR\_PASSPORT |  | MCAR\_PASSPORT | VARCHAR(250) |  |  |
| MCAR\_IDCARD |  | MCAR\_IDCARD | VARCHAR(250) |  |  |

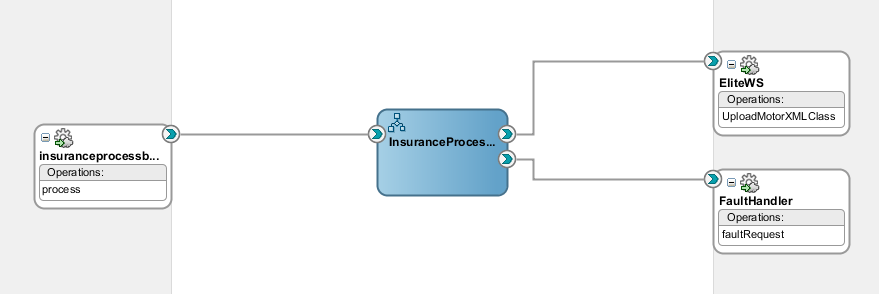
**Response**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Source Data Entity (Elite WS)** | **Required** | **Target Data Entity(Siebel)** | **Data Type & Length** | **Default Value** | **Comments** |
| MCAR\_APPNUM | Y | MCAR\_APPNUM | NUMBER(18) |  |  |
| MCAR\_APPLIN | Y | MCAR\_APPLIN | NUMBER(5) |  |  |
| MCAR\_REQSEQNo | Y | MCAR\_REQSEQNo | NUMBER(5) |  |  |
| MCAR\_POLNUM | Y | MCAR\_POLNUM | VARCHAR(14) |  |  |
| MCAR\_STATUS | Y | MCAR\_STATUS | VARCHAR(1) |  |  |
| MCAR\_REMARKS | Y | MCAR\_REMARKS | VARCHAR(2000) |  |  |

## Technical Design Information

1. The intent of this section is to identify the table, columns, and source values that are required to support the above data elements. Refer to the Physical Database Design (IM.040), to identify the existing tables where the above attributes are located.

### Composite Diagram



### Technical Program Flow

* Get Input Payload from Siebel
* Pass it to BPEL Receive activity
* Payload should contain the XML Data elements in order as per business rule design in 3.1
* Validate the Operation Types and Mandatory Fields

The mandatory fields depending on Operation type:

* Mandatory fields in case of Operation type = ‘I’‘:
  + MCAR\_REQSEQNO and MCAR\_POLNUM
  + MCAR\_EFFDATE and MCAR\_EXPDATE
  + MCAR\_FIRSTNAME\_AR and MCAR\_LASTNAME\_AR
  + MCAR\_IQAMAID
  + MCAR\_MAKE and MCAR\_MODEL
  + MCAR\_BODYTYPE and MCAR\_USAGE
  + MCAR\_HSPOWER and MCAR\_YEAR and MCAR\_NBPLACE
  + MCAR\_ENGINENO and MCAR\_CHASSNO
  + MCAR\_AREA
  + MCAR\_DRIVER\_NAME\_1 and MCAR\_DRIVER\_DOB\_1
  + MCAR\_REPAIR
  + MCAR\_SUMINS
  + MCAR\_IKAMA and MCAR\_PASSPORT and MCAR\_IDCARD
* Mandatory fields in case of Operation type ‘M’
  + MCAR\_APPNUM
  + MCAR\_APPLIN
* Mandatory fields in case of Operation type ‘C’
  + MCAR\_APPNUM
  + MCAR\_APPLIN
* Mandatory fields in case of Operation type ‘U’
  + MCAR\_APPNUM
  + MCAR\_APPLIN
  + MCAR\_GOVSEQNBR
  + MCAR\_PLATENo
* Mandatory fields in case of Operation type ‘R’
  + MCAR\_APPNUM
  + MCAR\_APPLIN

After successful validation, Derive username, password, group Id from InsuranceAccessCredentials.dvm.

Example:

dvm:lookupValue('oramds:/apps/dvm/ InsuranceAccessCredentials.dvm','Field','userName','Value','userName')

* Transform Insurance Request Payload to Elite WS
* Invoke the Elite webservice
* Get the response and assign back to Siebel.
* Incase of any validation failure, reply to Siebel with valid Status and error message using

MCAR\_STATUS and MCAR\_REMARKS

## Error Handling Considerations

EMAD Standard Fault Handler Service is used to capture all system related failure/error information.

## Web Security Considerations

SOA Admin will receive the Elite WS URL along with username, password and Group ID.

This will be configured through SiebelAccessCredentials.dvm in SOA Server.

1. The intent is to design the rules that are necessary to verify the format, length, relationships etc. of the attributes listed above.

## Performance Considerations

As per Oracle SOA Integration and EMAD standards.

# Integration Design for Insurance Status Update

1. The intent of this section is to design the services between the components and the interfaces with external systems for each Use Case. Refer to DS.080 Software Component Design and focus on calling arguments (i.e., service signature) and logic definition.

The Integration service provides facility to receive Insurance status and pass it to Siebel and OLFM to record the insurance status.

## Business Rule Design

Elite Service (Third party) will use this integration Service to send the insurance status

1. Receive the Input payload of Insurance Status from Elite

2. Validate the input.

3. Check the status=’C’ (Cancellation), then Update AP invoice staging table in OLFM and Invoke CP to create AP Invoice.

4. Otherwise, Update Siebel with status details

## Integration Design - Quick Reference

|  |  |
| --- | --- |
| SOA Project Name | **InsuranceStatusUpdate** |
| Service Type | One Way |
| Source - Target System | Elite - Siebel |
| **Inbound SOA Service Design** | Input Payload Schema is attached in 4.3.1 |
| Outbound SOA Reference Design | Siebel Web service, Fault Handler |
| Frequency | On Demand |
| Volume | Single/Batch |

## Data Sources

### Request/Response Information

|  |  |
| --- | --- |
| **Type** | **Details** |
|
| Request |  |

### Domain Value Mapping(DVM)

None

### Mapping Information

**Request**

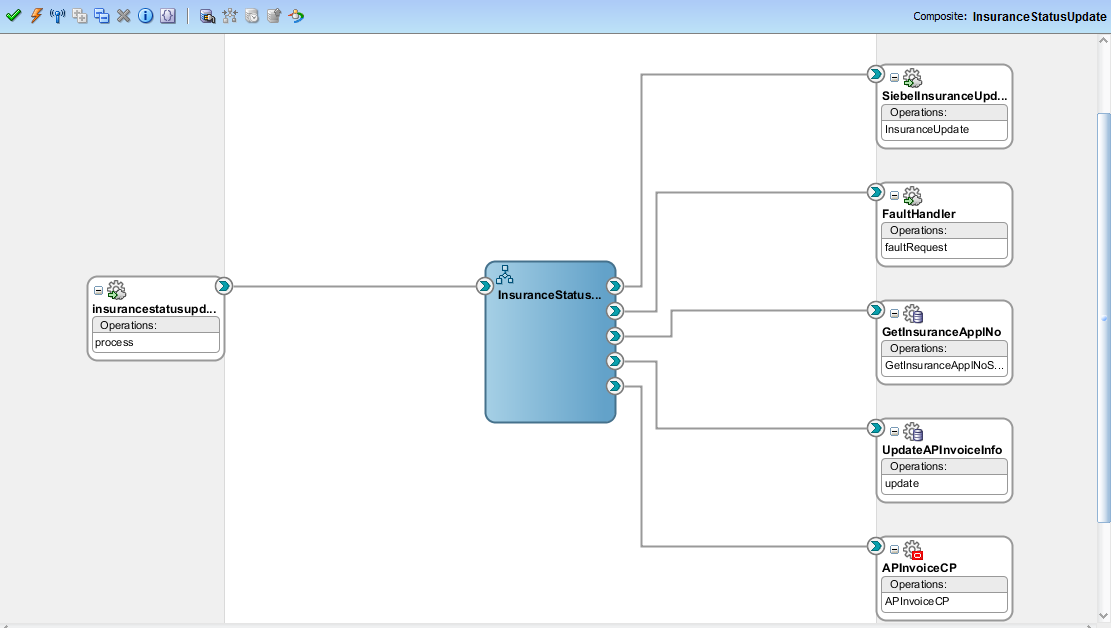
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Source Data Entity (EliteWS)** | **Required** | **Target Data Entity(Siebel/OLFM)** | **Data Type & Length** | **Default Value** | **Comments** |
| MCAR\_APPNUM | Y | MCAR\_APPNUM | NUMBER(18) |  | Application No. |
| MCAR\_APPLIN | Y | MCAR\_APPLIN | NUMBER(5) |  | Application Sequence No. |
| MCAR\_STATUS | Y | MCAR\_STATUS | VARCHAR(1) |  | Status |
| MCAR\_CANCELLATION\_DATE | N | MCAR\_CANCELLATION\_DATE | VARCHAR2(110 |  | Cancellation Date with format MM/DD/YYYY for OLFM |
| MCAR\_CANCELLATION\_CHARGE | N | MCAR\_CANCELLATION\_CHARGE | NUMBER(18,2) |  | Cancellation Charge for OLFM |
| MCAR\_REINSPOLNUM | Y | MCAR\_REINSPOLNUM | VARCHAR(25) |  | Insurance Co, Policy No. |
| MCAR\_REINSENDNUM | Y | MCAR\_REINSENDNUM | VARCHAR(25) |  | Insurance Co. Endorsement No. |
| MCAR\_PREMIUM | Y | MCAR\_PREMIUM | NUMBER(18,2) |  | Price |
| MCAR\_POLICY | Y | MCAR\_POLICY | VARCHAR(250) |  | Policy Document (PDF) |
| MCAR\_REMARKS |  | MCAR\_REMARKS | VARCHAR2(2000) |  | Remarks |

* “MCAR\_APPNUM “and MCAR\_APPLIN “: key unique between both Systems Elite and Raya.
* “MCAR\_STATUS “: Status is equal to “I” (Issued)or “E” (Error) with friendly description of the error in “MCAR\_REMARKS”.
* “MCAR\_REINSPOLNUM” and “MCAR\_REINSENDNUM” are the policy number received by the insurance company.
* “MCAR\_PREMIUM“ is the car premium calculated by Elite System.
* “MCAR\_POLICY “ is the map path of the scanned printout policy (PDF Format).

## Technical Design Information

1. The intent of this section is to identify the table, columns, and source values that are required to support the above data elements. Refer to the Physical Database Design (IM.040), to identify the existing tables where the above attributes are located.

### Composite Diagram



### Technical Program Flow

* Receive Insurance Status Update Payload from Elite
* Check the Status
* If status is ‘C’, then update the OLFM staging table XXAMG\_RY\_AP\_INVOICES\_STG\_T

with Cancellation Date and Cancellation Charge using Application Sequence No and Application Number

Then invoke AP Invoice Creation Concurrent program in OLFM

* If status is not ‘C’, then Invoke Siebel Service to Send Insurance Status update along with SiebelAccessCredentials.dvm
* Handle the fault if any and call standard Fault handler service

## Error Handling Considerations

EMAD Standard Fault Handler Service is used to capture all system related failure/error information.

## Web Security Considerations

This integration service will be authenticated by user name and password from SOA Server.

The below WS security policy has to be attached to endpoint Service

**“Oracle/wss\_username\_token\_service\_policy”**

1. The intent is to design the rules that are necessary to verify the format, length, relationships etc. of the attributes listed above.

## Performance Considerations

As per Oracle SOA Integration and EMAD standards.

# Integration Design for Claim Request

1. The intent of this section is to design the services between the components and the interfaces with external systems for each Use Case. Refer to DS.080 Software Component Design and focus on calling arguments (i.e., service signature) and logic definition.

The Integration service provides facility to create a claim against Insurance with the third party. Here Elite system will act as third party where in the claim request is sent by Siebel. The Elite Service will be called by using Username & password authentication. The business functionality and validation rules are covered in Elite Technical Specification.

## Business Rule Design

Siebel will invoke the Claim Request Service to create a claim.

1. Receive the Input payload of Claim Request

2. Validate and Transform the input to Elite Service

3. Get the response from Elite Service and pass it to Siebel.

## Integration Design - Quick Reference

|  |  |
| --- | --- |
| SOA Project Name | **InsuranceClaimProcess** |
| Service Type | Synchronous |
| Source - Target System | Siebel - Elite |
| **Inbound SOA Service Design** | SOA WSDL Interface(Request and Response XSD) is derived from Elite Service wsdl (attached in 6.3.1) |
| Outbound SOA Reference Design | Elite Web service, Fault Handler |
| Frequency | On Demand |
| Volume | Single/Batch |

## Data Sources

### Request/Response Information

|  |  |
| --- | --- |
| **Type** | **Details** |
|
| Request |  |
| Response |

### Domain Value Mapping (DVM)

|  |  |
| --- | --- |
| **Name** | **Description** |
|
| InsuranceAccessCredentials | Elite WS access credentials to create Insurance/Claim |

This is already created in MDS as part of InsuranceRequestProcess integration.

We will use the same for ClaimRequestProcess Integration as well.

### Mapping Information

**Request (**All fields are mandatory)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Source Data Entity (Siebel)** | **Required** | **Target Data Entity(Elite WS)** | **Data Type & Length** | **Default Value** | **Comments** |
| CLM\_POLNUM | Y | MCAR\_OPERATION | VARCHAR(14) |  | Main Policy No. |
| CLM\_PLATE | Y | MCAR\_REQSEQNO | VARCHAR(50) |  | Car Plate No. |
| CLM\_DECLDATE | Y | MCAR\_POLNUM | VARCHAR(10) DD/MM/YYYY |  | Claims Declaration Date |
| CLM\_CLMDATE | Y | MCAR\_APPNUM | VARCHAR(10) DD/MM/YYYY |  | Claims Date |
| CLM\_LOCATION | Y | MCAR\_APPLIN | VARCHAR(150) |  | Location |

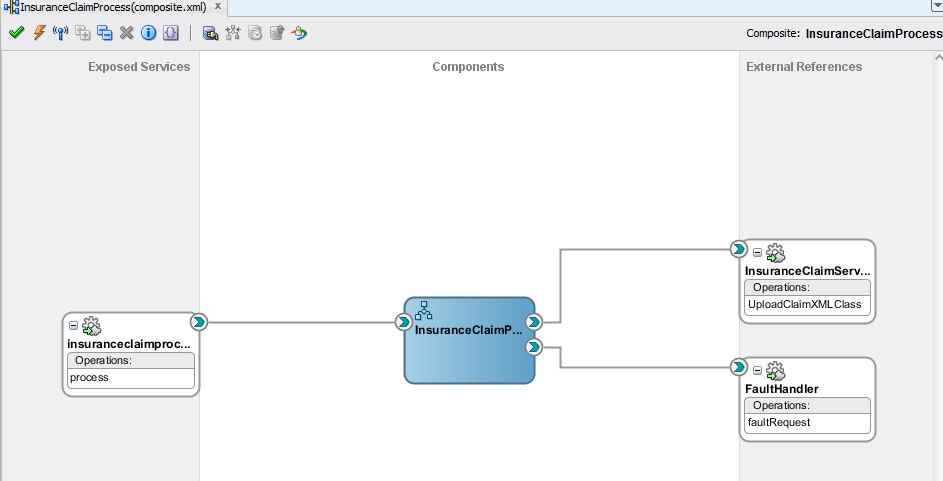
**Response**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Source Data Entity (Elite WS)** | **Required** | **Target Data Entity(Siebel)** | **Data Type & Length** | **Default Value** | **Comments** |
| CLM\_YEAR | Y | CLM\_YEAR | NUMBER(4) |  | Claim Year |
| CLM\_NUM | Y | CLM\_NUM | NUMBER(5) |  | Claim Number |
| CLM\_STATUS | Y | CLM\_STATUS | VARCHAR(1) |  | Status |
| CLM\_REMARKS | N | CLM\_REMARKS | VARCHAR2(2000) |  | Remarks |

## Technical Design Information

1. The intent of this section is to identify the table, columns, and source values that are required to support the above data elements. Refer to the Physical Database Design (IM.040), to identify the existing tables where the above attributes are located.

### Composite Diagram



### Technical Program Flow

* Get Input Payload from Siebel
* Pass it to BPEL Receive activity
* Validate Mandatory Fields
* After successful validation, Derive username, password, group Id from InsuranceAccessCredentials.dvm. Incase of any validation failure, reply to Siebel with valid Status and error message using

CLM\_STATUS and CLM\_REMARKS

Example dvm:lookupValue("InsuranceAccessCredentials.dvm","Field","userName","Value","userName")

* Transform Claim Request Payload to Elite WS
* Invoke the Elite webservice
* Get the response and assign back to Siebel.

## Error Handling Considerations

EMAD Standard Fault Handler Service is used to capture all system related failure/error information.

## Web Security Considerations

SOA Admin will receive the Elite WS URL along with username, password and Group ID.

The same will be configured in SiebelAccessCredentials.dvm which is in MDS.

1. The intent is to design the rules that are necessary to verify the format, length, relationships etc. of the attributes listed above.

## Performance Considerations

As per Oracle SOA Integration and EMAD standards.

# Integration Design for Claim Status Update

1. The intent of this section is to design the services between the components and the interfaces with external systems for each Use Case. Refer to DS.080 Software Component Design and focus on calling arguments (i.e., service signature) and logic definition.

The Integration service provides facility to receive Insurance Claim status and pass it to Siebel to record the insurance status.

## Business Rule Design

Elite Service (Third party) will use this integration Service to send the Insurance claim status

1. Receive the Input payload of Insurance Status from Elite

2. Validate and Transform the input to Siebel Service.

## Integration Design - Quick Reference

|  |  |
| --- | --- |
| SOA Project Name | **ClaimStatusUpdate** |
| Service Type | One Way |
| Source - Target System | Elite - Siebel |
| **Inbound SOA Service Design** | Input Payload Schema is attached in 7.3.1 |
| Outbound SOA Reference Design | Siebel Web service, Fault Handler |
| Frequency | On Demand |
| Volume | Single/Batch |

## Data Sources

### Request/Response Information

|  |  |
| --- | --- |
| **Type** | **Details** |
|
| Request |  |

### Domain Value Mapping(DVM)

None

### Mapping Information

**Request**

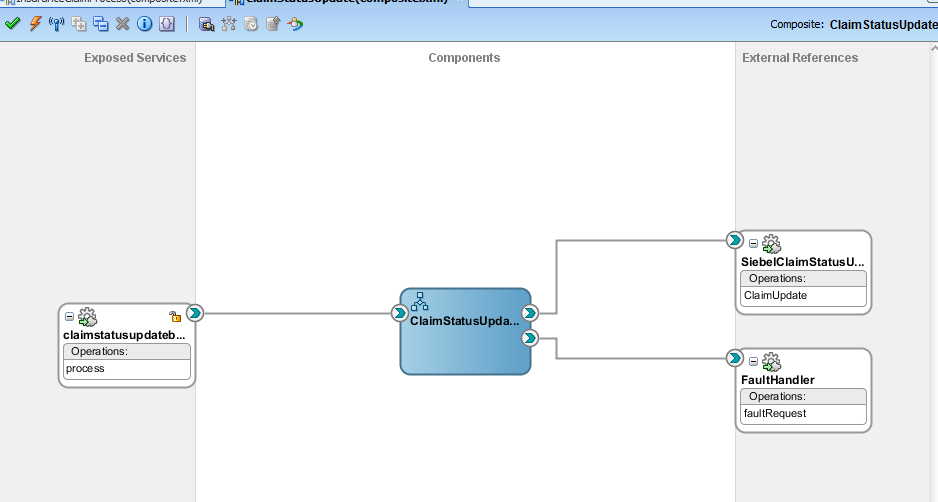
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Source Data Entity (EliteWS)** | **Required** | **Target Data Entity(Siebel)** | **Data Type & Length** | **Default Value** | **Comments** |
| CLM\_YEAR | Y | CLM\_YEAR | NUMBER(4) |  | Claim Year |
| CLM\_NUM | Y | CLM\_NUM | NUMBER(5) |  | Claim Number |
| CLM\_STATUS | Y | CLM\_STATUS | VARCHAR(25) |  | Status |
| CLM\_STATDET | Y | CLM\_STATDET | NUMBER(3) |  | Details Status |
| CLM\_REMARKS | N | CLM\_REMARKS | VARCHAR2(2000) |  | Remarks |

CLM\_STAUS and CLM-STATDET details are covered in Elite Technical Specification.

## Technical Design Information

1. The intent of this section is to identify the table, columns, and source values that are required to support the above data elements. Refer to the Physical Database Design (IM.040), to identify the existing tables where the above attributes are located.

### Composite Diagram



### Technical Program Flow

* Receive Claim Status Update Payload from Elite
* Transform to Siebel Service Input Variable
* Invoke Siebel Service to Send Claim Status.
* Handle the fault if any, and invoke standard Fault Handler Service.

## Error Handling Considerations

EMAD Standard Fault Handler Service is used to capture all system related failure/error information.

## Web Security Considerations

This integration service will be authenticated by user name and password from SOA Server.

The below WS security policy has to be attached to endpoint Service

**“Oracle/wss\_username\_token\_service\_policy”**

1. The intent is to design the rules that are necessary to verify the format, length, relationships etc. of the attributes listed above.

## Performance Considerations

As per Oracle SOA Integration and EMAD standards.

# Integration Design for Cancellation request

1. The intent of this section is to design the services between the components and the interfaces with external systems for each Use Case. Refer to DS.080 Software Component Design and focus on calling arguments (i.e., service signature) and logic definition.

The Integration service provides facility to Cancel the Opportunity(Contract) for a Customer. Here Siebel will act as Source system and EBS-OLFM will act as target system. The business functionality and validation rules are covered Siebel, OLFM Cancellation Design documents.

## Business Rule Design

Siebel will invoke the Opportunity Cancellation to cancel the Invoice, PO, Insurance and payment which was created earlier during Opportunity booking. To cancel this, Siebel would pass SOA Request Id, Customer Iqama and Contract Number to EBS. SOA Request ID will act as unique key between Siebel and OLFM

1. Receive the Input payload from Siebel along with SOA Request Id, Customer Iqama ID and Contract Number

2. Validate and Transform the input to EBS Concurrent Program

3. Invoke eBS Concurrent program to process the cancellation request.

4. Get the response from eBS about the Process Submission.

## Integration Design - Quick Reference

|  |  |
| --- | --- |
| SOA Project Name | **CancellationRequest** |
| Service Type | Synchronous |
| Source - Target System | Siebel - EBS |
| **Inbound SOA Service Design** | SOA WSDL Interface(Request and Response XSD) (attached in 4.3.1) |
| Outbound SOA Reference Design | Oracle Application Adapter, Fault Handler, Siebel Outbound Web Service |
| Frequency | On Demand |
| Volume | Single/Batch |

## Data Sources

### Request/Response Information

|  |  |
| --- | --- |
| **Type** | **Details** |
|
| Request |  |
| Response |

### Domain Value Mapping (DVM)

None.

### Mapping Information

**Request**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Source Data Entity (Siebel)** | **Required** | **Target Data Entity(eBS)** | **Data Type & Length** | **Default Value** | **Comments** |
| OrgId | Y | OrgId | NUMBER |  | eBS Organization ID |
| UserName | Y | UserName | VARCHAR2(100) |  | eBS User Name |
| ResponsibilityName | Y | ResponsibilityName | VARCHAR(100) |  | eBS Responsibility Name |
| InterfaceMode | Y | InterfaceMode | VARCHAR2(10) |  |  |
| InterfaceStatus | Y | InterfaceStatus | VARCHAR2(10) |  |  |
| SOARequestId | Y | SOARequestId | VARCHAR2(25) |  |  |
| CustomerIqamaId | Y | CustomerIqamaId | VARCHAR2(25) |  |  |
| ContractNumber | Y | ContractNumber | VARCHAR(100) |  |  |
| InsuranceCompany | N | Insurance\_Company | VARCHAR2(60) |  |  |
| Application Number | N | Application Number | VARCHAR2(25) |  |  |
| Application Sequence No | N | Application Sequence No | VARCHAR2(25) |  |  |
| Insurance Start Date | N | Insurance Start Date | VARCHAR2(11) |  | MM/DD/YYYY |
| Insurance End Date | N | Insurance End Date | VARCHAR2(11) |  | MM/DD/YYYY |
| Attribute1 | Y | SupplierCancelled? | VARCHAR2(1) |  | Value will be ‘Y’ or ‘N’ |
| Attribute2 | N |  |  |  |  |
| Attribute3 | N |  |  |  |  |
| Attribute4 | N |  |  |  |  |
| Attribute5 | N |  |  |  |  |
| Attribute6 | N |  |  |  |  |
| Attribute7 | N |  |  |  |  |
| Attribute8 | N |  |  |  |  |
| Attribute9 | N |  |  |  |  |
| Attribute10 | N |  |  |  |  |

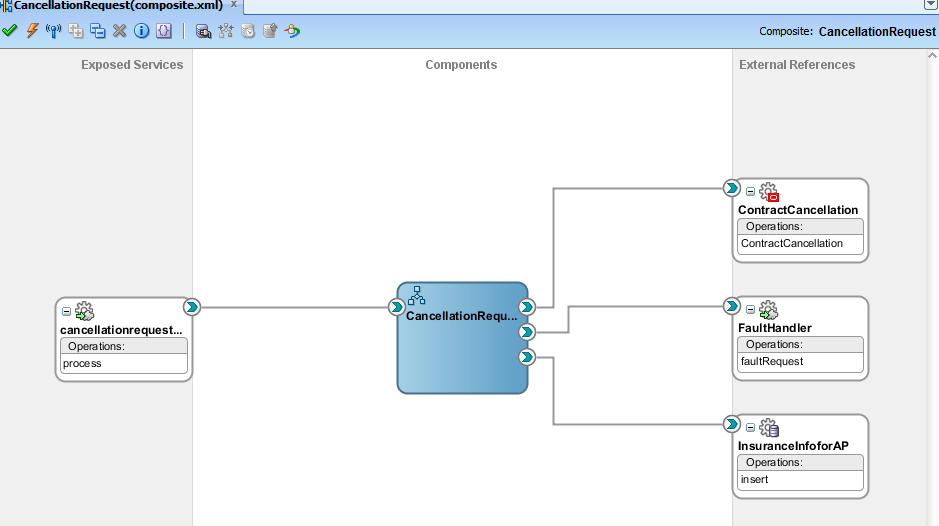
**Response**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Source Data Entity (Elite WS)** | **Required** | **Target Data Entity(Siebel)** | **Data Type & Length** | **Default Value** | **Comments** |
| SOARequestId | Y | SOARequestId | VARCHAR2(25) |  |  |
| CustomerIqamaId | Y | CustomerIqamaId | VARCHAR2(25) |  |  |
| ContractNumber | Y | ContractNumber | VARCHAR(100) |  |  |
| StatusCode | Y | StatusCode | VARCHAR(1) |  |  |
| StatusDescription | N | StatusDescription | VARCHAR(2000) |  |  |

## Technical Design Information

1. The intent of this section is to identify the table, columns, and source values that are required to support the above data elements. Refer to the Physical Database Design (IM.040), to identify the existing tables where the above attributes are located.

### Composite Diagram



### Technical Program Flow

* Get Input Payload from Siebel
* Pass it to BPEL Receive activity
* Payload should contain the XML Data elements in order as per business rule design in 7.1
* Validate all required fields and throw an error message If any of the required field is blank
* Transform Input Payload to Oracle Apps Adapter Input
* Invoke Oracle Apps Concurrent Program(AMG Raya Cancellation Program) for Cancellation Request through Oracle Apps Adapter .
* Insert a Insurance record(Applincation Sequence Number, Application Number, Insurance Company, InsuranceStart and Insurance End date in XXAMG\_RY\_AP\_INVOICES\_STG\_T with status ‘OPEN’

|  |  |
| --- | --- |
| Integration | OLFM(XXAMG\_RY\_AP\_INVOICES\_STG\_T) |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

* Get the response and assign status back to Siebel.

## Error Handling Considerations

EMAD Standard Fault Handler Service is used to capture all system related failure/error information.

## Web Security Considerations

As per Oracle SOA Integration and EMAD standards.

1. The intent is to design the rules that are necessary to verify the format, length, relationships etc. of the attributes listed above.

## Performance Considerations

As per Oracle SOA Integration and EMAD standards.

# Integration Design for Cancellation Status

1. The intent of this section is to design the services between the components and the interfaces with external systems for each Use Case. Refer to DS.080 Software Component Design and focus on calling arguments (i.e., service signature) and logic definition.

The Integration service provides facility to Send the Status of Opportunity Cancellation request from Oracle eBS to Siebel on each Step such as Invoice, PO, Payment and Insurance Cancellation of requested Customer. Here Oracle EBS will act as Source system and Siebel will act as target system. The business functionality and validation rules are covered Siebel, OLFM Cancellation Design documents.

## Business Rule Design

OLFM will raise a business event at end of each step in the cancellation process.

1. Receive the event through Oracle Application adapter along with SOA Request Id, Interface Code, Contract Number and Status

2. Validate and Transform the input to to Siebel outbound

3. If the Interface code is XXAMG\_RY\_RTV\_PKG, then invoke Insurance cancellation request and update the response in Ap invoice interface staging table XXAMG\_RY\_AP\_INVOICE\_STG\_T for the application sequence number and application number of Insurance based on SOA request Id

4. Send response to Siebel with Step code and Status.

## Integration Design - Quick Reference

|  |  |
| --- | --- |
| SOA Project Name | **CancellationRequest** |
| Service Type | Synchronous |
| Source - Target System | Siebel - EBS |
| **Inbound SOA Service Design** | SOA WSDL Interface(Request and Response XSD) (attached in 4.3.1) |
| Outbound SOA Reference Design | Oracle Application Adapter, Fault Handler, Siebel Outbound Web Service |
| Frequency | On Demand |
| Volume | Single/Batch |

## Data Sources

### Request/Response Information

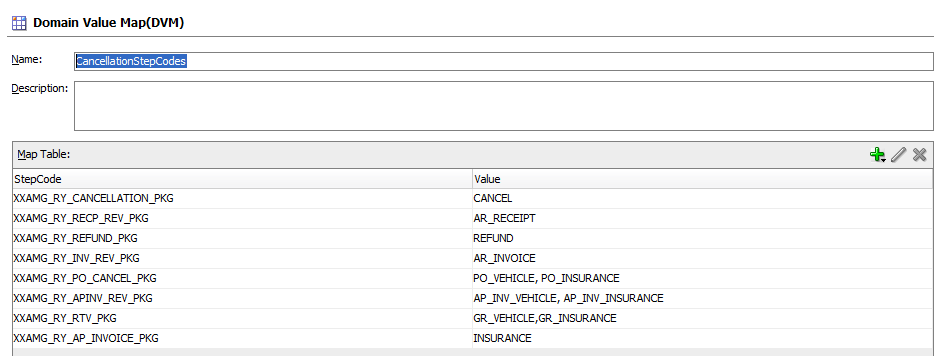
|  |  |
| --- | --- |
| **Type** | **Details** |
|
| Request |  |
| Response |

### Domain Value Mapping (DVM)

1) DVM: SiebelOutboundNotifyCredentials.dvm will be used to access credentials and pass it to Siebel Outbound Webserivce while responding back.

This is available in MDS.

2) DVM: CancellationStepCodes.dvm will be used to capture the step code(stage code) of cancellation process like below.



### Mapping Information

**Request:**

A Oracle Business event(XXAMG.RY.CANCELLATION.EVENT) will be fired at end of each step and the same will be captured through Oracle Application Adapter with below parameters.

Contract\_Number

Interface\_Code ( Step Code)

Status

SOA Request ID

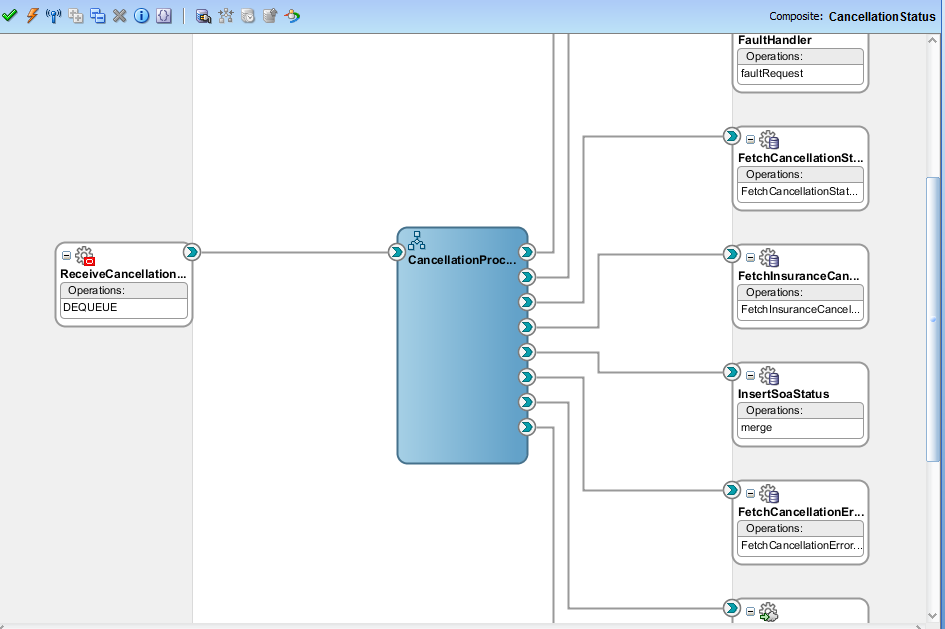
**Response**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Source Data Entity (OLFM)** | **Required** | **Target Data Entity(Siebel)** | **Data Type & Length** | **Default Value** | **Comments** |
| SOA\_RQST\_ID | Y | INTERFACE\_STAGE\_ID | VARCHAR2(60) |  |  |
| CONTRACT\_NUMBER | Y | CONTRACT\_NUMBER | VARCHAR2(120) |  |  |
| ‘Cancellation\_After\_Payment’ | Y | NOTIFICATION\_TYPE | VARCHAR(50) |  |  |
| INTERFACE\_CODE | Y | INTERFACE\_CODE | VARCHAR(30) |  |  |
| INTERFACE\_STATUS\_CODE | Y | INTERFACE\_STATUS\_CODE | VARCHAR(1) |  |  |
| ERROR\_MSG |  | ERROR\_MSG | VARCHAR2(500) |  |  |
| ATTRIB1 |  | ATTRIB1 | VARCHAR2(150) |  |  |
| ATTRIB2 |  | ATTRIB2 | VARCHAR2(150) |  |  |
| ATTRIB3 |  | ATTRIB3 | VARCHAR2(150) |  |  |
| ATTRIB4 |  | ATTRIB4 | VARCHAR2(150) |  |  |
| ATTRIB5 |  | ATTRIB5 | VARCHAR2(150) |  |  |
| ATTRIB6 |  | ATTRIB6 | VARCHAR2(150) |  |  |
| ATTRIB7 |  | ATTRIB7 | VARCHAR2(150) |  |  |
| ATTRIB8 |  | ATTRIB8 | VARCHAR2(150) |  |  |
| ATTRIB9 |  | ATTRIB9 | VARCHAR2(150) |  |  |
| ATTRIB10 |  | ATTRIB10 | VARCHAR2(150) |  |  |

## Technical Design Information

1. The intent of this section is to identify the table, columns, and source values that are required to support the above data elements. Refer to the Physical Database Design (IM.040), to identify the existing tables where the above attributes are located.

### Composite Diagram



### Technical Program Flow

* 1. Receive the business event : XXAMG.RY.CANCELLATION.EVENT through Oracle Application Adapter
  2. Loop through each element and get the value of SOA Request ID, Contract Number, Interface Code and Status
  3. If Status is not ‘0’, then retrieve the error message from XXMAG\_RY\_ERR\_STG\_T by using SOA Request Id and send that to Siebel.
  4. If Status is 0, then verify the Interface code as “XXMAG\_RY\_CANCELLATION\_PKG’ and then check the Insurance Process is completed in XXAMG\_RY\_AP\_INVOICE\_STG\_T (status code=’PROCESSED’) using SOA Request Id.
  5. If the above is success, then send the response to Siebel as “Opportunity Cancellation is completed’, otherwise the send the response as ‘Waiting for Insurance Cancellation’.
  6. When the interface code is ‘XXAMG\_RY\_RTV\_PKG’, then invoke the insurance cancellation using existin web service: Insurance request process using Operation Type ‘C” with Application Sequence Number and Applicaton Number which was stored in XXMG\_RY\_AP\_INVOICE\_STG\_T using SOA Request ID.
  7. Once Insurance Request Cancellation is done, then update the response back to XXAMG\_RY\_AP\_INVOICE\_STG\_T table with SOA Request ID in attribute4, attribut5.
  8. Assign the response to Siebel

## Error Handling Considerations

EMAD Standard Fault Handler Service is used to capture all system related failure/error information.

## Web Security Considerations

SOA Admin will receive the Elite WS URL along with username, password and Group ID.

This will be configured through SiebelAccessCredentials.dvm in SOA Server.

1. The intent is to design the rules that are necessary to verify the format, length, relationships etc. of the attributes listed above.

## Performance Considerations

As per Oracle SOA Integration and EMAD standards.

# Quality of Service Design Considerations

1. The intent of this section is to document the design considerations for the component regarding the non-functional requirements. Refer to the Supplemental Requirements (RD.055). If the Software Component Design (DS.080) is not available, use the sections below. Add additional sections for various Supplemental Requirement types.

## Restart Strategy

* As per EMAD Standards.

## Crash Recovery

* As per EMAD Standards

## Security

* As per EMAD Standards

## Performance

* As per EMAD Standards

# Quality of Service Design Considerations

1. The intent of this section is to document the design considerations for the component regarding the non-functional requirements. Refer to the Supplemental Requirements (RD.055). If the Software Component Design (DS.080) is not available, use the sections below. Add additional sections for various Supplemental Requirement types.

## Restart Strategy

* As per EMAD Standards.

## Crash Recovery

* As per EMAD Standards

## Security

* As per EMAD Standards

## Performance

* As per EMAD Standards

# Installation Considerations

1. Add to or modify this list as appropriate. Provide additional details where necessary to facilitate the creation of the installation routines.

Installation scripts must be prepared to perform the following actions in an automated way:

1. .

# Open and Closed Issues

1. Add open issues that you identify while writing or reviewing this document to the open issues section. As you resolve issues, move them to the closed issues section and keep the issue ID the same. Include an explanation of the resolution.  
     
   When this work product is complete, any open issues should be transferred to the project- or process-level Issue Log (Manage focus area) and managed using a project level Issue Form (Manage focus area). In addition, the open items should remain in the open issues section of this work product, but flagged in the resolution column as being transferred.

## Open Issues

| ID | Issue | Resolution | Responsibility | Target Date | Impact Date |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

## Closed Issues

| ID | Issue | Resolution | Responsibility | Target Date | Impact Date |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

# Appendix

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Doc. S.No | Component | Document Description | Reference | Comments |
|  |  |  |  |  |