**Document Attributes**

| **Attribute** | **Value** |
| --- | --- |
| Application ID /  Application Name | 296357a - |
| Owner | Ravali Yemineni |
| Owner Contact Information | +91 9686501189 ry553f |
| Other Attribute |  |

**Revision History**

The following table lists the revision history of this document:

| **Author** | **Date** | **Version #** | **Revision Description** |
| --- | --- | --- | --- |
| Ravali Yemineni | 08/09/17 | 0.01 | Initial draft for 296357a |
| Ravali Yemineni | 08/30/17 | 0.02 | Updated after official review with SalesExpress – Change summary:   1. InquireCsatResults Get method : internal logic changed on display of open/ close. |
| Ravali Yemineni | 09/15/17 | 0.03 | Changed the naming convention of possible values for searchField in CsatList get method and recordSorting as below  ‘SURVEY\_DATE’  ‘SURVEY\_ID’  ‘SURVEY\_STATUS’  ‘SERVACTPLAN’ ( this is included for recordSorting along with the above one’s) |
| Ravali Yemineni | 09/21/17 | 0.04 | Striked for archiving statement when we delete a survey.As this is expected to be part of 1802 release. |
| Ravali Yemineni | 10/09/2017 | 0.05 | Descoped api method QuestionActionPlan Delete as per CR 172983 |
| Ravali Yemineni | 01/16/2018 | 0.06 | ST Defect 397074 updated under CsatData Post operation. |

**Table of Contents**

[**Overview** 4](#_Toc351987618)

[**Problem Statement** 4](#_Toc351987619)

[**Design Decisions** 4](#_Toc351987620)

[**Alternative Designs** 5](#_Toc351987621)

[**Assumptions/Risks** 5](#_Toc351987622)

[**Traceability Matrix** 12](#_Toc351987623)

[**Pre-Production Disaster Recovery Planning** 13](#_Toc351987624)

[**Other Plans and References** 13](#_Toc351987625)

[**Acceptance & Approvals** 15](#_Toc351987626)

[**Appendix A: JMS Requirements** 17](#_Toc351987627)

[**Public Queues:** 17](#_Toc351987628)

**Overview**

In order to provide a complete view of a customer/account from pre-sale to delivery, Customer’s order/project detail data need to be added into customer database and integrated with GDB. Customer Database should be accessed by APIs with a Microservice oriented GUI for data updates and retrieval.

**Problem Statement**

This HLD is about everything that is needed to implement microservice ManageCSAT.

**Design Decisions**

### Database

#### HLD-296357a-CSAT-DBA-010 [ Schema ]

Ensure setting up a new schema to store data for Csat and related information.

Suggested Schema Name : csat

Any database objects referenced in this document should be under this new scheam unless it is explicitly specified otherwise.

#### HLD-296357a-CSAT-DBA-020 [ Tables ]

Ensure the following tables are created under the schema specified in *HLD-296357a-CSAT-DBA-010 [ Schema ]* .

##### CSAT

|  |  |  |  |
| --- | --- | --- | --- |
| **CUSTOMER** | | | |
| **Name** | **Data Type** | **Nullable** | **Comments** |
| Id | NUMBER(10) | N | CSAT\_SEQ  PK |
| Id\_organization | NUMBER (10) | N |  |
| survey\_date | Datetime | N |  |
| id\_csat\_status | Number(10) | N | Default New should be passed. Its id needs to be picked from CSAT\_STATUS |
| respondent\_name | Varchar(100) | Y |  |
| respond\_statusdate | Datetime | N |  |
| csat\_external\_surveyId | VARCHAR(20) | Y |  |
| created\_by | Varchar2(20) | N |  |
| created\_date | Datetime | N |  |
| updated\_by | Varchar2(20) | N |  |
| updated\_date | Datetime | N |  |
| delete\_Flag | char | N |  |
|  |  |  |  |

##### CSAT\_STATUS

|  |  |  |  |
| --- | --- | --- | --- |
| **CSAT\_STATUS** | | | |
| **Name** | **Data Type** | **Nullable** | **Comments** |
| ID | NUMBER(10) | N | CSAT\_STATUS\_SEQ  PK |
| STATUS\_NAME | VARCHAR(20) | N | New,Open,Complete |

##### CSAT\_QA

|  |  |  |  |
| --- | --- | --- | --- |
| **CSAT\_QA** | | | |
| **Name** | **Data Type** | **Nullable** | **Comments** |
| ID | NUMBER(10) | N | CSAT\_QA\_SEQ  PK |
| ID\_CSAT | NUMBER(10) | N | FK |
| QUESTION | VARCHAR2(500) | N |  |
| RATING | NUMBER(2) | N |  |
| COMMENT | VARCHAR2(200) | Y |  |
| OVERALL\_ACTION\_STATUS\_ID | NUMBER(10) | N |  |
| CREATED\_BY | VARCHAR2(20) | N |  |
| CREATED\_DATE | DATETIME | N |  |
| UPDATED\_BY | VARCHAR2(20) | N |  |
| UPDATED\_DATE | DATETIME | N |  |
| DELETE\_FLAG | CHAR | N |  |

##### CSAT\_QA\_ACTIONS

|  |  |  |  |
| --- | --- | --- | --- |
| **CSAT\_QA\_ACTIONS** | | | |
| **Name** | **Data Type** | **Nullable** | **Comments** |
| ID | NUMBER(10) | N | CSAT\_QA\_ACTIONS\_SEQ  PK |
| ID\_CSAT\_QA | NUMBER(10) | N | FK |
| ID\_CSAT | NUMBER(10) | N | FK |
| ACTION\_NAME | VARCHAR(20) | N |  |
| OBJECTIVES | VARCHAR(500) | N |  |
| ACTION\_STATUS | VARCHAR(20) | N |  |
| IMPACT | VARCHAR(500) | N |  |
| START\_DATE | DATETIME | N |  |
| PROJECTED\_CLOSURE | DATETIME | N |  |
| OWNERSHIP | VARCHAR2(20) | N |  |
| ACTION\_DESCRIPTION | VARCHAR(200) | N |  |
| NOTES | VARCHAR2(500) | N |  |
| CREATED\_BY | VARCHAR2(20) | N |  |
| CREATED\_DATE | DATETIME | N |  |
| UPDATED\_BY | VARCHAR2(20) | N |  |
| UPDATED\_DATE | DATETIME | N |  |
| DELETE\_FLAG | CHAR | N |  |

##### CSAT\_DELETE

|  |  |  |  |
| --- | --- | --- | --- |
| **CSAT\_DELETE** | | | |
| **Name** | **Data Type** | **Nullable** | **Comments** |
| ID | NUMBER(10) | N | ID\_ CSAT\_DELETE\_SEQ  PK |
| ID\_CSAT | NUMBER(10) | N | FK |
| DELETE\_REASON | VARCHAR2(200) | Y |  |
| DELETED\_BY | VARCHAR2(20) | N |  |
| DELETED\_DATE | DATETIME | N |  |

#### HLD-296357a-CSAT-DBA-030 [ Sequences ]

Ensure the following sequences are created under the schema specified in *HLD-296357a-CSAT -DBA-010 [ Schema ]* for the primary keys defined above.

|  |  |  |  |
| --- | --- | --- | --- |
| **SEQUENCES** | | | |
| **Name** | **Start Value** | **Increment By** | **Comments** |
| CSAT\_DELETE\_SEQ | 1 | 1 |  |
| CSAT\_QA\_ACTIONS\_SEQ | 1 | 1 |  |
| CSAT\_QA\_SEQ | 1 | 1 |  |
| CSAT\_STATUS\_SEQ | 1 | 1 |  |
| CSAT\_SEQ | 1 | 1 |  |

### Result Set Cache

This section describes the process to cache the result set from a query, or a microservice call for various purposes, for example, session/user data, sorting/re-sorting, pagination, avoiding from repeated queries.

The implementation of caches could be determined to facilitate fast access, sorting and pagination.

**Note : To ge generic mentioned on hld for caching of the resultset, but the design decision is left to dev team as per the convenience. So striking it off.**

#### ~~HLD-296357a-CsatList-Cache-020 [ CsatList ]~~

~~For any new search/query of CsatList API, a result set ID will be generated from ID\_RESULT\_SET\_SEQ sequence for the result set. The result set should be cached for the following purposes:~~

1. ~~Pagination of the result set.~~
2. ~~Sorting/re-sorting from the result set~~
3. ~~Avoid from repeated queries with the same result set against database.~~

### Csat List – GET

#### HLD-296357a-Csat- CsatList-GET-010 [ Process ]

This section describes the process to provide response based on CSATList API request. The response has dependency on different requests as below and the same api method is applicable on survey action plan landing screen

http://<server>:<port>/restservices/Csat/v1/service/CsatList/{orgid}

1. ~~If UserSelection.WildcardSearch is present, a surveyList structure should be provided in response.~~
2. ~~If resultSetId is populated, request is asking for next page’s data, or re-sorting the result set. UserSelection will be ignored and response structure (UserCustomer structure or Customer structure) should be determined based on the structure in~~ *~~HLD-296357a-CsatCsatListr-Cache-020 [ Customer List ]~~* ~~. If there is no cached result set, throw an “Invalid Input” exception.~~
3. ~~If UserSelection.WildcardSearch is present and resultSetId is not populated in request, follow the sections to query the database to prepare the result set for response :~~
4. *~~HLD-296357a-Csat-CsatList-GET-050 [ SurveyId ]~~* ~~if searchField is ‘Survey Id’.~~
5. *~~HLD-296357a-Csat-CsatList-GET-060 [ SurveyDate ]~~* ~~if searchField is ‘Survey Date’.~~
6. *~~HLD-296357a-Csat-CsatList-GET-070 [ SurveyStatus ]~~* ~~if searchField is ‘Survey Status’.~~
7. ~~The result set from above should be cached as described in~~ *~~HLD-296357a-Csat-CsatList-Cache-020 [ CsatList ]~~* ~~for further processing and subsequent requests.~~
8. In case that result set is empty, a “No Data Found” exception should be thrown.
9. Follow *HLD-296357a-Csat-CsatList-GET-030 [ Record Range ]* to select requested record in the range for response.
10. Follow the following data mapping and AID for response construction:

|  |  |  |
| --- | --- | --- |
| **Data Element in Microservice** | **Database Table and Column** | **Comments** |
| **SequencesurveyList start** |  |  |
| csatId | CSAT.id |  |
| externalSurveyId | CSAT.csat\_external\_surveyid |  |
| surveyDate | CSAT.survey\_date |  |
| surveyStatus | CSAT\_STATUS.csat\_status\_name |  |
| surveyActionPlanIndicator |  | On query result we have flag indicator by having logic to check whether any action plans exisits for a given csat id.  CSAT\_QA\_ACTIONS |
| RecordRange | *CSATRecordRange* |  |
|  |  |  |
| **Sequence surveyList End** |  |  |
| Response | *CSATResponse* | Only for failures |

#### HLD-296357a-Csat-CsatList-GET-020 [ Record Sorting ]

This section describes the data sorting based on RecordSorting structure in request. ~~The data sorting should be executed on the result set for the resultSetId (from RecordRange structure) cached in~~ *~~HLD-296357a-Csat-CsatList-Cache-020 [ Csat List ].~~*

|  |  |  |
| --- | --- | --- |
| **sortColumn Value in RecordSorting** | **Query column and Column Sort On** | **Comments** |
| ‘SURVEY\_ID’ | CSAT.csat\_external\_surveyId |  |
| ‘SURVEY\_DATE’ | CSAT.survey\_date |  |
| ‘SURVEY\_STATUS’ | CSAT.survey\_status |  |
| ‘SERVACTPLAN’ |  | On query result we have flag indicator by having logic to check whether any action plans exisits for a given csat id.  CSAT\_QA\_ACTIONS |

Note: By default the newly uploaded survey(i.e the surveystatus = ‘Open) should be the first record with the recent survey date as top while displaying on the grid. This is applicable only when no sorting column and sort type is passed as url query parameter. In other cases, display of records will be taken care as per query url parameters.

#### HLD-296357a-Csat-CsatList-GET-030 [ Record Range ]

This section describes the data pagination based on RecordRange structure in request. ~~The data pagination should be executed on the result set for the resultSetId cached in~~ *~~HLD-296357a-Csat-CsatList-Cache-020 [ CsatList ]~~.*

Data pagination process should be executed after Record Sorting process if RecordSorting is specified in the request.

A record number should be generated for each record, for example, first record has record number 1, second record has 2, and so on. Based on RecordRange.startRecord and Record.Range.endRecord, only the records with record number in between should be returned in response.

Note : SE team agreed that mandatorily they will send startRecord and EndRecord values will be send as part of request parameters.

#### HLD-296357a-Csat-CsatList-GET-060 [ search - SURVEY\_ID ]

This section describes the query construction based on searchValue and searchField ‘SURVEY\_ID from request. All the fields defined in CSATList structure in response (refer to AID for details) should be selected.

Only the CSAT records with csat\_external\_survey\_id in SURVEY\_ID, and its alias matched with searchValue from request will be selected. In addition, partial search should be supported, for example, if searchValue has value ‘ABC’, any CSAT records with CSAT.csat\_external\_survey\_id starting with ‘ABC’ should selected. If searchValue has value ‘ABC%’, any CSAT records with CSAT.csat\_external\_survey\_id containing ‘ABC’ should be selected.

Ensure CSAT.csat\_external\_survey\_id in cached in *HLD-296357a-Csat-CsatList-Cache-010 [ csatList ].*

Note: 1. all the comparisons described above should be case insensitive.

2. If cahing is not done as part of design decision then the search parameters will be part of get url method.

#### HLD-296357a-Csat-CsatList-GET-070 [Search - SURVEY\_DATE ]

This section describes the query construction based on searchValue and searchField ‘SURVEY\_DATE’ from request. The query should select out records from CSAT table based on csat.survey\_date field.

All the fields defined in csatList structure in response (refer to AID for details) should be selected.

Only the CSAT records with survey\_date matched with searchValue from request will be selected.No partial search is supported. Only exact search on survey date is considered.

Ensure customer.id\_organization is in the orgId cached in *HLD-296357a-Csat-CsatList-Cache-010 [ CsatList ].*

Note :If cahing is not done as part of design decision then the search parameters will be part of get url method.

#### HLD-296357a-Csat-CsatList-GET-080 [ search - SURVEY\_STATUS ]

This section describes the query construction based on searchValue and searchField ‘SURVEY\_STATUS from request. All the fields defined in CSATList structure in response (refer to AID for details) should be selected.

Only the CSAT records with surveyStatus in Survey\_Satus, and its alias matched with searchValue from request will be selected. No partial search is supported. Only exact search on survey status is considered.

Note : SearchValue of SearchColumn for survey status as of now will be of either ‘Open’ / ‘Completed’ . Care should be taken to retrieve the PK of the inputted search value from CSAT\_status tabe and query across CSAT table for the records matching the retrieved Csat\_status.id

Ensure CSAT.csat\_external\_survey\_id is in the surveyId cached in *HLD-296357a-Csat-CsatList-Cache-010 [ csatList ].*

*-------------------------------------------------------------------------------------------------------*

Note :If cahing is not done as part of design decision then the search parameters will be part of get url method.

### ServiceActionPlanData – GET

#### HLD-296357a-Csat-ServiceActionPlanData- -GET-010 [ Process ]

This section describes the process to provide response based on ServiceActionPlanData - GET API request. This API is retrieving a detail record for a customer.

Take csatId, questionId(optional) from request and query across CSAT\_QA and CSAT\_QA\_ACTIONS table for the corresponding record.

If input had questionId as parameter passed then retrieve the inputted questions start block and corresponding Actions block on the response structure

http://<server>:<port>/restservices/Csat/v1/service/ServiceActionPlanData/{csatId}?questionId=

In case that no record is returned from above, a “No Data Found” exception should be thrown.

Construct the response based on the data mapping below and AID:

|  |  |  |
| --- | --- | --- |
| **Data Element in Microservice** | **Database Table and Column** | **Comments** |
| csatId | CSAT\_QA.id\_csat |  |
| Questions Start |  |  |
| questionId | CSAT\_QA.id |  |
| Question | CSAT\_QA.question |  |
| Actions Start |  |  |
| actionId | CSAT\_QA\_ACTIONS.id |  |
| Action | CSAT\_QA\_ACTIONS.action\_name |  |
| Actions End |  |  |
| QuestionActionPlans End |  |  |
| Response | *CSATResponse* | Only on failures |

### QuestionActionPlanDetails – GET

#### HLD-296357a-Csat-QuestionActionPlanDetails- -GET-010 [ Process ]

This section describes the process to provide response based on QuestionActionPlanDetails - GET API request. This API is retrieving a action detail record for a action.

Take actionId from request and query across CSAT\_QA\_ACTIONS table for the corresponding record.

http://<server>:<port>/restservices/Csat/v1/service/QuestionActionPlanDetails/{actionId}

In case that no record is returned from above, a “No Data Found” exception should be thrown.

Construct the response based on the data mapping below and AID:

|  |  |  |
| --- | --- | --- |
| **Data Element in Microservice** | **Database Table and Column** | **Comments** |
| csatId | CSAT\_QA\_ACTIONS.id\_csat |  |
| externalSurveyId | CSAT.csat\_external\_surveyId |  |
| Objectives | CSAT\_QA\_ACTIONS.id |  |
| actionStatus | CSAT\_QA\_ACTIONS.Action\_status |  |
| Impact | CSAT\_QA\_ACTIONS.impact |  |
| actionDescription | CSAT\_QA\_ACTIONS.action\_description |  |
| startDate | CSAT\_QA\_ACTIONS.startdate |  |
| projectedClosure | CSAT\_QA\_ACTIONS.Projectedclosure |  |
| Ownership | CSAT\_QA\_ACTIONS.Ownership |  |
| Notes | CSAT\_QA\_ACTIONS.Notes |  |
| Response | *CSATResponse* |  |

### ~~QuestionActionPlan – DELETE~~ Got descoped as per CR 172983

#### ~~HLD-296357a-Csat-QuestionActionPlan- -DELETE -010 [ Process ]~~

~~This section describes the process to provide response based on QuestionActionPlan - DELETE API request. This API will delete inputted action from the corresponding table CSAT\_QA\_ACTIONS.~~

~~Take actionId from request and query across CSAT\_QA\_ACTIONS table for soft delete of corresponding record.~~

~~http://<server>:<port>/restservices/Csat/v1/service/QuestionActionPlan /{actionId }/{deletedBy}~~

~~In case that no record is returned from above, a “No Data Found” exception should be thrown.~~

~~Construct the request based on the data mapping below and AID:~~

|  |  |  |
| --- | --- | --- |
| **~~Data Element in Microservice~~** | **~~Database Table and Column~~** | **~~Comments~~** |
| ~~sessionId~~ | ~~-~~ |  |
| ~~transactionId~~ | ~~-~~ |  |
| ~~actionId~~ | ~~-~~ |  |
| ~~deletedBy~~ | ~~Logged in attuid from the request~~ |  |

~~Construct the response as the following mapping:~~

|  |  |  |
| --- | --- | --- |
| **~~Data Element in Microservice~~** | **~~Database Table and Column~~** | **~~Comments~~** |
| ~~Response~~ | ~~CSATResponse~~ | ~~Only in case of failures~~ |

### CsatInfoDetails – GET

#### HLD-296357a-Csat-CsatInfoDetailst- -GET-010 [ Process ]

This section describes the process to provide response based on CsatInfoDetails - GET API request. This API is retrieving a basic survey info details for a survey.

Take csatId from request and query across CSAT table for the corresponding record.

http://<server>:<port>/restservices/Csat/v1/service/CsatInfoDetails/{csatId}

In case that no record is returned from above, a “No Data Found” exception should be thrown.

Construct the response based on the data mapping below and AID:

|  |  |  |
| --- | --- | --- |
| **Data Element in Microservice** | **Database Table and Column** | **Comments** |
| csatId | CSAT.id |  |
| surveyId | CSAT.external\_survey\_id |  |
| surveyDate | CSAT.surveyDate |  |
| surveryStatus | CSAT. id\_csat\_status |  |
| respondentName | CSAT.respondent\_name |  |
| respondentStatusDate | CSAT.respond\_statusdate |  |
| Response | *CSATResponse* | Reference to AID |

### InquireCsatResults – GET

#### HLD-296357a-Csat-InquireCsatResults- -GET-010 [ Process ]

This section describes the process to provide response based on InquireCsatResults - GET API request. This API is retrieving a csat result details for a survey.

Take csatId from request and query across CSAT\_QA table for the corresponding record.

http://<server>:<port>/restservices/Csat/v1/service/InquireCsatResults/{csatId}

In case that no record is returned from above, a “No Data Found” exception should be thrown.

Construct the response based on the data mapping below and AID:

|  |  |  |
| --- | --- | --- |
| **Data Element in Microservice** | **Database Table and Column** | **Comments** |
| csatId |  |  |
| Sequence Questions Start |  |  |
| questionId | CSAT\_QA.id\_csat\_qa |  |
| Question | CSAT\_QA.question |  |
| Rating | CSAT\_QA.rating |  |
| sapIndicatorFlag | True, False | Have inner join on CSAT\_QA\_ACTIONS table for questionId . If found as not null then have query logic column field value as ‘True’ else retrive ‘False’ |
| overallSAPStatusFlag | True, False | Logic to check whether all the actions associated to questionId in CSAT\_QA\_ACTIONS are closed or cancelled then have ‘True’ (display as ‘Close’ selected on radiobutton).  ‘False’ (display as ‘Open selected on radiobutton) when one or more actions for the question is in status as ‘Open’ or ‘Planned’. |
| Comments | CSAT\_QA.comments |  |
| Sequence Questions End |  |  |
| Response | CSATResponse | Only in case of failures |
|  |  |  |

### RemoveCsat – DELETE

#### HLD-296357a-Csat-RemoveCsat- -DELETE-010 [ Process ]

This section describes the process to provide response based on RemoveCsat - DELETE API request. This API is retrieving a comments record for a question.

Step 1 : Take csatId from request and query across CSAT, CSAT\_QA and CSAT\_QA\_ACTIONS table for the corresponding record to perform soft delete on survey table, survey question and survey question actions tables. Make sure all these tables get updated on delete flag accordingly.

Step 2 : Take deletedBy, csatId and deletedReason from input api request and insert into CSAT\_DELETE table

~~Step 3 : Archive the deleted records from CSAT, CSAT\_QA and CSAT\_QA\_ACTIONS tables by having nightly job .~~

http://<server>:<port>/restservices/Csat/v1/service/RemoveCsat/{Csatid}/deletedby?deletedReason=

Construct the request based on the data mapping below and AID:

|  |  |  |
| --- | --- | --- |
| **Data Element in Microservice** | **Database Table and Column** | **Comments** |
| Request | CSATRequest |  |
| csatId |  |  |
| deletedBy |  |  |
| deletedReason |  |  |

Construct the response as the following mapping:

|  |  |  |
| --- | --- | --- |
| **Data Element in Microservice** | **Database Table and Column** | **Comments** |
| Response | CSATResponse | Only in case of failures |

### CsatData– POST

#### HLD-296357a-Csat-CsatData- -POST-010 [ Process ]

This section describes the process to provide response based on csat Data - POST API request. This API is creating/ updating a record for a customer in different scenarios

1. Through upload file , creates survey record for a customer/orgid.
2. Editing the survey info screen details.
3. Editing survey results screen information.

Update database table as the following mapping, and only update the fields which were sent in the request, even if the field has blank/null value. However, if the field is not sent in the request, the corresponding value in the table should not be touched.

|  |  |  |
| --- | --- | --- |
| **Data Element in Microservice** | **Database Table and Column** | **Comments** |
| surveyId | CSAT.id | PK value gerenrated from CSAT table  If null— then its add record.  If not null - then its to update a record. |
| externalCsatId | CSAT.csat\_external\_surveyId |  |
| surveyDate | CSAT.id\_customer\_status |  |
| survey\_status\_id | CSAT.id\_csat\_status | If SurveyId is null from the api request, then have value from the api which is equivalent to status = ‘Open’ on CSAT\_STATUS table. |
| respondentName | CSAT.respondent\_name |  |
| respondStatusDate | CSAT.respond\_statusdate |  |
| createdBy |  | If surveyId is null from the request then have logged in attuid as value in CSAT.Created\_by  If SurveyId is not null from the api request then have logged attuid as value stored in CSAT.Updated\_by.  Use this same attribute of createdby to store in CSAT\_QA.CreatedBy field if survey id of input request is null. |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *QA Start* |  | *-* | *0* | *50* | *1.0* |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| questionId | CSAT\_QA.id |  | 0 | 1 | Internal PK of csat\_qa  If it is null treated as new entry else it is to be updated. |
| Question | CSAT\_QA.question |  | 1 | 1 |  |
| Rating | CSAT\_QA.rating | Range 1 to 10 | 0 | 1 |  |
| Comment | CSAT\_QA.comments |  | 0 | 1 |  |
| updatedBy | CSAT\_QA.updated\_by |  | 0 | 1 | Logged in attuid needs to be stored in CSAT\_QA.updated\_by when survey results are getting modified. |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *QA End* |  | *-* | *0* | *50* | *1.0* |

ST Defect 397074:

Note : For the scenario like, a survey is deleted through /RemoveCsat operation and the user is trying to upload a survey having the same external survey Id which is already existing in the db records, then while inserting into database, a validation check needs to be performed such that system should allow the same external survey Id when the db contains records with delete flag as active (‘Y’) .

Otherwise , the system should not allow the insertion of survey with the same external id when existing in database with no set for deleteflag. In other words, we should not allow duplicate entry into db when deletegflag is not active.

Construct the response as the following mapping:

|  |  |  |
| --- | --- | --- |
| **Data Element in Microservice** | **Database Table and Column** | **Comments** |
| csatId | CSAT.id |  |
| Response | CSATResponse | Only in case of failures |

### QuestionActionPlanDetails – POST

#### HLD-296357a-Csat-QuestionActionPlanDetails- -POST-010 [ Process ]

This section describes the process to provide response based on QuestionActionPlanDetails - POST API request. This API is creating/ updating a record for a action plan of selected question through different scenarios

1. Through create/edit pop up screen of action for the selected question.
2. Through edit screen of SAP tab info.

Update database table as the following mapping, and only update the fields which were sent in the request, even if the field has blank/null value. However, if the field is not sent in the request, the corresponding value in the table should not be touched.

Validation on Ownership value : Call below microservice api method to check whether user provided registered system attuid or not.

http://<server>:<port>/restservices/ManageUserProfile/v1/service/UserDetails?userId=1001

If the response of userdetails get api method from is not null then trigger QuestionActionPlanDetails -> Post method. If userdetails get api method response has error code. Then Sales express team should through message as invalid user and the subsequent post method QuestionActionPlanDetails should not be triggered.

|  |  |  |
| --- | --- | --- |
| **Data Element in Microservice** | **Database Table and Column** | **Comments** |
| csatId | CSAT.id |  |
| actionId | CSAT\_QA\_ACTIONS.Id | If actionId is null then it is new entry.  If actionId is not null then it is to edit. |
| questionId | CSAT\_QA\_ACTIONS.id\_csat\_qa |  |
| actionName | CSAT\_QA\_ACTIONS. action\_name |  |
| Objectives | CSAT\_QA\_ACTIONS.objectives | If SurveyId is null from the api request, then have hard code value from the api which is equivalent to status = ‘New’ on CSAT\_STATUS table. |
| actionStatus | CSAT\_QA\_ACTIONS. action\_status |  |
| Impact | CSAT\_QA\_ACTIONS.impact |  |
| actionDescription | CSAT\_QA\_ACTIONS. action\_description | If surveyId is null from the request then have logged in attuid as value in CSAT.Created\_by  If SurveyId is not null from the api request then have logged attuid as value stored in CSAT.Updated\_by.  Use this same attribute of createdby to store in CSAT\_QA.CreatedBy field if survey id of input request is null. |
| startDate | CSAT\_QA\_ACTIONS.startdate |  |
| projectedClosure | CSAT\_QA\_ACTIONS.projectedclosure |  |
| Ownership | CSAT\_QA\_ACTIONS.ownership |  |
| Notes | CSAT\_QA\_ACTIONS.notes |  |
| createdBy | CSAT\_QA\_ACTIONS.created\_by  CSAT\_QA\_ACTIONS.updated\_by | If actionId is null store the logged in attuid in CSAT\_QA\_ACTIONS.created\_by else in CSAT\_QA\_ACTIONS.updated\_by |

Construct the response as the following mapping:

|  |  |  |
| --- | --- | --- |
| **Data Element in Microservice** | **Database Table and Column** | **Comments** |
| actionId | CSAT.id |  |
| Response | CSATResponse | Only in case of failures |

### Value List – GET

#### HLD-296357a-CSAT-Value-List-GET-010 [ Process ]

This section describes the process to provide response based on Value List - GET API request. This API is retrieving a list of values for an object.

http://<server>:<port>/restservices/Csat/v1/service/ValueList/{objectName}

First determine the table name to query data from based on objectName value as below. If the objectName is not in the list below, throw an exception “Invalid Input”.

|  |  |  |
| --- | --- | --- |
| **objectName** | **Database Table** | **Comments** |
| ‘csat status’ | CSAT\_STATUS |  |

Construct database query based on the following mapping or filters. If only objectName is populated in request, all records from the table determine above should be selected. If any optionId and optionValue is populated, only records satisfying the values sent in request should be selected. If no records returned from the query, an exception “No Data Found” should be thrown.

|  |  |  |
| --- | --- | --- |
| **Data Element in ValueList Sequence** | **Database Table and Column** | **Comments** |
| objectName | - | Determine the <table> name from above |
| optionId | <table>.id | Ensure <table>.id equals to optionId value from request only if it is populated |
| optionValue | <table>.name or <table>.status for ‘Status’ only | Ensure <table>.name/status equals to optionValue value from request only if it is populated |
|  |  |  |

Construct the response based on the data mapping below and AID:

|  |  |  |
| --- | --- | --- |
| **Data Element in Microservice** | **Database Table and Column** | **Comments** |
| objectName |  | Echo from request |
| Sequence OptionList Start |  |  |
| optionId | <table>.id |  |
| optionValue | <table>.name/status |  |
| optionSeqNo | <table>.sequence\_no |  |
| Sequence OptionList End |  |  |
| Response | CSATResponse |  |
|  | - |  |

### Value List – POST

#### HLD-296357a-Csat-Value-List-POST-010 [ Process ]

This section describes the process to provide response based on Value List - POST API request. This API is add/update values into a list of values for an object.

First determine the table name to query data based on objectName value as described in *HLD-296357a-Csat Value-List-GET-010 [ Process ]*. If the table name can’t be determined, throw an exception “Invalid Input”.

|  |  |  |
| --- | --- | --- |
| **Data Element in ValueList Sequence** | **Database Table and Column** | **Comments** |
| objectName | - | Determine the <table> name from above |
|  |  |  |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Choice OptionList Start | | |  |  | 1 | 100 |  |
| Sequence UpdateOption Start | |  |  | 1 | 1 |  |
| optionId | Integer |  | 0 | 1 |  |
| optionValue | String (100) |  | 1 | 1 |  |
| optionSeqNo | Integer |  | 0 | 1 |  |
| Sequence UpdateOption End | |  |  | 1 | 1 |  |
| Sequence AddOption Start | |  |  | 1 | 1 |  |
| optionValue | String (100) |  | 1 | 1 |  |
| optionSeqNo | Integer |  | 0 | 1 |  |
| Sequence NewOption End | |  |  | 1 | 1 |  |
| Choice OptionList End | | |  |  | 1 | 100 |  |

|  |  |  |
| --- | --- | --- |
| Action | - | ***AddOption sequence will be appeared.***  If action = ‘Add’, insert record of each optionId, optionValue, and optionSeqNo into the <table>  ***UpdateOption sequence will be appeared.***  If action = ‘Update’, update the record of <table> for optionId. If optionId is not in the <table>, throw an exception of “No Data Found”. |
| optionId | <table>.id | Match with <table>.id for action ‘Update’ |
| optionValue | <table>.name or <table>.status for ‘Status’ only | For action = ‘Update’, update <table>.name/status if it is populated for each optionId |
| optionSeqNo | <table>.sort\_index | For action = ‘Update’, update <table>.sort\_index if it is populated for each optionId. |

Construct the response as the following mapping:

|  |  |  |
| --- | --- | --- |
| **Data Element in Microservice** | **Database Table and Column** | **Comments** |
| Response | CSATResponse | Only in case of failures |

### Value List – DELETE

#### HLD-296357a-Csat-Value-List-DELETE-010 [ Process ]

This section describes the process to provide response based on Value List - POST API request. This API is to delete values into a list of values for an object.

http://<server>:<port>/restservices/Csat/v1/service/ValueList/{objectName}/{oprtionId}

First determine the table name to query data based on objectName value as described in *HLD-296357a-Csat-Value-List-GET-010 [ Process ]*. If the table name can’t be determined, throw an exception “Invalid Input”.

|  |  |  |
| --- | --- | --- |
| **Data Element in ValueList Sequence** | **Database Table and Column** | **Comments** |
| objectName | - | Determine the <table> name from above |
| optionId | <table>.id | Delete <table> record base on optionId (<table>.id ), |

Construct the response as the following mapping:

|  |  |  |
| --- | --- | --- |
| **Data Element in Microservice** | **Database Table and Column** | **Comments** |
| Response | CSATResponse | Only in case of failures |

**Alternative Designs**

*Describes alternative designs, evaluation criteria, risks and issues that were considered in choosing the design. If a Solution Approach was developed for this project and the design deviates in its approach from that recommended in the Solution Approach, the HLD should explain the difference in approach and the rationale for the change.*

None Identified.

**Assumptions/Risks**

*Identifies design assumptions, issues, constraints, and risks and mitigation strategies for each.*

*Refer to the Project Workflow Module for Risks/Issues associated with this Project.*

Assumptions

Risk/ Constraints

1. Any scope changes that impact the initial work efforts must be communicated in a timely manner or it may affect meeting project deliverables milestones.

**Solution Design**

*SRT REQPRO NOTE: In order to utilize the auto-tagging feature within SRT ReqPro and enable easier tagging of design elements, please document the design elements in the following table:*

*Note: Req. ID should be used to identify the Design Element ID.*

(Examples in red text)

|  |  |  |
| --- | --- | --- |
| **Req. ID** | **Requirement Description** | **Trace-To** |
| HLD.#.PID. Application Impacted  Example:  HLD.1. 179864.CSI-NM | **InquireCablePairTerminalDetails – New**   1. Component – M2E 2. Description of change:CSI shall create a new InquireCablePairTerminalDetails SPM to provide a list of all the terminals that a particular cable pair appears in, the terminal type, status of the pair in each of the terminals, its binding post/color in each of the terminals, and loop related characteristics of the pair.   The minimal required input is wirecenter, employee identification, cable and pair.  Priority is optional. If it isn’t input by the client it will be populated with “I” for immediate.  Refer to the Application Interface Design for the all request (input data)/response (output data) schemas for the service. The response will have information about the cable pair, and information about the cable pair in each of the terminals in which it appears.  CSI will call the INQ APP (Multiple Appearance Inquiry) transaction in LFACS. | *SR Req. ID* |
| HLD.#.PID. Application Impacted | **InquireFacilityAssignmentDetails – New**   1. Component – M2E 2. Description of change:CSI shall create a new InquireFacilityAssignementDetails SPM to provide the current data for facilities and/or loops and is a good source for obtaining pending service order data associated with a facility. This information may be used for record verification and for obtaining service order data associated with the facility. The Outside Plant Facility Assignment inquiry provides read only access to LFACS inventory data. No updating functionality is provided.   There is different ways to call this transaction to require different information;   * Facility address call provides a current representation of a facility address with its associated loop data.   What is minimal required; Wirecenter, employee identification, house number, street name   * Cable and Pair call provides a current representation of a cable pair with its’ associated loop data.   What is minimal required; Wirecenter, employee identification, cable, pair   * Circuit ID call provides a current representation of a circuit id with its' associated loop data.   What is minimal required; Wirecenter, employee identification, circuit identification.  Refer to the AID for the all request (input data)/response (output data) schemas for the service. The response will have all associated data of the facility as it currently exists in the database.  CSI will be calling LFACS INQ FASG transaction. | *SR Req. ID* |
| HLD.#.PID. Application Impacted | **InquireAffiliateAccountProfile – New**   1. Component - M2E 2. Description of change – The purpose of InquireAffiliateAccountProfile is to provide customer account information from BCAM to CSI-SPM, which will include DTV account association.    * + CSI-Interface      + The input for this new service will be BTN or BAN. CSI-SPM will call BCAM and request the data. Response from BCAM to CSI-SPM will include account data for BTN, BAN, or Telco Pending Order as well as DTV data.      + The communication between Client 🡪 CSI-SPM 🡪 BCAM will all be in the standard schema format.      + If an error is generated from BCAM then CSI-SPM will receive a failed message from the BCAM and return that failed message to the client in standard schema format. Specific details will be apart of the Application Interface design and schemas      + Data Elements/tags/field names      + Request and response elements will be detailed as part of Application Interface design and schemas   Business Rules -   1. This new service InquireAffiliateAccountProfile will call the InquireWirelineRegion service to retrieve the wireline region details. 2. This new service InquireAffiliateAccountProfile will return the Account Information and DTV account data to client. | *SR Req. ID* |
| HLD.#.PID. Application Impacted | **InquireCombinedBillingEligibility – Enhanced**   1. Component – M2E 2. Description of change – InquireCombinedBillingEligibility will verify if the ATT account is eligible for combined billing with DTV via the iCBT (integrated Combined Billing Tool) API based on the customer’s Wireline Telco 13 digit BTN (Billing Telephone Number) and MODE.  * Add a new value for “D” (DirecTV, used to set DTVIndicator) for the MODE. MODE value of “D” is used to determine DTV Indicator. Will pass the DTV Indicator to iCBT, and iCBT will use the indicator in the determination if the account is eligible for combined billing. * WirelineCombinedBilling **-** This structure allows the client to indicate the AT&T Billing Account Number + Customer Code (13 digits). * Mode -A mode for combined bill to denote type of combined bill requests.   Wireless/Wireline, or Both or Wireline only or DTV  For DirecTV the MODE of “D” is required.  If the MODE is equal to a value of “D”, set the DTV Indicator to true.  Otherwise, set the DTV Indicator to false.  Request and response data elements will be detailed as part of Application Interface Design and Schemas.  Business Rules   1. The existing service uses the iCBT API 2. This existing service InquireCombinedBillingEligibliity will return if combined billing eligible to the client |  |
| HLD.#.PID. Application Impacted | **InquireUnifiedCustomerProfileByServiceLocation - Enhanced**     1. Component - M2E 2. Description of change:CSI shall enhance an existing service, *InquireUnifiedCustomerProfileByServiceLocation*, is called by client to retrieve the customer and service profiles based upon a wireless subscriber number or a wireline telephone number or a wireline account number and now for this project, based upon a U-verse BAN as well. This interface will continue to retrieve customer and service profiles from CCR via VRI. 3. Response elements will be detailed as part of Application Interface design and schemas. | *SR Req. ID* |
| HLD.#.PID. Application Impacted  Example:  HLD.1. 179864. CSFOBPM | **InquireUnifiedCustomerServiceProfile - Enhanced**   1. Component - BPM 2. Description of change:CSI shall enhance an existing service, *InquireUnifiedCustomerServiceProfile*, that returns a Customer Profile and Customer Service Profile for a subscriber number or a wireless telephone number or a wireline account number device and now for this project, based upon a U-verse BAN as well. If response has U-verse account only, we will send DTV eligibility as not eligible. 3. Request and response elements will be detailed as part of Application Interface design and schemas. | *SR Req. ID* |
| HLD.#.PID. Application Impacted. AdapterName  Example:  HLD.1. 179864.CSI-Adapter.ACIS | **Adapter Requirements**  The following is an example of how we would capture Adapter requirements.  **ACIS adapter – Enhanced or New**   1. Component – Adapter 2. **Method(s) – Adapter method Name fetchCSRForMidwest The adapter method name is the name of the API, interface, etc that the adapter will use to obtain data from the source system. In the case of multiple handlers in a single adapter, each handler’s information impacted would be listed in the corresponding section.**   This information can be found on existing adapters at the following link  <https://operations.web.att.com/sites/CSD_DC/CSD%20Project%20Tracking%20Tool/Lists/CSD%20Adapters/Default%20View.aspx>  **C.** Description of change:  Description of the change should be inserted here.  The fetchCSRForMidwest method will be enhanced to include credit information.   1. **Connection Details**  * We would work with development in identifying this information. Examples would be EMBUS, MQSERIES or WebService.  1. **Downstream Interface(s)**   XYTACP01  **Processing Rules:**  Any unique processing that the adapter will perform will need to be documented especially where a new adapter is being created which could include a new source system. Again requirements will be working with development for this information. | *SR Req. ID* |
| HLD.#.PID. Application Impacted  Example:  HLD.1. 179864.CSI | **Version Proxy**   * 1. Standard Version Proxy map will be provided for the existing interfaces. | *SR Req. ID* |
| HLD.#.PID. Application Impacted  Example:  HLD.1. 179864. PartnerProfile | **Partner Profile**  *If there are any service policies or partner profile changes necessary to fulfill the business logic for this project, they must be documented in this section. The need to update the Partner Profile, in most cases, comes as a result of needing to define a new Service Policy that would allow dynamic rules to be executed differently for different partners/clients.*  The following is an example of a partner profile change:   1. Component – SPM Product & Offer Management – InquireWirelineRegion 2. Description of change: A new service policy will be created. The service policy will be called ‘IWR-SKIP-BCAM-ZIPCODE-VALIDATION’. This service policy will control the call to skip the AAV/BCAM call when zip code validation is requested and a zip code is not returned from CCR. With this flag set to true, CSI will bypass call to AAV and return the response from CSI. This will be considered as a zip code match. | *SR Req. ID* |
| HLD.#.PID. Application Impacted. AdapterName | **Mapping Upgrades**  If there are request or response schema changes to CSI services that impact the CDM file (for common/shared data elements), the developer assigned to the project needs to run an impact analyzer script (a.k.a. Schema Analyzer tool) to determine the impacted services sharing the common data elements.  This step should happen on or before the CSI HLD internal review with the Design Assurance team.  Any impacted services found, should be added in the HLD as being impacted for mapping upgrades. The owners of the identified impacted services should be informed and engaged by the PM so they can upgrade their data maps.  **InquireAffiliateAccountProfile – Enhanced**   1. Component - SPM 2. Description of change – The **InquireAffiliateAccountProfile** schema request and/or response is impacted by data map upgrades due to shared structure/element definitions in the Cingular Data Model (CDM) file. The common data structure/element being updated is the **StructureName or DataElementFieldName**. The change is being made due to:  * The **StructureName** or **DataElementFieldName** is being changed from Required to Optional (or vice versa) or * The **DataElementFieldName** size is being increased/decreased or * The **DataElementFieldName**’s data type is being changed from Numeric to Alphanumeric , String to Numeric, etc * An optional/required structure/Data element being introduced impacting the private schema invoked by **InquireAffiliateAccountProfile** which needs to be absorbed by the service or * Any other reason for the data map change reason | *SR Req. ID* |

*The* [SRT\_RM\_ReqPro\_Auto\_Tagging](https://cps.web.att.com/CPSWorkplace/getContent?id=current&vsId=%7B67070CE6-CDED-4932-8F1D-F6073BA40DF8%7D&objectStoreName=IT-Architecture.__.Planning.__.and.__.Integration&objectType=document&guestId=servicesguest) *Reference Document provides information on how to use auto-tagging within SRT ReqPro.*

*If the table is not used to capture requirements, please remove it from document.*

**System Agreements <CSI>**

**Note:** Although an attempt was made to capture all changes resulting from this design, unanticipated design changes may occur after this document has been turned over that may affect the impact on systems or applications.

This lists clients that have been identified as having impact.

*Instructions:  Only list the impacted clients within this project that are requesting these enhancements.*

|  |  |  |
| --- | --- | --- |
| **Client** | **Client of which System** | **Interface Name** |
| eElections IVR | CS FOBPM | ProcessSelfServiceCPNIElection |
| OPSS | CS FOBPM | ProcessSelfServiceCPNIElection |
|  |  |  |
| OPUS | CSI-Customer Care | InquireCPNIDetails |
| OPUS-Lite | CSI-Customer Care | InquireCPNIDetails |
| PDC | CSI-Customer Care | InquireCPNIDetails |
| PDC-Lite | CSI-Customer Care | InquireCPNIDetails |
| System Xi | CSI-Customer Care | InquireCPNIDetails |

**Traceability Matrix**

*Insert link or lists each requirement and references how it is addressed by the design. This will provide visibility on how well the design meets requirements. In any cases where some requirements are not addressed in the design, they should be called out in this section to enable the audience to better understand the design's level of traceability.*

*NOTE: The design element identifiers must be transferred over to the Requirements Traceability Matrix in order to complete the High Level Design.*

|  |  |
| --- | --- |
| **Requirement ID** | **Design Element Identifier** |
| *FR-1.1* | *HLD - 1* |

**Pre-Production Disaster Recovery Planning**

*As a reminder, for each new or modified application refer to* [ITSC Integrated Policies and Standards](http://itup.it.att.com/ittools/itmap/resources/cfm/itup/1_ProcessElement.cfm?xPEName=ITSC%20Integrated%20Policies%20and%20Standards). *Review disaster recovery requirements for DR Plans, related documentation, and exercise requirements and ensure all DR requirements are met and reflected in MOTS before an application is entered into production. Refer to* [IT Service Continuity Capability Guidance](http://itsc.web.att.com/itup.htm) *for more detailed instructions. If pre-production Disaster Recovery is not needed for this project, place an NA in this section with a business reason (i.e., OS version upgrade or Database version upgrade).*

*If not applicable for this project:*

N/A – This project does not impact the current Application Impact Analysis or Disaster Recovery Plan for the impacted Common Services (CSI) applications. A Pre-Production Disaster Recovery Exercise is not required for this project.

**Other Plans and References**

*Provide links to other plans at your discretion.*

|  |
| --- |
|  |
| **Reference** | | **Location** |
| Project Plan | | Place project document link here |
| Business Requirements Specifications | | Place project document link here |
| System Requirements | | Place project document link here |
| Application Interface Design | | Place project document link here |
| Uses Case document | | Place document link here |
| Requirements Traceability Guidelines | | <http://itup.it.att.com/ittools/itmap/itup/Method/mth_Requirements%20Traceability%20Guidelines.doc?CFID=190824&CFTOKEN=b5469ca5b19769e3-1AF3D333-D526-555C-731247FCB455064E> |
| Requirements Traceability Matrix | | Place project document link here |
| Any other documents that you see useful... (Example backends HLD and AID, Solution Approach, etc…) | | Place projects document link here |

**Acceptance & Approvals**

**Overview**

*Use this section to capture approvals in the event that electronic approvals via the PRISM Project Workflow Module will not be used.*

The Approvers of this work product agree that this document is acceptable and complete to the best of their knowledge and will be used by the project team as an official deliverable for the project. It is further agreed that this document can now be baselined and any changes to these sections from this point forward must follow the Managing Change in the IT UP.

Embed evidence of approval in the review table below, or use the PRISM Approval Functionality in the Project Workflow Module Workflow Template View.

**Approvers**

**PLEASE NOTE: This does not replace the Review Record. This is simply to gather the email approvals. The original email must be included and include the exact document name, version reviewed and being approved.**

|  |  |  |  |
| --- | --- | --- | --- |
| **ATTUID and Name** | **Role** | **Group/Application** | **Version Approved, Approval Date and Approval Evidence** |
| *attuid – name* | Solution Architect | IT Architecture and Engineering | **Embed the email approval and make sure the exact doc name, version reviewed and approved.** |
| *attuid – name* | Architect | ACSI |  |
| *attuid – name*  *(for each impacted ACSI application)* | Developer | *Group/Application* |  |
| *attuid – name*  *(for Data Layer)* | Development Team Lead | *Group/Application* |  |
| *attuid – name*  *(for each impacted non-ACSI application)* | Developer | *Group/Application* |  |
| *Send to the TA\_TestDesign distribution list.*  *Remove if no CST-Walton application impacts.* | Test Architect | CST - Walton  (Test team for CSI – Customer Care, CSI - Order & Subscription Management, CSI – Product & Offer Management, CAM, ATLAS, SWOT BE, VRI, DITREX (GIS gateway), EAI, Jackcache and Data Layer) |  |
| *Send to the CST-Agarwal Test Architect assigned to the project.*  *Remove if no CST-Agarwal application impacts.* | Test Architect | CST - Agarwal  (Test team for CSI-Network Management, CSI-Workforce Management, CSI-Credit & Validation, CSI-Trouble Management, EBTA, CS FOBPM, CS BOBPM and Remedy Fallout Manager) |  |
| RF9578 - Bob Farmer  *Remove if no CSI or FOBPM impacts.* | Production Support Manager | Production Support | *Bob is only a reviewer (not approver).* |
| BK5747 - Brian Knop  *Remove if no CSI or FOBPM impacts.* | Production Support Manager | Production Support | *Brian is only a reviewer (not approver).* |

**Appendix A: JMS Requirements**

**Public Queues:**

| Unique ID | 1 |
| --- | --- |
| Queue Name | pub.m2e.inquiretelcoloopdetails.request |
| Status (CRUD) | Create |
| Message Size (KB) | 50Kb\* |
| Message Rate (x per y) | 1 Transaction Per Second (TPS)  22,500/day (7,500 x 3 contracts) |
| Message Expiration (sec) | 120 Seconds |
| Requirement Text | Public request queue |

| Unique ID | 2 |
| --- | --- |
| Queue Name | pub.m2e.inquiretelcoloopdetails.response |
| Status (CRUD) | Create |
| Message Size (KB) | 50Kb\* |
| Message Rate (x per y) | 1 Transaction Per Second (TPS)  22,500/day (7,500 x 3 contracts) |
| Message Expiration (sec) | 120 seconds |
| Requirement Text | Public response queue |

*\*The Message Size can be calculated as follows:*

1. *Open the schema in Altova XMLSpy.*
2. *On the menu, select DTD/Schema.*
3. *In the drop-down list, select Generate Sample XML File…*
4. *Select your desired options and select OK.*
5. *Right-click on the tab with the generated XML and select Save As. Save the generated XML file to your C: drive.*
6. *On your C: drive, look at the file properties of the generated XML file. The file size is documented in the properties.*