SYSTEM DELIVERABLE

System Ref.	**AGCC**
System Name	**Delivery Service**

AGCC - DELIVERY SERVICE

FUNCTIONAL SPECIFICATIONS – UNIT PROCESS

TABLE OF CONTENTS

DOCUMENT CONTROL

Document Control: Revision History

Revision	Author	Date	Description
0.1	Sudheer Kumar		Main Structure
0.2	Manikandan		
0.3	Kumaravelu Gopal	16/02/2021	RBICCNA-3019 : DEV - Use new email body.
0.4	Kumaravelu Gopal	25/02/2021	RBICCNA-3205 DEV - Axon - Put communications with owner '01'

			records in Manual, RBICCNA-3206 DEV - Put communications AG7K to manual.
0.5	Kodavati Sai Sireesha	28/06/2024	Updated as per the code

1. INTRODUCTION

This document describes delivery service.

The main advantage of this document is to understand the internals of delivery service. For each document, delivery should be happen in the AGCC application.

The Document will focus on below topics:

- Document delivery.

2. HIGHLEVEL DESCRIPTION

This document describes delivery service.

In high level, for each document, for each document, delivery should be happen in the AGCC application.

3. DETAILED DESCRIPTION

3.1 Delivery Service:

Delivery Service is used to deliver the document for each communication. In AGCC application, there is a module called Delivery Service which is used for the delivery of the document for each communication.

Delivery Service has a implementation, which will deal with the delivery of the document, for delivery of the document we need to provide Deliver Request, which contains below information.

- The deliver header.
- The communication type.

Again the deliver header included below information:

- The event flow.
- The output format.
- The store pages.
- The archive flag for archive the document in digis or not.
- The origin type.
- The language.
- The generator client.
- The user.
- The Channel.
- The sub channel.
- The total email.

Communication type is included below information:

- The external ID.
- The internal ID.
- The bundle group.
- The object reference.
- The isAutoReply.
- The preference.
- The masterSwitchCode.
- The email address.
- The number of copy.
- The isLegalRepresentative.
- The isPimsEfficy.
- The isSubjectFormatted.
- The isFirstDocObjectFilled.
- The address Type.
- The email number etc.

  Delivery Service returns Deliver Response which includes below information.
- The communication external id.
- The archive id list.
- The renewal.
- The communication type.

- In deliver process , we will get communication type from the Deliver Request.
- Get Delivery Header from the Deliver Request again.
- Get work Flow from the Delivery Header.
- Set auto-reply in communication type, if work flow is auto-reply.
- Get user Type from the Delivery Header.
- Get event Flow from session data.
- Get Delivery Type from communication type.
- Set Communication Id from get External ID of communication type.
- Get Lang from the Delivery Header.
- Get Logo from the Delivery Header.
- Get Origin from the Delivery Header.
- Get User Type from the Delivery Header.
- Get Audit by using User type.
- Validate the Deliver Request.

3.1.1 Over rule sub channel for DVV:

- ➢ Get channel Type, sub Channel Type from communication type.
- ➢ If brand is DVV and channel type is LOCAL PRINT and sub Channel Type is EVENT then set Sub Channel of delivery in communication type as PRINT.
- ➢ If brand is DVV and channel type is EMAIL, and sender is null, then create party type as Email Address Type and set email as “noreply@dvvlap.be”.

-If Origin is Efficy and addressee is not null and external Id of addressee is not null then set Identifier Type as PSI code in addressee.

3.1.2 Registering or Re-Initializing communication:

- Get the communication from Communication table,
- If communication is null from data base, Registering communication.
- Else if communication entity of communication status is error then update Status in communication table.

- Else if communication status is already delivered then in deliver response set Communication External Id, set communication type and return deliver response.
- Else Communication is in intermediate status which is applicable for auto reply only.
- Get id from communication table and set in communication type id.
- Set Creation Time value in communication type.

3.1.3 Updating communication Thread table:

- First get communication from communication thread table.
- If communication is not null then get communication Thread Order and get Communication Thread ID.
- Else Insert communication into Communication Thread table.

Looping through all archived documents.
Get archive Uri from each Document Type.
If archive Uri is not blank :

- get extract Type Id from each Document Type.
- get extract External Id from each Document Type.
- get Document from table using external Id.
- If Document is null then Creates the document from meta data.
- Else get status from Document ,then throw Deliver Service Exception.
- If Document is Non Functional Document then set Contains Only Functional Document to false.
- Set Contains Only Email Document.

Else if channel is Messaging Channel and is Payload Data Available or is Email Body.

- If object and product code not null, then get product code.
- Build Document From Doc Type and store in Document Entity.
- Fill Subject only if Channel - Email and Sub Channel - 13.
- Get payload from the document type.
- If payload not null then get email Content from payload.
- If email Content is not blank, get Payload Content Buffer from payload is of Content Buffer then store content buffer in email Content.

If document entity is not null,

- then add communication document to communication document entry list.
- Add document entry also to document entry list.
- For document type set Document Order.
- Add all document entries in document Order Map ,add external id and increment order count are key value pairs.
- If object entity is null and document owner is AGCC then get object entity from document entity and store in object entity.
- Update document Type model.(Populate the data base values to model to reduce the multiple DB calls)
- If document Order is zero, is To Be Included then set Policy Holder, set Role Within Policy Or Claim, set Intermediary and create Bank Event User Type Generator using generate method.
- Store first Document Entity as document entity and first Document as document type
- Set Addressee Policy Holder in deliver Response.
- Else skip the 1st and making 2nd documents as 1st.

If document Type status not null then,

- Get document Status from document type.

- If document Status is equal to AUTO_REPLY_NOK and is Event Flow Register Communication and auto Reply Update Document is true then make auto Reply Update Document to false , update the status in communication table with EXTENSION_NOK_DOC, and also set communication as EXTENSION_NOK_DOC
- Else if document Status is equal to AUTO_REPLY_OK and is Event Flow Register Communication and auto Reply Update Document is true then make auto Reply Update Document to false , update the status in communication table with EXTENSION_OK_DOC, and also set communication as EXTENSION_OK_DOC.
-

If document type is To Be Included,

- Increment document order count.

3.1.4 Over Rule BHCP Channel :

- Iterate each document from document entry list,
- Get DocType from each document.
- If OutGoingDocTypeTemplate of DocType is not empty,
- If getOutGoingDocTypeTemplate size is 1 iterate OutGoingDocTypeTemplate.
    - If logo is blank in email body save template digital option.
    - Else
        - If OutGoingDocTypeTemplate logo and Document logo are same
        - If Product Code is blank, then in email body save template digital option.
    - Else
        - If OutGoingDocTypeTemplate Product Code and Document Product Code are same , then in email body save template digital option.
    - Else if getOutGoingDocTypeTemplate size greater than 1,
        - Get OutGoingDocTypeTemplate from table, if it is null then save in OutGoingDocTypeTemplate.
        - If OutGoingDocTypeTemplate is not null then get mail body from template digital option.
    - If channel type is EBIB and mail body is not empty and index is equal to 0, then
        - Get sub Channel Type from communication type.
        - If it is Mail Body
            - Is Sub Channel Belins then Scriptura creates bank mail body.
            - Else throw with error message "Sub channel 13 is expected".
        - Else if sub channel is bank scriptura then set Bank Mail Created By Bank to true.
        - Else origin Type is QIS, set channel is ARCHIVE.
        - Else throw error message as "Communication doesn't contain any Bank mail document"
    - If channel type is EBIB and mail body is not empty and index is equal to 0 and origin is QIS then
        - set channel is ARCHIVE.

3.1.5 Overrule Communication Channel :

- Get list of overrule RB Docs, from configuration.
- Get list of overrule Zhivago Docs from configuration.
- Get channel Type, sub Channel Type from communication.

- Check is Bank Mail From Axon.
- Iterate Document entity list.
    - If document is not null then
        - Get OutGoingDocTypeTemplate from docType and is not empty.
        - If OutGoingDocTypeTemplate size is 1 then for each OutGoingDocTypeTemplate, check logo is blank, then mail body is get from template digital option.
        - Else if logo is present and product code is blank then mail body is get from template digital option.
        - Else Product Code is equal from both OutGoingDocTypeTemplate and Document then mail body is get from template digital option.
- Else if getOutGoingDocTypeTemplate size greater than 1,
    - Get OutGoingDocTypeTemplate from table, if it is null then save in OutGoingDocTypeTemplate.
    - If OutGoingDocTypeTemplate is not null then get mail body from template digital option.
- Get document External Type from docType.
- If it is from index =0 and is Bank Mail From Axon, then check if overrule RB Docs contains document External Type or mail body is no.
    - If document Entity List size is 1 then document External Type is RB2023, and Creator User ID is "createClaimHistoryCertificates" then stop.
    - Set Channel to ARCHIVE.
- Else if ZHIVAGO_CHECK.
- Loop through all the Zhivago docs and overrule.
- Check is Health Connect, channel Type should be health connect and sub channel type is Zhivago.
- Check overrule Zhivago Docs contains document External Type or is Health Connect then call Zhivago Delivery Service then over Rule To Zhivago Channel.
- Set Skip Messaging Document if it is central print or sub channel is B Post then stop.

- Overrule of channel to Central Print when total size of documents in communication exceed the Maximum Limit**.**

- If origin type is QIS then call Auto Reply Service then update Uudoc Type.
- Registering Communication Document and Document Instance in data base.

3.1.6 Deliver Renewal Communication:

Check list of communication documents to be is Renewal Communication.

If communication document is In Error then update status in communication table with ERROR. In deliver response set Communication External Id and also set Communication, finally return deliver response.

Else set Renewal flag to true in deliver response, set renewal in communication, set Communication External Id and set communication also. Update Status to PROCESSING_PENDING, return Deliver Response.

If deliver response is renewal and communication type is error then stop renewal, return deliver response.

-If object entity is null and object is no null from document Entity List and Identifier is not null.

3.1.7 Builds the object entity:

Get policy Holder, addressee from document Type.
Declare originTypeEnum.
Populate a new object entity.
Get mutation Type and set mutation Type in object entity.
Set Endorsement Number in object entity.
Interpretation of object into archive level.
Case : When object Type is CLAIMS,POLICY & POLICYGROUP set Archive Type, set Archive External Id.
Case : When object Type is PROPOSAL, get predefined Policy Number from common data.
If predefined Policy Number is not null and Sign Request Document Identifier then set Archive Type as POLICY, set Archive External Id.
If Predefined Policy Number Exists then set Archive Type as POLICY.
Else set Archive Type as PERSON.
If is Origin I series set Archive External Id,
Case : QUOTE,PERSON set Archive Type, get policy holder, set Archive External Id.
Set Segment in object Entity.
Set Audit in object Entity.
After building object Entity, check whether it is existed in data base or not, if it returns null then add into object table,
Create ObjectType set value, identifier, archive object then return.
Case : When object Type is EXTERNAL_POLICY_NUMBER set Archive Type, set Archive External Id.

3.1.8 Builds the Document from the Doc Type:

Below the fields are consider as inputs for build document.

Document Type, Communication Type, Audit, Product Code, Object Entity, Document (First Doc Entity).

Step 1: Get the channel and sub channel from the communication type.

Then Get the Origin from the session and Doc system key is message format ("origin.{0}.agccdocsystem" with Origin Code).

Step 2: **If** channel type is **EMAIL** and sub channel is not **BELINS_SCRIPTURA(13)** or **BELINS_SCRIPTURA_SIMILAR(14).**

- Determines the document external type (from the document Type table) based on the archive Object Type (Object table)
  - If , Object empty then document external type value is "**EMAILBODYCUSTOMER**"
  - Otherwise, If the case archive Type is **CLAIMS** then document external type value is "**EMAILBODYCLAIM**"
  - OR the case archive Type is **PERSON** then document external type value is "**EMAILBODYCUSTOMER**"
  - OR the case archive Type is **POLICY** and origin is "**AG58**" then document external type value is "**EMAILBODYPOLICYLIFE**"

        - OR the case archive Type is **POLICY only**, then document external type value is "**EMAILBODYPOLICYLIFE**"
    - return this doc external type value for email body document.

Step 3: **Otherwise**, If channel type not **SMS** and external type is not empty, then doc external type value assign from document type object.

- Otherwise, If **UUDOCTYPE** is not empty, then assign the doc external type value is **UUDOCTYPE**.

- Otherwise throw the Deliver Service Exception "**Document externalType is mandatory**".

- If UU Doc type is not empty, Gets the document parameter by passing uudocTypeid from DocType table.
    - **Query**: select * from docType where uuDocTypeId=? and validFromDate=? order by validFromDate asc; (fetch first single record).

Populate the template : Its return the **OutGoingDocTypeTemplate** record, based on the id, logo and product code from the "**outgoingdoctypetemplate**" table.
Otherwise, If doc external type value not empty, Get the document parameter by passing uudocTypeid from **DocType** table.
Otherwise, Get the document parameter is "**UNKNOWN**".

Create document :
create the document Passing following inputs: **UuDocTypeId, DocType, DocSubType , Logo, generate, UUID ("UUID("02", "arch://uuid/")"), Language, AGCC("00"), Document status (INITIAL(10)), CONSTANT_00,docSystemKey,CONSTANT_00, FLAG_Y, origin.**

Sample Code:

```
        Document document = new Document(docParam.getUuDocTypeId(),
docParam.getDocType(), docParam.getDocSubType(),
        docType.getLogo(), uuidService.generate(), DocumentTypeEnum.UUID,
docType.getLanguage(),DocumentOwnerEnum.AGCC,
DocumentStatusEnum.INITIAL, CONSTANT_00, getValue(docSystemKey),
        CONSTANT_00, FLAG_Y, originType.getOriginCode());
```

➢ Get the payload and do the enrichment perform:

- Get the payload, If payload not empty, then get the email body for enrichment.

- If email body and MyDvv Sign Request (If channel is email and sub channel is Belins scriptura (13)) both are true, then proceed the enrichment process.

        - If we have party role 12 then do the enrichment else, skip the enrichment and skipping enrichment temporarily for CONFIRMCL,CMBCSCL and do the perform enrichment then return the enrichment response.

- Get the **AGENT**, **EMAILLADDRESSE**, **ADDRESSEE** and **POLICYHOLDER** from the enrichment response and set into the Document Type.

- Get the common data passing input as Document Type and Doc Type.

- Get the common data content buffer get from the pay load and append with content buffer then assigned to message content.

- **Write email ORSMS to file** passing input as document external id and message content.

➢ Set the below the fields for create Document:

- isMyDvvSignRequest is true (if channel is email and sub channel is Belins scriptura(13)) then set the **SourceExternalId**,**IsAddresseePolicyHolder**, based on **policyHolder** (13) then register the object.
- Set following fields in to document object :Origin, Direct Debit, Audit,
    - Address, Party, Object, Product Code, DocType and GenerationTimestamp.
- if ((**UUDOC_TYPE_276**.equals(UuDocTypeId) OR **UUDOC_TYPE_277**.equals(UuDocTypeId) OR
  UUDOC_TYPE_278.equals(UuDocTypeId) AND firtstdocEntity != null)

    - Set following fields in to document object : RoleWithinPolicyOrClaim, AgencyIdentifier, AgencyIdentifierType, Logo, Language, Object, ProductCode, IsAddresseePolicyHolder.

➢ **Once set above all fields in the document object then Create the new document in the document table.**

- If ((**UUDOC_TYPE_276.equals**(UuDocTypeId) OR **UUDOC_TYPE_277.equals**(UuDocTypeId) OR
  UUDOC_TYPE_278.equals(UuDocTypeId) {

    - If **ConfidentialAccess** is true, then Creating additional data [FieldName:{} FieldValue:{}] in **DocumentAdditionalData** table.

    - If **MedicalAccess** is true then Creating additional data [FieldName:{} FieldValue:{}] in **DocumentAdditionalData** table.

  }
- Set the **ArchiveID**("arch://uuid/" + ExternalId) and **EmailBody** is true in Document Type object.

➢ Finally , Build the document from the doc type.

➢ If any error occurred meanwhile, then return the DeliverServiceException message("Exception occurred during mailbody/email/sms document creation").

3.1.9 Build Parent Receiver :

Get channel type, addressee.
Create Receiver.
If addressee is not null,

- Create Persistable Party And Address Entity Populator, set user type.
- Get PartyEntityResponse from populator.
- Iterate Address from PartyEntityResponse.
    - Email address needs to be set when the channel is Email.
    - If channel type is EMAIL, then set Receiver Address.
    - Else address type is SOPHARTY then set Receiver Address.

- Set is logical, is physical, PartyRoleType as Addressee in Receiver and return.

If channel type is Email, then call inform ex Handler and my bo Handler.
Get build Receiver List.
If parent Receiver is not null then build Receiver List is not empty, and communication is Routing Through Intermediary then set Is Physical to N.
Set Communication in parent Receiver.

If is-Restart flag is true then create Receiver Decider, if existing Receiver is null create Receiver in data base. Else set Id to parent Receiver from existing Receiver id.
Else create parent Receiver entry in data base.

Finally update communication.

If messaging Document is not null, then for messaging Document, set Party, set Address, set Object and then update messaging Document in Data base.

If channel type is Central Print and parent Receiver address is not null, and also communication sub channel is null then set sub channel is b post for delivery.

If and parent Receiver address is SOPHARTY, then set sub channel as SOPHARTY.

If channel type is Central Print and is Address Sheet Required, Check if address sheet has been already created or not if it is not created, generate Address Sheet Document Request.

Once we call central print service for address sheet generation, it will give response, if the response does not contains any error, we will get address sheet document from document table using response given by external id.

Else throw error Address sheet generation failed in Scriptura.

If generated address Sheet is not null, then create communication document, then add it to communication Document Entity List.

Add Address sheet should be first in the list.

If receiver List is not empty and parent Receiver is not null, iterate receiver List, set communication for each receiver, set Parent Receiver, set Audit from Parent Receiver.

If is Restart flag is true then create Receiver Decider, if it is null then in table add Receiver Decider, else add id to receiver, else create receiver in receiver in table.

3.1.10 Registering bundle:

Determine Is Claim Object is true or false.

Determine the Is Document owner extern is true or false.

If addressee not null and address external id is not null, then get addressee PSI.

Populate bundle. Always create new bundle in table irrespective of bundle status.

One of the field, Manual Document Addition flag will set below the condition in to the Bundle table:

If the condition Is Origin Axon , Is Object Claim, and Is Document owner extern OR Event flow is register-communication OR Event flow is delivery and Is Claint then set the Manual Document Addition Flag is "Y" , Otherwise, Flag is "N".

Add extra documents for Central Print.

Register document instances in data base.

Document should be available in Scriptura's cache for central print.

Update status in bundle and document instance in data base.

Populating document additional data from DB.
While spilt of mails when we come across the "mail too big"
If query gives result

- Then 1 or more documents has more than 8MB (DocSize from DOCUMENT table > 8000000)
    - Then Externalid of COMMUNICATION table is 'mail too big' case
        - Then CommunicationExternalID out of Renewal process
        - Set the communication status to 83
        - SO during renewal the communication with status 99 will be only picked
        - Response send should be NOK

3.1.11 Trigger channel based delivery:

Get origin from session data.
Get channel type from communication type, follow the below steps if channel type is matched.

Case : **LOCALPRINT:**
Check sub channel required local print, create Local Print Request, call local Print Delivery Service, get Local Print Response.

If Local Print Response is not null, iterate over Local Print Response Model ,store the document id and file name from Local Print Response Model in map and it to communication type.

If local print response has any error set communication type status code as FAILURE and throw Communication failed in Local Print delivery.

Else set status of communication type as SUCCESS and stop.

Case : **FOD, ESERVICES, ARCHIVE, CENTRALPRINT, HEALTHCONNECT, PUSHNOTIFICATION**

For all the above cases set status as of communication type as SUCCESS and stop.

Case : **EBIB**

Get List of Document Type from communication type.

If list of document type is not empty, iterate over document type list.

If document not contains email body, increment total Documents Attached.

If total Documents Attached are greater than DOCUMENTS ATTACHED LIMIT FOR BANKMAIL then stop.

Set Status as FAILED in communication type.

Throw error message: Communication failed in Bank mail delivery as total number of documents attached is beyond maximum limit

Else set status as of communication type as SUCCESS.

case **SMS :**

create SMS Deliver Request, call **SMS** delivery service and get SMS Deliver Response.
If sms response not null and the value of sms response is SUCCEESS,

Set Status as FAILED in communication type. Throw error message : Communication failed in SMS delivery, else set status as of communication type as SUCCESS.

Case **EMAIL**:

Create Email Request, iterate document Entity List, create Email Document Request, call email Delivery Service, if email Delivery Response is not null and it has error then Set Status as FAILED in communication type, then throw Communication event failed at email delivery. else set status as of communication type as SUCCESS.

End.

For External policy reference, update object data for communication event in case of External policy.

For Channel base bank event, call Bank Event Decider.

Checks if it is delivered for Local print, Event and print.

3.1.12 Trigger Channel Based Bank Event:

case **FOD:** set communication type status as SUCCESS.

Case **EBIB**: check the flag is-Bank-Mail-Needed is true,create Bank Email Event Request. Call deliver Bank Mail Event service and get IBankMailEventResponse, if it has any errors then set communication type status as FAILURE, throw error message as Communication event failed at bank event delivery. Else set communication type status as SUCCESS.

Case **Health Connect:**

Zhivago delivery for bank we use this case, create HealthConnectEventRequest and call deliverHealthConnectEvent service then get IHealthConnectEventResponse. , if it has any errors then set communication type status as FAILURE, then throw error message as Communication event failed at health connect delivery. Else set communication type status as SUCCESS.

Case **ARCHIVE,EMAIL,SMS,CENTRALPRINT,LOCALPRINT,PUSHNOTIFICATION, ESERVICES:**

This case is for Normal Bank event, create BankCommunicationEventRequest, call deliverBankCommunicationEvent and get BankCommunicationEventResponse. if it has any errors then set communication type status as FAILURE, then throw error message as Communication event failed at bank mail delivery. . Else set communication type status as SUCCESS.

If any exception occurred while trigger channel based then throw error message as Exception occurred in bank event.

End.

Change the document visibility in Digis, for this call perform Confirm Document method is used to set the document visible in Digis Tree.

If channel type is FOD then call FOD Delivery Service for store Additional Data.

When we get any exception related to communication event service then throw error message as : Exception occurred in communication event Service.

When any exception occurred while calling delivery, then throw error message as : Exception Occurred in Delivery Service.

When any exception occurred while calling central print, then throw error message as : Exception Occurred in Central Print Delivery Service.

If any other exception occurred , then throw Error occurred while performing DB operation.

Finally set Communication External Id, set Communication in deliver response and . Return deliver response.