Requirements/

**DWCORE Data for Capture Existing Industrial**

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 |
|  |  |  |  |
|  |  |  |  |

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 6

1.1 Purpose of the Design Specification 6

1.1.1 Business Functional Requirements 6

1.1.2 BFR Change Log 6

1.1.3 Technical Design Requirements 6

1.1.4 TDR Change Log 6

2 Assumptions 7

3 Technical Design 7

3.1 Referenced Documents 7

3.1.1 Usability Standards 7

3.1.2 Design Standards 7

3.2 Process Flow and Logical Model 7

3.3 Functional Logic 7

3.4 Record Counts 11

3.5 Index on the Tables 11

3.6 Batch Job Schedule and Dependency 11

3.7 Data Sources & Mapping 11

3.8 Physical Data Model 11

3.9 Framework Model 11

3.10 Validation/Error Handling 11

3.11 Interfaces 11

4 Report Changes 11

5 Technical Architecture 11

5.1 Infrastructure Considerations 11

5.2 Data Retention 12

5.3 High Availability 12

5.4 Backup, Rollback and Recover 12

6 Other Design Specifications 12

6.1 Build/Configure Standards 12

6.2 Policies and Procedures 12

6.3 Security Design 12

6.3.1 New or Existing Security 12

6.3.2 Hierarchal Data Access 12

6.3.3 Infrastructure 12

6.4 Environmental 12

7 Appendix 12

# 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. Data about existing industrial customers 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. | See the appendix for the original prototype query from James Shrenk of the pricing team. |

### BFR Change Log

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

### Technical Design Requirements

| **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 |  |
| 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? |

### TDR Change Log

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

# Assumptions

# Technical Design

## Referenced Documents

### Usability Standards

### 

### Design Standards

## Process Flow and Logical Model

Review w/ James

### Data Flow



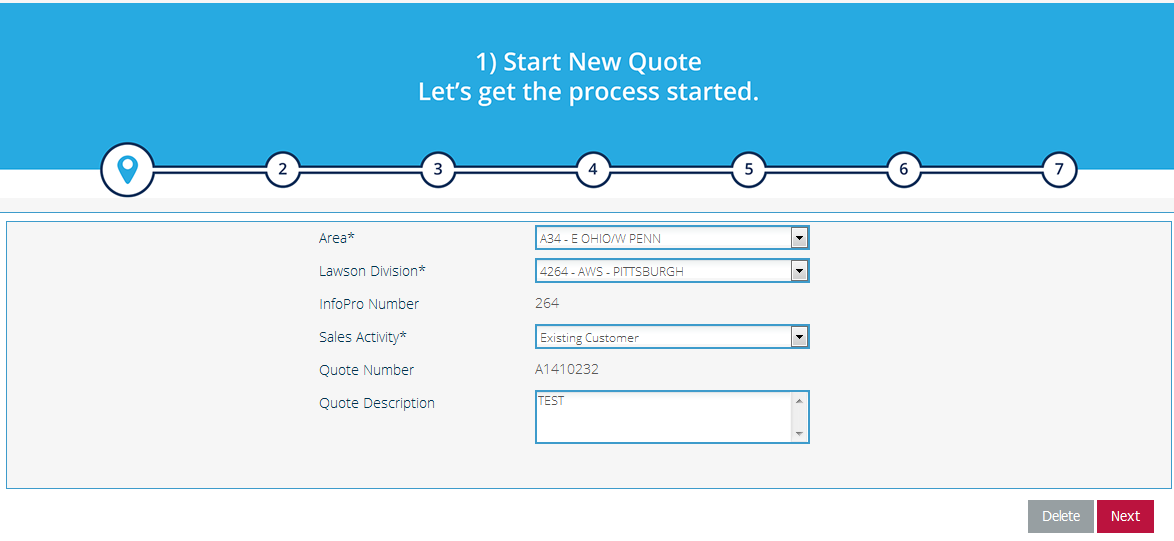
### Application flow

#### 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)



#### Select Service Offering – View Existing Services



SEE BELOW INSERT B

SEE BELOW INSERT A

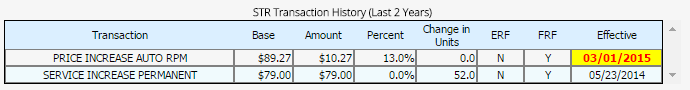
INSERT “A”



INSERT “B” (need to build out query to include charge code for small,how is it getting charged (monthly, perservice, by the ton, etc) )



Current STR for small



#### Select Service Offering – Service Change



SEE BELOW INSERT C

SEE BELOW INSERT D

INSERT “C”

Columns in blue not be shown and are for design specification only

Questions:

on Small not all dropdown fields default to "No Change", some default to the value

for non drop downs do we still want default of "No Change", the type of the column may not allow it

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Current Service | Current Service Notes | 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 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" |
| Container Type | Open Top | read only | No Change | drop down defaulted to "No Change" |
| Service Level | Scheduled | read only | On Call | Service Level is currently called Frequency |
| Customer Owned | No | read only | No Change | currently a check box, be we leave as is? |
| Average hauls per month (Actual) | 4.1 | read only  most recent 12 months | 4.1 | where do we get this data? |
| Disposal Site Name | Imperial L/F | read only | Imperial L/F | do we give them an option to change, if so how? |
| Average Tons per Haul (Actual) | 5.2 | read only  most recent 12 months | 5.2 | where do we get this data? |
| Average Minutes per Haul (Actual) | 82 | read only  per InfoPro Route Data | 82 | where do we get this data? |
| Geo-coded Minutes per Haul | 76 | read only - Bing Maps | 76 | where do we get this data? |

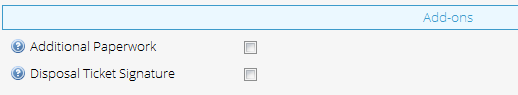
Waste Type will need to select the waste type (since a single container can have many waste types) – based on disposal ticket and disposal site

Need a mapping for all these fields (it is a different then small cont)

Average hauls & avg tons per haul - only support integer not float.

Add geo-code number to new service avg minues (if current avg min per haul is > than geo-code minutes) then leave current avg.

INSERT “D”



#### 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

## Functional Logic

### SQL Scripts

#### 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

## Framework Model

No Changes.

## Validation/Error Handling

No Changes.

## Interfaces

Not Applicable.

# Report Changes

# 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