Requirements/Design Specification

**Existing Large Container**

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Revision** | **Description** | **Author** |
| 11/03/2014 | 1.0 | Initial Version with Business Requirements | Rich Simon |
| 12/11/2014 | 1.1 | Initial Version with Technical Requirements | Doug Bloebaum |
| 2/9/2015 | 1.2 | Merged Business and Technical Documents | Roger |
| 2/16/2015 | 1.3 | Added flow charts | Roger |
| 3/26/2015 | 1.4 | Addressed Container, STR tables and Service Criteria (Insert C) within section 3.2.2.5 | Roger |
| 4/16/2015 | 1.5 | Additions based on requirments meetings | Roger |

The information furnished herein by Republic Services Inc. is proprietary and confidential to Republic Services Inc. personnel and is not to be duplicated, published, or disclosed to any third party in whole or in part without permission from Republic Services, Inc.

© Copyright 2012, Republic Services Inc. - All rights reserved.

Table of contents

1 Business Requirements 7

1.1 Purpose of the Design Specification 7

1.1.1 Business Functional Requirements 7

1.1.2 BFR Change Log 8

1.1.3 Technical Design Requirements ETL (R & SSIS) – tackle this after pricing 9

1.1.4 Technical Design Requirements ETL (R & SSIS) 9

1.1.5 TDR Change Log 9

2 Assumptions 9

3 Requirement Notes 10

4 Technical Design 11

4.1 Referenced Documents 11

4.1.1 Usability Standards 11

4.1.2 Design Standards 11

4.2 Process Flow and Logical Model 11

4.2.1 Data Flow 11

4.2.2 Application flow 12

4.3 Pricing Logic – work in progress 22

4.3.1 Pricing Overview 22

4.3.2 Pricing Calculations 23

4.4 Data Logic – Work in Progress 24

4.4.1 New Required Data 24

4.4.2 Logic for producing required data 24

4.4.3 SSIS Packages 29

4.5 Record Counts 29

4.6 Index on the Tables 29

4.7 Batch Job Schedule and Dependency 29

4.8 Data Sources & Mapping 29

4.9 Physical Data Model 29

4.10 Framework Model 30

4.11 Validation/Error Handling 30

4.12 Interfaces 30

5 Downstream Impacts 30

5.1 InfoPro Upload 30

5.2 TIBCO/BI Reporting 30

6 Technical Architecture 30

6.1 Infrastructure Considerations 30

6.2 Data Retention 30

6.3 High Availability 30

6.4 Backup, Rollback and Recover 30

7 Other Design Specifications 30

7.1 Build/Configure Standards 30

7.2 Policies and Procedures 30

7.3 Security Design 30

7.3.1 New or Existing Security 30

7.3.2 Hierarchal Data Access 30

7.3.3 Infrastructure 30

7.4 Environmental 31

8 Appendix 31

8.1 Original prototype query (James Shrenk) 31

# Business Requirements

## Purpose of the Design Specification

The Republic Services Group “Capture” project produced a Configure-Price-Quote (CPQ) Software as a Service (SaaS) system based on a product from BigMachines, Inc (BMI). The base product was later acquired by Oracle and renamed to Oracle CPQ Cloud. The purpose of Capture is to allow RSG sales associates to produce professional-looking, accurate quotes for commercial and industrial customers. Phase 1 of the Capture release supported:

* New commercial customers
* Existing commercial customers
* New industrial customers

The next major phase of Capture will add support for:

* Existing industrial customers

This document will describe a design for gathering data from the BIDW DWCORE database about existing industrial customers so that they can be quoted accurately in the Capture tool. This data will be used by routines written in the R language, and will also be sent to the Capture platform to be used in real time during the quotation process.

### Business Functional Requirements

| **Business Functional Requirement** | **Notes** |
| --- | --- |
| 1. Existing industrial customer data from the BIDW DWCORE environment must be made available to the Capture BMIDM environment and to the Capture platform in order to allow quotes to be generated for existing industrial customers. | Updates to ETL process to bring data in are needed:  See Section 3.4 for information on changes and sample queries to be used in the ETL process |
| Account Types Supported | * Permanent, Temporary, Seasonal * Move from the quote level to the container level for small and large * Add verbiage for initial and renewal terms. (see section 4.2.2.8) * Add account type to line item grid for all pages, not just pricing (see section 4.2.2.8) |
| Supported Frequencies | * On Call * Scheduled |
| Sales Activities Supported | * New * Service Change * Rate Adjustment * Price Increase - Personally Secured * Price Increase - Contractual * Rollback due to PI * Rollback of Current Rates * Rollback due to competitive bid * Close Site (same reasons as container) * Close Account (same reasons as container) * Close Container Group * Lost to Competitor * Service Issues * Closed Business * Competitor Pricing * Price Increase * Change Of Owner |
| 1. Other Functionality | * Rate Restriction * Commission * Turning Fees on/off |
| 1. Service Revenue Screen | SR fields are below. They should support only positive numbers and follow same logic small container uses (see GH ticket #337) – do not allow negative rates   * Dry run * Relocate * Removal * Washout |
| 1. Disposal Site | Ability for the sales rep to change the disposal site. See section 3.2.2.5 Insert C |
| Generate Docs – CSA | * For seasonal we use the Perm CSA * Need the option to generate 2 CSA's per quote. There are divisions with multiple perm & temp CSAs. (See section 4.2.2.9) * Proposal – large existing has billing types, need to include? |
| 1. Compensation | Not available yet, will not address as part of scope. |
| Approvals | No logic changes |
| Reporting | * Ability to look at profitability of a customer at the industrial level |

### BFR Change Log

| **Change Item** | **Notes** |
| --- | --- |
|  |  |

### Technical Design Requirements ETL (R & SSIS) – tackle this after pricing

| **Technical Design Requirement** | **Notes** |
| --- | --- |
| 1. Data extraction processes must be written in SSIS and must coexist with the existing automated SSIS batches which support the Capture environment. |  |
| 1. Any additional processing added by this effort must not cause the overall Capture nightly data extraction and load process to finish after 07:00 ET. |  |
| 1. A new table named stg\_account\_status\_ind must be created in the BMIDM database to hold data about existing industrial containers |  |
| 1. A new SSIS process must be designed and developed to extract data from the BIDW DWCORE environment and load it into the BMIDM stg\_account\_status\_ind table |  |
| 1. A new table named (account\_status\_ind) must be created in the BMIDM database to hold output at the container group grain produced by R processes | See table structure in section 3.9 |
| 1. A new SSIS process must be designed and developed to join stg\_account\_status\_ind with (account\_status\_ind) to produce a CSV file suitable for upload to the Capture applications account\_status\_ind data table. | What happens to CSV once produced? |

### Technical Design Requirements ETL (R & SSIS)

|  |  |
| --- | --- |
| 1. Doc Engine |  |

### TDR Change Log

| **Change Item** | **Notes** |
| --- | --- |
|  |  |

# Assumptions

1. Existing industrial items that cannot be changed. If changes is desired close the container and open a new one.
   * Container Type
   * Compactor
   * Waste Type??
2. Charge type/method will not be displayed
3. DEL/REM will only be shown on line item grid, if quantity changes (should we default to $0 - Brittany)
4. For non MSW, need additional DSPs for every waste type. Will need to map every DSP to a waste type. We have a couple options
   1. Option 1 – Pick the DSP with blank waste material type. If not found pick the first disposal option.
   2. Option 2 – Support multiple disposal types / charge codes (This is out of scope for the current phase)
5. Compensation for existing containers is out of scope
6. Cost will not change

# Requirement Notes

* Look into if the STR is using Estimated vs Actual hauls. - ANSWER - Is calculated based on actual hauls per Marieke 4/2
* Is there a minimum Haul per month - Rich to follow up? ANSWER - Use the true number, round to nearest tenth all the down to 0.1 4/2/15
* Moving Account Type to Config (Rich to follow up with Mike) - ANSWER - Yes 4/14
* If move, where in config - will need multiple CSAs Brittany to check with Randy if we can put Perm and Temp on same CSA. - ANSWER - for seasonal we use the Perm CSA, and we need the option to generate 2 CSA's per quote 4/14
* Will the current division configuration determine if a second CSA option is needed or will the Existing Large customers require new CSAs that don’t currently exist in Capture? ANSWER – the current quote configuration will determin if both a perm and temp CSA is needed.
* Should we default REM/DEL to $0 - ANSWER - we will not change for either, will charge for the Haul (commercial we charge for DEL and $0 on the REM) 4/14
* Ask Rob if the line items (in line item grid) are variable. Can we dynamically show charge codes if needed. ANSWER - They are dynamic. There is a chunk of code that generates them. This code looks at a variety of different tables. Attached is a doc that explains how the lines are generated 4/14
* Billing type - Troy to forward Roger, excel doc. Roger will build a table based on the data (haul + disposal, min tonnage, etc…) Done 4/15, see figure 3.1
* Compensation - will there be differences Rand? No, this is out of scope for large existing 4/15
* May need to re-word term and rate restriction fields on pricing page – Yes, see section 4.2.2.8 4/15

Open Questions

* Suggestion: Per Al & Brittany - Existing large (flat rate)- Use 8 ton Max haul rate otherwise change to Haul + Disposal
  + Anything over 8 ton re-price - decided no 4/14
  + Create a division level table to identify the max tons for each division, with a starting value of 8 tons? - add on agenda for next meeting.
  + How many over 25 tons, over 9999 - John to follow up -
* 4/15/15 suggestion - Not permitting ownership change in Industrial? Brittany to check STR and see how many actually occur.
* Run a qry – find all industrial customers for last 12 months, look at the site and spank it against Capture data (John)
* James and/or Beckie to provide an overview of how the following are calculated
  + Cost
  + Floor
  + Average
  + Target
* 1) we need to come up with a plan for what to do with existing (work in progress) quotes when we take this live in production – moving the account type to container level will cause the same issue we ran into with moving competitor for work in progress quotes.
  + Build in a mass update for existing quotes
* 2) lets ask the group what they think about getting a call together with 6 -10 reps to run a simple POC demo by them to get there opinion.
  + Build in part of testing plan with UAT and after 1st development cycle
* Build out use cases for example for next meeting (roger/james)
* Build out pricing logic
* What is our large existing pricing strategy?

# Technical Design

## Referenced Documents

### Usability Standards

### Design Standards

## Process Flow and Logical Model

### Data Flow



### Application flow

#### Sign In Screen

No Changes – should remain the same

#### Start New Quote

Step 1) Start New Quote for Existing Customer – (No Changes)



#### Select Existing Customer

Step 2 - Enter Customer Site Information (No Changes)



Move to container config screen. Keep the name Account Type (Put toward top of config screen for new & existing containers)

#### Select Service Offering – View Existing Services



SEE BELOW INSERT B

SEE BELOW INSERT A

INSERT “A”

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Container** | | | | | | | | | |
| Container  Grp | Qty | Size | Schedule | Compactor | Est Monthly  Revenue | Total  Revenue | Avg Hauls  per Month | Avg Tons  per Haul | Avg Service  Min per Haul |
| 1 | 1 | 40 | 1 X Week | N | $1,150 | $1,500 | 4.1 | 5.2 | 82 |
| 2 | 1 | 4 | 1 X Week | N | $350 | $380 | - | - | - |

|  |  |
| --- | --- |
| **Field Sourcing Information** | |
| Container Grp | Small - No Change Large - Pull from account\_status\_ind.container\_grp\_nbr |
| Qty | Small - No Change Large - Pull from account\_status\_ind.container\_cnt |
| Size | Small - No Change Large - Pull from account\_status\_ind.container\_size |
| Schedule (lifts /period) | Small - No Change Large - Pull from account\_status\_ind.?? |
| Compactor | Small - No Change Large - Pull from account\_status\_ind.has\_compactor |
| Estimated Monthly Revenue | Small - No Change Large - Calculated |
| Total Revenue | Small - No Change Large - Pull from account\_status\_ind.container\_cnt |
| Avg Hauls per Month | Large Only - small show a "-" |
| Avg Tons per Haul | Large Only - small show a "-" |
| Avg Service Minutes per Haul | Large Only - small show a "-" |

INSERT “B”

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **STR Transaction History (Last Two Years)** | | | | | | | |
| Transaction | Core | Amount | Percent | Change  in Units | ERF | FRF | Effective |
| Price Decrease Rollback of PI | $364.59 | -$78.58 | -17.73% | 0 | N | N | 12/1/2004 |
| Price Increase Auto RPM | $443.17 | $89.20 | 25.20% | 0 | N | N | 11/1/2004 |

|  |  |
| --- | --- |
| **Field Sourcing Information** | |
| Transaction | Small - No Change Large - Pull from??? |
| Core | Small - No Change Large - Pull from??? |
| Amount | Small - No Change Large - Pull from??? |
| Percent | Small - No Change Large - Pull from??? |
| Change in Units | Small - No Change Large - Pull from??? |
| ERF | Small - No Change Large - Pull from??? |
| FRF | Small - No Change Large - Pull from??? |
| Effective | Small - No Change Large - Pull from??? |

#### Select Service Offering – Service Change

**Keep the same as small container**



SEE BELOW INSERT C

SEE BELOW INSERT D

INSERT “C”

* Columns in blue will not be shown and are for design specification only
* The following data will be pulled from the account\_status\_ind table.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Current Service | Current Service Notes and source column | New Service | New Service Notes |
| Waste Type | Solid Waste | read only | No Change | drop down defaulted to "No Change" |
| Quantity | 1 | read only | 1 | free form - type integer |
| Container Size | 40 | read only | No Change | drop down defaulted to "No Change" |
| Container Code | RO | read only | No Change | drop down defaulted to "No Change" |
| Account Type | Permanent | read only | Permanent | Can this be changed? No for the time being\* |
| Container Type | Open Top | read only | No Change | Cannot be changed (help text to outline process to change container type) |
| Frequency | Scheduled | read only | No Change | drop down defaulted to "No Change" |
| Customer Owned Container | Yes | read only (Yes/No) | Existing value | Show regardless for all container types |
| Compacter Owned | No | read only (Yes/No) | Existing value | Only show if stationary compactor (container type) is selected |
| Hauls per Month (Actual) | 4.2 | read only  float to the tenth all the way to 0.1 | Existing value | If frequency changes allow edit  If scheduled (read only) we calculate  (See Figure 3.1) |
| Disposal Site Name | Imperial L/F | read only | Imperial L/F | Read/Write, use the geo-coded minutes to price  If current site not found, default to 1st site (add help text if site is not found). Use same table, map layout and logic as New Industrial. See Figure 3.2 below. |
| Tons per Haul (Actual) | 5.2 | read only  float to the tenth all the way to 0.1 | 5 | Free form  Get from Account\_status\_ind table |
| Minutes per Haul (Actual) | 82 | read only  per InfoPro Route Data | 82 | Free form  Get from Account\_status\_ind table  (if current avg min per haul is > than geo-code minutes) then leave current avg |
| Geo-coded Minutes per Haul | 76 | read only - Bing Maps | 76 | Generated in quote. Allow override values for all three option (desired disposal site, cust site time override, round trip drive time override, disposal time override, adjust total time (read only) |
| UOM | Per Ton | read only | No Change | drop down defaulted to "No Change" |
| Billing Type | Haul + Disposal | read only | No Change | drop down defaulted to "No Change".  See figure 3.3 for Bill Types and associated fields to display for each Bill Type |
| Comp Asset Value | Do Not Display | Cannot pull this from InfoPro | Text Field | $ field, Leave blank, optional for rep to enter  Only Display if compactor container type selected (Self-Contained Compactor or Stationary Compactor) |
| One-Time Installation Charge | Do Not Display | Cannot pull this from InfoPro | Text Field | $ field, Leave blank, optional for rep to enter  Only Display if compactor container type selected (Self-Contained Compactor or Stationary Compactor) |
|  |  |  |  |  |

Figure 3.1 - Hauls per month calculations if frequency changes

|  |  |  |  |
| --- | --- | --- | --- |
| **Industrial Service Level Changes (Existing)** | | | |
| **Current Status** | **Future Status** | **Calculation Action** | **How** |
| On-Call | Scheduled | "Capture" to perform Calculation – Field Read Only | Future Pick-ups per week times 52/12 |
| On-Call | On-Call | "Capture" to perform Calculation – Field Read Only | Average Hauls per month prior 12 months (actuals) |
| Scheduled | On-Call | User to supply estimate # of hauls per month | Data field open (whole number >= 1) |
| Scheduled | Scheduled | "Capture" to perform Calculation – Field Read Only | Future Pick-ups per week times 52/12 |

Figure 3.2 – New Industrial Disposal Site Map Functionality



Figure 3.3 – Ton per Haul Fields to display for selected Billing Type (add table of UOM rules & add to disposal)

|  |  |  |  |
| --- | --- | --- | --- |
| **Billing Type (Drop Down)** | **Ton / Haul Fields to Include** | | |
| Haul + Disposal | Hauls per Month | Tons per Haul |  |
| Haul + Minimum Tonnage | Hauls per Month | Tons per Haul | Minimum Tons/Haul |
| Flat Rate + Overage | Hauls per Month | Tons Included in Haul Rate |  |

INSERT “D”



* Columns in blue will not be shown and are for design specification only
* The following data will be pulled from the account\_status\_ind table.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Current Service | Current Service Notes and source column | New Service | New Service Notes |
| Additional Paperwork | Yes | read only (Yes/No) | Existing value | Always show  Checkbox (editable) |
| Disposal Ticket Signature | No | read only (Yes/No) | Existing value | Always show  Checkbox (editable) |
| Washout On Every Haul | No | read only (Yes/No) | Existing value | Always show  Checkbox (editable) |

#### Select Service Offering – Rate Adjustment

This section is all read only once Rate Adjustment (radio button). The rate adjustment will take place on the pricing page



SEE BELOW INSERT E

INSERT “E”

|  |  |  |
| --- | --- | --- |
|  | Current Service | Current Service Notes |
| Waste Type | Solid Waste | read only |
| Quantity | 1 | read only |
| Container Size | 40 | read only |
| Container Code | RO | read only |
| Container Type | Open Top | read only |
| Service Level | Scheduled | read only |
| Customer Owned | No | read only |
| Average Hauls per Month | 4.1 | read only - actual |
| Disposal Site Name | Imperial L/F | read only |
| Average Tons per Haul | 5.2 | read only - actual |
| Average Minutes per Haul | 82 | read only - actual |
| Geo-Coded Minutes per haul | 76 | read only |

#### Select Service Offering – Close Container Group



SEE ABOVE INSERT E

#### Pricing Page

Add Account Type (Perm/Temp/Seasonal)

Add note if there are perm & temp containers . “Init/Renewal terms only apply to perm containers”



#### Generate Docs

Business Rules for CSA generation

|  |  |  |  |
| --- | --- | --- | --- |
| Existing Business | Line of Business | Scenario | CSA |
| Permanent | Industrial | Adding temp (large) | Show temp CSA (do not display the perm cans) |
| Temporary | Industrial | Adding temp (large) | Show new container on temp CSA |
| Temporary | Industrial | Adding perm  (large or small) | Show perm CSA |
| Perm or Temp | Industrial | Adding/Changing temp and perm service | Generate temp and perm CSAs |

We will need the ability to do print multiple CSA’s on the Gen Docs page. Display:

* CSA Version drop down box when there are multiple permenant CSAs available and a perm container is quoted
* Temp CSA Version drop down box when there are multiple temporary CSAs available and a temp container is quoted
* On Gen Docs Page, if CSA checkbox is selected, display a dropdown to allow options of showing/printing Perm, Temp or Both; if multiple container types are quoted.



Add “Perm” in front of CSA Version



Temp

#### Finalize Page

How are the transaction reason code calculated? Any updates needed to the underlying tables?

## Pricing Logic

### Pricing Overview

Existing Container Pricing Overview

* Service change
  + Hold margin constant between current service configuration and revised service configuration
  + Allow for division configured mark up/down at Floor, Average, and Target levels
  + When current price is below new business floor, allow division configured parameter to recover some or all of the difference
  + If price is below floor we want to try to recover some price
  + Price is above target??

**Scenario 1**

Customer has a haul rate of $240, disposal rate of $35.

Want to go from OC to Scheduled Service



**Scenario 1**

Customer contracts Republic to change service level from O/C to 1 time per week service on his 30 yard Open Top Container. Customer’s current Haul Rate is $ 240.00 per haul and Disposal Rate of $ 35.00/ton with $120.00 month rental rate.

Note, we currently do not create guardrails for rentals, but the rental revenue is being applied to the overall financial returns, so losing the rental could cause the haul rate to balloon up. The following was included just for the sake of conversation.

New Business Guardrails For this Customer

$240

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Haul Rate | $135.00 | | $175.00 | | $220.00 | | $265.00 | |
|  |  |  |  |  |  |  |  |  |
|  | Cost | | Floor | | Average | | Target | |
|  |  |  |  |  |  |  |  |  |
| Dispoasl Rate | $23.00 | | $35.00 | | $37.00 | | $39.00 | |
|  |  |  |  |  |  |  |  |  |
|  | Cost | | Floor | | Average | | Target | |
|  |  |  |  |  |  |  |  |  |
|  | $50.00 | | $100.00 | | $120.00 | | $150.00 | |
| Rent Charge |  |  |  |  |  |  |  |  |
|  | Cost | | Floor | | Average | | Target | |

Gaurdrails Should Be?

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Haul Rate | $135.00 | |  | |  | |  | |
|  |  |  |  |  |  |  |  |  |
|  | Cost | | Floor | | Average | | Target | |
|  |  |  |  |  |  |  |  |  |
| Dispoasl Rate | $23.00 | |  | |  | |  | |
|  |  |  |  |  |  |  |  |  |
|  | Cost | | Floor | | Average | | Target | |
|  |  |  |  |  |  |  |  |  |
|  | $50.00 | |  | |  | |  | |
| Rent Charge |  |  |  |  |  |  |  |  |
|  | Cost | | Floor | | Average | | Target | |

**Scenario # 2**

Customer contracts Republic to change service level from O/C to 1 time per week service on his 30 yard Open Top Container. Customer’s current Haul Rate is $ 240.00 per haul and Disposal Rate of $ 35.00/ton with $ 120.00 month rental rate and has a competitive bid more a local hauler where the quote is has the haul rate of $ 180.00/haul, disp. rate of $ 36.00 and free rental on the container.

New Business Guardrails For this Customer

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Haul Rate | $135.00 | | $175.00 | | $220.00 | | $265.00 | |
|  |  |  |  |  |  |  |  |  |
|  | Cost | | Floor | | Average | | Target | |
|  |  |  |  |  |  |  |  |  |
| Dispoasl Rate | $23.00 | | $35.00 | | $37.00 | | $39.00 | |
|  |  |  |  |  |  |  |  |  |
|  | Cost | | Floor | | Average | | Target | |
|  |  |  |  |  |  |  |  |  |
|  | $50.00 | | $100.00 | | $120.00 | | $150.00 | |
| Rent Charge |  |  |  |  |  |  |  |  |
|  | Cost | | Floor | | Average | | Target | |

Gaurdrails Should Be?

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Haul Rate | $135.00 | |  | |  | |  | |
|  |  |  |  |  |  |  |  |  |
|  | Cost | | Floor | | Average | | Target | |
|  |  |  |  |  |  |  |  |  |
| Dispoasl Rate | $23.00 | |  | |  | |  | |
|  |  |  |  |  |  |  |  |  |
|  | Cost | | Floor | | Average | | Target | |
|  |  |  |  |  |  |  |  |  |
|  | $50.00 | |  | |  | |  | |
| Rent Charge |  |  |  |  |  |  |  |  |
|  | Cost | | Floor | | Average | | Target | |

* Rollback
  + Price Rollback of Current Rates (General Save)
    - Floor is lowest of cost + division configured margin or customer’s current revenue (if lower than cost)
    - Target is higher of customer’s current revenue or new business floor
    - Division can configure % of contribution dollars to save at each guardrail level
  + Price rollback to PI
    - Retain an increasing percentage of the PI at each guardrail
    - Intent is that floor should never be below the customer’s previous rate and Target never above their current rate with the PI
    - Use only for PI issues; if customer is concerned about overall price, use rollback due to price
  + Price rollback to competitive bid
    - Take the higher of competitive bid vs new business floor and use as the floor price
    - If Customer is paying less than floor, use their current revenue as floor instead
    - Average and Target guardrails retrain an additional % of the gap between the customer’s current price and the calculated floor
    - Months remaining in the contract can be used as a basis of demurrage (markup on floor)
* Close Container Group
  + No affect on pricing

Large Exising Pricing Formulas

* Service Change
* Rollback
  + Price Rollback of current rates
    - Floor
      * apply(cbind(revenue,(cost\_to\_serve\_month + (save\_base\_margin\_adj \* curr\_margin\_dollars))),1,min)
      * min(Direct cost + floor margin to save, Current Revenue)
    - Ave
      * apply(cbind(cost\_to\_serve\_month,(price\_base\_adj + abs(((revenue - apply(cbind(price\_base\_adj,cost\_to\_serve\_month),1,max))) \* save\_targ\_pct\_retain))),1,max)
      * Max(Direct Cost,Calculated floor price + abs(Current revenue - Max(Calculated floor price,Direct cost)) \* Avg Retain %
    - Target
      * apply(cbind((cost\_to\_serve\_month/(1-base\_margin\_new)),(price\_base\_adj + (abs((revenue -apply(cbind(price\_base\_adj,cost\_to\_serve\_month),1,max))) \* save\_str\_pct\_retain))),1,max)
      * Max(Newbusiness floor,Calculated floor price + abs(Current revenue - Max(Calculated floor price,Direct cost)) \* Target Retain %)
  + Price rollback to PI
  + Price rollback to competitive bid

apply(cbind(revenue,(cost\_to\_serve\_month + (save\_base\_margin\_adj \* curr\_margin\_dollars))),1,min)

min(Direct cost + floor margin to save, Current Revenue)

What if we cannot identify the disposal site? 4/15

Suggestion –

* Look in sites table first to see if it’s there
* If it is not default Third Party
* (ensure all Republic sites are in the table first)

Pricing is categorized into

* Service
* Rental
* Disposal
* One Time Charges
  + Delivery
  + Removal
  + Others…

Categories that identify revenue

         Haul Codes - 3PF, REC, REG, RE4

         Disposal Codes - DIS, DSP, DSR, EOF, HHW, OVR, RDS, RMF, VRY

         Rental Codes – CMR, CPL, DM2, DM3, DM5, DM7, DMR, LEA, MLF, R02, R31, RC3, REC, REG, REN

Other Categories James has (pass through, just on line item grid as FYI, not in monthly totals or proposal or CSA, and are editable on line item grid)

         Commodity Charge Codes

         Fee Charge Codes

         Rebate CC

         Service

         Bulk

         Credits

         Maintenance – washouts

         Site Charge Codes – Liners, etc

         Other

         Tax

If multiple codes do we show them separately and sum up (For example two REGs - my vote yes).

### Pricing Calculations

Service Calculations

|  |  |
| --- | --- |
| Cost |  |
| Floor | always at their current price |
| Average |  |
| Target | max(floor \* (1+ spread),market rate) |
| Price | always set to average |
| ERF |  |
| FRF |  |
| Total |  |

Rental Calculations

|  |  |
| --- | --- |
| Cost |  |
| Floor |  |
| Average |  |
| Target |  |
| Price |  |
| ERF |  |
| FRF |  |
| Total |  |

Disposal Calculations

|  |  |
| --- | --- |
| Cost |  |
| Floor |  |
| Average |  |
| Target |  |
| Price |  |
| ERF |  |
| FRF |  |
| Total |  |

One Time Charges

|  |  |
| --- | --- |
| Cost |  |
| Floor |  |
| Average |  |
| Target |  |
| Price |  |
| ERF |  |
| FRF |  |
| Total |  |

## Data Logic – Work in Progress

### New Required Data

This section will highlight new data fields to be brought in and will need to updated in Mapping Document

New Tables

* account\_status\_ind – need DDL

Rates should be pulled from rpt\_rate\_hist

Current filters on the rates? Don’t think so.

Current filters on accounts?

### Logic for producing required data

use DWCORE

-- Step 1: produce Fact\_Invoice\_Detail temp table -- drop table ##fid

SELECT fid.receipt\_nbr,

fid.invoice\_detail\_pk,

fid.revenue\_period\_sk,

fid.invoice\_dt\_sk,

fid.invoice\_from\_dt\_sk,

fid.invoice\_to\_dt\_sk,

fid.invoice\_amt,

fid.container\_qty,

fid.disposal\_qty,

fid.haul\_qty,

fid.qty\_billed,

fid.disposal\_unit\_rate\_amt,

fid.disposal\_ticket\_nbr,

fid.invoice\_freq\_nbr,

fid.disposal\_unit\_of\_measure,

fid.acct\_fee\_sk,

fid.invoice\_note,

fid.corp\_hier\_sk,

fid.acct\_sk,

dcg.Infopro\_Div\_Nbr,

dcg.Acct\_Nbr,

dcg.Site\_Nbr,

dcg.Container\_Grp\_Nbr,

dcg.Orig\_Start\_Dt,

dcg.is\_Container\_Owned,

dcg.is\_Oncall,

dcg.has\_Compactor,

dcg.Container\_Cd,

dcg.Container\_Nm,

dcg.Excess\_Ton\_Amt,

s.Employee\_SK,

s.Postal\_Cd + s.Postal\_Cd AS zip,

s.LATITUDE,

s.LONGITUDE,

s.Original\_Open\_Dt,

s.Contract\_Term,

s.Contract\_Status,

s.Effective\_Dt,

s.Expiration\_Dt,

s.Contract\_Nbr,

fid.container\_grp\_sk\_2,

fid.charge\_cd\_sk,

fid.site\_sk,

fid.service\_cd\_sk,

fid.service\_dt\_sk,

fid.rate\_sk,

h.Cur\_Div\_Nbr,

h.Cur\_Infopro\_Div\_Nbr,

h.Cur\_LOB\_Category,

dcc.Charge\_Cd,

dcc.Charge\_Cd\_Desc,

dcc.Charge\_Method,

dcc.Charge\_Method\_Desc,

dcc.Charge\_Typ,

dcc.Charge\_Typ\_Desc,

a.Acct\_Type

INTO ##fid\_t

FROM Fact\_Invoice\_Detail fid

INNER JOIN Dim\_Container\_Grp dcg

ON fid.Container\_Grp\_SK\_2 = dcg.Container\_Grp\_SK

INNER JOIN Dim\_Site s

ON fid.Site\_SK=s.Site\_SK

INNER JOIN Dim\_Acct a

ON fid.Acct\_SK=a.Acct\_SK

INNER JOIN Dim\_Corp\_Hier h

ON fid.Corp\_Hier\_SK = h.Corp\_Hier\_SK

INNER JOIN Dim\_Charge\_Cd dcc

ON fid.Charge\_Cd\_SK=dcc.Charge\_Cd\_SK

WHERE dcg.container\_cd IN ( 'IR', 'RO', 'RE', 'RS', 'SC' )

AND dcg.container\_size BETWEEN 10 AND 100

AND fid.is\_deleted = 0

AND fid.is\_updated = 0

AND h.Sub\_LOB BETWEEN 100 AND 199 -- 71,040,858 rows in 5m35s; 8m09s (into); 34m09s; 54m15s; 49m43s

select \* into ##fid from ##fid\_t where revenue\_period\_sk BETWEEN 201310 AND 201409 -- 18,822,008 rows in 2m40s

drop table ##fid\_t

-- Step 2: produce Fact\_Service\_Detail temp table -- drop table ##fsd

SELECT fsd.load\_seq\_nbr,

fsd.service\_route\_minutes,

fsd.service\_dump\_minutes,

fsd.service\_miles\_qty,

fsd.is\_disposal\_allow,

fsd.special\_handling\_cd AS fsd\_special\_handling\_cd,

fsd.has\_compactor AS fsd\_has\_compactor,

fsd.container\_size AS fsd\_container\_size,

fsd.service\_cd\_sk,

fsd.service\_dt\_sk,

fsd.Container\_Grp\_SK\_2,

dcg.has\_compactor AS dcg\_has\_compactor,

dcg.container\_size AS dcg\_container\_size,

dcg.special\_handling\_cd AS dcg\_special\_handling\_cd,

dcg.Infopro\_Div\_Nbr,

dcg.Acct\_Nbr,

dcg.Site\_Nbr,

dcg.Container\_Grp\_Nbr

INTO ##fsd

FROM fact\_service\_detail fsd

INNER JOIN dim\_corp\_hier h

ON fsd.corp\_hier\_sk = h.corp\_hier\_sk

INNER JOIN dim\_container\_grp dcg

ON fsd.container\_grp\_sk\_2 = dcg.container\_grp\_sk

WHERE fsd.Container\_Size BETWEEN 10 AND 100

AND fsd.service\_dt\_sk BETWEEN 20131001 AND 20140930

AND fsd.is\_Updated = 0

AND fsd.is\_Deleted = 0

AND dcg.container\_cd IN ( 'IR', 'RO', 'RE', 'RS', 'SC' )

AND h.Sub\_LOB BETWEEN 100 AND 199 -- 13m44s 5,249,254 rows; 32m30s 5,249,254 rows with joins; 12m4s INTO; 5,053,977 rows in 19m56s; 5,053,977 in 16m16s

-- Step 3: produce Fact\_Landfill\_Detail temp table -- drop table ##fld

SELECT dcg.infopro\_div\_nbr,

dcg.Acct\_Nbr,

dcg.Site\_Nbr,

dcg.Container\_Grp\_Nbr,

fld.disposal\_cd,

fld.disposal\_price\_cd,

fld.disposal\_time\_start,

fld.disposal\_time\_end,

fld.disposal\_ticket\_nbr,

fld.Process\_Dt\_SK

INTO ##fld

FROM fact\_landfill\_detail fld

INNER JOIN dim\_container\_grp dcg

ON fld.container\_grp\_sk = dcg.Container\_Grp\_SK

INNER JOIN dim\_corp\_hier h

ON fld.corp\_hier\_sk = h.corp\_hier\_sk

WHERE fld.Process\_Dt\_SK BETWEEN 20131001 AND 20140931

AND fld.is\_deleted = 0

AND fld.is\_updated = 0

AND dcg.container\_cd IN ( 'IR', 'RO', 'RE', 'RS', 'SC' )

AND dcg.container\_size BETWEEN 10 AND 100 -- 2m24s 4,231,604 rows; 4,231,604 rows in 15s; 4,231,635 rows in 44s

-- Step 4: produce Fact\_Sales\_Activity temp table

SELECT dcg.infopro\_div\_nbr,

dcg.acct\_nbr,

dcg.site\_nbr,

dcg.container\_grp\_nbr,

fsa.sales\_activity\_period\_sk,

trc.txn\_cd,

trc.reason\_cd,

trc.txn\_reason\_desc,

dc.competitor\_nm

into ##fsa

FROM dbo.fact\_sales\_activity AS fsa

INNER JOIN dim\_txn\_reason\_cd AS trc

ON fsa.txn\_reason\_cd\_sk = trc.txn\_reason\_cd\_sk

INNER JOIN dbo.dim\_competitor AS dc

ON fsa.competitor\_sk = dc.competitor\_sk

INNER JOIN dim\_container\_grp dcg

ON fsa.container\_grp\_sk=dcg.container\_grp\_sk

WHERE trc.txn\_cd = 1

AND fsa.is\_deleted = 0

AND fsa.is\_updated = 0

AND dcg.container\_cd IN ( 'IR', 'RO', 'RE', 'RS', 'SC' )

AND dcg.container\_size BETWEEN 10 AND 100

AND dcg.orig\_start\_dt BETWEEN '2013-10-01' AND '2014-09-30' -- 340,727 rows in 37s; 4m58s; 340,757 rows in 1m49s

create index ##fid\_i ON ##fid(Infopro\_Div\_Nbr,acct\_nbr,Site\_Nbr,Container\_Grp\_Nbr,Invoice\_Dt\_SK,Disposal\_Ticket\_Nbr,Service\_Dt\_SK,Service\_Cd\_SK) -- 19s

create index ##fld\_i ON ##fld(Infopro\_Div\_Nbr,Acct\_Nbr,Site\_Nbr,Container\_Grp\_Nbr,process\_dt\_sk,Disposal\_Ticket\_Nbr) -- 3s

create index ##fsd\_i ON ##fsd(infopro\_div\_nbr,Acct\_Nbr,Site\_Nbr,Container\_Grp\_Nbr,Service\_Dt\_SK,Service\_Cd\_SK) -- 3s

create index ##fsa\_i ON ##fsa(Sales\_Activity\_Period\_SK,Infopro\_Div\_Nbr,Acct\_Nbr,Site\_Nbr,Container\_Grp\_Nbr) -- 1s

DECLARE @account\_status\_ind TABLE(

);

INSERT INTO @account\_status\_ind

select fid.Cur\_Div\_Nbr,

fid.Cur\_Infopro\_Div\_Nbr,

fid.Acct\_Nbr,

fid.Site\_Nbr,

fid.zip,

fid.LATITUDE,

fid.LONGITUDE,

fid.Container\_Grp\_Nbr,

fid.is\_Container\_Owned,

fid.is\_Oncall,

fid.Charge\_Cd,

fid.Charge\_Cd\_Desc,

fid.receipt\_nbr,

Row\_number() OVER (partition BY fid.invoice\_detail\_pk ORDER BY fsd.load\_seq\_nbr) AS row1,

fid.charge\_method,

fid.charge\_method\_desc,

fid.charge\_typ,

fid.charge\_typ\_desc,

fld.disposal\_cd,

fld.disposal\_price\_cd,

fld.disposal\_time\_start,

fld.disposal\_time\_end,

fid.[revenue\_period\_sk],

fid.[invoice\_dt\_sk],

fid.[invoice\_from\_dt\_sk],

fid.[invoice\_to\_dt\_sk],

fid.[invoice\_amt],

fid.[container\_qty],

fid.[disposal\_qty],

fid.[haul\_qty],

fid.[qty\_billed],

fid.[disposal\_unit\_rate\_amt],

fid.[disposal\_ticket\_nbr],

fid.[invoice\_freq\_nbr],

fid.[disposal\_unit\_of\_measure],

fsd.service\_route\_minutes,

fsd.service\_dump\_minutes,

fsd.service\_miles\_qty,

fsd.is\_disposal\_allow,

fid.orig\_start\_dt,

fsa.txn\_cd,

fsa.reason\_cd,

fsa.txn\_reason\_desc,

fsa.competitor\_nm,

CASE

WHEN Datediff(m, fid.orig\_start\_dt, CONVERT(DATETIME, CONVERT(VARCHAR(8), fid.invoice\_dt\_sk), 112)) <= 6 THEN 1

ELSE 0

END AS new\_customer\_cg\_flag,

COALESCE(fsd.fsd\_special\_handling\_cd, fsd.dcg\_special\_handling\_cd) AS special\_handling\_cd,

CASE

WHEN fsd.fsd\_special\_handling\_cd IN ( 'A', 'D', 'S', 'X', '6', '4' ) THEN 1

ELSE 0

END AS special\_waste,

COALESCE(fsd.fsd\_has\_compactor, fsd.dcg\_has\_compactor) AS has\_compactor,

COALESCE(fsd.fsd\_container\_size, fsd.dcg\_container\_size) AS container\_size,

fid.Container\_Cd,

fid.Container\_Nm,

fid.Cur\_LOB\_Category,

CASE

WHEN fid.Acct\_Type IN ( 'P', 'B', 'C', 'E', 'L' ) THEN 'Permanent'

WHEN fid.Acct\_Type = ( 'S' ) THEN 'Seasonal'

WHEN fid.Acct\_Type = ( 'I' ) THEN 'Intercompany'

WHEN fid.Acct\_Type IN ( 'T', 'X' ) THEN 'Temporary'

ELSE fid.Acct\_Type

END AS acct\_type,

fid.Excess\_Ton\_Amt,

drh.Rate\_Eff\_Dt,

daf.Fee\_Type,

daf.Fee\_Type\_Desc,

daf.is\_Locked,

CASE

WHEN fid.Acct\_Fee\_SK > 0 THEN 1

ELSE 0

END AS is\_fee,

CASE

WHEN fid.Invoice\_Note = '@ERFONFRF@' THEN 1

ELSE 0

END AS is\_erf\_on\_frf,

fid.Original\_Open\_Dt AS site\_open\_dt,

fid.Contract\_Term,

fid.Contract\_Status,

fid.Effective\_Dt AS contract\_start\_dt,

fid.Expiration\_Dt AS contract\_end\_dt,

fid.Contract\_Nbr,

e.Employee\_EIN

into ##result

from ##fid fid

LEFT OUTER JOIN ##fld fld

ON fid.Infopro\_Div\_Nbr=fld.Infopro\_Div\_Nbr

AND fid.acct\_nbr=fld.Acct\_Nbr

AND fid.Site\_Nbr=fld.Site\_Nbr

AND fid.Container\_Grp\_Nbr=fld.Container\_Grp\_Nbr

AND fid.Invoice\_Dt\_SK=fld.process\_dt\_sk

AND fid.Disposal\_Ticket\_Nbr=fld.Disposal\_Ticket\_Nbr

LEFT OUTER JOIN ##fsd fsd

ON fid.Infopro\_Div\_Nbr=fsd.infopro\_div\_nbr

AND fid.Acct\_Nbr=fsd.Acct\_Nbr

AND fid.Site\_Nbr=fsd.Site\_Nbr

AND fid.Container\_Grp\_Nbr=fsd.Container\_Grp\_Nbr

AND fid.Service\_Dt\_SK=fsd.Service\_Dt\_SK

AND fid.Service\_Cd\_SK=fsd.Service\_Cd\_SK -- type 1 dimension, so this is ok

LEFT OUTER JOIN ##fsa fsa

ON DATEPART(YEAR,fid.Orig\_Start\_Dt)\*100 + DATEPART(MONTH,fid.Orig\_Start\_Dt)=fsa.Sales\_Activity\_Period\_SK

AND fid.Infopro\_Div\_Nbr=fsa.Infopro\_Div\_Nbr

AND fid.Acct\_Nbr=fsa.Acct\_Nbr

AND fid.Site\_Nbr=fsa.Site\_Nbr

AND fid.Container\_Grp\_Nbr=fsa.Container\_Grp\_Nbr

LEFT OUTER JOIN dim\_rate\_hist drh

ON fid.Rate\_SK=drh.rate\_sk

LEFT OUTER JOIN dim\_acct\_fee daf

ON fid.Acct\_Fee\_SK=daf.Acct\_Fee\_SK

LEFT OUTER JOIN dbo.dim\_employee e

ON fid.Employee\_SK=e.Employee\_SK -- 21,353,629 rows in 8m22s

DELETE FROM ##result WHEre row1!=1 -- get rid of duplicates

--select count(\*) from ##result where row1=1 -- 18,822,008 rows

### SSIS Packages

An SSIS package must be developed to produce the account\_status\_ind table, update certain columns within it from the (TBD) table, and produce a CSV output file to send to Capture. Its flow is as follows:

The “Process Report XLS File” Data Flow Task looks like:



## Record Counts

Record count of account\_status\_ind from 20131001 to 20140930 is approximately 19 million.

## Index on the Tables

TBD

## Batch Job Schedule and Dependency

TBD

## Data Sources & Mapping

For detailed mapping information, please refer to the [Enterprise Mapping Document](http://itpmo-2010projects/BI%20Program%20Strategy/3.0%20Design/Mapping/Enterprise%20Mapping%20Document.xls).

## Physical Data Model

See the [Enterprise Mapping Document](http://itpmo-2010projects/BI%20Program%20Strategy/3.0%20Design/Mapping/Enterprise%20Mapping%20Document.xls) for column and data type details. Also see the test tables created in devbmisql01.BMIDM

Preliminary account\_status\_ind table structure is as follows

James

## Framework Model

No Changes.

## Validation/Error Handling

No Changes.

## Interfaces

Not Applicable.

# Downstream Impacts

## InfoPro Upload

## TIBCO/BI Reporting

TBD

# Technical Architecture

## Infrastructure Considerations

No changes to infrastructure.

## Data Retention

No changes to data retention.

## High Availability

Not Applicable.

## Backup, Rollback and Recover

No changes to backup and recovery procedures.

# Other Design Specifications

## Build/Configure Standards

Reference Aldon procedures for BI.

## Policies and Procedures

Conforms to all published IT policies and procedures.

## Security Design

### New or Existing Security

No Changes.

### Hierarchal Data Access

No Changes.

### Infrastructure

No Changes.

## Environmental

No Additional environmental requirements.

# Appendix

## Original prototype query (James Shrenk)

SELECT a.\*

FROM (SELECT b.cur\_div\_nbr,

b.cur\_infopro\_div\_nbr,

c.acct\_nbr,

f.site\_nbr,

f.postal\_cd AS zip,

d.container\_grp\_nbr,

d.is\_container\_owned,

d.is\_oncall,

e.charge\_cd,

e.charge\_cd\_desc,

a.receipt\_nbr,

Row\_number()

OVER (

partition BY a.invoice\_detail\_pk

ORDER BY h.load\_seq\_nbr) AS row1,

e.charge\_method,

e.charge\_method\_desc,

e.charge\_typ,

e.charge\_typ\_desc,

g.disposal\_cd,

g.disposal\_price\_cd,

g.disposal\_time\_start,

g.disposal\_time\_end,

a.[revenue\_period\_sk],

a.[invoice\_dt\_sk],

a.[invoice\_from\_dt\_sk],

a.[invoice\_to\_dt\_sk],

a.[invoice\_amt],

a.[container\_qty],

a.[disposal\_qty],

a.[haul\_qty],

a.[qty\_billed],

a.[disposal\_unit\_rate\_amt],

a.[disposal\_ticket\_nbr],

a.[invoice\_freq\_nbr],

a.[disposal\_unit\_of\_measure],

h.service\_route\_minutes,

h.service\_dump\_minutes,

h.service\_miles\_qty,

h.is\_disposal\_allow,

d.orig\_start\_dt,

j.txn\_cd,

j.reason\_cd,

j.txn\_reason\_desc,

j.competitor\_nm,

CASE

WHEN Datediff(m, d.orig\_start\_dt, CONVERT(DATETIME, CONVERT(VARCHAR(8), a.invoice\_dt\_sk), 112)) <= 6 THEN 1

ELSE 0

END AS new\_customer\_cg\_flag,

CASE

WHEN h.[special\_handling\_cd] IS NULL THEN d.special\_handling\_cd

ELSE h.special\_handling\_cd

END AS special\_handling\_cd,

CASE

WHEN h.special\_handling\_cd IN ( 'A', 'D', 'S', 'X',

'6', '4' ) THEN 1

ELSE 0

END AS special\_waste,

CASE

WHEN h.[has\_compactor] IS NULL THEN d.has\_compactor

ELSE h.has\_compactor

END AS has\_compactor,

CASE

WHEN h.[container\_size] IS NULL THEN d.container\_size

ELSE h.container\_size

END AS container\_size,

d.container\_cd,

d.container\_nm,

b.cur\_lob\_category,

CASE

WHEN c.acct\_type IN ( 'P', 'B', 'C', 'E', 'L' ) THEN 'Permanent'

WHEN c .acct\_type = ( 'S' ) THEN 'Seasonal'

WHEN c.acct\_type = ( 'I' ) THEN 'Intercompany'

WHEN c.acct\_type IN ( 'T', 'X' ) THEN 'Temporary'

ELSE c.acct\_type

END AS acct\_type,

d.excess\_ton\_amt,

k.rate\_eff\_dt,

m.fee\_type,

m.fee\_type\_desc,

m.is\_locked,

CASE

WHEN a.acct\_fee\_sk > 0 THEN 1

ELSE 0

END AS is\_fee,

CASE

WHEN a.invoice\_note = '@ERFONFRF@' THEN 1

ELSE 0

END AS is\_erf\_on\_frf,

f.original\_open\_dt AS site\_open\_dt,

f.contract\_term,

f.contract\_status,

f.effective\_dt AS contract\_start\_dt,

f.expiration\_dt AS contract\_end\_dt,

f.contract\_nbr,

n.employee\_ein

FROM [dbo].[fact\_invoice\_detail] AS a

INNER JOIN dbo.dim\_corp\_hier AS b

ON a.corp\_hier\_sk = b.corp\_hier\_sk

INNER JOIN dbo.dim\_acct AS c

ON a.acct\_sk = c.acct\_sk

INNER JOIN dbo.dim\_container\_grp AS d

ON a.container\_grp\_sk\_2 = d.container\_grp\_sk\_2

INNER JOIN dbo.dim\_charge\_cd AS e

ON a.charge\_cd\_sk = e.charge\_cd\_sk

INNER JOIN dbo.dim\_site AS f

ON a.site\_sk = f.site\_sk

LEFT JOIN dbo.fact\_landfill\_detail AS g

ON a.acct\_sk = g.acct\_sk

AND a.disposal\_ticket\_nbr = g.disposal\_ticket\_nbr

AND a.site\_sk = g.site\_sk

AND a.container\_grp\_sk\_2 = g.container\_grp\_sk\_2

AND a.invoice\_dt\_sk = g.process\_dt\_sk

LEFT JOIN dbo.fact\_service\_detail AS h

ON a.container\_grp\_sk\_2 = h.container\_grp\_sk\_2

AND a.service\_cd\_sk = h.service\_cd\_sk

AND a.service\_dt\_sk = h.service\_dt\_sk

INNER JOIN dbo.dim\_service\_cd AS i

ON a.service\_cd\_sk = i.service\_cd\_sk

LEFT JOIN (SELECT a.acct\_sk,

a.site\_sk,

a.container\_grp\_sk,

a.sales\_activity\_period\_sk,

b.txn\_cd,

b.reason\_cd,

b.txn\_reason\_desc,

c.competitor\_nm

FROM dbo.fact\_sales\_activity AS a

INNER JOIN dbo.dim\_txn\_reason\_cd AS b

ON a .txn\_reason\_cd\_sk = b.txn\_reason\_cd\_sk

INNER JOIN dbo.dim\_competitor AS c

ON a.competitor\_sk = c.competitor\_sk

WHERE b.txn\_cd = 1

AND a.is\_deleted = 0

AND a.is\_updated = 0

AND a.is\_merged = 0

AND a.is\_split = 0) AS j

ON Datepart (year, d.orig\_start\_dt) \* 100 + Datepart(month, d.orig\_start\_dt) = j.sales\_activity\_period\_sk

AND a.acct\_sk = j.acct\_sk

AND a.site\_sk = j.site\_sk

AND a.container\_grp\_sk\_2 = j.container\_grp\_sk

LEFT JOIN dbo.dim\_rate\_hist AS k

ON a.rate\_sk = k.rate\_sk

LEFT JOIN dbo.dim\_acct\_fee AS m

ON a.acct\_fee\_sk = m.acct\_fee\_sk

LEFT JOIN dbo.dim\_employee AS n

ON f.employee\_sk = n.employee\_sk

WHERE 1 = 1

AND sub\_lob BETWEEN 100 AND 199

AND a.is\_deleted = 0

AND a.is\_updated = 0

AND a.is\_split = 0

AND a.is\_merged = 0

AND c.is\_franchise = 0

AND c.is\_national\_account = 0

AND d.container\_cd IN ( 'IR', 'RO', 'RE', 'RS', 'SC' )

AND d.container\_size BETWEEN 10 AND 100

AND a.revenue\_period\_sk BETWEEN 201310 AND 201409) AS a

WHERE a.row1 = 1