CRM (Customer Relationship Management)

Solution Design Document

October 4, 2012

Prepared For:

ANN INC.

ANN TAYLOR | LOFT

Contact:

Mark Beening

Technical Lead

mark.bening@epsilon.com

847.592.1151

TABLE OF CONTENTS

Purpose of the Document

The System Design Document (SDD) is the technical description used to construct and maintain the Customer Relationship Management (CRM) solution being implemented by Epsilon for Phase 1. The System Design Document contains references to additional documents that contain portions of the technical description. This document is primarily intended for ANN INC and the Epsilon development organization. This document is also useful reference material for anyone who has a stake in understanding the solution design such as business systems analysts, production analysts, developers, and project managers.

By providing sign-off on this document, ANN INC. confirms that they have reviewed this document, that the information contained herein is accurate and complete, and that any solution delivered to meet these requirements will thus meet their requirements for that solution. The change control process defined in the Master Services Agreement must be used to update the contents of this document after the first version has been approved by both ANN INC. and Epsilon.

Document approval should be provided in **Approvals.**

Introduction

The System Design Document outlines the system architecture, data models, and the source to target mappings as part of the Build Phase 1 development efforts to be undertaken by Epsilon on behalf of ANN, INC for the CRM Solution. The design defined in this document represents the complete scope of the implementation project to which Epsilon shall adhere unless otherwise agreed to by both ANN INC and Epsilon.

The System Design Document is based on the business requirements detailed in the Requirements Definition Document (see **Supporting Documents**) agreed to between ANN INC and Epsilon. The RDD is used to define all objects and rules into the System Design Document from which the database build originates. Each requirement identified in the RDD is directly related to one or more detailed design considerations that describe how the solution addresses each business requirement.

Anything not addressed in this document or the Requirements Definition Document is out of scope and not included in the final Build Phase 1 delivery. By signing this document, ANN INC confirms they have reviewed this document, that the information contained herein is accurate and complete, and that any solution delivered to meet this design will meet the Final Acceptance criteria as described in the Build Phase Statement of Work (SOW) for the CRM Solution.

Supporting Documents

Document Title	Document Filename	Document Description
Requirements Definition Document (RDD)	BRD_v10_ANN_20120619_Final_CRM_Approved.docx	Requirements for the build of the CRM Solution.
Interface Agreements	ANN INC InterfaceAgreement Layouts 20120924.xlsx	Inbound feed descriptors
Source to Staging Target Mapping	ANN Inc Source to Stage Mapping 20120924.xlsx	Details the rules for populating source feeds into staging tables.
Staging to Account Work Target Mapping	ACCOUNT Stage to Work Mapping 20120924.xlsx	Details the rules for updating the Data Warehouse with data loaded into the staging tables.
Staging to AW Work Target Mapping	AuditWorks TXN Stage to Work Mapping_20120924.xlsx	Account/Agility Source transformations for building Customer subject area in Data Warehouse.
AW Work to Data Warehouse Target Mapping	AuditWorks TXN Work to Data Warehouse Mapping 20120924.xlsx	AuditWorks POS Transaction Transformations
ATG Staging to Data Warehouse Target Mapping	ATG TXN Stage to Data Warehouse Mapping 20120924.xlsx	ATG E-Commerce Transaction Transformations
MarketWorks Transaction Migration to Data Warehouse Target Mapping	MarketWorks TXN Stage to Data Warehouse Mapping 20120924.xlsx	MarketWorks Migration Transaction Transformations
MarketWorks Product Migration to Data Warehouse Target Mapping	MW Product Stage to Data Warehouse Mapping 20120924.xlsx	MarketWorks Migration Product Transformations
MarketWorks Campaign Migration to Data Warehouse Target Mapping	DW Historical Campaign Taxonomy Mappings 20120924.xlsx	MarketWorks promotion history migration mappings
Reference Table Staging to Data Warehouse Target Mapping	Reference Tables Stage to Data Warehouse Mapping_final 20120924.xlsx	Data Warehouse Transformations other sources: SDL, CheetahMail, BazaarVoice, RightNow, KBM, Experian
SAP Staging to Data Warehouse Target Mapping	SAP Product Stage to Data Warehouse Mapping_20120924.xlsx	SAP Product, Store, and Promo Data Warehouse transformations
Data Warehouse Campaign Load	DW Campaign Taxonomy Mappings 20120924.xlsx	Loading campaign data from SAS into the DW
Data Warehouse to Data Mart Target Mapping	DM Campaign Taxonomy Mappings 20120924.xlsx DM Dimension Mappings 20120924.xlsx DM Fact Mappings 20120924.xlsx DM Summary Mappings 20120924.xlsx DM Xref Mappings 20120924.xlsx	Details the rules for updating the Data Mart with data loaded into the Data Warehouse.
Data Warehouse Physical Model	ANN_DW_MODEL_20120924.docx	ER Diagram for the Data Warehouse.

Document Title	Document Filename	Document Description
Data Mart Physical Model	ANN_DM_MODEL_20120924.docx	ER Diagram for the Data Mart.

Document Revision History

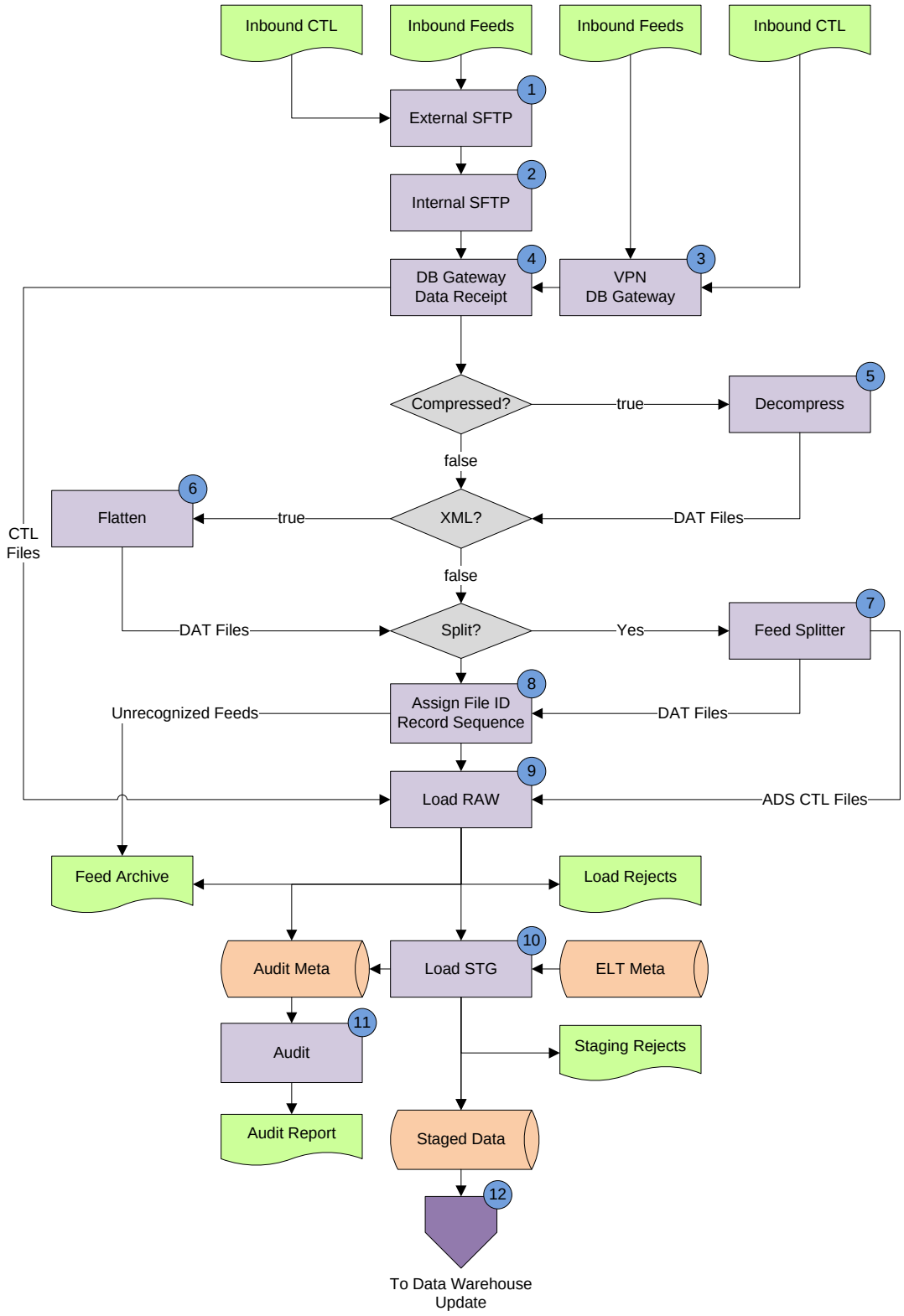
Date	Author	Version	Summary of Change
7/29/2012	Mark Beening	1.0	Initial draft compiled for review with ANN INC
8/7/2012	Mark Beening	1.1	Edits from on-site SDD review
9/18/2012	Mark Beening	1.2	Edits from ANN INC feedback: Dan Williams, Sue Lee, Suresh Kancherla, Geoff Oliver, Dianne Binford
9/24/2012	Mark Beening	1.3	Added detailed RMB process flows
10/3/2012	Mark Beening	1.4	Incorporated final SDD feedback modifications

Implementation/Update Design Specification

Data Receipt and Staging Design

The data receipt and staging design section describes how the solution addresses requirements related to inbound and outbound feeds. This phase of the solution begins with the external data providers in terms of how data is fed into the system and what the processes are that get the data into a state in which it is ready to be loaded into the Data Warehouse. It explains the scenarios that occur if expected feed Interface Agreement requirements are not met as well as the process auditing mechanisms put in place to monitor ongoing quality of the update process.

Process Flow



Process Flow Annotations

1. Inbound feeds from Vendor sources include ADS (non-PCI feeds), BazaarVoice, CheetahMail, Experian, KBM, Omniture, RightNow, and SDL. These are posted by 6am EST to a dedicated feed landing zone for each vendor. The feed transmissions are encrypted via SSH and authentication is performed by vendor public key sharing.
2. Once the transmission of a Vendor feed completes, it is immediately moved to an internal SFTP counterpart positioned inside the Epsilon firewall. Once this operation is complete, the transferred data is no longer accessible from any external (internet facing) source.
3. Inbound feeds from ANN include SAP, ADS (PCI feeds), ATG, AuditWorks, and Marketworks. These are posted by 1am EST to the a dedicated feed landing zone within the dedicated ANN production environment. This transfer is performed via SSH encryption with dual-level encryption provided by the Corp/Corp VPN tunnel. Once a feed transfer completes, it is immediately moved to the inbound feed loading zone where it joins the vendor feeds and awaits processing.
4. As feeds are posted to the ANN-dedicated DB Gateway server loading zone the file names are appended with a unique timestamp to facilitate the ability to trap and appropriately respond to feeds that were posted multiple times (either intentionally or unintentionally). The data files are provided in various formats appropriate to each data provider as defined in the Interface Agreements. Whenever available within the source systems, control files are delivered as an additional layer of quality control intended for validation of the data transfer process as well as source system feed extract accuracy. At 1am EST, the load process is initiated to accept all of the daily, weekly, monthly, quarterly, and annual feeds that have arrived for processing that day.
5. Inbound feeds are recognized and decompressed using the appropriate supported utility (GZIP, PKZIP, RAR).
6. Feeds are provided in XML format from ATG and BazaarVoice. Custom utilities have been developed to flatten well-formed XML structures and generate delimited-file feeds for each node while preserving the hierarchical nature of the data through the assignment of node relationship IDs. The number of flattened node structures that get generated are variable based on the XSD format.
7. Several feeds provided into the solution have multiple structures based upon a Record Type indicator within the data itself. In order to load this data for processing in the database, each type must be split out into a separate feed. This applies to AuditWorks, SDL Event feed, and ADS. Additionally, ADS provides data control mechanisms within the feed that are parsed out and leveraged as control files within our load process.
8. The file naming convention is parsed out to determine its intended purpose in the update. At this time, each file is assigned a file ID as well as a unique record identifier that is appended to each record and is used to track an original source record through the entire update process. And further facilitate research tasks within the database.
9. Each feed is loaded into the database into a schema that temporarily houses the data in it's raw format. These are tables that exactly mirror the source feed format defined in the Interface Agreements except that all data types are defined a Text Only. Any rejects that occur at this stage are the result of data exceeding the maximum allowable length of one or more fields in the source layout. This results in the rejection of the entire record. A rejects file accumulates these and includes a descriptor that indicates which field caused the reject.
10. The Staging process loads each feed into a separate schema that also mirrors the source feed structure except that the data types on these tables match those of the fields they will eventually map into in the Data Warehouse structures. Any rejects that occur during this stage are the result

of invalid data type conversions. These are also posted out to reject files with appropriate reason codes that can be used for later analysis. Note: Before the subsequent update occurs, the Staging tables are reinitialized to prepare for the next load. All successfully loaded data is trimmed of whitespace and converted to uppercase except where explicitly noted (i.e. email addresses are lowercase). SAS CI provides a mechanism to convert data into mixed/proper casing. This is defined by the user at campaign extraction time.

11. The Load/Staging audit process performs several checks to verify that data has been prepared for update into the Data Warehouse is accurate. This includes:
    a. Verification that all expected feeds are accounted for
    b. Source Extract vs. Actual File Counts (via Control File validation)
    c. Data gap/overlap analysis for incremental feeds driven on dates (i.e. Transaction)
    d. Multiple feeds posted for a single type
    e. Source feed Primary Key verification
    f. Record rejection rates within tolerance defined by source and feed
    g. Field-Level Data Profiling and Trending
12. Assuming no automated audit failures occur on a feed identified as Critical to the update, the process continues on to the Data Warehouse Update process, otherwise the update process is halted and the anomalies are immediately escalated to the source system.

1. Data Transfer

<table>
<tr><th></th><th>Design</th></tr>
<tr><td>BU1.1</td><td>Daily Incremental will represent data including all activity through the end of business hours TBD. Specific details to be defined during the Design Phase and captured in the Solution Design Document</td></tr>
<tr><td>BU1.1.1</td><td>Daily incremental feeds are expected to include all data spanning the end of the previous business cycle through end of the current business cycle. These are extracted and transmitted to Epsilon at the end of that cycle. The trigger to indicate the universe of data that is extracted incrementally varies by source and feed, but is typically the change of one or more attributes associated to a feed's primary key. This is referred to as the Activity Date.</td></tr>
<tr><td>BU1.1.2</td><td>The update cycle will conform to the following schedule:
Day 1 – feeds arrive by 6am EST. This is based on customer activity that occurred the previous day
Day 2 – mart available 7am EST (6am EST on Monday). This provides updated customer activity to the user community 1 day in arrears (Sunday transactions are available at the start of business on Tuesday morning)</td></tr>
<tr><td>BU1.1.3</td><td>The External Retail Matchback feeds (REA, NZA, TAA) are returned and loaded on the day following their extraction</td></tr>
<tr><td>BU1.1.4</td><td>Where sources provide an Activity Date, an audit is performed as part of the load process to identify potential gaps in the data relative to the previous load(s). The following feeds include an Activity Date and perform this daily audit:
<table>
<tr><th>Source</th><th>Type</th><th>Freq</th><th>Activity Date</th></tr>
<tr><td>RIGHTNOW</td><td>Call Center Incident/Customer</td><td>Daily</td><td>CREATED_DT</td></tr>
<tr><td>BV</td><td>Product Review ANN Taylor</td><td>Daily</td><td>CRDATE</td></tr>
<tr><td>BV</td><td>Product Review LOFT</td><td>Daily</td><td>CRDATE</td></tr>
<tr><td>SDL</td><td>Bulk Mail Send Events</td><td>Daily</td><td>TIMESTAMP</td></tr>
<tr><td>SDL</td><td>Event Based Mail Send Events</td><td>Daily</td><td>TIMESTAMP</td></tr>
<tr><td>SDL</td><td>Open Events</td><td>Daily</td><td>TIMESTAMP</td></tr>
<tr><td>SDL</td><td>Click Events</td><td>Daily</td><td>TIMESTAMP</td></tr>
<tr><td>SDL</td><td>Unsubscribe Events</td><td>Daily</td><td>TIMESTAMP</td></tr>
<tr><td>SDL</td><td>Transaction Events</td><td>Daily</td><td>TIMESTAMP</td></tr>
<tr><td>SDL</td><td>Subscriptions</td><td>Daily</td><td>DATETIME_SUB</td></tr>
<tr><td>SDL</td><td>Un-Subscriptions</td><td>Daily</td><td>DATETIME_UNSUB</td></tr>
<tr><td>SDL</td><td>Change of Demographics</td><td>Daily</td><td>DATE_CHANGED</td></tr>
<tr><td>SAP</td><td>Associate Reference</td><td>Daily</td><td>DATE_TIME</td></tr>
<tr><td>SAP</td><td>Promotion</td><td>Daily</td><td>DATE_TIME</td></tr>
<tr><td>OMNI</td><td>Visits ANN Taylor</td><td>Daily</td><td>VISIT_DATE</td></tr>
<tr><td>OMNI</td><td>Visits Loft</td><td>Daily</td><td>VISIT_DATE</td></tr>
<tr><td>OMNI</td><td>Visit by Product Category ANN Taylor</td><td>Daily</td><td>VISIT_DATE</td></tr>
<tr><td>OMNI</td><td>Visit by Product Category Loft</td><td>Daily</td><td>VISIT_DATE</td></tr>
<tr><td>ATG</td><td>ECOM Transaction - Orders</td><td>Daily</td><td>LASTMODIFIEDDATE</td></tr>
<tr><td>ATG</td><td>ECOM Transaction - Returns</td><td>Daily</td><td>CREATED_DATE</td></tr>
<tr><td>ATG</td><td>ECOM Catalog Requests</td><td>Daily</td><td>REQUESTED_DATE</td></tr>
<tr><td>AW</td><td>POS Transaction Header (Type H)</td><td>Daily</td><td>TRANSACTION_DATE</td></tr>
<tr><td>AW</td><td>POS Transaction Line (Type L)</td><td>Daily</td><td>TRANSACTION_DATE</td></tr>
<tr><td>AW</td><td>POS Merchandise Detail (Type M)</td><td>Daily</td><td>TRANSACTION_DATE</td></tr>
<tr><td>AW</td><td>POS Discount Detail (Type D)</td><td>Daily</td><td>TRANSACTION_DATE</td></tr>
<tr><td>AW</td><td>POS Return Detail (Type R)</td><td>Daily</td><td>TRANSACTION_DATE</td></tr>
<tr><td>AW</td><td>POS Authorization Detail (Type A)</td><td>Daily</td><td>TRANSACTION_DATE</td></tr>
<tr><td>AW</td><td>POS Customer Detail (Type C)</td><td>Daily</td><td>TRANSACTION_DATE</td></tr>
</table></td></tr>
</table>

	AW	POS Expanded Customer Detail (Type E)	Daily	TRANSACTION_DATE	
	AW	POS Line Notes (Type N)	Daily	TRANSACTION_DATE	

BU1.1.5

The following daily source/feeds extract incremental feeds through a process that does not persist as an attribute of the record. For these the *Activity Date* is the current daily system load date. They are not subject to the Activity Date audit defined above.

Source	Type	Freq	Activity Date
ADS	Customer Daily PLCC	Daily	sysdate
ADS	Customer Daily CO Brand	Daily	Sysdate
ADS	Customer Xref Daily PLCC	Daily	Sysdate
ADS	Customer Xref Daily COBRAND	Daily	Sysdate
SDL	IID Keys	Daily	Sysdate
CM	Loader Exception	Daily	Sysdate
SDL	Alterian Aggregate	Daily	Sysdate
ATG	ECOM Customer Profile	Daily	sysdate
EXP	Name/Zip Address Append	Daily	sysdate

BU1.1.6

No weekly feeds have an available activity date.

BU1.1.7

The following weekly source/feeds extract incremental feeds through a process that does not persist as an attribute of the record. For these the *Activity Date* is the current daily system load date. They are not subject to the Activity Date audit defined above.

Source	Type	Freq	Activity Date
OMNI	SAINT File ANN Talyor	Daily	Sysdate
OMNI	SAINT File Loft	Daily	Sysdate
EXP	Reverse Email Append	Weekly	Sysdate
EXP	Trade Area Append	Weekly	Sysdate

BU1.1.8

Where sources provide an Activity Date, an audit it performed as part of the load process to identify potential gaps in the data relative to the previous load(s). The following feeds include an Activity Date and perform this weekly audit:

Source	Type	Freq	Activity Date
SAP	Location Hierarchy (Store)	Weekly	DATE_TIME
SAP	Vendor	Weekly	DATE_TIME
SAP	Style Attributes	Weekly	DATE_TIME
SAP	Product Hierarchy	Weekly	DATE_TIME

BU1.1.9

The following monthly source/feeds extract incremental feeds through a process that does not persist as an attribute of the record. For these the *Activity Date* is the current daily system load date. They are not subject to the Activity Date audit defined above.

Source	Type	Freq	Activity Date
ADS	Customer Monthly PLCC (type 0)	Monthly	sysdate
ADS	Customer Monthly PLCC (type 7)	Monthly	sysdate
ADS	Customer Monthly PLCC (type 9)	Monthly	sysdate
ADS	Customer Monthly COBRAND (type 0)	Monthly	sysdate
ADS	Customer Monthly COBRAND (type 7)	Monthly	sysdate
ADS	Customer Monthly COBRAND (type 9)	Monthly	sysdate

BU1.1.10

No annual/quarterly feeds have an available activity date.

BU1.1.11

The following annual/quarterly source/feeds extract incremental feeds through a process that does not persist as an attribute of the record. For these the *Activity Date* is the current daily system load date. They are not subject to the Activity Date audit defined above.

Source	Type	Freq	Activity Date
KBM	Demographic Append (ongoing)	Annual / Quarterly	sysdate

BU1.1.12

Migration data with a viable activity date should reflect the data date accurately in history. An audit is performed to insure no migration data overwrites any ongoing incremental data.

Source	Type	Freq	Activity Date
CRM	Promotion History	n/a	PROMO_DATE
CRM	Taxonomy	n/a	PROMO_DATE

BU1.1.13

Migration data without a native activity date leverages a hardcoded date that represents the date the data was extracted (typically the latest transaction date in the migration dataset). This is used to compare data from incremental sources with that from MarketWorks.

Note: the primary keys for these target tables (ACCOUNT, PRODUCT, TXN_HEADER, TXN_ITEM, TXN_ITEM, RMB_IDENTIFIER, TENDER) are designed just that they will not overwrite each other.

Source	Type	Freq	Activity Date
MW	Address	n/a	migration data date
MW	Class	n/a	migration data date
MW	Color	n/a	migration data date
MW	Customer	n/a	migration data date
MW	Customer_Alternate_Key	n/a	migration data date
MW	Customer_Xref	n/a	migration data date
MW	Department	n/a	migration data date
MW	Division	n/a	migration data date
MW	Product_Xref	n/a	migration data date
MW	Style	n/a	migration data date
MW	Style_Xref	n/a	migration data date
MW	Tender	n/a	migration data date
MW	Tender_Mapping	n/a	migration data date
MW	TI_Transaction_Detail_Error	n/a	migration data date
MW	TI_Transaction_Header_Error	n/a	migration data date
MW	TI_Transaction_Tender_Error	n/a	migration data date
MW	TI_Transaction_Coupon_Error	n/a	migration data date
MW	Transaction_Coupon	n/a	migration data date
MW	Transaction_Detail	n/a	migration data date
MW	Transaction_Header	n/a	migration data date
MW	Attribute_Grouping	n/a	migration data date
MW	Attribute_List	n/a	migration data date
MW	Customer_Attribute	n/a	migration data date
MW	Email_POS_Arch	n/a	migration data date
MW	Transaction_Detail_XREF	n/a	migration data date

BU1.2

All feeds included in the daily update are made available to Epsilon by TBD PM EST daily for the update to be available by TBD AM EST in the CRM. Specific details to be defined during the Design Phase and captured in the Solution Design Document

BU1.2.1

The load process is scheduled to run daily at 1am EST. At this time, all feeds that have been transmitted are identified by naming convention. All daily feeds are identified and loaded in raw format into the staging area in preparation for load into the data warehouse.

Source	Type	Freq
ADS	Customer Daily PLCC	Daily
ADS	Customer Daily CO Brand	Daily
ADS	Customer Xref Daily PLCC	Daily
ADS	Customer Xref Daily COBRAND	Daily

RIGHTNOW	Call Center Incident/Customer	Daily
BV	Product Review ANN Taylor	Daily
BV	Product Review LOFT	Daily
SDL	IID Keys	Daily
SDL	Bulk Mail Send Events	Daily
SDL	Event Based Mail Send Events	Daily
SDL	Open Events	Daily
SDL	Click Events	Daily
SDL	Unsubscribe Events	Daily
SDL	Transaction Events	Daily
SDL	Subscriptions	Daily
SDL	Un-Subscriptions	Daily
SDL	Change of Demographics	Daily
CM	Loader Exception	Daily
SDL	Alterian Aggregate	Daily
SAP	Associate Reference	Daily
SAP	Promotion	Daily
OMNI	Visits ANN Taylor	Daily
OMNI	Visits ANN Loft	Daily
OMNI	Visit by Product Category ANN Taylor	Daily
OMNI	Visit by Product Category Loft	Daily
OMNI	SAINT File ANN Talyor	Daily
OMNI	SAINT File Loft	Daily
ATG	ECOM Customer Profile	Daily
ATG	ECOM Transaction - Orders	Daily
ATG	ECOM Transaction - Returns	Daily
ATG	ECOM Catalog Requests	Daily
AW	POS Trasaction Header (Type H)	Daily
AW	POS Trasaction Line (Type L)	Daily
AW	POS Merchandise Detail (Type M)	Daily
AW	POS Discount Detail (Type D)	Daily
AW	POS Return Detail (Type R)	Daily
AW	POS Authorization Detail (Type A)	Daily
AW	POS Customer Detail (Type C)	Daily
AW	POS Expanded Customer Detail (Type E)	Daily
AW	POS Line Notes (Type N)	Daily
EXP	Name/Zip Address Append	Daily

BU1.2.2

Feeds identified as weekly are loaded on received on **Saturday by 1am**
The Experian feeds are returned as soon as the external append process completes.

Source	Type	Freq
SAP	Location Hierarchy (Store)	Weekly
SAP	Vendor	Weekly
SAP	Style Attributes	Weekly
SAP	Product Hierarchy	Weekly
EXP	Reverse Email Append	Weekly
EXP	Trade Area Append	Weekly

BU1.2.3

Feeds identified as monthly are loaded on the **First Saturday of the Month by 1am**

Source	Type	Freq
ADS	Customer Monthly PLCC (type 0)	Monthly
ADS	Customer Monthly PLCC (type 7)	Monthly
ADS	Customer Monthly PLCC (type 9)	Monthly
ADS	Customer Monthly COBRAND (type 0)	Monthly
ADS	Customer Monthly COBRAND (type 7)	Monthly

<table>
<tr><td></td><td>
<table>
<tr><td>ADS</td><td>Customer Monthly COBRAND (type 9)</td><td>Monthly</td></tr>
</table>
</td></tr>
<tr><td>BU1.2.4</td><td>Feeds identified as annual are loaded prior to UAT with full refreshes performed relative to acceptance. (quarterly incremental for Demographic is also relative to acceptance.)
<table>
<tr><th>Source</th><th>Type</th><th>Freq</th></tr>
<tr><td>KBM</td><td>Demographic Append (ongoing)</td><td>Annual Refresh / Quarterly Changes</td></tr>
<tr><td>ANN</td><td>Fiscal Calendar</td><td>Annual</td></tr>
</table>
</td></tr>
<tr><td>BU1.2.5</td><td>Initial Demographic load may be done from a pre-processed dataset provided for Marketworks. In this case, the marketworks customer_ID will be resolved to Postal Contact ID as a one-time load.</td></tr>
<tr><td>BU1.3</td><td>Epsilon will establish SFTP inbound and outbound transfer sites for ANN INC.’s data center</td></tr>
<tr><td>BU1.3.1</td><td>ANN, INC’s inbound data transfers are managed via SSH public key authentication through the Corp/Corp VPN into the Database Gateway Server, PC1UANNTETL03</td></tr>
<tr><td>BU1.3.2</td><td>ANN, INC’s inbound data is PUSHED into the Epsilon-hosted system for the following sources:
MarketWorks – Migration Data
AuditWorks – POS Transaction Data (includes PCI)
ATG – Ecommerce Transaction Data (includes PCI)
SAP – Product, Store, Associate Data (includes PCI)
ADS – PCI feeds (ADS account/CC cross-reference)</td></tr>
<tr><td>BU1.4</td><td>Epsilon will establish SFTP inbound and outbound data transfers with ADS</td></tr>
<tr><td>BU1.4.1</td><td>ADS inbound data transfers are managed through the shared SFTP infrastructure through the anntads user authenticated via SSH public key</td></tr>
<tr><td>BU1.4.2</td><td>ADS inbound data is PUSHED into the Epsilon hosted system</td></tr>
<tr><td>BU1.4.3</td><td>ADS outbound data transfers are managed through the shared SFTP infrastructure. SSH key authentication is performed through the retc0152 user hosted on est.alliancedata.com</td></tr>
<tr><td>BU1.4.4</td><td>CheetahMail outbound data transfers are PUSHED into the CheetahMail-hosted system, est.alliancedata.com:/to_alliance</td></tr>
<tr><td>BU1.5</td><td>Epsilon will establish SFTP inbound and outbound data transfers with RightNow</td></tr>
<tr><td>BU1.5.1</td><td>RightNow inbound data transfers are managed through the shared SFTP infrastructure through the anntrtnow user authenticated via SSH public key</td></tr>
<tr><td>BU1.5.2</td><td>RightNow inbound data is PUSHED into the Epsilon hosted system</td></tr>
<tr><td>BU1.6</td><td>Epsilon will establish SFTP inbound and outbound data transfers with BazaarVoice</td></tr>
<tr><td>BU1.6.1</td><td>BazaarVoice inbound data transfers are managed through the shared SFTP infrastructure through the anntbv user authenticated via SSH public key</td></tr>
<tr><td>BU1.6.2</td><td>BazaarVoice inbound data is PUSHED into the Epsilon hosted system</td></tr>
<tr><td>BU1.7</td><td>Epsilon will establish SFTP inbound and outbound data transfers with CheetahMail</td></tr>
<tr><td>BU1.7.1</td><td>CheetahMail inbound data transfers are managed through the shared SFTP infrastructure through the anntcm user authenticated via SSH public key</td></tr>
<tr><td>BU1.7.2</td><td>CheetahMail inbound data is PUSHED into the Epsilon hosted system</td></tr>
<tr><td>BU1.7.3</td><td>CheetahMail outbound data transfers are managed through the shared SFTP infrastructure. SSH key authentication is performed through the Epsilon user hosted on tt.cheetahmail.com</td></tr>
<tr><td>BU1.7.4</td><td>CheetahMail outbound data transfers are PUSHED into the CheetahMail-hosted system, tt.cheetahmail.com:/to_cheetah</td></tr>
<tr><td>BU1.8</td><td>Epsilon will establish SFTP inbound and outbound data transfers with SAP</td></tr>
<tr><td>BU1.8.1</td><td>SAP data transfers are managed through the ANN, INC VPN protocol addressed in BU1.3.1</td></tr>
</table>

BU1.9	Epsilon will establish SFTP inbound and outbound data transfers with Omniture
BU1.9.1	Omniture inbound data transfers are managed through the shared SFTP infrastructure through the anntomni user authenticated via SSH public key
BU1.9.2	Omniture inbound data is PUSHED into the Epsilon hosted system
BU1.10	Epsilon will establish SFTP inbound and outbound data transfers with ATG
BU1.10.1	ATG data transfers are managed through the ANN, INC VPN protocol addressed in **BU1.3.1**
BU1.11	Epsilon will establish SFTP inbound and outbound data transfers with KBM Group
BU1.11.1	KBM inbound data transfers are managed through the shared SFTP infrastructure through the anntkbm user authenticated via SSH public key
BU1.11.2	KBM inbound data is PUSHED into the Epsilon hosted system
BU1.11.3	KBM outbound data transfers are managed through the shared SFTP infrastructure. Named user authentication is being leverage through the **Epsilon3701** user hosted on securexfer.kbm1.com
BU.1.11.4	KBM will provide SSH key authentication functionality as soon as the technology becomes available on their systems.
BU1.11.5	KBM outbound data transfers are PUSHED into the KBM-hosted system, securexfer.kbm1.com:/inbound
BU1.11.1	Epsilon will establish SFTP inbound and outbound data transfers with Experian
BU1.11.1.1	Experian inbound data transfers are managed through the shared SFTP infrastructure through the anntexp user authenticated via SSH public key
BU1.11.1.2	Experian inbound data is PUSHED into the Epsilon hosted system
BU1.11.1.3	Experian outbound data transfers are managed through the shared SFTP infrastructure. SSH key authentication is performed through the **eema1151** user hosted on STM.EXPERIAN.COM
BU1.11.1.4	Experian outbound data transfers are PUSHED into the Experian-hosted system, stm.experian.com:to_xpn
BU1.11.2	Epsilon will establish SFTP inbound and outbound data transfers with SDL
BU1.11.2.1	SDL inbound data transfers are managed through the shared SFTP infrastructure through the anntalt user authenticated via SSH public key
BU1.11.2.2	SDL inbound data is PUSHED into the Epsilon hosted system
BU1.11.2.3	SDL outbound data transfers are managed through the shared SFTP infrastructure. SSH key authentication is performed through the **neyAD2YeDA74u** user hosted on xfer.alterianconnect.com
BU1.11.2.4	SDL outbound data transfers are PUSHED into the Experian-hosted system, defined in in **Appendix G – External Access**
BU1.12	Epsilon will provide ANN CRM with SFTP capabilities for outbound data transfers with ANN print vendors
BU1.12.1	SFTP access is provided through the Corp/Corp VPN tunnel into a dedicated data landing zone on the Database Gateway (PC1UANNETL03)
BU1.12.2	Data extracted from the Campaign Management/Business Intelligence tool and made available to the client by the following process: 1. Data extracted from SAS CI, SAS VDD, or Business Objects is posted to a dedicated data directory on the SAS Server. 2. The directory is polled persistently such that as long as a data transfer is not currently in progress, it will immediately transfer the file to the ANN accessible data area. Files needing to be pushed to the client accessible site are identified as those lacking a '*.sent' file extension. 3. These files are pushed via SFTP to the database gateway server. Once successful transmission has been verified via a source/target checksum operation, the source file is renamed as with a '*.sent' extension 4. After a period of 1 week, the '*.sent' files are deleted.

	5. A User then logs into the Database Gateway host and retrieves the dataset via an SFTP pull.
BU1.12.3	Per business requirement 1.12, this requirement refers to all vendors
BU1.13	ANN INC. requires SFTP user IDs and passwords into the file landing environment
BU1.13.1	Named users are maintained for all external sources. SSH public key validation is the preferred method of authentication; however passwords can be supplied by the Epsilon Support Team upon request.
BU1.13.2	Initial named service accounts have been setup as follows: • ANN, INC - anntaylor • ADS – anntads • BazaarVoice – anntbv • CheetahMail – anntcm • Experian – anntexp • KBM – anntkbm • Omniture – anntomni • RightNow – anntrn • SDL – anntalt
BU1.13.3	Named SFTP user accounts for individual SAS/BO users are setup as part of the user creation process.
BU1.14	For all inbound and outbound automated SFTP data transfers, authentication is done via SSH keys with each external system including all ANN INC. sources as well as ADS, RightNow, BazaarVoice, CheetahMail, SAP, Omniture, ATG, KBM Group and Experian
BU1.14.1	Inbound SSH public keys have been shared with Epsilon from the following sources: • ADS • ANN, INC • BazaarVoice • CheetahMail • Experian • KBM Group • Omniture • RightNow • SDL
BU1.14.2	Outbound SSH and RSH public keys have been shared with the following sources: • ADS • CheetahMail • Experian • SDL
BU1.14.3	The following Outbound source is unable to support SSH key authentication at this time. They are in the process of upgrading their infrastructure and will migrate us as the technology becomes available. In the meantime, we authenticate via named user/password combination: • KBM Group
BU1.15	Data files will be zipped using GZIP or PKZIP compression
BU1.15.1	The load process identifies file compression method by named extension and linux ‘file’ utility

	• Gzip • Unzip Note: the solution also supports multi-part RAR as the compression method used for MarketWorks.
BU1.15.2	Unsupported compression methods or supported methods requiring a password to uncompress result in a failed load process for that feed an invoking the Process Exception **Escalation Procedure**
BU1.16	Data files are transmitted using SSL encryption
BU1.16.1	All inbound and outbound feeds transferred over the internet or through the Corp/Corp VPN tunnel are performed via SFTP protocol on Port 22. This performs SSH encryption in real time as the data is transferred. External transfers are managed through one of the following hosts: • ASLAN – Internet • PC1UANNTETL03 – VPN
BU1.16.2	All internal Epsilon intra-system transfers are also performed via SFTP on Port 22.
BU1.17	Historical data files from MarketWorks will conform to the following naming convention; ▪ TABLENAME.dmp ▪ TABLENAME – for example (ADDRESS, CLASS, COLOR, etc..) ▪ Tables will arrive as native oracle table dumps
BU1.17.1	Marketworks Non-PCI migration data is provided in Oracle Dump format. The DMP files are loaded into the ANN_MIGRN_USER schema in the Data Warehouse node. Table names match the DMP file names • ADDRESS • ATTRIBUTE_GROUPING • ATTRIBUTE_LIST • CLASS • COLOR • CUSTOMER • CUSTOMER_ATTRIBUTE • CUSTOMER_XREF • DEPARTMENT • DIVISION • PRODUCT_XREF • STORE • STYLE • STYLE_XREF • TENDER • TENDER_MAPPING • TI_TRANSACTION_DETAIL_ERROR • TI_TRANSACTION_HEADER_ERROR • TRANSACTOIN_DETAIL • TRANSACTION_HEADER
BU1.17.2	MarketWorks PCI feeds have the credit-card values hashed and are provided as delimited text feeds. They are loaded though the standard mechanism developed for ANN INC described in this section. • CUST_ALT_KEY

	• EMAIL_POS_ARCH • TRAN_TEND_ERR

BU1.17.3	The source database from which the DMP files are extracted bears the following attributes:
	• Database Instance Name = MW01
	• Oracle Version = Oracle Database 10g Enterprise Edition Release 10.2.0.4.0 - 64bit
	• Tablespace Name =
	  ○ MWDATA – All table except those noted below
	  ○ MWDATA2 – TRANSACTION_DETAIL
	  ○ MWDATA3 – TI_TRANSACTION_HEADER_ERROR
	  ○ MWDATA5 – TI_TRANSACTION_DETAIL_ERROR
	• Export Method = EXP

BU1.18	Historical Marketworks data will not provide a control file. Epsilon will receive table counts from ANN INC. once the transfers are complete, for verification purposes.
BU1.18.1	Once all MarketWorks data has been loaded into the database, row counts of each table are provided and audited against the source counts. The load count much match 100% in order to proceed with the migration data load into the Data Warehouse.
BU1.19	Incoming data file from ADS will conform to the following naming convention; ▪ ANN_ADS_CUSTDAILYPLCC_yyyymmdd.dat ▪ ANN_ADS_CUSTDAILYCBCC_yyyymmdd.dat ▪ ANN_ADS_XREFDAILYPLCC_yyyymmdd.dat ▪ ANN_ADS_XREFDAILYCBCC_yyyymmdd.dat ▪ ANN_ADS_CUSTMTHLYPLCC_yyyymmdd.dat ▪ ANN_ADS_CUSTMTHLYCBCC_yyyymmdd.dat ○ 001 is the file counter, it gets incremented by '1' when the source file is split into multiple files for the same filename/ date combination: Occurs when a single file is too large to transfer to Epsilon
BU1.19.1	The ADS feed handling process leverages a mechanism used for all feeds. The feed source (position 2) and type (position 3) is parsed out from the file name. This is used to identify the load schema as well as assign and track the appropriate data warehouse transformations throughout the update.
BU1.19.2	The feed date and sequence is parsed out from the file name. This is intended to represent the date that the data was extracted from the source system. It is used as the data Activity Date when a true data creation/modification date is not available at the record level in the source feed.
BU1.19.3	The feed sequence is used to load data in sequence, should a feed need to be delivered in multiple parts. In the event that PKs of a source feed are duplicated this sequence is leveraged to insure "last in wins". The sequence is also used to differentiate multi-part feeds that should all be loaded versus unintentional multi-sent, resent, or replacement feeds that should be trapped and escalated as part of the load audit process defined in **Appendix H – Audit Reports**
BU1.19.4	Feeds not conforming the agreed upon naming convention are rejected and flagged as part of the load audit process.
BU1.19.5	The ADS feeds are identified through adherence to the following naming convention: ▪ ANN_ADS_CUSTDAILYPLCC_yyyymmdd.dat ▪ ANN_ADS_CUSTDAILYCBCC_yyyymmdd.dat ▪ ANN_ADS_XREFDAILYPLCC_yyyymmddhh24miss.dat ▪ ANN_ADS_XREFDAILYCBCC_yyyymmddhh24miss.dat

	▪ ANN_ADS_CUSTMTHLYPLCC_yyyymmdd.dat ▪ ANN_ADS_CUSTMTHLYCBCC_yyyymmdd.dat
BU1.20	ADS will not provide control files. Epsilon will receive record type '9' for every file, which is a trailer record that can be used for verification purposes.
BU1.20.1	The ADS control file handling process leverages a mechanism used for all feeds. A control file in the agreed upon naming convention is identified and loaded into the file_detail meta data table.
BU1.20.2	The record quantity and/or uncompressed byte count is compared to each file's received and loaded count for validation purposes. Any of the following conditions would invoke the load exception escalation process described in **Appendix F – Escalation procedure** • Missing/Malformed Control File • Control File record count does not equal Data File record count • Control File byte count does not equal Data File byte count
BU1.20.3	The feed record quantity is extracted from the Customer Monthly feed identified as Record Type = 9. This is loaded into the file_detail meta-data table as the file record count for later load validation.
BU1.20.4	The feed record quantity is extracted from the Customer Daily feed identified as Record Type = 9. This record is provided for each division. They are rolled up and loaded into the file_detail meta-data table as the file record count for later load validation.
BU1.21	Incoming data file from RightNow will conform to the following naming convention; ▪ ANN_RIGHTNOW_INCIDENT_yyyymmdd_001.dat o 001 is the file counter, it gets incremented by '1' when the source file is split into multiple files for the same filename/ date combination: Occurs when a single file is too large to transfer to Epsilon
BU1.21.1	For RightNow data feed handling, see BU1.19.1 through BU1.19.4
BU1.21.2	The RightNow feeds are identified through adherence to the following naming convention: ▪ ANN_RIGHTNOW_INCIDENT_yyyymmdd_001.dat
BU1.22	Incoming control files are used to verify successful data transfer and staging. These feeds from RightNow will correspond to an appropriate data file and will conform to the following naming convention; ▪ ANN_RIGHTNOW_INCIDENT_yyyymmdd_001.ctl Every data source will have a control file, even if the source was transmitted in multiple files, there would only be one control file. Contents should be one record for every file the control file is managing: Filename\ bytes uncomp\ # records.
BU1.22.1	For RightNow control file handling, see BU1.20.1 through BU1.20.2
BU1.23	Incoming data file from BazaarVoice will conform to the following naming convention; ▪ ANN_BV_RATINGSAT_yyyymmdd_001.xml ▪ ANN_BV_RATINGSLOFT_yyyymmdd_001.xml o 001 is the file counter, it gets incremented by '1' when the source file is split into multiple files for the same filename/ date combination: Occurs when a single file is too large to transfer to Epsilon
BU1.23.1	For BazaarVoice data feed handling: see BU1.19.1 through BU1.19.4
BU1.23.2	The BazaarVoice feeds are identified through adherence to the following naming

	convention: ▪ ANN_BV_RATINGSAT_yyyymmdd_001.xml ▪ ANN_BV_RATINGSLOFT_yyyymmdd_001.xml
BU1.23.3	The BazaarVoice XML is flattened using a customized process to extract the RATINGS data. A tab-delimited feed reflecting the layout described in the Interface Agreement is generated for each brand.
BU1.24	BazaarVoice will not be sending control files. For Epsilon to verify this file for each update an email with counts will need to be sent.
BU1.24.1	BazaarVoice does not provide a control file. The process must assume the transferred and loaded quantity is accurate.
BU1.24.2	Over time, we will evaluate a mechanism in attempt to set feed size thresholds based upon historical trending.
BU1.25	Incoming data file from CheetahMail will conform to the following naming convention; ▪ loader_yyyymmddhhmiss_001.gz ○ 001 is the file counter, it gets incremented by '1' when the source file is split into multiple files for the same filename/ date combination: Occurs when a single file is too large to transfer to Epsilon
BU1.25.1	For CheetahMail data feed handling: see BU1.19.1 through BU1.19.4
BU1.25.2	The CheetahMail feeds are identified through adherence to the following naming convention: ▪ loader_yyyymmddhhmiss_001.dat
BU1.26	CheetahMail will not be sending control files. For Epsilon to verify this file for each update an email with counts will need to be sent.
BU1.26.1	CheetahMail does not provide a control file. The process must assume the transferred and loaded quantity is accurate.
BU1.26.2	Over time, we will evaluate a mechanism in attempt to set feed size thresholds based upon historical trending.
BU1.27	Incoming data file from SAP will conform to the following naming convention; ▪ ANN_SAP_STORE_yyyymmdd_001.dat ▪ ANN_SAP_ASSOCIATE_yyyymmdd_001.dat ▪ ANN_SAP_VENDOR_yyyymmdd_001.dat ▪ ANN_SAP_STYLE_yyyymmdd_001.dat ▪ ANN_SAP_PRODUCT_yyyymmdd_001.dat ▪ ANN_SAP_PROMO_yyyymmdd_001.dat ○ 001 is the file counter, it gets incremented by '1' when the source file is split into multiple files for the same filename/ date combination: Occurs when a single file is too large to transfer to Epsilon
BU1.27.1	For SAP data feed handling: see BU1.19.1 through BU1.19.4
BU1.27.2	The SAP feeds are identified through adherence to the following naming conventions ▪ ANN_SAP_STORE_yyyymmdd_001.dat ▪ ANN_SAP_ASSOCIATE_yyyymmdd_001.dat ▪ ANN_SAP_VENDOR_yyyymmdd_001.dat ▪ ANN_SAP_STYLE_yyyymmdd_001.dat ▪ ANN_SAP_PRODUCT_yyyymmdd_001.dat ▪ ANN_SAP_PROMO_yyyymmdd_001.dat

BU1.28	Incoming control files are used to verify successful data transfer and staging. These feeds from SAP will correspond to an appropriate data file and will conform to the following naming convention; ▪ ANN_SAP_STORE_yyyymmdd_001.ctl ▪ ANN_SAP_ASSOCIATE_yyyymmdd_001.ctl ▪ ANN_SAP_VENDOR_yyyymmdd_001.ctl ▪ ANN_SAP_STYLE_yyyymmdd_001.ctl ▪ ANN_SAP_PRODUCT_yyyymmdd_001.ctl ▪ ANN_SAP_PROMO_yyyymmdd_001.ctl Every data source will have a control file, even if the source was transmitted in multiple files, there would only be one control file. Contents should be one record for every file the control file is managing: Filename\ bytes uncomp\ # records
BU1.28.1	For SAP control file handling, see BU1.20.1 through BU1.20.2
BU1.29	Incoming data file from Omniture will conform to the following naming conventions; ▪ ANN_OMNI_VISITAT_yyyymmdd_001.dat ▪ ANN_OMNI_VISITLOFT_yyyymmdd_001.dat ▪ ANN_OMNI_VISITCATGAT_yyyymmdd_001.dat ▪ ANN_OMNI_VISITCATGLOFT_yyyymmdd_001.dat ▪ ANN_OMNI_SAINTAT_yyyymmdd_001.dat ▪ ANN_OMNI_SAINTLOFT_yyyymmdd_001.dat o 001 is the file counter, it gets incremented by '1' when the source file is split into multiple files for the same filename/ date combination: Occurs when a single file is too large to transfer to Epsilon
BU1.29.1	For Omniture data feed handling: see BU1.19.1 through BU1.19.4
BU1.29.2	The Omniture feeds are identified through adherence to the following naming conventions ▪ ANN_OMNI_VISITAT_yyyymmdd_001.dat ▪ ANN_OMNI_VISITLOFT_yyyymmdd_001.dat ▪ ANN_OMNI_VISITCATGAT_yyyymmdd_001.dat ▪ ANN_OMNI_VISITCATGLOFT_yyyymmdd_001.dat ▪ ANN_OMNI_SAINTAT_yyyymmdd_001.dat ▪ ANN_OMNI_SAINTLOFT_yyyymmdd_001.dat
BU1.30	Incoming control files are used to verify successful data transfer and staging. These feeds from Omniture will correspond to an appropriate data file and will conform to the following naming convention; ▪ ANN_OMNI_VISITAT_yyyymmdd_001.ctl ▪ ANN_OMNI_VISITLOFT_yyyymmdd_001.ctl ▪ ANN_OMNI_VISITCATGAT_yyyymmdd_001.ctl ▪ ANN_OMNI_VISITCATGLOFT_yyyymmdd_001.ctl ▪ ANN_OMNI_SAINTAT_yyyymmdd_001.ctl ▪ ANN_OMNI_SAINTLOFT_yyyymmdd_001.ctl Every data source will have a control file, even if the source was transmitted in multiple files, there would only be one control file. Contents should be one record for every file the control file is managing:

	Filename\ bytes uncomp\ # records.
BU1.30.1	For Omniture control file handling, see BU1.20.1 through BU1.20.2
BU1.31	Incoming data file from ATG will conform to the following naming convention; ▪ ANN_ATG_PROFILE_yyyymmdd_001.xml ▪ ANN_ATG_TRANSACTIONS_yyyymmdd_001.xml ▪ ANN_ATG_RETURNS_yyyymmdd_001.xml ▪ ANN_ATG_CATREQ_yyyymmdd_001.xml o 001 is the file counter; it gets incremented by '1' when the source file is split into multiple files for the same filename/ date combination: Occurs when a single file is too large to transfer to Epsilon
BU1.31.1	For ATG data feed handling: see BU1.19.1 through BU1.19.4
BU1.31.2	The ATG feeds are identified through adherence to the following naming convention ▪ ANN_ATG_PROFILE_yyyymmdd_001.xml ▪ ANN_ATG_TRANSACTIONS_yyyymmddhh24miss_001.xml ▪ ANN_ATG_RETURNS_yyyymmdd_001.xml ▪ ANN_ATG_CATREQ_yyyymmdd_001.xml
BU1.31.3	The ATG XML feed is processed through a utility to flatten out each node an maintain the hierarchical relationships. These can later be rejoined and normalized into the CRM Data Warehouse structures.
BU1.31.4	The flattened nodes are loaded into the staging schema as follows: • Profile o RAW_ATG_PROFILE o RAW_ATG_BILLING_ADDRESS o RAW_ATG_CREDIT_CARD o RAW_ATG_SHIPPING_ADDRESS • Transaction o RAW_ATG_ORDER o RAW_ATG_PRICE_INFO o RAW_ATG_ADJUSTMENT o RAW_ATG_TAX_PRICE_INFO o RAW_ATG_COMMERCE_ITEM o RAW_ATG_CONFIG_COMMERCE_ITEM o RAW_ATG_CURRENT_PRICE_DETAIL o RAW_ATG_DETAIL_ITEM_PRICE_INFO o RAW_ATG_ELECTRONIC_SHIPPING_GR o RAW_ATG_GIFTLIST_HANDLING_INST o RAW_ATG_GIFT_CARD o RAW_ATG_GIFT_CERTIFICATE o RAW_ATG_HANDLING_INSTRUCTION o RAW_ATG_HARDGOOD_SHIPPING_GR o RAW_ATG_ITEM_PRICE_INFO o RAW_ATG_LOCAL_STORE o RAW_ATG_MANUAL_PRICING_ADJUST o RAW_ATG_ORDER_PRICE_INFO

	○ RAW_ATG_PAYMENT_GROUP ○ RAW_ATG_PAY_ITEM_REL ○ RAW_ATG_PAY_ORDER_REL ○ RAW_ATG_RELATED_ORDERS ○ RAW_ATG_RELATIONSHIP ○ RAW_ATG_SHIPPING_GROUP ○ RAW_ATG_SHIPPING_PRICE_INFO ○ RAW_ATG_SHIP_ITEM_REL ○ RAW_ATG_STORE_CREDIT ○ RAW_ATG_SUBSKU_COMMERCE_ITEM • Return ○ RAW_ATG_RETURN_REQUEST ○ RAW_ATG_RETURN_ITEM ○ RAW_ATG_REFUND_METHOD • Catalog Request ○ RAW_ATG_CATALOG_REQUEST ○ RAW_ATG_CATALOG_PREFERENCES
BU1.31.5	Each RAW table includes the following attributes used to maintain XML hierarchical relationships: • INT_DTL_ID – current XML node ID • PAR_DTL_ID – parent XML node ID • INT_ID – top-level node ID • XML_FILE_ID – file tracking ID • XML_RECORD_NBR – streamed element count within a file
BU1.32	Incoming control files are used to verify successful data transfer and staging. These feeds from ATG will correspond to an appropriate data file and will conform to the following naming convention; ▪ ANN_ATG_PROFILE_yyyymmdd_001.ctl ▪ ANN_ATG_TRANSACTIONS_yyyymmdd_001.ctl ▪ ANN_ATG_RETURNS_yyyymmdd_001.ctl ▪ ANN_ATG_CATREQ_yyyymmdd_001.ctl Every data source will have a control file, even if the source was transmitted in multiple files, there would only be one control file. Contents should be one record for every file the control file is managing: Filename\ bytes uncomp.
BU1.32.1	Appropriate to XML data, ATG provides only the XML uncompressed byte count.
BU1.32.2	For ATG control file handling, see BU1.20.1 through BU1.20.2
BU1.33	Incoming data file from AuditWorks will conform to the following naming convention; ▪ ANN_AW_ TRANSACTIONS_yyyymmdd_001.dat ○ 001 is the file counter; it gets incremented by '1' when the source file is split into multiple files for the same filename/ date combination: Occurs when a single file is too large to transfer to Epsilon
BU1.33.1	For AuditWorks data feed handling: see BU1.19.1 through BU1.19.4
BU1.33.2	The AuditWorks feeds are identified through adherence to the following naming convention ▪ ANN_AW_ TRANSACTIONS_yyyymmddhh24miss_001.dat

BU1.33.2	The AuditWorks file sequence attribute spans days. It is expected that we would receive one and only one feed each day, so this sequence can be used for validation such that • There are not gaps in sequence relative to the previous day's sequence • Multiple feeds are not delivered within the same day Failure of either condition results in the invocation of the **Audit Escalation process** defined in the **Appendix**
BU1.33.3	The AuditWorks hierarchical POS record structure is split by record type and loaded into the following tables • RAW_AW_AUTH_DTL • RAW_AW_CUST_DTL • RAW_AW_DISC_DTL • RAW_AW_EXPCUST_DTL • RAW_AW_LINE_NOTES • RAW_AW_MERCH_DTL • RAW_AW_RETURN_DTL • RAW_AW_TXN_HEADER • RAW_AW_TXN_LINE
BU1.34	Incoming control files are used to verify successful data transfer and staging. These feeds from AuditWorks will correspond to an appropriate data file and will conform to the following naming convention; ▪ ANN_AW_ TRANSACTIONS_yyyymmdd_001.ctl Every data source will have a control file, even if the source was transmitted in multiple files, there would only be one control file. Contents should be one record for every file the control file is managing: Filename\ bytes uncomp\ # records.
BU1.34.1	The sum of all AuditWorks split/loaded feeds should be validated against the record count in the control file as a whole
BU1.34.2	For AuditWorks control file handling, see BU1.20.1 through BU1.20.2
BU1.35	Incoming data file from KPM Group or Experian will conform to the following naming conventions; ▪ ANN_KBM_DEMOGRAPHIC_yyyymmdd_001.dat (or EXP) ▪ ANN_KBM_REA_yyyymmdd_001.dat (or EXP) ▪ ANN_KBM_NZA_yyyymmdd_001.dat (or EXP) ▪ ANN_KBM_TRADEAREA_yyyymmdd_001.dat (or EXP) o 001 is the file counter; it gets incremented by '1' when the source file is split into multiple files for the same filename/ date combination: Occurs when a single file is too large to transfer to Epsilon
BU1.35.1	For KBM data feed handling: see BU1.19.1 through BU1.19.4
BU1.35.2	The KBM feeds are identified through adherence to the following naming convention ▪ ANN_KBM_DEMOGRAPHIC_yyyymmdd_001.dat
BU1.35.3	The original requirement has been updated to receive the following feeds from Experian • REA • NZA

	• TRADEAREA
BU1.36	Incoming control files are used to verify successful data transfer and staging. These feeds from KPM Group or Experian will correspond to an appropriate data file and will conform to the following naming convention; ▪ ANN_KBM_ DEMOGRAPHIC _yyyymmdd_001.ctl (or EXP) ▪ ANN_KBM_REA_yyyymmdd_001.ctl (or EXP) ▪ ANN_KBM_NZA_yyyymmdd_001.ctl (or EXP) ▪ ANN_KBM_TRADEAREA_yyyymmdd_001.ctl (or EXP) Every data source will have a control file, even if the source was transmitted in multiple files, there would only be one control file. Contents should be one record for every file the control file is managing: Filename\ bytes uncomp\ # records.
BU1.36.1	For KBM control file handling, see BU1.20.1 through BU1.20.2
BU1.37	Incoming data file from Experian will conform to the following naming conventions; ▪ ANN_EXP_MOSIAC_yyyymmdd_001.dat o 001 is the file counter; it gets incremented by '1' when the source file is split into multiple files for the same filename/ date combination: Occurs when a single file is too large to transfer to Epsilon
BU1.37.1	For Experian data feed handling: see BU1.19.1 through BU1.19.4
BU1.37.2	The Experian feeds are identified through adherence to the following naming convention ▪ ANN_EXP_REA_yyyymmdd_001.dat ▪ ANN_EXP_NZA_yyyymmdd_001.dat ▪ ANN_EXP_TRADEAREA_yyyymmdd_001.dat ▪ ANN_EXP_MOSIAC_yyyymmdd_001.dat
BU1.38	Incoming control files are used to verify successful data transfer and staging. These feeds from Experian will correspond to an appropriate data file (and will conform to the following naming convention; ▪ ANN_EXP_MOSIAC_yyyymmdd_001.ctl Every data source will have a control file, even if the source was transmitted in multiple files, there would only be one control file. Contents should be one record for every file the control file is managing: Filename\ bytes uncomp\ # records.
BU1.38.1	For Experian control file handling, see **BU1.20.1** through **BU1.20.2**
BU1.39	Incoming mail files from CRM Team will conform to the following naming conventions; ▪ tbd_yyyymmdd_001.dat o 001 is the file counter; it gets incremented by '1' when the source file is split into multiple files for the same filename/ date combination: Occurs when a single file is too large to transfer to Epsilon
BU1.39.1	The CRM (promotion history) feeds are identified through adherence to the following naming convention ▪ ANN_CRM_PROMO_yyyymmdd_001.dat ▪ ANN_CRM_TAX_yyyymmdd_001.dat
BY1.39.2	For Promotion History data handling see **BU28.1**
BU1.40	Incoming control files are used to verify successful data transfer and staging. These

	feeds from CRM Team will correspond to an appropriate data file and will conform to the following naming convention; ▪ tbd_yyyymmdd_001.ctl Every data source will have a control file, even if the source was transmitted in multiple files, there would only be one control file. Contents should be one record for every file the control file is managing: Filename\ bytes uncomp\ # records
BU1.40.1	Promotion History and Taxonomy control files are generated by the Epsilon list processing team as defined through the control file specification leveraged for the other data sources listed herein.
BU1.41	Incoming data file from SDL will conform to the following naming convention; ▪ ANN_SDL_IIDKEYS_yyyymmdd_001.dat ▪ ANN_SDL_DATA_yyyymmdd_001.dat ▪ ANN_SDL_SUBS_yyyymmdd_001.dat ▪ ANN_SDL_UNSUBS_yyyymmdd_001.dat ▪ ANN_SDL_DEMO_yyyymmdd_001.dat ▪ ANN_SDL_AGGREGATE_yyyymmdd_001.dat o 001 is the file counter, it gets incremented by '1' when the source file is split into multiple files for the same filename/ date combination: Occurs when a single file is too large to transfer to Epsilon
BU1.41.1	For SDL data feed handling: see BU1.19.1 through BU1.19.4
BU1.41.2	The SDL feeds are identified through adherence to the following naming convention ▪ ANN_SDL_IIDKEYS_yyyymmdd_001.dat ▪ ANN_SDL_DATA_yyyymmdd_001.dat ▪ ANN_SDL_SUBS_yyyymmdd_001.dat ▪ ANN_SDL_UNSUBS_yyyymmdd_001.dat ▪ ANN_SDL_DEMO_yyyymmdd_001.dat ▪ ANN_SDL_AGGREGATE_yyyymmdd_001.dat
BU1.42	Incoming control files are used to verify successful data transfer and staging. These feeds from SDL will correspond to an appropriate data file and will conform to the following naming convention; ANN_ SDL _IIDKEYS_yyyymmdd_001.ctl ▪ ANN_ SDL _DATA_yyyymmdd_001. ctl ▪ ANN_ SDL _SUBS_yyyymmdd_001. ctl ▪ ANN_ SDL _UNSUBS_yyyymmdd_001. ctl ▪ ANN_ SDL _DEMO_yyyymmdd_001. ctl ▪ ANN_ SDL _AGGREGATE_yyyymmdd_001. ctl Every data source will have a control file, even if the source was transmitted in multiple files, there would only be one control file. Contents should be one record for every file the control file is managing: Filename\ bytes uncomp\ # records
BU1.42.1	For SDL control file handling, see BU1.20.1 through BU1.20.2
BU1.43	Where possible, data sources unable to provide data for any of the above sources will transmit an empty (zero byte) file with the appropriate naming convention as well as an associated control file indicating that the intention was to send no data for the specified

	feed.
BU1.43.1	As discuss in BU1.20.2, a missing control file (for a feed expecting one) results in an audit failure. In the event that there is no data to transmit on any given day, the expectation is that each source system will still generate and transmit an empty file, and a control file indicating zero uncompressed bytes and zero records were provided to the load process.
BU1.44	Incoming data from Agility will conform to the following naming convention; ▪ ANN_AGD_FROMAGILITY_yyyymmdd.dat ▪ ANN_AGR_FROMAGILITY_yyyymmdd.dat ▪ ANN_AGD_INDXREF_yyyymmdd.dat ▪ ANN_AGR_INDXREF_yyyymmdd.dat
BU1.44.1	For Agility data feed handling: see BU1.19.1 through BU1.19.4
BU1.44.2	The Agility feeds are identified through adherence to the following naming convention ▪ ANN_AGD_FROMAGILITY_yyyymmdd.dat ▪ ANN_AGR_FROMAGILITY_yyyymmdd.dat ▪ ANN_AGD_INDXREF_yyyymmdd.dat ▪ ANN_AGR_INDXREF_yyyymmdd.dat
BU1.45	Outbound Agility file conform to the following naming convention; ▪ ANN_AGD_TOAGILITY_yyyymmdd.dat ▪ ANN_AGR_TOAGILITY_yyyymmdd.dat ▪ ANN_AGR_STORE_yyyymmdd.dat
BU1.45.1	Agility outbound feeds are prepared with the specified naming convention and the extraction date equal to the date as of the beginning of the load process.
BU1.45.2	Agility outbound feeds are extracted daily and processed in-line within the update process.
BU1.46	Outbound ADS files conform to the following naming convention; ▪ ANN_EPS_CUSTOMER_yyyymmdd_001.dat ▪ ANN_EPS_PROMO_yyyymmdd_001.dat ▪ ANN_EPS_TRANSDTL_yyyymmdd_001.dat ▪ ANN_EPS_TRANSHDR_yyyymmdd_001.dat ▪ ANN_EPS_TRANSCPN_yyyymmdd_001.dat ▪ ANN_EPS_PRODUCT_yyyymmdd_001.dat
BU1.46.1	ADS outbound feeds are prepared with the specified naming convention and the extraction date equal to the date as of the beginning of the load process. ▪ ANN_EPS_CUSTOMER_yyyymmdd_001.dat ▪ ANN_EPS_PROMO_yyyymmdd_001.dat ▪ ANN_EPS_TRANSDTL_yyyymmdd_001.dat ▪ ANN_EPS_TRANSHDR_yyyymmdd_001.dat ▪ ANN_EPS_TRANSCPN_yyyymmdd_001.dat ▪ ANN_EPS_PRODUCT_yyyymmdd_001.dat
BU1.46.2	ADS outbound feeds are extracted as part of the weekly processed executing on Saturday after the daily update completes
BU1.46.3	The customer data feed is a full extract of all customers from the Data Mart. The customer entity in this case is identified by the mart Individual level. See **Customer** for more information on the customer Individual level. Through this mechanism the best address for an account is delivered. In the case where multiple ADS accounts are linked to a single individual, the most currently updated account is assigned to the Individual
BU1.46.4	The transaction header, detail, and coupon feeds extracted reflect all new or modified transactions that have been loaded into the database within the previous weekly cycle.
BU1.46.5	The product feed is a full extract of all historical and ongoing product data denormalized at the SKU level

<table>
<tr><td>BU1.46.6</td><td>The promotion feed reflects all outbound Direct Mail communication that has had been loaded as part of the more recent weekly cycle. Because ANN INC is performing order attribution manually, we are unable to post to the response_flag indicating that a transaction resulted from the promotion effort.</td></tr>
<tr><td>BU1.46.7</td><td>The diagram in Extracts / Audits / Acceptance Design illustrates the relationship of the ADS outbound feeds.</td></tr>
<tr><td>BU1.47</td><td>Outbound CheetahMail files conform to the following naming convention;
ANN_EPS_POSSUBS_yyyymmdd.dat
ANN_EPS_ADSSUBS_yyyymmdd.dat
ANN_EPS_CUSTPROFILE_yyyymmdd.dat
ANN_EPS _EMAILHYGIENE_yyyymmdd_001.dat</td></tr>
<tr><td>BU1.47.1</td><td>CheetahMail outbound feeds are prepared with the specified naming convention and the extraction date equal to the date as of the beginning of the load process.
ANN_EPS_POSSUBS_yyyymmdd.dat
ANN_EPS_ADSSUBS_yyyymmdd.dat
ANN_EPS_CUSTPROFILE_yyyymmdd.dat
ANN_EPS _EMAILHYGIENE_yyyymmdd_001.dat</td></tr>
<tr><td>BU1.47.2</td><td>CheetahMail outbound feeds are extracted as part of the daily process that executes once the daily mart refresh has been built and the daily mart SLA has been met</td></tr>
<tr><td>BU1.47.3</td><td>The POS Subscribers feed is generated as an extract from the Data Warehouse for any NEW email addresses that were inserted from the AuditWorks Transaction feed as part of this daily cycle. These are identified through the REC_INS_DT on ACCOUNT_EMAIL where ACCT_SOURCE_CD = “AW”.</td></tr>
<tr><td>BU1.47.4</td><td>The store attribute of the POSSUBS is the store at which the transaction occurred, leveraging the following brand_cd decode:
AT = APOS
LOFT = LPOS
LOS = OPOS
ATF = FPOS
In the event that a new email address is linked to transactions associated with multiple brands, the email address is duplicated for each brand store

In the event that a transaction cannot be identified with one of the four brands above, the email is excluded from the subscriber feed and an audit failure is distributed for escalation.</td></tr>
<tr><td>BU1.47.5</td><td>The subscriber list (PID) assigned is based on the Store ID brand as follows:
<table>
<tr><th>PID</th><th>Description</th><th>Brand</th></tr>
<tr><td>1617026943</td><td>Ann Taylor Stores</td><td>AT</td></tr>
<tr><td>1617027553</td><td>LOFT</td><td>LOFT</td></tr>
<tr><td>1617027907</td><td>Ann Taylor Factory</td><td>ATF</td></tr>
<tr><td>2071576052</td><td>LOFT Outlet Stores</td><td>LOS</td></tr>
<tr><td>2089582708</td><td>Ann Taylor Canada</td><td>AT</td></tr>
<tr><td>2089600494</td><td>LOFT Canada</td><td>LOFT</td></tr>
</table></td></tr>
<tr><td>BU1.47.6</td><td>The ADS subscribers feed is generated as an extract from the Data Warehouse from the ADS Customer Daily or Monthly feed, and include both:
New email addresses – identified via
account_email.ACCT_SOURCE_CD = ‘ADS’
account_email.REC_INS_DT = ${current cycle}
Preference change email addresses – identified via
account_email.ACCT_SOURCE_CD = ‘ADS’
brand_email.em_optout_ind = 0 and brand_email.last_optin_dt = ${current cycle}</td></tr>
</table>

<table>
<tr><td></td><td>
brand_emal_hist doesn't contain any previous OPT-IN for the email address under any PID
</td></tr>
<tr><td>BU1.47.7</td><td>The store attribute of the ADSSUBS is derived through the following condition:
if DIV_NBR = 177 or (IDV_NBR = 088 and CC_LOGO = 002) then LOFT else AT</td></tr>
<tr><td>BU1.47.8</td><td>The subscriber list (PID) assigned is based on the Store ID brand as follows:
<table>
<tr><th>PID</th><th>Description</th><th>Brand</th></tr>
<tr><td>1617026943</td><td>Ann Taylor Stores</td><td>AT</td></tr>
<tr><td>1617027553</td><td>LOFT</td><td>LOFT</td></tr>
</table>
</td></tr>
<tr><td>BU1.47.9</td><td>After the ADSSUBS feed has been delivered to CheetahMail, and the subsequent update is processed, an audit is performed to verify that the requested email addresses were returned by CheetahMail on the SUBS feeds and loaded into BRAND_EMAIL. This is done as a post-DW update validation.</td></tr>
<tr><td>BU1.47.10</td><td>The CheetahMail customer profile and transaction feeds are extracted as described in Appendix J – CheetahMail/SDL Extracts as well as the DW to Mart Source to Target documents</td></tr>
<tr><td>BU1.47.12</td><td>The Customer Profile CUSTOMER_NUMBER attribute is assigned as the mart INDIV_ID</td></tr>
<tr><td>BU1.47.13</td><td>The Email Hygiene feed is extracted based on new email addresses linked to BRAND_EMAIL (i.e. sourced from CM) that have been modified through the Email Hygiene address correction process.</td></tr>
<tr><td>BU1.47.14</td><td>The Email Hygiene Store/Brand is based on decoding the PID(s) to which an existing CheetahMail-linked account is associated as follows:
<table>
<tr><th>PID</th><th>BRAND_CD</th><th>Description</th></tr>
<tr><td>1617026943</td><td>AT</td><td>Ann Taylor Stores</td></tr>
<tr><td>1617027553</td><td>LOFT</td><td>LOFT</td></tr>
<tr><td>1617027907</td><td>ATF</td><td>Ann Taylor Factory</td></tr>
<tr><td>2071576052</td><td>LOS</td><td>LOFT Outlet Stores</td></tr>
<tr><td>2068023233</td><td>LOFT</td><td>Celebrations Bride Outreach</td></tr>
<tr><td>2072079322</td><td>LOFT</td><td>LOFT Loves Teachers</td></tr>
<tr><td>2078200572</td><td>LOFT</td><td>LiveLoveLOFT</td></tr>
<tr><td>2072079327</td><td>ENT</td><td>Network for Style (can be completed ignored; 52 people a long time ago)</td></tr>
<tr><td>1950820576</td><td>ENT</td><td>Temp List (probably unnecessary to have on Epsilon?)</td></tr>
<tr><td>2088200651</td><td>ENT</td><td>one of the credit card lists (LOFT PLCC, LOFT MC, AT PLCC, or AT MC)</td></tr>
<tr><td>2088200983</td><td>ENT</td><td>one of the credit card lists (LOFT PLCC, LOFT MC, AT PLCC, or AT MC)</td></tr>
<tr><td>2088200991</td><td>ENT</td><td>one of the credit card lists (LOFT PLCC, LOFT MC, AT PLCC, or AT MC)</td></tr>
<tr><td>2088200995</td><td>ENT</td><td>one of the credit card lists (LOFT PLCC, LOFT MC, AT PLCC, or AT MC)</td></tr>
</table>
</td></tr>
<tr><td>BU1.47.15</td><td>The Email Hygiene PID provided for ALL the PIDs a CheetahMail-sourced corrected email address was linked.</td></tr>
<tr><td>BU1.47.16</td><td>The PID encountered on the Loader Exception feeds is often not populated. Only valid PIDs (though undeliverable) are loaded to BRAND_EMAIL preferences. The remaining CheetahMail feeds all have valid PIDs. If a new PID is introduced in the CM data before Epsilon is provided with the appropriate brand designation, the PID is assigned to the ENT brand where it can be later re-aligned as appropriate.</td></tr>
<tr><td>BU1.48</td><td>Outbound append files conform to the following naming convention;

ANN_EPS_DEMOGRAPHIC_yyyymmdd_001.dat
ANN_EPS_MOSAIC_yyyymmdd_001.dat
ANN_EPS_REA_yyyymmdd_001.dat
ANN_EPS_NZA_yyyymmdd_001.dat
ANN_EPS_TRADEAREA_yyyymmdd_001.dat

</td></tr>
<tr><td>BU1.48.1</td><td>Experian outbound feeds are prepared with the specified naming convention and the extraction date equal to the date as of the beginning of the load process.

ANN_EPS_DEMOGRAPHIC_yyyymmdd_001.dat

</td></tr>
</table>

	▪ ANN_EPS_MOSAIC_yyyymmdd_001.dat ▪ ANN_EPS_REA_yyyymmdd_001.dat ▪ ANN_EPS_NZA_yyyymmdd_001.dat ▪ ANN_EPS_TRADEAREA_yyyymmdd_001.dat
BU1.48.2	The following Experian outbound feed is extracted as part of the daily in-line retail matchback process that as part of the update cycle • Name/Zip Append
BU1.48.3	The following Experian outbound feeds are extracted as part of the daily in-line retail matchback process through a weekly batching of unidentified transactions as defined by the **Retail Matchback External Process** • Reverse Email Append • Trade Area Append
BU1.48.4	The following Experian outbound feed is extracted as part of the daily process that executes once the daily mart refresh has been built and the daily mart SLA has been met • Demographic • MOSAIC
BU1.48.5	The external retail matchback extracts (NZA, REA, TAA) are sourced from the RMB_IDENTIFIER table for each external matching method in an effort to assign an address for account linkage to a transaction. The data used to build the RMB_IDENTIFIER varies between MarketWorks, AuditWorks, and ATG. See the transaction **Stage to Data Warehouse** mapping documents for details.
BU1.48.6	The Demographic and MOSAIC append extracts are sourced from the POSTAL_CONTACT table. All unique postal contacts (individuals at an address) are extracted where they are linked to accounts that are associated with purchases occurring with the most recent **36 months**
BU1.49	Outbound SDL files conform to the following naming convention; • ANNTAYLOR_AS_TRANS_ yyyymmdd.txt • ANNTAYLOR_AW_TRANS_ yyyymmdd.txt • ANNTAYLOR_FS_TRANS _ yyyymmdd.txt • ANNTAYLOR_LS_TRANS _ yyyymmdd.txt • ANNTAYLOR_LW_TRANS _ yyyymmdd.txt • ANNTAYLOR_OS_TRANS _ yyyymmdd.txt
BU1.49.1	SDL outbound feeds are prepared with the specified naming convention and the extraction date equal to the date as of the beginning of the load process • ANNTAYLOR_AS_TRANS_yyyymmdd.txt • ANNTAYLOR_AW_TRANS_yyyymmdd.txt • ANNTAYLOR_FS_TRANS_yyyymmdd.txt • ANNTAYLOR_LS_TRANS_yyyymmdd.txt • ANNTAYLOR_LW_TRANS_yyyymmdd.txt • ANNTAYLOR_OS_TRANS_yyyymmdd.txt
BU1.49.2	SDL outbound feeds are extracted as part of the daily process that executes once the daily mart refresh has been built and the daily mart SLA has been met
BU1.49.3	All transaction data is rolled up at the email address level through their relationship to one or more individuals and accounts. Where an individual is linked to multiple email addresses, those transactions are re-stated for each email address.
BU1.50	Outbound DM files are configurable within SAS Campaign Manager
BU1.50.1	SAS Campaign Manager temporarily stores all campaign data in a table on the mart database server. This table contains denormalized campaign promotion history and taxonomy attributes. Prior to daily load into the Data Warehouse this table is normalized and extracted into delimited feeds that are audited and loaded via the standard

	mechanisms into the Data Warehouse and mart.
BU1.50.1	SAS CI is used to define a consistent promotion history output feed with the following naming convention for direct mail as: • ANN_CRM_DMPROMO_${CMPGN_ID}_yyyymmdd.dat
BU1.50.2	Promotion history files are generated in TAB-delimited format with the following attributes:

INDIV_ID	NUMBER (13)
MAIL_HH_ID	NUMBER (13)
MAIL_INDIV_ID	NUMBER (13)
MAIL_GAID	NUMBER (32)
MAIL_ACCT_NBR	VARCHAR2 (20 Char)
HH_ID	NUMBER (13)
GAID	NUMBER (32)
CAMPAIGN_ID	NUMBER (20)
CAMPAIGN_CD	VARCHAR2 (20 Char)
BRAND_CD	VARCHAR2 (4 Char)
MODEL_ID	NUMBER (10)
MODEL_VERSION_ID	VARCHAR2 (10 Char)
MODEL_SCORE	NUMBER (12,7)
MODEL_SEG	VARCHAR2 (20 Char)
MODEL_ID_2	NUMBER (10)
MODEL_VERSION_ID_2	VARCHAR2 (10 Char)
MODEL_SCORE_2	NUMBER (12,7)
MODEL_SEG_2	VARCHAR2 (20 Char)
SEGMENT	NUMBER (3)
KEYCODE	VARCHAR2 (8 Char)
VERSION_CD	VARCHAR2 (10 Char)
ACE_GEO_LAT	VARCHAR2 (10 Char)
ACE_GEO_LNG	VARCHAR2 (11 Char)
CLOSEST_STORE_NUM1	NUMBER (10)
CLOSEST_STORE_NUM2	NUMBER (10)
CLOSEST_STORE_NUM3	NUMBER (10)
OFFER_CD_1	VARCHAR2 (10 Char)
OFFER1_BARCODE_1	NUMBER (12)
OFFER1_BARCODE_2	NUMBER (12)
OFFER_CD_2	VARCHAR2 (10 Char)
OFFER2_BARCODE_1	NUMBER (12)
OFFER2_BARCODE_2	NUMBER (12)
OFFER_CD_3	VARCHAR2 (10 Char)
OFFER3_BARCODE_1	NUMBER (12)
OFFER3_BARCODE_2	NUMBER (12)
OFFER_CD_4	VARCHAR2 (10 Char)
OFFER4_BARCODE_1	NUMBER (12)
OFFER4_BARCODE_2	NUMBER (12)
POSTAL_CONTACT_ID	NUMBER (10)
ASSIGNED_STORE	NUMBER (10)
PRIMARY_STORE_NUMBER	NUMBER (10)
MAIL_CONTROL_FLG	VARCHAR2 (1 Char)
MISC_CODE_N1	NUMBER (5)
MISC_CODE_N2	NUMBER (5)
MISC_CODE_N3	NUMBER (5)
MISC_CODE_N4	NUMBER (5)
MISC_CODE_N5	NUMBER (5)

MISC_CODE_C1	VARCHAR2 (10 Char)	
MISC_CODE_C2	VARCHAR2 (10 Char)	
MISC_CODE_C3	VARCHAR2 (10 Char)	
MISC_CODE_C4	VARCHAR2 (10 Char)	
MISC_CODE_C5	VARCHAR2 (10 Char)	

BU1.50.3

SAS CI generates a corresponding direct mail campaign taxonomy record for each Keycode that is loaded to a table in the mart. New records are extracted and loaded back into the Data Warehouse on the subsequent update.

The Taxonomy feed conforms to the following naming convention

- ANN_CRM_DMTAXONMY_${CMPGN_ID}_yyyymmdd.dat

The feed consists of the following attributes:

CAMPAIGN	CAMPAIGN_ID	NUMBER (20)
CAMPAIGN	CAMPAIGN_CD	VARCHAR2 (20 Char)
CAMPAIGN	BRAND_CD	VARCHAR2 (4 Char)
CAMPAIGN	COUNTRY_CD	NUMBER (1)
CAMPAIGN	CAMPAIGN_NM	VARCHAR2 (80 Char)
CAMPAIGN	CAMPAIGN_EFFECTIVE_DT	DATE (7)
CAMPAIGN	CAMPAIGN_EXPIRATION_DT	DATE (7)
CAMPAIGN	CALENDAR_YEAR	NUMBER (4)
CAMPAIGN	CAMPAIGN_TYPE	VARCHAR2 (20 Char)
CAMPAIGN	SELECT_DT	DATE (7)
CAMPAIGN	INHOME_DT	DATE (7)
CAMPAIGN	CHANNEL_CD	VARCHAR2 (1 Char)
CAMPAIGN	FISCAL_YEAR	NUMBER (4)
CAMPAIGN	FISCAL_QUARTER	NUMBER (1)
CAMPAIGN	FISCAL_MONTH	NUMBER (2)
CAMPAIGN	SOURCE	VARCHAR2 (3 Char)
CAMPAIGN_VERSIONS	VERSION_CD	VARCHAR2 (10 Char)
CAMPAIGN_VERSIONS	VERSION_CD_DESC	VARCHAR2 (80 Char)
CAMPAIGN_VERSIONS	ESTIMATED_CPP	NUMBER (10)
CAMPAIGN_VERSIONS	FINAL_CPP	NUMBER (10)
CAMPAIGN_VERSIONS	FORMAT	VARCHAR2 (80 Char)
CAMPAIGN_OFFERS	CAMPAIGN_ID	NUMBER (20)
CAMPAIGN_OFFERS	CAMPAIGN_CD	VARCHAR2 (20 Char)
CAMPAIGN_OFFERS	OFFER_CD	VARCHAR2 (10 Char)
CAMPAIGN_OFFERS	OFFER_DESC	VARCHAR2 (80 Char)
CAMPAIGN_OFFERS	COUPON_CD	VARCHAR2 (12 Char)
CAMPAIGN_OFFERS	COUPON_POS_NBR	NUMBER (1)
CAMPAIGN_OFFERS	REDEMPTION_DIVISION	VARCHAR2 (20 Char)
CAMPAIGN_COUPONS	COUPON_NM	VARCHAR2 (80 Char)
CAMPAIGN_COUPONS	EFFECTIVE_DT	DATE (7)
CAMPAIGN_COUPONS	EXPIRATION_DT	DATE (7)

BU1.50.4

SAS CI is used to define a variable campaign output format appropriate to each print vendor fulfilling the campaign.

The campaign feed is generated with a campaign identifier as follows:

- ANN_CRM_DMCMPGN_${CMPGN_ID}_yyyymmdd.dat

<table>
<tr><td>BU1.51</td><td>Outbound EM files are configurable within SAS Campaign Manager</td></tr>
<tr><td>BU1.51.1</td><td>SAS CI is used to define a consistent promotion history output feed with the following naming convention for direct mail as:
ANN_CRM_EMPROMO_${CMPGN_ID} yyyymmdd.dat</td></tr>
<tr><td>BU1.51.2</td><td>Promotion history files are generated in TAB-delimited format with the following attributes:
<table>
<tr><th>Field</th><th>Data Type</th><th>Description</th></tr>
<tr><td>EMAIL_ID</td><td>NUMBER(10)</td><td>The email address being promoted</td></tr>
<tr><td>BRAND_CD</td><td>VARCHAR2(4 CHAR)</td><td>Brand associated with the campaign (AT, LOFT, ATF, LOS</td></tr>
<tr><td>PROMOTION_DT</td><td>DATE</td><td>Date of the promotion</td></tr>
<tr><td>KEYCODE</td><td>VARCHAR2(20 CHAR)</td><td>Campaign tracking code</td></tr>
</table></td></tr>
<tr><td>BU1.51.3</td><td>SAS CI generates a corresponding email campaign taxonomy record for each Keycode</td></tr>
<tr><td>BU1.51.4</td><td>SAS CI is used to define a variable campaign output format appropriate to each Email Service Provider fulfilling the campaign.

The campaign feed is generated with a campaign identifier as follows:
ANN_CRM_EMCMPGN_${CMPGN_ID}_yyyymmdd.dat</td></tr>
<tr><td>BU1.52</td><td>Daily Incremental data from all sources will be cut off from at a specific time each day, and sent to Epsilon. Specific details to be defined during the Design Phase and captured in the Solution Design Document. Adjustments on cutoff and SFTP times may happen after full integration testing to be able to meet daily SLA’s.</td></tr>
<tr><td>BU1.52.1</td><td>All daily, weekly, monthly, quarterly, and annual feeds are expected to be full transferred by 6am EST for inclusion in the update process.</td></tr>
<tr><td>BU1.52.2</td><td>The following feeds are in-line process feeds and are processed 24 hours after the initial cutoff
Omniture Visit
Omniture Visit Category
Experian MOSAIC
Experian Name/Zip Append
Experian Reverse Email Append
Experian Trade Area Append
KBM Demographics</td></tr>
<tr><td>BU1.52.3</td><td>Any feed arriving during or after the cutoff defined in BU1.52.1 are excluded from the update cycle and escalated with the source system as defined in Appendix F – Escalation Procedures. Any missing feed identified as critical result in an immediate fatal termination of the update process and escalation with the source for resolution. The following feeds are identified as critical.
AW Transactions
ATG Transactions
ADS Daily/Monthly Customer</td></tr>
</table>

2. Data File Layout

	Requirement

BU2.1	All data providers will need to convert their corresponding data feeds to the defined template standard layouts. These layouts are defined the Interface Agreement
BU2.1.1	Any feed that does not conform to the documented Interface Agreements results in the rejection of that feed, and audit failure and an escalation with the source system as defined in **Appendix F – Escalation Procedures**.
BU2.2	MW data will be provided as native oracle dumps
BU2.2.1	Please refer to **BU1.17** for MarketWorks DMP feed delivery expectations.
BU2.3	ADS data files and control files are Fixed ASCII
BU2.3.1	Please refer to **BU1.19** and the Interface Agreements for ADS fixed ASCII feed delivery expectations
BU2.4	RightNow data files and control files are pipe delimited
BU2.4.1	Please refer to **BU1.21** and the Interface Agreements for RightNow pipe-delimited feed delivery expectations
BU2.5	BazaarVoice data files are XML format
BU2.5.1	Please refer to **BU1.23** and the Interface Agreements for BazaarVoice XML feed delivery expectations
BU2.6	CheetahMail data files are pipe delimited
BU2.6.1	Please refer to **BU1.25** and the Interface Agreements for CheetahMail delimited feed delivery expectations
BU2.7	SAP data files and control files are pipe delimited
BU2.7.1	Please refer to **BU1.27** and the Interface Agreements for SAP pipe-delimited feed delivery expectations
BU2.8	Omniture data files and control files are pipe delimited
BU2.8.1	Please refer to **BU1.29** and the Interface Agreements for Omniture pipe-delimited feed delivery expectations
BU2.9	ATG data files are XML format
BU2.9.1	Please refer to **BU1.31** and the Interface Agreements for ATG XML feed delivery expectations.
BU2.9.2	Any modification to the XSD hierarchical relationship will require a Change Request. Any additional elements created within a node are evaluated and may require a Change Request.
BU2.10	AW data files and control files are pipe delimited
BU2.10.1	Please refer to **BU1.33** and the Interface Agreements for AuditWorks pipe-delimited feed delivery expectations
BU2.11	KBM data files and control files are pipe delimited
BU2.11.1	Please refer to **BU1.35** and the Interface Agreements for KBM pipe-delimited feed delivery expectations
BU2.12	Records with fields whose data exceeds the maximum field lengths specified in the Interface Agreement are excluded from the load and subsequent update process
BU2.12.1	Record-level rejects occur in the update process at the following stages • **Load** – any attribute with a field value exceeding the maximum data length specified in the Interface Agreement is rejected • **Staging** – any attribute failing the data type conversion from VARCHAR() to the appropriate data warehouse data type (DATE, NUMERIC, etc) as specified in the Interface Agreement is rejected • **Data Warehouse** – in order to enforce the uniqueness of a primary key, de-duplication results in rejected records
BU2.12.2	Record-level rejects are extracted by type and reason for possible escalation and remediation with the source system
BU2.12.3	Record-level reject counts are aggregated for each feed and evaluated as part of **the Staging Quality Audit process**
BU2.13	Experian data files and control files are pipe delimited

BU2.13.1	Please refer to **BU1.37** and the Interface Agreements for Experian pipe-delimited feed delivery expectations
BU2.14	SDL data files and control files are pipe delimited
BU2.14.1	Please refer to **BU1.41** and the Interface Agreements for SDL pipe-delimited feed delivery expectations

3. Initial/Historical

	Design
BU3.1	MW will provide historical Client / transactional data from 4/1/2001
BU3.1.1	All MarketWorks tables data are loaded completely into the ANN_MIGRN_USER schema mirroring exactly how they exist in MarketWorks today. The rules for identifying the subset of data that survives in the CRM Data Warehouse is described in **Customer**
BU3.1.2	It is expected that the MarketWorks transaction and related Customer data is to date back as far as 4/1/2001 and contain continuous transaction data through the migration cutover date
BU3.1.3	Once the migration data has been validated as having been accurately loaded into the Data Warehouse its original source data is archived off to tape backup.
BU3.2	ADS will provide up to 120 months of historical Client / transactional data, or as much that is available in the source system not to exceed 120 months
BU3.2.1	All ADS migration data we receive is loaded into the Staging tables. The Customer Monthly feed LAST_PURCH_DT is evaluated against the migration cutover date to identify any accounts aged past 120 months.
BU3.2.2	Depending on volume of overage, a decision is made to either truncate customer and associated cross-reference data or evaluate the potential need for additional storage post-implementation.
BU3.2.3	Once the migration data has been validated as having been accurately loaded into the Data Warehouse its original source data is archived off to tape backup.
BU3.3	RightNow will provide up to 60 months of historical Client data, or as much that is available in the source system not to exceed 60 months
BU3.3.1	All RightNow migration data we receive is loaded into the Staging tables. The incident CREATE_DT is evaluated against the migration cutover date to identify data aged past 60 months.
BU3.3.2	Depending on volume of overage, a decision is made to either truncate call center incidents and associated cross-reference data or acquire additional storage post-implementation.
BU3.3.3	Once the migration data has been validated as having been accurately loaded into the Data Warehouse its original source data is archived off to tape backup.
BU3.4	BazaarVoice will provide up to 9 months of historical Client data, or as much that is available in the source system not to exceed 9 months
BU3.4.1	All BazaarVoice migration data we receive is loaded into the Staging tables. The review CRDATE is evaluated against the migration cutover date to identify data aged past 9 months.
BU3.4.2	Depending on volume of overage, a decision is made to either truncate call center incidents and associated cross-reference data or acquire additional storage post-implementation.
BU3.4.3	Once the migration data has been validated as having been accurately loaded into the Data Warehouse its original source data is archived off to tape backup.
BU3.5	CheetahMail will provide historical Client data from 11/1/2009
BU3.5.1	All CheetahMail migration data we receive is loaded into the Staging tables. The activity SENT_DT or RESPONSE_DT is evaluated against the migration cutover date to identify data aged past 11/1/2009.
BU3.5.2	Depending on volume of overage, a decision is made to either truncate call center incidents and associated cross-reference data or acquire additional storage post-implementation.

BU3.5.3	Once the migration data has been validated as having been accurately loaded into the Data Warehouse its original source data is archived off to tape backup.
BU3.6	SAP will provide up to 24 months of store/ product data, or as much that is available in the source system not to exceed 24 months
BU3.6.1	All SAP migration data we receive is loaded into the Staging tables. The Store, Associate, Product DATE_TIME is evaluated against the migration cutover date to identify data aged past 24 months
BU3.6.2	Depending on volume of overage, a decision is made to either truncate call center incidents and associated cross-reference data or evaluate the potential need for additional storage post-implementation.
BU3.6.3	Once the migration data has been validated as having been accurately loaded into the Data Warehouse its original source data is archived off to tape backup.
BU3.7	Omniture will provide historical web behavior data from 10/15/2011
BU3.7.1	All Omniture Web Tracking Visit, Visit Category and associated SAINT reference migration data we receive is loaded into the Staging tables. The VISIT_DATE is evaluated against the migration cutover date to identify data aged past 10/15/2011
BU3.7.2	Depending on volume of overage, a decision is made to either truncate call center incidents and associated cross-reference data or evaluate the potential need for additional storage post-implementation.
BU3.7.3	Once the migration data has been validated as having been accurately loaded into the Data Warehouse its original source data is archived off to tape backup.
BU3.8	ATG historical Ecommerce orders from 1/1/2009
BU3.8.1	ATG historical transaction data has been determined to be out of scope
BU3.9	CRM team will provide up to 24 months of historical mail file data, or as much that is available in the source system not to exceed 24 months
BU3.9.1	Direct Mail promotion history is provided by the CRM team from the mail and control files used to originally deploy these campaigns. The campaigns are expected to have no deployment dates aged past 24 months relative to the migration cutover date.
BU3.9.2	Each mail file is run through a data conversion process managed through a separate SOW specific to this effort. The conversion process standardize into a consistent file format for the number of campaign feeds allowed for in the SOW.
BU3.9.3	Depending on volume of overage, a decision is made to either truncate call center incidents and associated cross-reference data or evaluate the potential need for additional storage post-implementation.
BU3.9.4	Once the migration data has been validated as having been accurately loaded into the Data Warehouse is original source data is archived off to tape backup.
BU3.10	CRM team will provide up to 24 months of historical taxonomy data, or as much that is available in the source system not to exceed 24 months
BU3.10.1	The CRM team provides corresponding campaign taxonomy data in tab-delimited format to provide the following attributes for each campaign keycode - TBD
BU3.11	Epsilon requires full initial load 5 weeks prior to UAT and then standard delta process thereafter
BU3.11.1	Epsilon requires the historical migration feeds 2 full weeks prior to UAT
BU3.11.2	Epsilon will then process two incremental loads in the two-week duration of SIT
BU3.12	Historical model score/ meta data will be limited to MW Attribute feeds
BU3.12.1	The model score mapping from the MarketWorks Customer Attribute tables is defined in the **Data Warehouse mapping documents**
BU3.13	CRM team will provide Demographic data from KBM Group for clients making a purchase in the past 36 months

BU3.13.1	Once all CUSTOMER and TRANSACTION subject-area data from all sources has been loaded into the Data Warehouse, run through AGILITY, and approved, all Postal Contacts with at least one transaction order date in the 36 months prior to the migration date are extracted for Demographic overlay processing at KBM.
BU3.13.2	Upon processing, all KBM demographic data returned is loaded and appended to the appropriate Postal Contacts in the Data Warehouse
BU3.14	CRM team will provide Customer Segmentation data (MOSAIC) from Experian for clients making a purchase in the past 36 months
BU3.14.1	Once all CUSTOMER and TRANSACTION subject-area data from all sources has been loaded into the Data Warehouse, run through AGILITY, and approved, all Postal Contacts with at least one transaction order date in the 36 months prior to the migration date are extracted for MOSAIC overlay processing at Experian.
BU3.14.2	Upon processing, all Experian demographic data returned is loaded and appended to the appropriate Postal Contacts in the Data Warehouse

4. Daily/Incremental Load

	Design
BU4.1	From a staging perspective, each full refresh file (from each source) is treated as a replacement for any prior feed. In the case where a source provides a full refresh feed twice within the same cycle, the most current feed will be applied in the warehouse update
BU4.1.1	As system storage capacity permits, the RAW tables retain the current and prior 2 iterations of loaded data identified by daily, weekly, monthly, quarterly, and annual load date in order to facilitate ongoing process analysis traceable back to original source data.
BU4.1.2	The STG (stage) tables are truncated completely upon each load and refreshed with only the current update cycle's dataset.
BU4.1.3	The staging process permits the staging of multiple files for the same template from a single source as part of a single cycle. The sequence indicator provided in the file naming convention (where available) is used to differentiate multi-part files from potentially duplicated or replaced feeds. Files of the same type with the same sequence are flagged for review as part of the Staging Audit process. In the event, a critical feed fails this audit the update process is immediately halted and the source system is contacted for resolution.
BU4.1.4	Multiple feeds are staged in sequential order such that any duplication of primary key data at the source results in a "last in wins" scenario.
BU4.2	Weekly profile reports of all ANN INC. data files will be placed on the Client portal
BU4.2.1	. The profile reports are collected and published weekly to the Client-hosted SharePoint portal

5. Staging Quality Audit

	Design
BU5.1	Data transfer and external feed generation are validated as part of the Epsilon feed load process by using control files. Each data file provided for the CRM has an accompanying control file containing a single record of rolled up file counts. These are validated once the data file is staged
BU5.1.1	Wherever possible each source provides a one to one control file corresponding to each file transmitted for load. This is used to verify that the data expected from the source system was extracted and transferred to Epsilon properly. They are a pipe-delimited single record file containing the following attributes • File Name • Uncompressed byte size • Record count

BU5.1.2

The following feeds provide a control file by which the RECORD COUNT is validated against the actual feed:

RIGHTNOW	INCIDENT
SDL	IIDKEYS
SDL	DATA
SDL	DATA
SDL	DATA
SDL	DATA
SDL	DATA
SDL	DATA
SDL	SUBS
SDL	UNSUBS
SDL	DEMO
SDL	AGGREGATE
SAP	STORE
SAP	ASSOCIATE
SAP	VENDOR
SAP	STYLE
SAP	PRODUCT
SAP	PROMO
OMNI	VISIT_AT
OMNI	VISIT_LOFT
OMNI	VISTCATG_AT
OMNI	VISTCATG_LOFT
OMNI	SAINT_AT
OMNI	SAINT_LOFT
AW	TRANSACTIONS
AW	TRANSACTIONS
AW	TRANSACTIONS
AW	TRANSACTIONS
AW	TRANSACTIONS
AW	TRANSACTIONS
AW	TRANSACTIONS
AW	TRANSACTIONS
AW	TRANSACTIONS
KBM	DEMOGRAPHIC
KBM	DEMOGRAPHIC
EXP	REA
EXP	NZA
EXP	TRADEAREA
EXP	MOSIAC
EXP	MOSIAC

BU5.1.3

The following feeds do not provide a control file, but do provide in-line feed indicators which can be leveraged for the same purpose of validating record counts:

ADS	CUSTMTHLYPLCC
ADS	CUSTMTHLYCOBRAND

BU5.1.3

The following feeds provide a control file by which the UNCOMPRESSED BYTE COUNT is validated against the actual feed:

ATG	PROFILE
ATG	TRANSACTIONS
ATG	RETURNS
ATG	CATREQ

<table>
<tr><td>BU5.1.4</td><td>If there is no data to transmit during a given cycle the solution expected to be delivered an empty data file with a corresponding control file indicating 0 bytes and/or 0 records</td></tr>
<tr><td>BU5.1.5</td><td>The following automated feeds do not provide any mechanism for us to validate data transmission accuracy<table><tr><td>ADS</td><td>XREFDAILYPLCC</td></tr><tr><td>ADS</td><td>XREFDAILYCOBRAND</td></tr><tr><td>BV</td><td>RATINGSAT</td></tr><tr><td>BV</td><td>RATINGSLOFT</td></tr><tr><td>CM</td><td>LOADER</td></tr></table></td></tr>
<tr><td>BU5.1.6</td><td>The following migrations feeds are validated as a one-time confirmation via written communication directly with the source data provider:<table><tr><td>MW</td><td>Address</td></tr><tr><td>MW</td><td>Class</td></tr><tr><td>MW</td><td>Color</td></tr><tr><td>MW</td><td>Customer</td></tr><tr><td>MW</td><td>Customer_Alternate_Key</td></tr><tr><td>MW</td><td>Customer_Xref</td></tr><tr><td>MW</td><td>Department</td></tr><tr><td>MW</td><td>Division</td></tr><tr><td>MW</td><td>Product_Xref</td></tr><tr><td>MW</td><td>Style</td></tr><tr><td>MW</td><td>Style_Xref</td></tr><tr><td>MW</td><td>Tender</td></tr><tr><td>MW</td><td>Tender_Mapping</td></tr><tr><td>MW</td><td>TI_Transaction_Detail_Error</td></tr><tr><td>MW</td><td>TI_Transaction_Header_Error</td></tr><tr><td>MW</td><td>TI_Transaction_Tender_Error</td></tr><tr><td>MW</td><td>Transaction_Coupon</td></tr><tr><td>MW</td><td>Transaction_Detail</td></tr><tr><td>MW</td><td>Transaction_Header</td></tr><tr><td>MW</td><td>Attribute_Grouping</td></tr><tr><td>MW</td><td>Attribute_List</td></tr><tr><td>MW</td><td>Customer_Attribute</td></tr><tr><td>MW</td><td>Email_POS_Arch</td></tr><tr><td>MW</td><td>Transaction_Detail_XREF</td></tr><tr><td>CRM</td><td>FISCAL</td></tr><tr><td>CRM</td><td>PROMO History</td></tr><tr><td>CRM</td><td>Campaign Taxonomy</td></tr></table></td></tr>
<tr><td>BU5.1.7</td><td>If a feed is missing an expected control file an audit failure occurs invoking the escalation procedure described in Appendix F – Escalation Procedures</td></tr>
<tr><td>BU5.1.8</td><td>If a control file byte count does not match that of the transmitted feed an audit failure occurs invoking the escalation procedure described in Appendix F – Escalation Procedures</td></tr>
<tr><td>BU5.1.9</td><td>If a control record count does not match that of the transmitted feed (using a linux line count ‘wc –l’ operation) an audit failure occurs invoking the escalation procedure described in Appendix F – Escalation Procedures</td></tr>
<tr><td>BU5.2</td><td>Record rejects will be handled per 6.12.1 of the SOW as follows;
Data validation issues will be generally categorized and researched as follows:
i. Data issue due to known defect that can be corrected through defect fix
ii. Data issue that is an allowable exception due to certain business rule differences (e.g. householding consolidation)
iii. Data issue without clear explanation in which case each issue will be researched and a root cause identified, and Customer and</td></tr>
</table>

	Epsilon will mutually agree on fix
BU5.2.1	Data is validated through several mechanisms: reject analysis through either data length, data types, referential integrity, Primary Key deduplication, filtering rules, trending analysis, as well as sample data scenario reviews. The solution leverages data profiling and auditing components in order to identify any data issues.
BU5.2.2	Where a data issue can be quantified in terms in relation to a threshold the determination is made based on whether the impacted feed should halt the update, proceed with exclusion of the feed, or proceed with all successfully included data.
BU5.3.3	As anomalies are trapped and assessed resolution is obtained through the appropriate mutually-agreed upon mechanism: • Defect resulting in the modification of Epsilon process • Defect resulting in the modification of source system extraction process • Change request resulting in the modification of Epsilon process • Change request resulting in the modification of source system extraction process • Allowable/Explainable scenario resulting the documentation of the root cause for future reference
BU5.3	Records that are rejected by the staging loader are trapped, audited, and tracked in staging reports. These can be invalid data types or data exceeding maximum field length
BU5.3.1	The tracking and aggregation of load/stage data rejects at the file-level are provided through the Staging Audit Report described in **Appendix H – Audit Reports**
BU5.3.2	The Staging Audit Report is distributed by email to the Epsilon ANN support team and posted to the Epsilon-hosted Sharepoint portal for client access.
BU5.3.3	Load statistics are found in the following tables • elt_owner.file_detail – load statistics • elt_owner.file_type – feed expectations • elt_owner.file_threshold – load tolerances • elt_owner.process_savestats – source/target upsert statistics • elt_owner.process_timeline – throughput statistics • elt_owner.process_reject – rejection detail • elt_owner.aug_stg_hist – load/staging historical summary
BU5.4	Files containing all rejected records will be available on SFTP server for download and analysis by data source support teams
BU5.4.1	Reject files are created with the following convention • load - ${source_feed_name}_data_length_reject.dat • stage - ${source_feed_name}_data_type_reject.dat
BU5.4.2	Load reject files are in the source format.
BU5.4.3	Stage reject files are in tab-delimited format and have a reject reason and field indicator appended to the end of each record.
BU5.4.4	Reject files are posted to the DB gateway in /jcs/data/${platform}/dms/ann/ann_dw/rejects they are retained for 90 days, or as long as storage on the gateway permits before they are archived to tape.
BU5.4.5	Reject files are posted to the outbound_rejects folder on the DB gateway accessible to ANN resources through the Corp/Corp VPN tunnel. Vendors are provided with reject files upon request.
BU5.5	Alert if multiple files were received from a single source for the same feed

BU5.5.1	The data transfer process appends a unique date/time to the end of each filename once the transfer completes. This coupled with the file sequence component of the file name provides the mechanism to allow for multi-part files, but also gain awareness as to when multiple copies of the same file are transmitted.
BU5.5.2	Multiple feeds are identified in the Staging Audit. As long as the feed is not identified as critical the load is permitted to proceed with a 'last in wins' rule by primary key across all feeds. Once the audit is reviewed, the determination is made that a subsequently loaded file was provided by the source in error, the rollback procedures are invoked to restore the appropriate data.
BU5.6	Any error/alert triggered for a critical file results in a halt of the update process pending resolution by the ANN INC. IT team
BU5.6.1	Any audit failure identified with a critical feed halts the update process immediately and the Source Escalation procedure described in the **Appendix** is invoked. These feeds have been identified as critical. • AW Transactions • ATG Transactions • ADS Daily/Monthly Customer
BU5.6.2	When the update process is halted, all jobs are forcibly terminated and the support team is paged with the direction to immediately review audit reports and process logs.
BU5.7	The system provides appropriate file tracking mechanisms to provide historical logging, status reporting, and summary information associated with each feed throughout the data transfer and staging phase
BU5.7.1	File tracking structures are identified in **BU5.3.3**
BU5.8	The system provides profiles reports on each feed loaded to the staging tables to be used in unit testing/ quality analysis
BU5.8.1	Profile reports are provided by aggregating staged data
BU5.8.2	Frequency Distributions are provided on all VARCHAR fields with less than 30 distinct values.
BU5.8.3	Blank/Non-Blank counts are provided on all fields
BU5.8.4	All file and field level aggregations are stored for historical trending and prior load variance purposes.
BU5.9	The system will provide alerts when files are missing for the update. The alert will go to the vendor and/or ANN Inc.
BU5.9.1	Feeds missing from their expected cycle are flagged in the Staging audit report
BU5.9.2	Control files missing for their data counterparts are flagged in the Staging audit report
BU5.9.3	The audit failure is distributed via email for proper escalation as well as posted to Sharepoint.
BU5.9.4	In the event that a critical feed is missing, the update process is halted and the support team is paged for immediate resolution.
BU5.9.5	Non-Critical feeds are escalated and caught up during the subsequent update cycle(s) once the missing data is provided.

Data/ Email Hygiene, PCI and Matching Consolidation (AGILITY) Requirements

6. Data/ Email Hygiene

#	Design

BU6.1	Upper case all fields upon load to the staging tables, except email address.
BU6.1.1	All fields except where explicitly noted below are converted up uppercase as data is loaded from the RAW tables to their STG counterparts. Generally data is only kept in original casing where mixed casing is required to maintain foreign key relationships.
BU6.1.2	Email Addresses are converted to lowercase as data is loaded from RAW to STG
BU6.2	Original name and address fields will be stored separately within the data warehouse and data mart from standardized fields returned for AGILTY process.
BU6.2.1	Original (source) address data is retained in the Data Warehouse in the ACCOUNT table.
BU6.2.2	Standardized/Corrected/COA'd address data is stored in the related POSTAL_CONTACT table.
BU6.3	Implement email address hygiene process is to correct malformed email addresses and incorrect domains whenever possible (on Ann's behalf) – contained in **Appendix D**
BU6.3.1	All inbound email addresses are passed through the email hygiene through their initial source mapping into WRK_SRC_ACCOUNT
BU6.3.2	Inbound email addresses from CheetahMail subscriber, unsubscriber, or demographic feeds are excluded from the Email Hygiene process
#	**Email Hygiene Design**
EH.1	Limited character removal and replacement will be included in the hygiene process.
EH1.1.1	Remove any spaces within the email address value.
EH1.1.2	Remove a 'dot' that exists in the local mailbox part directly before the @ symbol *(test.@blank.com).*
EH1.1.3	Remove a 'dot' that exists in the first position in the local mailbox part with text after it *(.test@blank.com).*
EH1.1.4	Remove a 'dot' that directly follows the @ symbol in the domain *(test@.blank.com).*
EH1.1.5	Remove a 'dot' that is the last position of the email address *(test@blank.com.).*
EH1.1.6	Where there is a 'repeated dot' in the email address change to a 'single dot' *(test@blank..com or test..1@blank.com).*
EH1.1.7	Where a comma or semi-colon exists in the domain before any top-level domains, change the comma or semi-colon to a 'dot'. For a complete list of top-level domains please see the NACM Source-to-Target Email Hygiene Reference tab "Top-level Domains" table. Here are some examples: ,com à .com ,edu à .edu ,net à .net ;gov à .gov ;biz à .biz ;mil à .mil
EH1.1.8	Remove 'mailto:' at the start of an email address *(mailto:test@blank.com).*
EH1.1.9	Remove 'www.' At the start of an email address *(www.test@blank.com).*

EH1.1.10	Correct a misspelling in the domain for the following: .comm à .com .nett à .net
EH1.1.11	Where a non-numeric or non-alpha (e.g. %, #, $) character exists in the local mailbox part or domain remove the bad character. See the NACM Source-to-Target Email Hygiene Reference tab “Limited Character Removal/Replacement” table for a complete list of bad characters and where they are addressed, e.g. local, domain or trailing/preceding *(test%@blank.com).*
EH.2	After the limited character removal and replacement, domain correction will be included in the hygiene process.
EH1.2.1	Each email address shall be compared to the list of bad domain names listed in the IN_DOMAIN column of the “Domain Correction” table on the NACM Source-to-Target Email Hygiene Reference tab to determine whether or not the email address has an identified bad domain that can be corrected.
EH1.2.2	Compare the domain part of the email address to the IN_DOMAIN value. Where there is a match create a new email_address by concatenating the <local mailbox part> with the <@> and the associated <OUT_DOMAIN> value in the “Domain Correction” table on the NACM Source-to-Target Email Hygiene Reference tab. Keep both the original and created version of the email address and apply the general rules in Section 8 above.
EH.3	After being processed by the limited character removal and domain correction, then the email will be evaluated for being valid or not (VALID_IND)
EH1.3.1	After all email addresses have been hygiene processed, whether or not they were corrected, evaluate each email for validity: • Where the local mailbox part is missing for an email address (local mailbox part is the portion before the @ symbol), set the EMAIL.VALID_IND = 0. • Where the domain part is missing for the email address (domain part is the portion after the @ symbol), set the EMAIL.VALID_IND = 0. • Where the ‘dot’ is missing in the domain, set the EMAIL.VALID_IND = 0. • Where the local mailbox part is > 64 characters long, set the EMAIL.VALID_IND = 0. • Else, set the EMAIL.VALID_IND = 1.
EH1.3.2	The UNDELIVERABLE indicator is also set to 0 indicating an invalid email address when encountered in the CheetahMail Loader exception feed for the following types: • STOP – Address, domain, or local part is blocked from subscribing • MAIL – Malformed or Illegal Email Address • SCHM – Database schema error (e.g. data to wide for column) • ERRO – Other (non-schema) database error CheetahMail (loader exception) is the source of truth for undeliverable identification. Epsilon does no derive deliverability in any other way. Epsilon does however set a separate VALID indicator as defined below.

EH1.3.3	The following strings also identify INVALID email addresses - null@% - nul@% - noemail@% - none@% - noemail@% - nothanks@% - noll@% - na@% - non@% - no@% - n@% - yahoo@yahoo.com - at@at.com - 123@aol.com - dfdf@jdfdj.com - xfvdfdf@sdfdfgd.com - fddvf@fdfd.com
EH.4	Design of the email hygiene process shall allow for future modifications to the domain correction and character replacement lists.
EH.4.1	The email hygiene correction logic can be modified on a go-forward basis through the submission of a Change Request. Historically processed email addresses would not be re-corrected as part of the logic modification except through the significant effort of realigning all of the historically established email dimensions which would be evaluated at the time such a task becomes desirable.
EH.5	Hygiened email addresses will be stored as the active email address in the data warehouse; therefore, automatically being considered for the best email address
EH1.5.1	Active email addresses are identified in the email table where active_email_id = email_id. Foreign key relationships are always realigned to point to the corrected/active email_id. These relationships are: • EMAIL_PROFILE • BRAND_EMAIL • EMAIL_BRAND_AGGREGATE • EMAIL_PROMOTION • EMAIL_RESPONSE • ACCOUNT_EMAIL
EH1.5.2	Fundamentally this leaves the original/erroneous email address with no relationship to any of the above tables except through the re-pointed active/corrected email ID. These email addresses are retained in the EMAIL table for reference purposes only. The solution also implements a table related to ACCOUNT_EMAIL for historical auditing purposes called ACCOUNT_EMAIL HIST that tracks the relationship of the original uncorrected email_id received for an ACCOUNT. Inserts are only made into this table when an email correction is made.
EH.6	Number of corrected emails to be available by source in regular audit reporting.

EH1.6.1	Number of corrected emails can be obtained through two mechanisms: 1. From EMAIL where email_id <> active_email_id 2. From ACCOUNT_EMAIL_HIST (distinct email_id) – this table can also be used to report corrections by date as well as number of iterative corrections per email address.
EH.7	Each email address will be assigned a unique EMAIL_ID by the hygiene process.
EH1.7.1	As an email address from any source is inserted into the EMAIL a check is performed on the standardized lower(email_address) field to see if the email address is already present. Any new inserts increment the email_id with the following sequence: • email_id_seq.nextval
EH.8	After all email addresses have been hygiene processed, whether or not they were corrected, evaluate each email for validity: 1) Where the local mailbox part is missing for an email address (local mailbox part is the portion before the @ symbol), set the EMAIL.VALID_IND = 0. 2) Where the domain part is missing for the email address (domain part is the portion after the @ symbol), set the EMAIL.VALID_IND = 0. 3) Where the 'dot' is missing in the domain, set the EMAIL.VALID_IND = 0. 4) Where the local mailbox part is > 64 characters long, set the EMAIL.VALID_IND = 0. 5) Else, set the EMAIL.VALID_IND = 1.
EH1.8.1	After all email addresses have been hygiene processed, whether or not they were corrected, the system will evaluate each email for validity: 1) Where the local mailbox part is missing for an email address (local mailbox part is the portion before the @ symbol), set the EMAIL.VALID_IND = 0. 2) Where the domain part is missing for the email address (domain part is the portion after the @ symbol), set the EMAIL.VALID_IND = 0. 3) Where the 'dot' is missing in the domain, set the EMAIL.VALID_IND = 0. 4) Where the local mailbox part is > 64 characters long, set the EMAIL.VALID_IND = 0. 5) Else, set the EMAIL.VALID_IND = 1.
BU6.4	All email addresses that were corrected due to the email hygiene process, we will sent back to CheetahMail to apply the changes to their email addresses.
BU6.4.1	Corrected email addresses are provided to CheetahMail for "subscription" through the ANN_EPS_EMAILHYGIENE_yyyymmdd.dat daily feed
BU6.4.2	These email addresses are identified via the following criteria 1. Email Address whose created_dt is since the last cycle 2. The email address is the result of the a change generated through the Email Hygiene process 3. Associated to one or more subscriber lists – successful join from EMAIL to BRAND_EMAIL on EMAIL_ID 4. Subscribed – where BRAND_EMAIL.opt_in_ind = 1 5. Corrected this update cycle – where EMAIL.modified_dt or BRAND_EMAIL.modified_dt equal to the current update cycle

BU6.4.3	The PID and Brand Code posted to the EMAILHYGIENE subscriber feed is the same as the original PID/Brand that prior to hygiene. In the event that an email address was subscribed to multiple PIDs the corrected email address is provided multiple times in the extract (once for each subscription).
BU6.5	Non-numeric data will be stripped from all phone number fields.
BU6.5.1	In the mapping to WRK_SRC_ACCOUNT all phone number mappings are passed through the following regular expression for cleansing non-numeric data; TRIM(REGEXP_REPLACE(a.HOME_PHN_NBR,'[^0-9]','')) AS home_phone_nbr,
BU6.5.2	This logic formats and validates phone number values CASE WHEN (phone_country_cd IN ('USA','CAN') or phone_country_cd is null) AND REGEXP_INSTR (phone_nbr,'^\(?([2-9][0-8][0-9])\)?[-.]?([2-9][0-9]{2})[-.]?([0-9]{4})$') = 0 THEN 0 ELSE 1 END AS valid_phone_ind
BU6.6	All numeric dollar fields will be loaded as a precision of 15,2 in stage/warehouse.
BU6.6.1	Data type conversions are done as part of the RAW to STG (stage) mapping phase. Where mapped to a numeric conversion is done via cast(${fieldName} as NUMERIC(15.2)) and then direct mapping to the Data Warehouse.
BU6.6.2	Cast conversion failures result in the rejection of the entire record as part of the Staging load.
BU6.7	Standardize all date and time formats based upon consistent inbound data types.
BU6.7.1	Inbound dates are provided in a variety of formats although consistent by source and feed.
BU6.7.2	Data type standardizations are done as part of the RAW to STG (stage) mapping phase. Dates and TimeStamp are converted such that the time component is truncated via trunc(cast(${fieldname as DATE)), except where explicitly noted otherwise.
BU6.7.3	CheetahMail/SDL email activity data retains the time component of the date value as part of the mapping.
BU6.7.4	Cast conversion failures result in the rejection of the entire record as part of the Staging load.
BU6.8	Accept only standard ASCII character sets.
BU6.8.1	Extended ASCII characters are accepted in support of international data.
BU6.8.2	The Oracle Data Warehouse and Mart nodes are configured for double-byte data in support of international character sets.
BU6.8.3	Interface agreement maximum data lengths are generally based on standard ASCII characters. This may result in data rejection based on load length subject to established rejection tolerances.
BU6.9	Trim all white space.
BU6.9.1	Every field is left and right trimmed of tab and space characters via TRIM(${fieldname})
BU6.10	Support for variable field delimiters.

BU6.10.1	Feed formats vary by source. The solution has been designed to accept each of these and load properly: Source / Format MarketWorks / Oracle DMP MarketWorks (PCI) / Tab-Delimited ADS / Fixed RightNow / Tab-Delimited BazaarVoice / XML SDL / Pipe-Delimited CheetahMail / Pipe-Delimited SAP / Pipe-Delimited Omniture (ClickStream) / Tab-Delimited Omniture (Visit) / Pipe-Delimited Omniture (SAINT) / Tab-Delimited ATG / XML AuditWorks / Pipe-Delimited KBM / Pipe-Delimited Experian / Pipe-Delimited Promotion History / Tab-Delimited

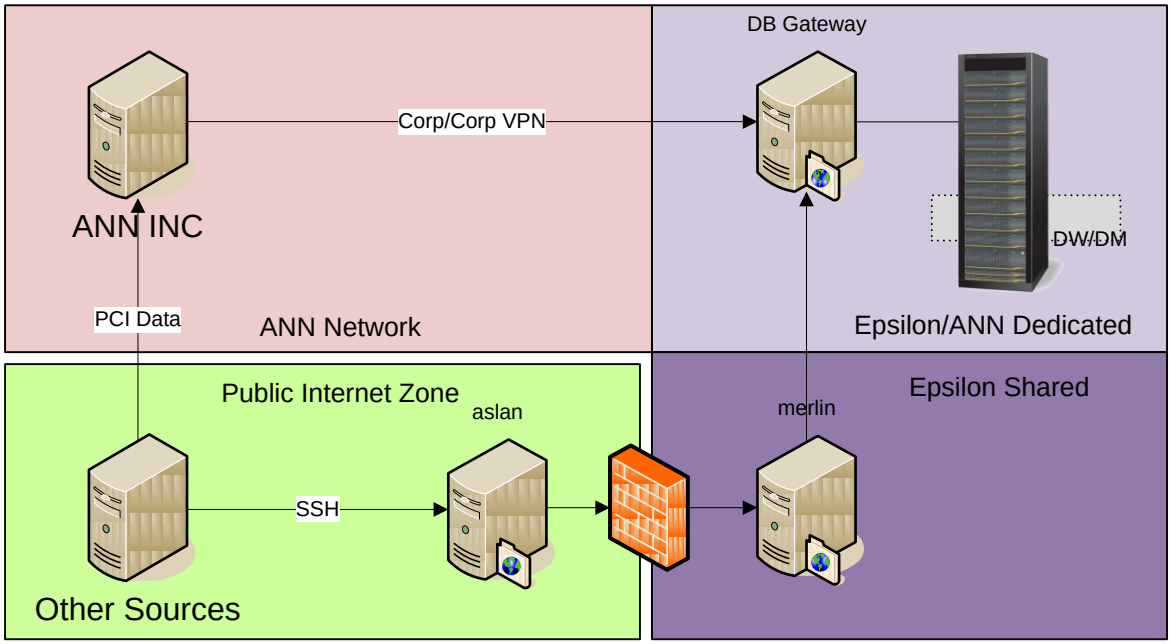
7. PCI

All handling of PCI data is done within the ANN INC infrastructure. This includes data receipt, initial RSA decryption, generating the hash surrogate value, feed regeneration, and posting to the Epsilon CRM infrastructure for processing. This includes the following sources:

- MarketWorks (PCI feeds only)
- ADS (XREF feeds only)
- AuditWorks
- ATG

All above sources are processed through the same decryption and hashing algorithm developed by ANN INC such that the hash value received by Epsilon retains the same distinct identifier quality associated with a Credit Card account. All use of this hash value is then done directly, facilitates our exact text matching needs for Retail Matchback, and protects ANN INC's customer private financial data from exposure within the CRM solution.

All feeds that have hashed Credit Card equivalent are sent over an SSH encrypted connection through a Corp/Corp VPN Tunnel accessible only by ANN INC and Epsilon.



#	Design
BU7.1	MarketWorks files that need to land in PCI environment to hash the credit card numbers: • Customer_Alternate_Key • TI_Transaction_Tender_Error • Email_POS_Arch • Email_POS_Arch3

BU7.1.1	MarketWorks feeds are hashed at ANN INC and provided in clear text through the double-encrypted transfers (SSH over SFTP through the secure VPN tunnel)
	• Customer_Alternate_Key
	  ○ Among other attributes, this feed provides Credit Card hash values as related to a MarketWorks Customer
	• TI_Transaction_Tender_Error
	  ○ MarketWorks unidentified transaction payment data. The Credit Card hash value is posted to the transaction RMB_IDENTIFIER table in the Data Warehouse for use in the Retail Matchback process to associate these transactions with Customers.
	• Email_POS_Arch & Email_POS_Arch3
	  ○ This data includes Email and Credit Card data obtained at retail stores. Both components are loaded into the Data Warehouse transaction RMB_IDENTIFIER table for later use in the Retail Matchback process.

BU7.1.2	The EMAIL_POS_ARCH does not have a direct relationship to MarketWorks transaction IDs, The All of the Names, Postal Codes, and Email Address on the POS feed are related to a CC#. This CC# is joined to the MW Tender ERROR table. Any name(s), postal code(s) or email address(s) associated with that CC# are assigned to the tender feed which ultimately is used to load the RMB_IDENTIFIER table for use in the retail matchback process.
BU7.2	ADS files that need to land in PCI environment to hash the credit card numbers: • Customer_Xref_Daily

BU7.2.1	ADS PCI feeds are provided to ANN INC in PGP encrypted format. ANN INC maintains the responsibility for the timeliness and quality related to receipt, processing, and transmission to Epsilon ahead of the pre-determined SLA.
BU7.2.1	ADS PCI feeds hashed at ANN INC and provided in clear text through the double-encrypted transfers (SSH over SFTP through the secure VPN tunnel) • Customer_Xref_Daily ○ This feed is used to provide and Credit Card equivalent attribute associated with an ADS account. It is later used to attempt to link to MarketWorks, AuditWorks, or ATG unidentified transactions through the Credit Card used on the order.
BU7.3	ATG files that need to land in PCI environment to hash the credit card numbers: • Transaction_Orders
BU7.3.1	ATG PCI feeds are provided to ANN INC with Credit Card fields encrypted via RSA. ANN INC maintains the responsibility for the timeliness and quality related to receipt, processing, and transmission to Epsilon ahead of the pre-determined SLA.

BU7.3.2	ATG PCI feeds hashed at ANN INC and provided in clear text through the double-encrypted transfers (SSH over SFTP through the secure VPN tunnel)
	• Transaction
	  ○ Credit Cards are used on ATG as part of the Retail Matchback process to build both transaction and account datasets
	• Profile
	  ○ The ATG profile feed contains credit card data in the form of billing preferences, however ATG account data is derived directly from the transaction as opposed to this feed.

BU7.4	AuditWorks files that need to land in PCI environment to hash the credit card numbers: • Transaction (Line Notes record types)
BU7.4.1	AuditWorks PCI feeds are provided to ANN INC with Credit Card fields encrypted via RSA. ANN INC maintains the responsibility for the timeliness and quality related to receipt, processing, and transmission to Epsilon ahead of the pre-determined SLA.
BU7.4.2	AuditWorks PCI feeds hashed at ANN INC and provided in clear text through the double-encrypted transfers (SSH over SFTP through the secure VPN tunnel) • Transaction (line notes record type) • Credit Cards are used on ATG as part of the Retail Matchback process to build both transaction and account datasets

8. AGILITY/ CDI

The AGILITY Identity Management (AIM) subsystem consists of a series of standard and customized processes leveraged for the purpose of providing the solution with the mechanism by which customer data is integrated across variant data sources. As part of this process AGILITY performs the following functions:

- Individual Name Parsing and Standardization for customized matching rules
- Address Field Parsing and Standardization for customized matching rules
- CASS & SERP Address Verification and Correction to optimize direct mail deliverability
- Proprietary Address Correction (PAC) provide missing address elements through high confidence matching to vast data assets

- Change of Address – both 48-month USPS as well as 15-year historical proprietary data repository based on self-reported move-tracking
- Standard Suppression Identification – DMA Pander, Prison, Deceased, Vulgar
- Standard Street-level and unit designated Address identification
- Maintain Client-specific data repository in support of customized identification of
    - Individuals
    - Households
- Leverage non-name/address attributes for additional consolidation
    - Email Address Hard Key
    - ADS Account Number Hard Key
- Best-Address selection (NCOA, PCOA, LACS, CASS, Source)
- Mailability Score and delivery point type indicators
- Distance to Store calculation

As part of the larger **Customer** phase of the Data Warehouse update, AGILITY is processed on a grid external to the CRM solution, but is performed as an in-line service as part of the daily update cycle.

The AGILITY process is provided with the set of distinct name, address, and hard key elements received that day. It then returns to the CRM solution standardized addressing information, AGILITY IDs, and a number of delivery point descriptive indicators. It is important to note that, the Data Warehouse maintains both the original source name and address information on the ACCOUNT table and the standardized/corrected name and address information on the POSTAL_CONTACT table. The mart then provides visibility exclusively to the standardized name and address by way of a Best Postal Contact.

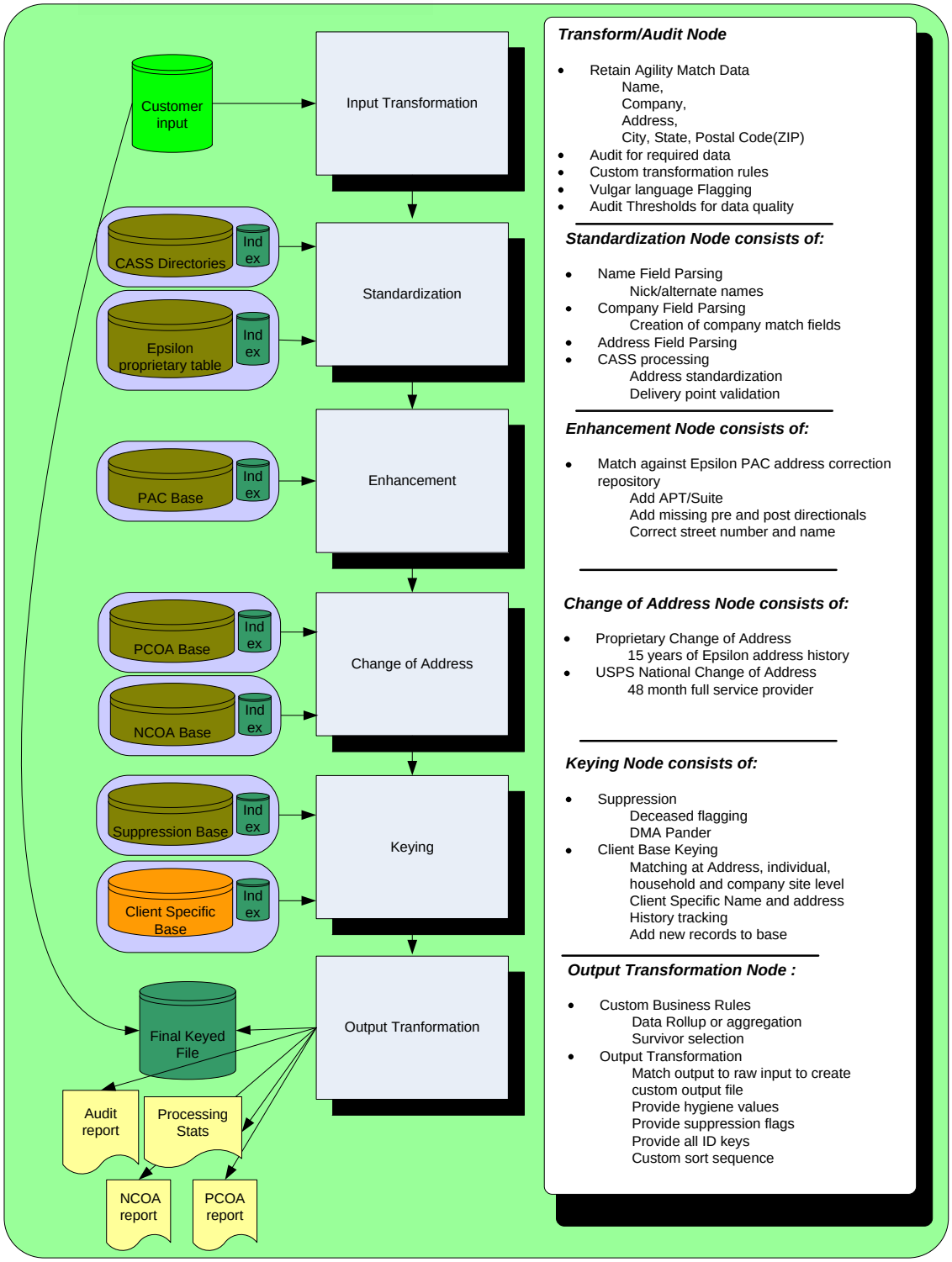
Quarterly, the established base of Postal Contacts is rerun through the entire Agility process as part of a base *refresh* or *swing*. This performs two primary functions:

- Pick up recent Change of Address / USPS data for Postal Contacts processed previously
- Provide new opportunities for Individual/Household level consolidation through
    - Modifications made to address data
    - Upgrades to the AGILITY system
- Re-calculation of nearest single store data by brand.

Note: SAS users can always derive store proximity through the LAT/LONG on the SUM_INDIVIDUAL (or SUM_HOUSEHOLD) tables vs. LAT/LONG on the LU_STORE table. LAT/LONG is assigned incrementally and refreshed quarterly. The LAT/LONG on the store is refreshed weekly.

The refresh process provides the CRM solution with updated address information as well as household and individuals cross-references in order to establish the updated linkages in the Data Warehouse.

Process Flow



#	Design
BU8.1	Files to process through Agility;

	• Return files from KBM or Experian (Name/ Zip Append, Reverse Email Append, Trade Area Append) • MW – Bill-to (Create name/ address file from Customer and Address tables – active addresses only) • MW – Ship-to (Create name/ address file from Customer and Address tables – active addresses only) • MW - Email_POS_Arch, Email_POS_Arch3 • ADS - Customer Daily and Customer Monthly files • RightNow - Customer Incident file • CheetahMail - Change of Demographic file • SAP – Associate_Reference file • SAP – Store File (for latitude and longitude assignment only) • ATG – Transaction Bill-to ATG – Transaction Ship-to • ATG – Customer Profile • ATG – Catalog_Requests file • AW – Transaction (Customer Detail record types)
BU8.1.1	As detailed in the **Customer** section of the Data Warehouse definition this following sources of data contribute to the Agility data flow: • Return files from KBM or Experian (Name/ Zip Append, Reverse Email Append, Trade Area Append) • MW – Bill-to (Create name/ address file from Customer and Address tables – active addresses only) • MW – Ship-to (Create name/ address file from Customer and Address tables – active addresses only) • MW - Email_POS_Arch • ADS - Customer Daily and Customer Monthly files • RightNow - Customer Incident file • SDL - Change of Demographic file • SAP – Associate_Reference file • SAP – Store File (for latitude and longitude assignment only) • ATG – Transaction Bill-to ATG – Transaction Ship-to • ATG – Customer Profile • ATG – Catalog_Requests file • AW – Transaction (Customer Detail record types)

BU8.1.2	Data that would not get AGILITY keys returned because they do not contain enough information to identify an individual are filtered from the process. In order to run through AGILITY a record must have at least one of the following:
	• Address Data
	  ○ Last Name populated
	  ○ Address Line (line1) populated
	  ○ Either a postal code or both city and state
	• Email Hard Key
	  ○ First Name populated
	  ○ Last Name populated
	  ○ Email Address populated
	• ADS Account Hard Key
	  ○ Last Name populated

	○ ADS Account Number populated
BU.8.1.3	Email Address that only have a zip code are not identified in the database as INDIVIDUALS. The postal code is stored on the BRAND_EMAIL table as provided on the SDL profile, however any "Stores Near You" calculation would need to be done within SAS based on the postal code (because postal code by itself is not enough to derive a LAT/LONG which is required for proximity calculations)
BU8.1.4	Once candidate data is identified from all sources, the process de-dupes the input file on exact matches of full name, all address components, all hard key values. This is done to manage the quantities processed through the AGILITY process particularly when transaction data is included among the AGILITY sources and expected to have identical address information across transactions.
BU8.1.5	The feeds are generated in pipe-delimited format with the following naming convention: • ANN_AGD_TOAGILITY_yyyymmdd.dat
BU8.1.6	The process later associates all de-deuped AGILITY data back to the original source records such that the keys that would have otherwise been applied through AGILITY directly benefit from the same address processing.
BU8.1.6	The feeds are returned in pipe-delimited format with the following naming convention: • ANN_AGD_FROMAGILITY_yyyymmdd.dat • ANN_AGD_INDXREF_yyyymmdd.dat
BU8.2	No records should be dropped during the consolidation process. IDs will be propagated to original incoming data, and consolidated fact summary tables will be created in the dart mart.
BU8.2.1	The AGILITY process is provided an extract of distinct name, address, and hardkey data to run through the process. The exact number of number of records is returned to the CRM solution for load into the Data Warehouse
BU8.2.2	Agility Extract and Load counts are retained in the FILE_DETAIL audit table. An audit check is performed as part of the Data Warehouse load to verify these quantities match.
BU8.2.3	An audit is performed on the daily XREF file that is provided for each execution. An audit is performed to verify that all new or remapped keys are included in the XREF. Transient keys may be found in the XREF and are ignored.
BU8.3	Domestic(US) address standardization and hygiene (see BU15.5)
BU8.3.1	US addresses are standardized via SAP ACE (Address Correction and Encoding)
BU8.4	Canadian address standardization and hygiene (see BU15.5)
BU8.4.1	Canadian addresses are standardized via StreetSweeper
BU8.4.2	In an effort to maximize effectiveness of the StreetSweeper application, Canadian addresses are cleansed prior to processing. This includes variable rules by source: • Conversion of province names to Codes • Blanking out malformed postal codes ○ Less than 6 non-whitespace characters ○ All numeric • Country code assignment based on province and/or postal code designation
BU8.4.3	The country code field is leverage by the Agility process to identify which records should process through Canadian (StreetSweeper) and US (ACE)
BU8.5	Matching/ keying including the assignment and ongoing maintenance of 3 unique identifiers: Address ID, Household ID, and Individual ID. All 3 identifiers are numeric, the Address ID is 32 bytes and the Household ID and Individual ID are each 13 bytes. The first 26 bytes of the Address ID identifies the physical

	address at the street level, and the last 6 bytes represents any secondary information such as apartment number. • An Individual is defined as a unique First Name and Last Name at an Address or a unique First Name and ADS Account ID or a unique First Name and Last Name and Email Address. • A Household is defined as a unique Last Name (Surname) at an Address. • Agility hygiene rules are set as outlined in **Appendix E**
BU8.5.1	The Ann Taylor Agility configuration is based upon Epsilon best practices for matching Retail Customer Data in an Agility Identity Management system. This configuration is one of the most often used configuration sets at Epsilon and is in place with nearly all of our users in the Retail Customer Data category. This configuration is moderately more conservative than the set one would use for Retail Prospect Data. Specifically, since using first name initials can sometimes lead to over-consolidation, all such first name initial matching is disabled with this configuration.
BU8.5.2	The native address ID is referred to as a Global Address ID (GAID), it's identification is performed through matching to an Epsilon global (implementation-rule-agnostic) data repository. It has two components: • Bytes 1-21 = Street Level Address Identifier • Bytes 22-32 = Secondary Address Identifier
BU8.5.3	The Address ID is converted to NUMERIC(32,0) in the data warehouse load
BU8.5.4	Should the solution need access to the street-level address identifier this is performed via the following function: cast(substr(trim(to_char(gaid, '00000000000000000000000000000000')), 1, 21) as numeric(21, 0)) as GAID21
BU8.5.5	The Individual ID is derived through a series of comparisons done with both address and name components. These identification rules are defined fully in the Business Requirements Document (BRD) Appendix
BU8.5.6	The Individual ID is converted to NUMERIC(13, 0) in the data warehouse load
BU8.5.7	The Household ID is derived through a series of comparisons done with both address and surname components. These identification rules are defined fully in the Business Requirements Document (BRD) Appendix
BU8.5.8	The Household ID is converted to NUMERIC(13, 0) in the data warehouse load
BU8.6	Hard Key matching will be leveraged for ANN including the following prioritized rules; ADS Account ID with First Name Email Address with First Name and Last Name Hard Key matching rules will be applied to all sources in priority order. If a particular source does not have a specific Hard Key value, then the rule will not apply. Agility Hard Key matching rules are set as outlined in **Appendix E**
BU8.6.1	When matching on ADS Account ID, Agility will use the Hard Key value along with the first name. The Hard Key value is an exact string match. So the value must match 100%. Once the string match is made the first names are analyzed using the Agility algorithms.
BU8.6.2	When matching on EMAIL, Agility will use the Hard Key value along with the full name. The Hard Key value is an exact string match. So the value must match 100%. Once the string match is made the names are analyzed using the Agility algorithms.
BU8.6.2	Hard Key matching is attempted on all available hard keys / name combination send to AGILITY on a record to attempt to identify all possible relationships.

<table>
<tr><td>BU8.7</td><td>If multiple values for a given hard key such as email address are present on given record during an update, only one value will be passed to Agility for keying purposes.
For instance; RightNow has an email address and 2 alternate email address fields. In priority order - email address will be the email address for hard key if it passes hygiene, if not alternate email address 1 will be used if it passes hygiene, if not alternate email address 2 will be used.</td></tr>
<tr><td>BU8.7.1</td><td>The ADS account feeds have a single ADS account number. It is the only field this hardkey is derived from.</td></tr>
<tr><td>BU8.7.2</td><td>The RightNow Incident feed supports up to 3 email addresses. The email selected for hardkey purposes is the first one found using the following lookup sequence:
EMAIL
EMAIL_ALT1
EMAIL_ALT2
Should multiple email addresses be found on a single incident record the additional email addresses are ignored for hard key individual identification purposes.</td></tr>
<tr><td>BU8.7.1</td><td>The ATG transaction Email addresses are selected as follows:
BillingAddress Node – use for ATG Billing Account build
ShippingAddress Node – use to ATG Shipping Account build
Order Node – email address ignored</td></tr>
<tr><td>BU8.7.2</td><td>The AuditWorks transaction email addresses are selected as follows
Customer Role = 1 – use for AW Billing Account build
Customer Role = 2 - use for AW Shipping Account build</td></tr>
<tr><td>BU8.7.3</td><td>The remaining account sources include a single email address to select from when populated:
ADS Monthly/Daily
SDL Subs/Demographics
ATG Profile
Experian Reverse Email Append</td></tr>
<tr><td>BU8.7.4</td><td>These account sources do not have an email address to provide
ATG Catalog Requestor
SAP Associate
Experian Name/Zip Append
Experian Trade Area Append</td></tr>
<tr><td>BU8.8</td><td>The address fields need to support international field lengths for future processing.</td></tr>
<tr><td>BU8.8.1</td><td>The AGILITY inbound layout includes these fields in support of future foreign address processing:
<table>
<tr><td>SRC_FIRM_NM</td><td>VARCHAR</td><td>50</td></tr>
<tr><td>SRC_ADDR_LINE_1</td><td>VARCHAR</td><td>65</td></tr>
<tr><td>SRC_ADDR_LINE_2</td><td>VARCHAR</td><td>65</td></tr>
<tr><td>SRC_ADDR_LINE_3</td><td>VARCHAR</td><td>65</td></tr>
<tr><td>SRC_ADDR_LINE_4</td><td>VARCHAR</td><td>65</td></tr>
<tr><td>SRC_CITY</td><td>VARCHAR</td><td>50</td></tr>
<tr><td>SRC_STATE_CD</td><td>VARCHAR</td><td>40</td></tr>
<tr><td>SRC_POSTAL_CD</td><td>VARCHAR</td><td>10</td></tr>
</table></td></tr>
</table>

SRC_COUNTRY_CD	VARCHAR	40	

The AGILITY outbound layout includes these fields in support of future foreign address processing:

BEST_ADDR_LINE_1	VARCHAR2(100CHAR)
BEST_ADDR_LINE_2	VARCHAR2(100CHAR)
BEST_ADDR_LINE_3	VARCHAR2(100CHAR)
BEST_ADDR_LINE_4	VARCHAR2(100CHAR)
BEST_CITY	VARCHAR2(50CHAR)
BEST_STATE_CD	VARCHAR2(50CHAR)
BEST_COUNTRY_CD	VARCHAR2(3CHAR)
BEST_POSTAL_CD	VARCHAR2(10CHAR)
BEST_ZIP4	VARCHAR2(4CHAR)
BEST_LOCALITY_NM	VARCHAR2(100CHAR)
BEST_CARRIER_ROUTE_CD	VARCHAR2(4CHAR)
BEST_DELIVERY_PT	VARCHAR2(2CHAR)
BEST_DELIVERY_CKDG	VARCHAR2(1CHAR)
BEST_DPBC	VARCHAR2(2CHAR)
BEST_LOT_NBR	VARCHAR2(4CHAR)
BEST_LOT_SEQ	VARCHAR2(1CHAR)
BEST_ACE_LATITUDE	VARCHAR2(10CHAR)
BEST_ACE_LONGITUDE	VARCHAR2(11CHAR)
BEST_ACE_REC_TYPE	VARCHAR2(2CHAR)
BEST_ACE_ERROR_CD	VARCHAR2(4CHAR)
BEST_ACE_ERROR_STATUS	VARCHAR2(6CHAR)
BEST_ACE_DPV_STATUS	VARCHAR2(1CHAR)
BEST_DPV_FTNOTE	VARCHAR2(12CHAR)
BEST_ACE_FOREIGN_CD	VARCHAR2(1CHAR)
BEST_BUSINESS_IND	VARCHAR2(1CHAR)
BEST_FOREIGN_IND	VARCHAR2(1CHAR)
BEST_DSF2_MATCH_LVL	VARCHAR2(1CHAR)
BEST_DSF2_VACANT	VARCHAR2(1CHAR)
BEST_DSF2_SEASON	VARCHAR2(1CHAR)
BEST_DSF2_RESBUS	VARCHAR2(1CHAR)
BEST_DSF2_DELTYPE	VARCHAR2(1CHAR)
BEST_DSF2_DELPTDRP	VARCHAR2(1CHAR)
BEST_MAIL_SCORE	VARCHAR2(1CHAR)
BEST_CASS_DELIVERABLE_IND	VARCHAR2(1CHAR)
BEST_PRISON_IND	VARCHAR2(1CHAR)
BEST_PANDER_IND	VARCHAR2(1CHAR)
BEST_DECEASED_IND	VARCHAR2(1CHAR)
BEST_APO_FPO_IND	VARCHAR2(1CHAR)
BEST_US_POSS_IND	VARCHAR2(1CHAR)
BEST_DPV_CMRA_IND	VARCHAR2(1CHAR)
BEST_DPV_FP_IND	VARCHAR2(1CHAR)
BEST_DPV_THROWBACK_IND	VARCHAR2(1CHAR)
BEST_FIPS_STATE	VARCHAR2(2CHAR)
BEST_FIPS_COUNTY	NUMBER(3)

BU8.8.2	
BU8.9	All household processing will be based on standardized addresses.
BU8.9.1	Household and Individual ID assignment is derived from name components and Global Address ID (GAID). The GAID is based on COA/Standardized address components.
BU8.10	NCOA/PCOA refresh of all active postal contacts will run 4X/ Year (Quarterly).

BU8.10.1	The AGILITY Refresh process executes quarterly starting with the first refresh performed after acceptance and quarterly based on that date
BU8.10.2	The refresh process includes an extract of all postal contacts in the Data Warehouse
BU8.10.3	The feeds are generated in pipe-delimited format with the following naming convention: • ANN_AGR_TOAGILITY_yyyymmdd.dat
BU8.10.4	The feeds are returned in pipe-delimited format with the following naming convention: • ANN_AGR_FROMAGILITY_yyyymmdd.dat • ANN_AGR_INDXREF_yyyymmdd.dat
BU8.11	NCOA/PCOA of all incoming name and address records as part of the daily update process.
BU8.11.1	All data processed through AGILITY is run through NCOA, Canadian COA, and PCOA as appropriate. This includes all daily incremental as well as quarterly refresh.
BU8.12	Post Agility hygiene reports (after each update) will be sent to the Client portal; DQA (Data Quality Audit) CASS.
BU8.12.1	The following AGILITY audit reports are automatically posted to the Epsilon-hosted ANN Sharepoint portal through the mechanism described below: • DQA (Data Quality Audit) including Duplicates Report • CASS (3553)

BU8.12.2	To post to the Client Portal, an email is sent using the following attributes:
	• To: ANN_SP_AUD@epsilon.com
	• From: jcsuser@epsilon.com
	• Subject: "Agility Audit"
	• Attachment(s): all reports to post
	  ○ ANN_DQA_yyyymmddhh24miss.xls
	  ○ ANN_agility_dupe_report_yyyymmddhh24miss.zip
	  ○ ANN_CASS_yyyymmddhh24miss.353
	
	This posts the attached reports to the following location: **Home > Document Repository > Agility Audit**

BU8.13	The system will store a cross reference of all Agility keys to customer account number relationship. This cross reference can be used to track merged and split accounts.
BU8.13.1	Each daily incremental update the AGILITY cross-reference is applied loaded into STG_AG_INDIV_XREF: • ORIG_INDIV_ID • NEW_INDIV_ID • ACTIVITY_DT • FILE_ID • RECORD_NBR • ALTERNATE_KEY • FILE_TYPE_CD • FILE_TS
BU8.13.2	The STG_AG_INDIV_XREF is used to establish the relationship between an ACCOUNT, POSTAL_CONTACT and INDIVIDUAL in the Data Warehouse. When this cross-reference is processed the IDs on these tables are aligned to relate to the current INDIVIDUAL.
BU8.13.3	The incremental and refresh cross-reference data is retained online for as long as storage capacity allows in STG_AG_INDIV_XREF_HIST to facilitate the auditing of historical ID

	migration. The structure of this table is identical to STG_AG_INDIV_XREF
BU8.13.4	The mart manifests the account to individual relationships in the form of a small reference table as a dimension of Individual that identifies each of their Account Sources and Account Numbers. The direct relationship of an ADS account as provided by MW is not retained however the link through the individual establishes the relationship of ADS and MW accounts (as well as all other account sources)

9. Distance to Store Assignment

#	Design
BU9.1	Epsilon will pass the ANN provided SAP Store reference file to Agility each week as well as a full refresh each quarter. This file includes the physical address of each Ann Taylor Store, Ann Factory Store, LOFT Store and LOFT Outlet Store. • Closed stores and Web (611,612,618,619) are excluded from this process.
BU9.1.1	AGILITY is provided with the incremental refresh of all open and NON-WEB stores with the following extract: select ${JOBID} as job_id, a.ALTERNATE_KEY as RECORD_NBR, a.BRAND_CD as BRAND_CD, a.STORE_NBR as STORE_NBR, a.STORE_NM as STORE_NM, a.STORE_ADDRESS_1 as ADDRESS_1, a.STORE_ADDRESS_2 as ADDRESS_2, a.STORE_CITY as CITY, a.STORE_STATE as STATE_CD, a.STORE_POSTAL_CD as ZIP_CODE, a.STORE_COUNTRY_CD as COUNTRY_CD, a.STORE_STATUS_CD as STATUS_CD, a.STORE_OPEN_DT as OPEN_DT, a.STORE_CLOSE_DT as CLOSE_DT from dw_owner.STORE a where a.STORE_NBR not in (611,612,618,619) and CLOSED_DT is not null;
BU9.1.2	The feed is sent in pipe-delimited format with the following naming conventions: • ANN_AGR_STORE_yyyymmdd.dat
BU9.2	Agility will assign the Latitude and Longitude for each store location. The process will assign roof-top level of precision.
BU9.2.1	Agility will assign the Latitude and Longitude for each store location. The process will assign roof-top level of precision. Each incremental and refresh update an updated STORE feed is send back to the CRM solution. The feed is returned in pipe-delimited format with the following naming conventions: • ANN_AGR_STOREUPDATE_yyyymmdd.dat

BU9.2.2	The following STORE attributes in the data warehouse are updated based on the specified store number: • STORE_GAID • STORE_LATITUDE • STORE_LONGITUDE
BU9.2.3	The full unit-designated GAID is stored in the Data Warehouse for each store. This provides the mechanism to identify either the unit or street-level GAID. The latitude and longitude is based on street-level address identification.
BU9.3	During the daily/ weekly Agility processing of each Client feed, the distance to closest store (any of the 4 store types) will be assigned including the store number. The distance assigned is a distance in miles "as the crow flies" and not an actual driving distance.
BU9.3.1	The Agility process uses the address, brand code, and store number of the STORE feed to append the following attributes • CLOSEST_STORE_NBR • CLOSEST_STORE_DIST • CLOSEST_AT_STORE_NBR • CLOSEST_AT_STORE_DIST • CLOSEST_LOFT_STORE_NBR • CLOSEST_LOFT_STORE_DIST • CLOSEST_ATFS_STORE_NBR • CLOSEST_ATFS_STORE_DIST • CLOSEST_LOS_STORE_NBR • CLOSEST_LOS_STORE_DIST
BU9.3.2	Proximity calculations for additional stores can be derived in SAS through lat/long identifiers on both the SUM_INDIVIDUAL/SUM_HOUSEHOLD tables vs LU_STORE
BU9.3.2	Distances are calculated in miles (numeric(10,2)) based on a straight-line lat/long relationship to the customer's address
BU9.3.3	These attributes of a customer are used to establish the POSTAL_CONTACT_STORE table in the data warehouse. This relates a POSTAL_CONTACT to a store for each brand
BU9.3.4	In the Data Mart, the closest store attributes by brand for the BEST postal contact are displayed to the end user on the SUM_INDIVIDUAL table. Similarly, the closest store attributes by brand for the Head of Household are displayed to the end user on the SUM_HOUSEHOLD table.
BU9.4	Every quarter there will be an Agility refresh that runs on the entire client table, at this time the closest store distance and store number append will be reapplied to all client records.
BU9.4.1	As part of the refresh process, the full store feed is extracted out to AGILITY. The existing store reference is removed and replaced with the new dataset updated with current Lat/Long data.

BU9.4.2	The refresh postal contact feed is then rerun through Address Standardization and COA processing and all store assignments are recalculated based on the update store reference. The updated attributes are then fed back into the CRM data warehouse and updated on the POSTAL_CONTACT table.

Data Warehouse Design

The Data Warehouse layer standardizes all of the source data from historical migration as well as incremental updates. Each feed is loaded into the database structures in one of the following ways:

Direct table mappings with minimal transformation (i.e. BazaarVoice Ratings)

Joining several feeds to into a single denormalized structure (i.e. SAP Product)

Through splitting a staged dataset amongst one or more target tables (i.e. Account Sources)

Through the significant restructuring of source data through a series of transformations, joins, and decision branching to arrive at database structures appropriate to a Marketing platform (i.e AW & ATG transaction data)

The above feed to table mappings are summarized in **Appendix B – Source to Target Overview**

In general, the attempt is made to retain the ability to reference native source data within these structures; however when standardizing similar data across several sources there is often the need to employ transformation rules in order to arrive at consistent definitions and optimize the use of available storage capacity. So, while each and every field is not preserved in its original format, the system maintains the most important native key definitions and source data used for complex transformation decisions should they be required for later reference.

The Data Warehouse update generally employs an *upsert* process regardless of whether the inbound feeds are incremental or full replacement. An *upsert* is defined deleting existing records in the data warehouse that match the key of an incoming update and then inserting all new and replacement records. Existing records are only replaced if the activity date of the incoming record is equal to or later than what already exists in the Data Warehouse. This is done to maintain historical referential integrity. There are some specific cases (i.e Transaction) where a matching key on an incoming record will force the deletion of records in several data warehouse tables prior to loading.

The Data Warehouse maintains several structures designed to facilitate the ability to audit the load process over time and address questions as they arise. These include:

Identifiers to indicate the source feed, record, job ID, and Date that originally inserted a row into a Data Warehouse table

Identifiers to indicate the source feed, record, job ID and Date that most recently updated a row in a Data Warehouse table

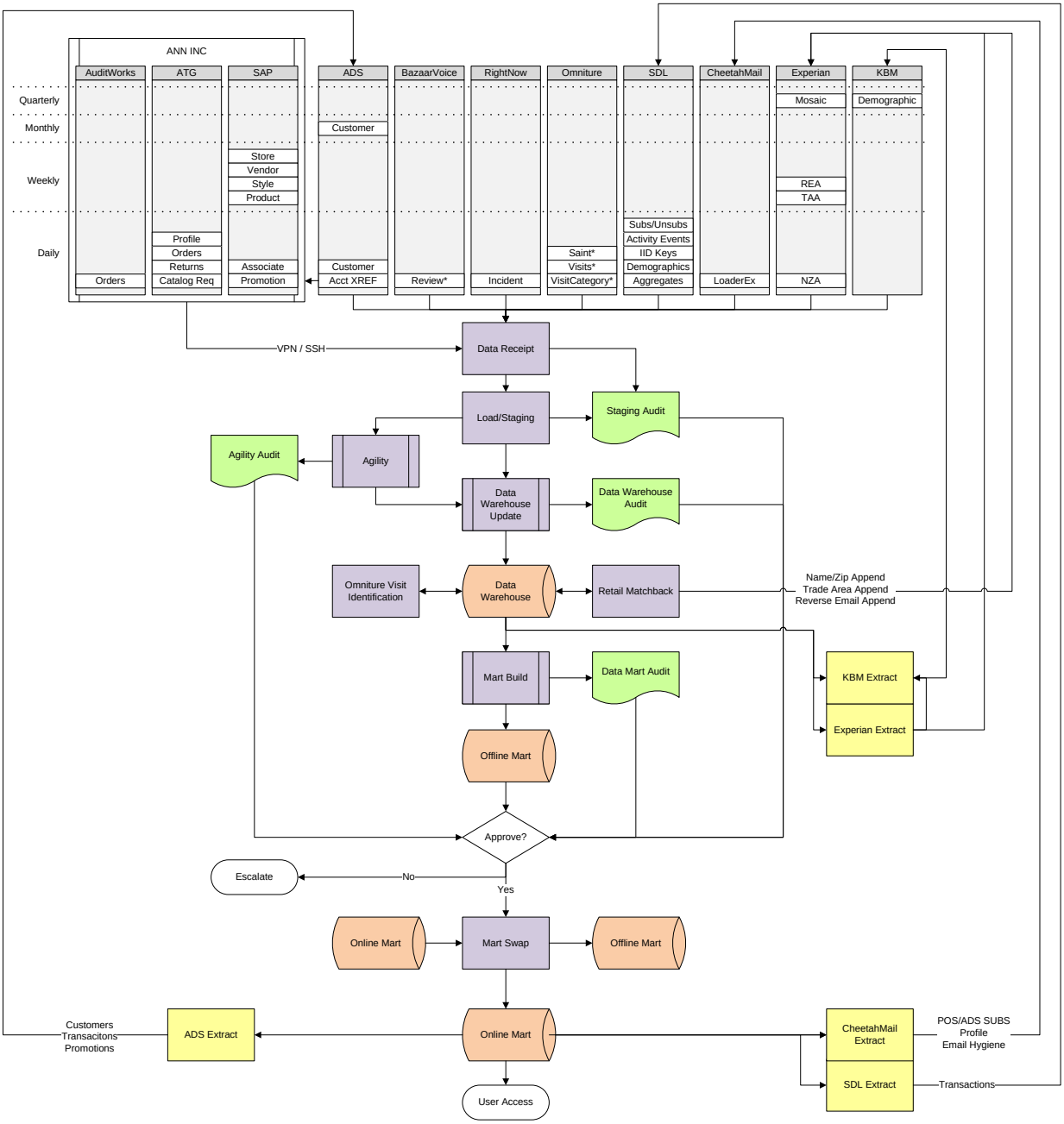
History tables to track each change to IDs or preferences

Audit tables to report on update process results

Audit tables of table and field profiles that can be used for trending and thresholding purposes

This section also describes pieces of the Data Mart build process where objects in a database subject area promoted as Fact or Dimension structures. In other words, for all those table except where the mart structures are completely derived from Data Warehouse facts (rollups).

Process Flow



Common Data Warehouse

#	Design
BU10.1	The system will allow for the updating of existing data as well as insertion of new data in the data warehouse.
BU10.1.1	Each source feed is loaded and mapped into one or more target Data Warehouse structures. Source feeds may also be joined on key relationships prior to Data Warehouse mapping. These mapping are fully documented in the supplementary **Stage to Data Warehouse Source to Target** documents.
BU10.1.2	Based on the Primary Key definition of the target structure, the ELT Framework identifies if the source key value already exists. If not, the record is *inserted* and any appropriate transformations are applied. If the key does exist, every attribute on the Data Warehouse table is *updated* based upon the source values and appropriate transformations, This methodology is generally referred to as *upsert* logic.
BU10.1.3	Upsert logic is applied to all Data Warehouse mappings except where explicitly instructed to otherwise completely truncate the target Data Warehouse table in favor of loading all of the full refresh source records. Upsert logic is generally preferred because replacing target structures can often lead to orphaned data in the Warehouse. Should ANN decide to only make the latest data visible to the end-user, these are easily identified filtered from data warehouse as part of the mart build process through the use of ELT attributes (last update job id, date, and file id) The ANN solution does not require the full reload of any target DW structure.
BU10.1.4	All ANN sources are resending the full set of transaction data (all lines) for both new and updated orders. In order to insure consistent relationships in the Data Warehouse for all transaction data, the system deletes all records from the following tables with a matching incoming transaction key prior to re-insertion from the source feed. • TXN_HEADER • TXN_ITEM • TXN_ITEM_DISCOUNT • TENDER • RMB_IDENTIFIER
BU10.1.5	When updating a Data Warehouse table all fields are updated except where a field is explicitly identified as “retained”. Retained fields are never overwritten and thus retain their value as determined on initial load. (i.e. derived first transaction date, or initial load status)
BU10.2	The system will load each feed instance and will store a unique identifier in the form of a timestamp on both the staging and data warehouse structures. This will enable refresh/replace and rollback functionality.
BU10.2.1	The ELT Framework maintains the following attributes on each record in the Data Warehouse in order to facilitate the potential need to audit data through the process and/or perform system database rollbacks: • ACTIVITY_DT – the value representing the relevancy date of the source record (i.e. created date). Where not available, the current load date is used. • CREATED_JOB_ID – the Job Control System (JCS) job ID that first inserted the record. Used to tie to overall metrics and logs of the update process. • CREATED_DT – the date the record was initially loaded • ORIG_ALTERNATE_KEY - concatenation of distinct feed file ID and Record ID for tying any record back to a specific feed instance. This is the file/record that was used for the initial insert • MODIFIED_JOB_ID – the latest Job Control System (JCS) job ID that upserted

the record.

- MODIFIED_DT – the latest date the record was upserted
- ALTERNATE_KEY – the unique file/record that last updated the Data Warehouse record

BU10.3

The system will identify brand upon loading data into the data warehouse. Brand will be identified as (AT,ATF,LOFT,LOS) this corresponds to level at which permissions are captured at.

Note: Ecom trans will be loaded with AT or LOFT brands with a channel code = web

BU10.3.1

The following Data Warehouse table include a BRAND identifier with derivation rules appropriate to each source feed as described in the **Stage to Data Warehouse Source to Target mapping documents**:

TABLE_NAME	Brand Source
ACCOUNT	MW - Transaction Division; Customer Segmentation Flag/Code, Attr Codes ADS - Division Number, CC Logo SDL - PID ATG - Cat Req Code, Site ID RN - Prod ID AW - SAP Store SAP - 'ENT' KBM - Extracted Brand
ACCOUNT_EMAIL	From ACCOUNT
ACCOUNT_EMAIL_HIST	From ACCOUNT
ACCOUNT_EVENT	From ACCOUNT
ACCOUNT_HARD_KEY	From ACCOUNT
ACCOUNT_PHONE	From ACCOUNT
BRAND_EMAIL	PID Decode
BRAND_EMAIL_AGGREGATE	PID Decode
CAMPAIGN_TAXONOMY	Keycode Decode
CSC_INCIDENT	Prod ID
EMAIL_OPTOUT_HIST	PID Decode
EMAIL_PROMOTION	Keycode/PID Decode
EMAIL_RESPONSE	PID Decode
MODEL_RUN	Model Definition
MODEL_SCORE_CURRENT	Model Definition
MODEL_SCORE_HISTORY	Model Definition
PROMO_HISTORY	Keycode Decode
REVIEW	Feed Naming Convention
RMB_IDENTIFIER	From TXN_HEADER
SITE_VISIT	From TXN_HEADER
SITE_VISIT_UNID	From TXN_HEADER
TENDER	From TXN_HEADER
TXN_HEADER	MW - Division AW - SAP Store ATG - Site ID
TXN_ITEM	From TXN_HEADER
TXN_ITEM_DISCOUNT	From TXN_HEADER
VISIT_MERCH_CATEGORY	ATG Order/ACCOUNT

BU10.3.2

The BRAND derived at the Data Warehouse layer is leveraged in the Data Mart build process to provide separate and distinct views of the solution both by Brand as well as at the Enterprise level.

BU10.3.3

The following Data Warehouse tables do not include a BRAND designation

TABLE_NAME

	ACCOUNT_SOURCE
	ACCT_NBR_XREF
	CM_ISSUE
	DEMOGRAPHIC
	DOMAIN_CORRECTION
	EMAIL
	EMAIL_PROFILE
	EMAIL_RESPONSE_TYPE
	FISCAL_CALENDAR
	GEOSEGMENT
	INDIVIDUAL
	MOSAIC
	OMN_CAMPAIGN_KEY_MAP
	PHONE
	PHONE_TYPE
	POSTAL_CONTACT (note: includes BRAND_CD as attribute based on BEST Account, but not included in Primary Key)
	POSTAL_CONTACT_ARCHIVE
	POSTAL_CONTACT_STORE
	PRODUCT
	PRODUCT_PROMO
	REF_DOMAIN
	RFM_MAP
	RMB_CURR_POOL
	RMB_ID_TYPE
	RMB_METHOD_VALUE
	RMB_POOL
	RMB_POOL_MEMBER
	STORE
	VISIT_MERCH_CATEGORY_UNID
BU10.4	The system will assign the individual identifier from AGILTY as the coupon Client account number for use in generating a barcode for direct marketing campaigns.
BU10.4.1	The system will assign the individual identifier from AGILTY hashed to an 8-byte number as the coupon Client account number for use in generating a barcode for direct marketing campaigns.
BU10.5	The system will keep all keys fields the same name across all tables.
BU10.5.1	All foreign key relationships in the Data Warehouse maintain the same field names or field name pre-appended with a role between tables (i.e. TXN_NBR in txn_header joins to TXN_NBR in txn_item). The one exception is FISCAL_CALENDAR.calendar_dt joins to any date field in the warehouse.

BU10.5.2	In the Data Mart, where multiple joins are done between two tables the field name unique define the join as follows: • f_txn_item.BILL_INDIV_ID = sum_individual.INDIV_ID • f_txn_item.SHIP_INDIV_ID = sum_individual.INDIV_ID • sum_individual.CLOSEST_AT_STORE_ID = store.STORE_ID • sum_individual.CLOSEST_LOFT_STORE_ID = store.STORE_ID • sum_individual.CLOSEST_ATF_STORE_ID = store.STORE_ID • sum_individual.CLOSEST_LOS_STORE_ID = store.STORE_ID • sum_household.CLOSEST_AT_STORE_ID = store.STORE_ID • sum_household.CLOSEST_LOFT_STORE_ID = store.STORE_ID • sum_household.CLOSEST_ATF_STORE_ID = store.STORE_ID • sum_household.CLOSEST_LOS_STORE_ID = store.STORE_ID • sum_household.HOH_INDIV_ID = sum_individual.INDIV_ID
BU10.6	System and Epsilon will facilitate ad-hoc data import processes (e.g sweepstakes, opt-in cards, etc) – Epsilon and ANN INC. will need to come up with a standard interface for these adhoc files. Anything outside if this layout will need to go to the list conversion group and a change request will be provided to ANN INC.

The system utilizes the following interface to map current and new feeds to the contact subject area in the Data Warehouse.

Target Table Column Name	Target Table Datatype	Business Rule
BRAND_CD	VARCHAR2(4)	
ACCT_SOURCE_CD	VARCHAR2(4)	"AHOC"
ACCT_NBR	VARCHAR2(20)	ELT ALTERNATE_KEY
ALT_ACCT_NBR	VARCHAR2(20)	
ACCT_STATUS_CD	VARCHAR2(1)	
CLIENT_ACCT_STATUS_CD	VARCHAR2(1)	
MAIL_OPTOUT_IND	NUMBER(1)	
RENT_OPTOUT_IND	NUMBER(1)	
NUCLEAR_OPTOUT_IND	NUMBER(1)	
LANGUAGE_CD	VARCHAR2(3)	
SRC_GENDER_CD	VARCHAR2(1)	
SRC_PREFIX	VARCHAR2(20)	
SRC_FIRST_NM	VARCHAR2(50)	
SRC_MIDDLE_NM	VARCHAR2(50)	
SRC_LAST_NM	VARCHAR2(50)	
SRC_SUFFIX	VARCHAR2(20)	
SRC_FULL_NM	VARCHAR2(80)	
SRC_FIRM_NM	VARCHAR2(50)	
SRC_ADDR_LINE_1	VARCHAR2(65)	
SRC_ADDR_LINE_2	VARCHAR2(65)	
SRC_ADDR_LINE_3	VARCHAR2(65)	
SRC_ADDR_LINE_4	VARCHAR2(65)	
SRC_CITY	VARCHAR2(50)	
SRC_STATE	VARCHAR2(40)	
SRC_COUNTRY_CD	VARCHAR2(3)	
SRC_POSTAL_CD	VARCHAR2(10)	
ADDR_TYPE_CD	VARCHAR2(4)	
ADDR_START_DT	DATE(0)	
ADDR_END_DT	DATE(0)	
MKTG_SOURCE_CD	VARCHAR2(10)	
SRC_DECEASED_IND	NUMBER(1)	
SRC_BIRTH_DT	DATE(0)	
ASSOCIATE_IND		
SRC_CREATE_DT	DATE(0)	
MAIL_REQUEST_SRC_CD	VARCHAR2(4)	
MAIL_REQUEST_DT	DATE(0)	
CA_PRIV_IND	NUMBER(1)	
EMAIL_ADDR	VARCHAR2(100)	
EMAIL_OPTOUT_IND	NUMBER(1)	Note: Email preferences on ACCOUNT sources are retained purely for reference. Actual email preferences (by PID) are identified exclusively through the SDL SUB/UNSUB feeds
EMAIL2_ADDR	VARCHAR2(100)	
EMAIL2_OPTOUT_IND	NUMBER(1))	See Note: EMAIL_OPTOUT_IND
EMAIL3_ADDR	VARCHAR2(100)	
EMAIL3_OPTOUT_IND	NUMBER(1)	See Note: EMAIL_OPTOUT_IND
SRC_ACCOUNT_KEY	VARCHAR2(25)	
HOME_PHONE_NBR	VARCHAR2(30)	
HOME_PHONE_OPTOUT_IND	NUMBER(1)	
WORK_PHONE_NBR	VARCHAR2(30)	
WORK_PHONE_OPTOUT_IND	NUMBER(1)	
MOBILE_PHONE_NBR	VARCHAR2(30)	
MOBILE_PHONE_OPTOUT_IND	NUMBER(1)	
SMS_OPTOUT_IND	NUMBER(1)	

BU10.6.2

MOBILE_PHONE_NBR	VARCHAR2(30)	
MOBILE_PHONE_OPTOUT_IND	NUMBER(1)	
SMS_OPTOUT_IND	NUMBER(1)	

BU10.6.3

The feed is provided in TAB-delimited format with the following naming convention:

- ANN_CRM_ACCOUNT_yyyymmdd_SEQ.dat

BU10.6.4

A unique ACCT_SOURCE_CD is assigned to identify where each dataset originated. These sources are ranked at the bottom of the best postal contact hierarchy

BU10.6.5

Any additional information about how/when the customer data was obtained would be stored in the ACCOUNT_EVENT table for possible future implementation.

BU10.7

ANN INC. will have the capability to do customer look-ups to the database via SAS.

BU10.7.1

Customer lookups are done on the INDIVIDUAL and HOUSEHOLD structures in the mart via SAS/Business Objects

BU10.8

ANN INC. will need to request a preference change, name change or address change (TBD on ticketing system) so these changes can be made to the database via the update process. Specific details to be defined during the Design Phase and captured in the Solution Design Document.

BU10.8.1

The level that ANN sees in the mart for an individual is based on a series of business rules applied from multiple source records. In order to enact the changes, a set of complex logic needs to be applied to all the underlying accounts for a postal_contact – otherwise the underlying records may force a different result when the mart is recalculated.

The system will support the ability for ANN to modify the name, address, and preference information associated with a Postal Contact by submitting a request through the production/operations ticketing system with an attached tab-delimited record containing the updated values.

BU10.8.2

Original data for the change record is obtained by the requestor from the SUM_INDIVIDUAL table in the Data Mart. The format for the change request record is as follows and includes ALL of the following attributes in the table below.

Field Name	Format	Precision	
BEST_POSTAL_CONTACT_ID	NUMBER	10	
PREFIX	VARCHAR2	20	
FIRST_NM	VARCHAR2	50	
MIDDLE_NM	VARCHAR2	50	
LAST_NM	VARCHAR2	50	
MATURITY_SUFFIX	VARCHAR2	20	
ADDR_LINE_1	VARCHAR2	65	
ADDR_LINE_2	VARCHAR2	65	
ADDR_LINE_3	VARCHAR2	65	
ADDR_LINE_4	VARCHAR2	65	
CITY	VARCHAR2	50	
STATE	VARCHAR2	40	
COUNTRY_CD	VARCHAR2	3	
POSTAL_CD	VARCHAR2	10	
HOME_PHONE_OPTOUT_IND	VARCHAR2	1	
WORK_PHONE_OPTOUT_IND	VARCHAR2	1	
MOBILE_PHONE_OPTOUT_IND	VARCHAR2	1	
MAIL_OPTOUT_IND	VARCHAR2	1	
RENT_OPTOUT_IND	VARCHAR2	1	
NUCLEAR_OPTOUT_IND	VARCHAR2	1	Used for global optout (suppression) of all channels

BU10.8.3	Changes to NAME & ADDRESS are made on POSTAL_CONTACT directly. Changes are made to the AGILITY POD by sending updated name/address information through the agility process with the existing agility INDIV_ID populated. Agility would in turn assign the same INDIV_ID and store the new address. A new GAID and HHID would be assigned for the new address along with the other address elements. In Agility multiple addresses and hard keys are linked to a single individual id over time. This process would add the new address into the chain. If that transaction came back in after the fact with the old address, you will get the same INDID. However if no COA was possible, meaning no NCOA/PCOA match, then the old address would still be returned. This would result in the previous HHID and GAID from before. The POD maintains all the address history to keep IDs persistent, but does not act as a COA service. If you send in a transaction with an ID and new address. If that individual matches an already existing record at that new address on the POD and the ID for that match is lower than the ID provided on the transaction. Agility will consolidate the provided ID to the lower ID from the match on the POD.
BU10.8.4	Preference changes are set at the INDIVIDUAL level in the data mart and serve to override any underlying ACCOUNT-based preferences.

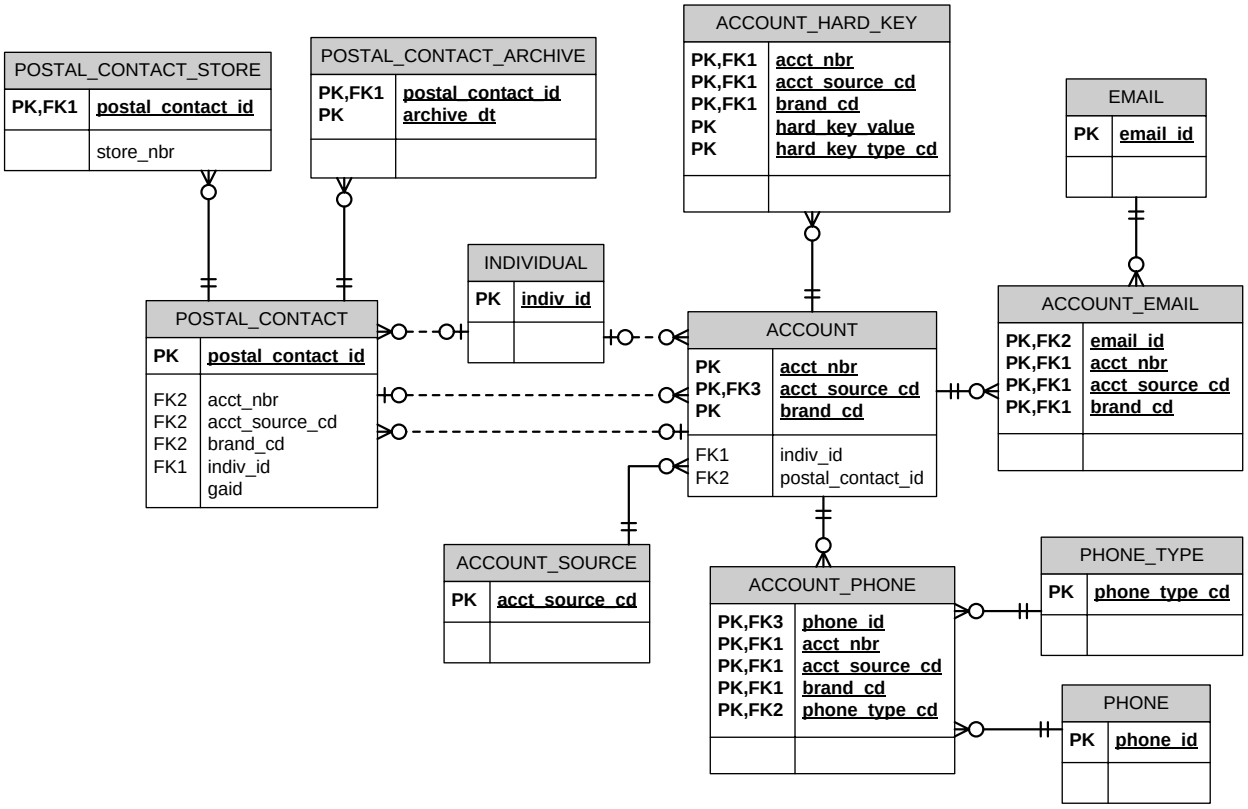
Customer Subject Area

The system maintains the customer subject area from a large data flow that includes the consolidation of contact data from all Customer sources into a single standardized dataset and processes the data through Agility for Postal Contact Creation and Individual/Household/Address relationship identification. The resulting data structures represent a fully normalized view of customer data with the flexibility to perform rollups and associations with underlying fact/dimension in the Data Warehouse.

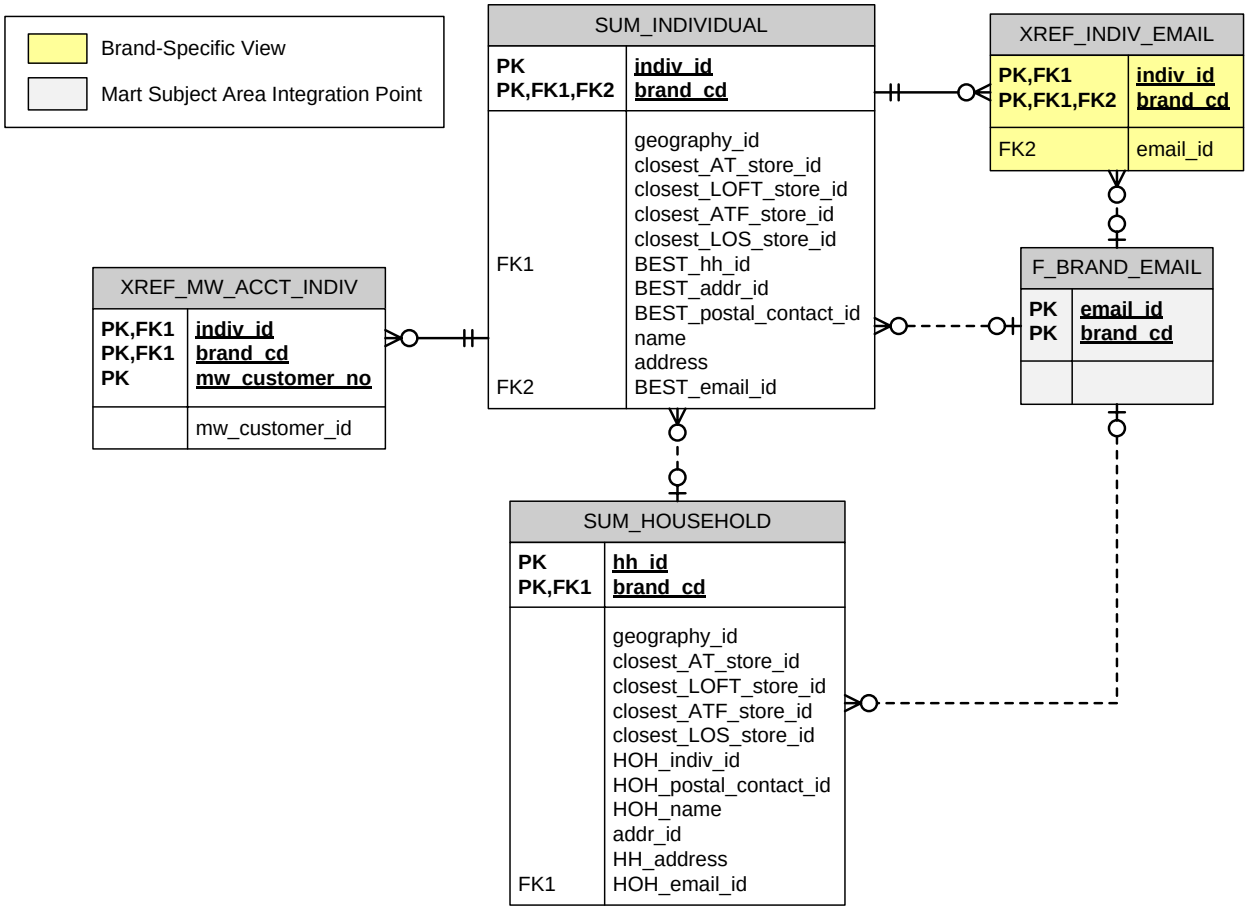
The Customer Subject area in the Data Warehouse consists of the following Data Structures:

Table Name	Description
ACCOUNT	Source-derived individual attributes and preference data
ACCOUNT_EMAIL	Shows all Email addresses linked to an Account
ACCOUNT_HARD_KEY	Agility Hard Key matching attributes (ADS Account ID, Email Address)
ACCOUNT_PHONE	Shows all Phone relationships links to an Account
ACCOUNT_SOURCE	Used for ranking Account sources
ACCOUNT_OPTOUT_HIST	Mail, Rent, Nuclear opt-out history tracking
ACCOUNT_MW_SEGMENT	Historical Marketworks preference segments
ACCT_NBR_XREF	Generated account numbers
INDIVIDUAL	A Person related to one or more physical locations
PHONE	Phone Numbers
PHONE_TYPE	Phone Type (home, work, mobile)
POSTAL_CONTACT	An individual at a physical location
POSTAL_CONTACT_ARCHIVE	Archive for postal contacts not selected for Data Warehouse load
POSTAL_CONTACT_STORE	Relates a Postal Contact to one or more "nearest" stores by brand
XREF_ACCOUNT_MW_CUSTOMER	Relates historical MW Customers to final MW Customers

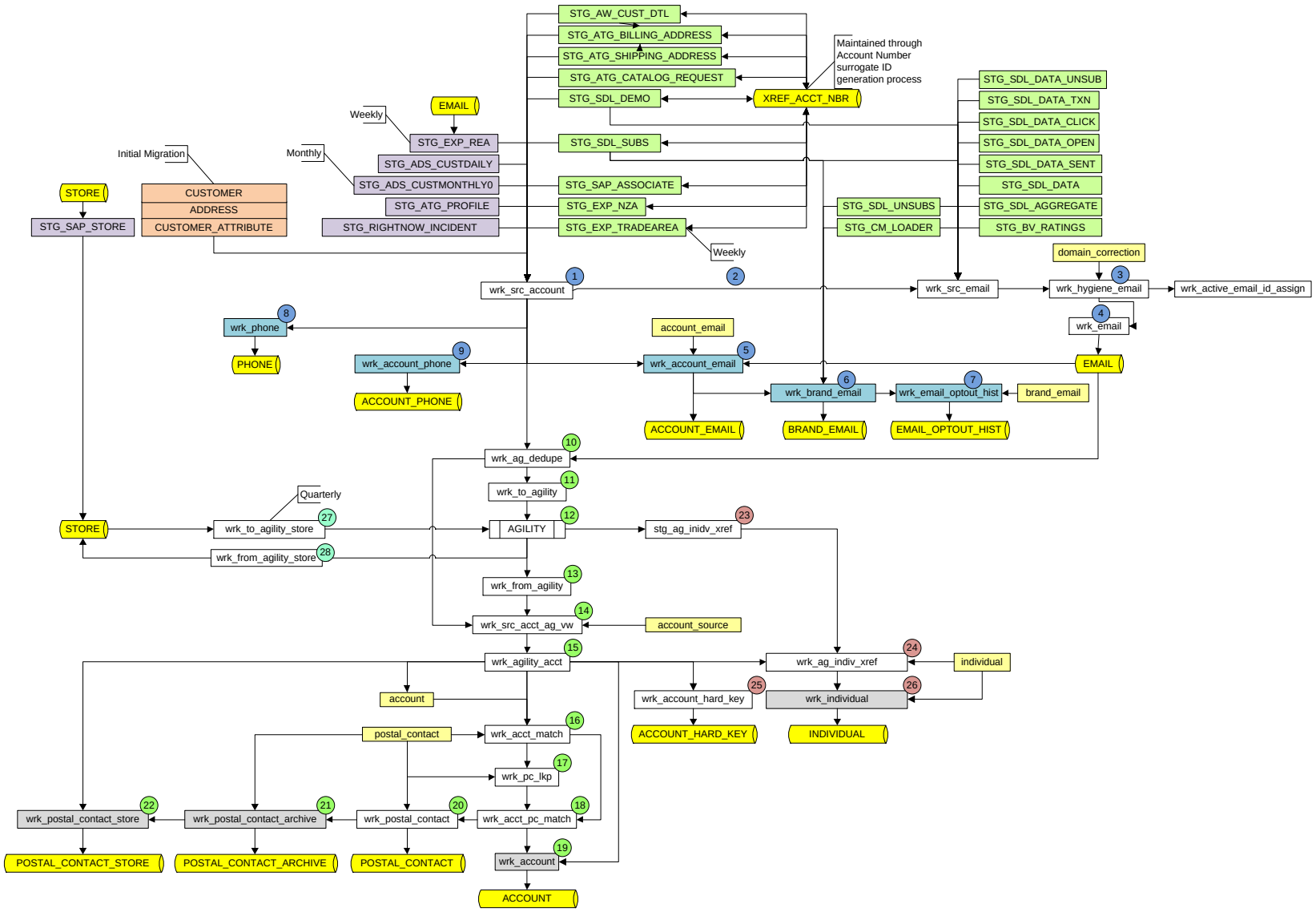
Customer Entity Key Relationships – Data Warehouse



Customer Entity Key Relationships – Data Mart



Customer Data Warehouse Update Process



Customer Data Warehouse Update Process Annotations

1. **WRK_SRC_ACCOUNT** - All ACCOUNT sources are loaded into a work table leveraging transformations specific to each source to derive standardized elements applicable across sources including: Brand code designation; name, address, email and phone identification; account preferences; unique account identifier; and additional customer attributes.
2. **WRK_SRC_EMAIL** - Email address from all sources are segmented off and consolidated along with the derivation of nuclear opt-out preferences at the email address level.
3. **WRK_HYGIENE_EMAIL** - Email Hygiene is performed to correct common typos and character sequences that invalidate an email address
4. **WRK_EMAIL** - A complete set of original and corrected email addresses included in the update are loaded into the Data Warehouse and all email deliverability attributes are updated. Newly identified email addresses are assigned a unique EMAIL_ID.
5. **WRK_ACCOUNT_EMAIL** - Build table consolidating all Email and Account related data most notably email preferences maintained at the Account level by the data sources. This preference data is for research and reference purposes only as master preference data is maintained at the BRAND EMAIL level with CheetahMail as the source.
6. **WRK_BRAND_EMAIL** - Build table consolidating all brand/list relationships from the source data and current email preference data from CheetahMail
7. **WRK_EMAIL_OPTOUT_HIST** - Maintain history of email preference changes received from ACCOUNT sources relative to what may have previously been loaded in the Data Warehouse
8. **WRK_PHONE -** Gather all of the standardized phone data from the variety of sources that provide phone numbers
9. **WRK_ACCOUNT_PHONE –** Build table relating Phone numbers to Accounts for load into the Data Warehouse
10. **WRK_AG_DEDUPE –** Extract all account information with sufficient data to be Agilitized:
    a. Last Name and Address Line 1 and ((City and State) or Postal Code)
    b. First Name and ADS Account ID
    c. Full Name and Email Address
11. **WRK_TO_AGILITY –** Select distinct Name/Address and assign ID for matchback
12. **AGILITY –** Perform AGILITY processing
    a. Address Standardization (CASS/SERP)
    b. NCOA/PCOA Processing
    c. Address ID Assignment
    d. Customized Household & Individual Assignment
    e. Customer Nearest Store Calculation
    f. Suppression Flagging
    g. Mailability Scoring
13. **WRK_FROM_AGILITY** – Load the Name/Address feed returned from AGILITY
14. **WRK_SRC_ACCT_AG_VW –** Match AGILITY output back to Source Account data
15. **WRK_AGILITY_ACCT –** De-dupe all accounts returned by the src_acct_ag_vw and place the results in a working table for further processing
16. **WRK_ACCT_MATCH –** Attempt to look up existing accounts and retrieve the postal contact information if available
17. **WRK_PC_LKP –** Populate the postal contact lookup for all accounts. It either finds an existing postal_contact_id or assigns a new one for each indiv_id and GAID combination encountered
18. **WRK_ACCT_PC_MATCH –** Look up the final postal_contact_id for each account and place the results in a working table for further processing
19. **WRK_ACCOUNT –** Build the final source ACCOUNT table for load into the Data Warehouse

20. **WRK_POSTAL_CONTACT –** Build the final POSTAL_CONTACT table for load into the Data Warehouse
21. **WRK_POSTAL_CONTACT_ARCHIVE –** Identify records whose Agility key information has changed as part of the process
22. **WRK_POSTAL_CONTACT_STORE –** Build the table used to relate a Postal Contact to the nearest store of each brand
23. **STG_AG_INDIV_XREF –** Load the AGILITY key consolidation cross-reference feed
24. **WRK_AG_INDIV_XREF –** Load an intermediate working table to combine all new indiv_ids detected during the account process and changes loaded via the latest agility link process
25. **WRK_ACCOUNT_HARD_KEY –** Build a table showing the specific hard key information (ADS account id or email address) sent to Agility for processing. An account can have multiple email addresses related to it but only one is used in the actual hard key matching process.
26. **WRK_INDIVIDUAL –** Build the table tying Accounts to Postal Contacts and identifying the intersecting Individual/Customer
27. **WRK_TO_AGILITY_STORE –** Create quarterly extract of Agility Store data
28. **WRK_FROM_AGILITY_STORE –** Update STORE attributes in data warehouse to facilitate customized distance calculations in the mart

11. Account

<table>
<tr><th>#</th><th>Design</th></tr>
<tr><td>BU11.1</td><td>The system will track customer information provided by various systems at a brand level as a consolidated and standardized object known as an ACCOUNT.</td></tr>
<tr><td>BU11.1.1</td><td>All customer contact data is included in the WRK_SRC_ACCOUNT table. This data is traceable back to the original source in the Data Warehouse through the native or derived ACCT_NBR attribute.</td></tr>
<tr><td>BU11.1.2</td><td>The system will attempt to associate each account with one or more brands through either native account attributes or in the case of Marketworks, transaction history and grouping/segmentation/permission attributes. If the system cannot identify an actual brand, the value is defaulted to Enterprise (ENT)</td></tr>
<tr><td>BU11.2</td><td>The system will track the source system that populated the data for an account via an account source code. Account source code values for ANN Inc are:
MW – MarketWorks
WEBB – Web Bill-to
WEBS – Web Ship-to
WEBP – Web Profile
WEBC – Web Catalog Requestor
RN – Call Center
AWB – AuditWorks Bill-to
AWS – AuditWorks Ship-to
REA – Reverse Email Append
NZA – Name ZIP Append
TAA – Trade Area Append
ADS – Alliance Data Systems
EMP - Employee
CM – CheetahMail</td></tr>
<tr><td>BU11.2.1</td><td>ACCOUNTs are traced back to their source systems through the following identifiers.

<table>
<tr><th>Source</th><th>ACCT_SRC_CD</th><th>ACCT_NBR</th></tr>
<tr><td>Marketworks</td><td>MW</td><td>CUSTOMER_ID</td></tr>
<tr><td>ATG Bill-to</td><td>WEBB</td><td>{xref}</td></tr>
<tr><td>ATG Ship-to</td><td>WEBS</td><td>{xref}</td></tr>
<tr><td>ATG Profile</td><td>WEBP</td><td>PROFILE_ID</td></tr>
<tr><td>ATG Catalog Requestor</td><td>WEBC</td><td>NVL(PROFILE_ID, {xref})</td></tr>
<tr><td>RightNow</td><td>RN</td><td>REF_NO</td></tr>
<tr><td>AuditWorks Bill-to</td><td>AWB</td><td>{xref}</td></tr>
<tr><td>AuditWorks Ship-to</td><td>AWS</td><td>{xref}</td></tr>
<tr><td>Reverse Email Append</td><td>REA</td><td>EMAIL_ID</td></tr>
<tr><td>Name/Zip Append</td><td>NZA</td><td>{xref}</td></tr>
<tr><td>Trade Area Append</td><td>TAA</td><td>{xref}</td></tr>
<tr><td>Alliance Data Systems</td><td>ADS</td><td>ACCT_NBR</td></tr>
<tr><td>ANN Associate</td><td>EMP</td><td>{xref}</td></tr>
<tr><td>SDL Demographic/Subscribers</td><td>CM</td><td>EMAIL_ID</td></tr>
</table>

Note: RN/CallCenter Catalog requestors are provided via ATG Catalog Requestor feed</td></tr>
<tr><td>BU11.2.2</td><td>The {xref} notation above refers to the ACCT_NBR_XREF Data Warehouse structure by which a unique numeric identifier is assigned to the unique combination of the following</td></tr>
</table>

	fields: • SRC_FIRST_NM • SRC_MIDDLE_NM • SRC_LAST_NM • SRC_ADDR_LINE_1 • SRC_ADDR_LINE_2 • SRC_ADDR_LINE_3 • SRC_ADDR_LINE_4 • SRC_CITY • SRC_STATE • SRC_COUNTRY_CD • SRC_POSTAL_CD • EMAIL_ADDR This is used for sources that do not have a native account number (or something similar). This data persists in the data warehouse so any time the system encounters the exact combination of these fields the same account number attribute value will be assigned.
BU11.3	The system will load MarketWorks historical customer table; (Segmentation Value D - which is the ADS Account ID) to an attribute of a MW account source. This is for historical tracking and/ or analysis purposes.
BU11.3.1	The MarketWorks ADS Account ID (segmentation value D) has been determined to not be entirely correct. The decision was made to unmap it from the ACCOUNT.ALT_ACCT_NBR field.
BU11.4	The system will load ADS customer migration and on-going feeds and will need to distinguish between PLCC and Cobrand and flag the brand as LOFT or AT PLCC cardholder.
BU11.4.1	The ADS PLCC vs Cobrand distinction is done via the DIV_CD • 176 = AT Cobrand • 177 = LOFT Cobrand • 88 = AT & LOFT PLCC
BU11.4.2	The ADS Brand Code is derived as follows: if DIV_NBR = 177 or (DIV_NBR = 088 and CC_LOGO = 2) then LOFT if DIV_NBR = 176 or (DIV_NBR = 088 and CC_LOGO = 1) then AT else ENT
BU11.5	Account sources will be ranked in the following order for use in determining which account will provide the information for the master Postal Contact record: *1 = Best* 1. SAP Associate Feed 2. ADS 3. ATG Bill-to, ATG Catalog Request 4.CheetahMail 5. AuditWorks Bill-to 6. RightNow 7.MarketWorks 8. AuditWorks Ship-to, ATG Ship-to 9. KBM or Experian Name/Zip Append 10. KBM or Experian Trade Area Append, Reverse Email Append
BU11.5.1	Account Source Ranking is stored in the ACCOUNT_SOURCE table in the data warehouse. It is used to assist in the identification of the BEST Postal Contact

	<table><tr><th>ACCT Source</th><th>Rank</th><th>Description</th></tr><tr><td>EMP</td><td>5</td><td>ANN INC (SAP) ASSOCIATES</td></tr><tr><td>ADS</td><td>10</td><td>ALLIANCE DATA SYSTEMS</td></tr><tr><td>WEBB</td><td>20</td><td>ECOMM (ATG) BILL-TO</td></tr><tr><td>WEBC</td><td>30</td><td>ECOMM (ATG) CATALOGUE REQUESTOR</td></tr><tr><td>CM</td><td>40</td><td>CHEETAHMAIL (SDL) DEMO/SUBS</td></tr><tr><td>AWB</td><td>50</td><td>POS (AUDITWORKS) BILL-TO</td></tr><tr><td>RN</td><td>60</td><td>CALL CENTER (RIGHT NOW)</td></tr><tr><td>MW</td><td>100</td><td>MARKETWORKS</td></tr><tr><td>WEBS</td><td>110</td><td>ECOMM (ATG) SHIP-TO</td></tr><tr><td>AWS</td><td>120</td><td>POS (AUDITWORKS) SHIP-TO</td></tr><tr><td>NZA</td><td>130</td><td>NAME/ZIP APPEND (KBM)</td></tr><tr><td>REA</td><td>135</td><td>REVERSE EMAIL APPEND (KBM)</td></tr><tr><td>TAA</td><td>140</td><td>TRADE AREA APPEND (KBM)</td></tr></table>
BU11.6	The system will maintain the following data if available in the source system at the account level: • Source system name and address • Agility standardized name components and gender • Direct mail, phone, email, 3rd party share, and nuclear opt-out permissions • Preferred language • Address type, start date and end date • Marketing source code • Source account number or system generated number if not available, this includes (ADS account ID, Customer ID, ATG profile ID, CM issue_id, ect…)
BU11.6.1	The Data Warehouse stores the original source system name and address in the Data Warehouse ACCOUNT table
BU11.6.2	The system stores the AGILITY standardized name/address components, gender, and related attributes in the Data Warehouse POSTAL_CONTACT table. In the Data Mart, the best Postal Contact is selected and included on both the Individual and Household levels.
BU11.6.3	The system stores Direct Mail, Nuclear, and Rental permissions provided by the source systems on the ACCOUNT table
BU11.6.4	When provided by the Account source, the system stores email permissions on the ACCOUNT_EMAIL table. The system store Email permissions provided by SDL/CheetahMail on the BRAND_EMAIL table where list ID = PID
BU11.6.5	The system stores Phone permissions for each phone type on the ACCOUNT_PHONE table.
BU11.6.6	In the Data Mart, the system determines the most restrictive Direct Mail, Nuclear, Rental, and Phone permissions from all accounts related to an individual id and uses those values on the Individual and household level summaries.
BU11.6.7	The system stores a provided language preference setting on the Data Warehouse ACCOUNT table as LANGUAGE_CD

BU11.6.8	The language preference in the Data Mart will reflect the value associated with the ACCOUNT used to populate the POSTAL_CONTACT selected as the best POSTAL_CONTACT for the individual.
BU11.6.9	The address type indicator is provided through AGILITY DSF2 processing. It is stored in the following fields on POSTAL_CONTACT • DSF2_RES_BUS_CD – Business/Residential Indicator • DSF2_DELIVERY_TYPE_CD – Delivery Type
BU11.6.10	The address start and end dates are stored in the Data Warehouse on the ACCOUNT table when provided by the source. The Change of Address (COA) date is stored on POSTAL_CONTACT.
BY11.6.11	The marketing source code is stored on the ACCOUNT table in the mktg_source_cd field. It is used to identify where the data provided had obtained the customer information. In the Data Mart, the system selects the mktg_source_cd associated with the ACCOUNT used to populate the POSTAL_CONTACT selected as the best POSTAL_CONTACT for the individual for the value stored at the individual level.
BU11.7	The system will provide a mechanism to link source transaction and event data to client information via a unique account key consisting of the brand, the account source, and an account number (either from native data or generated by the system).
BU11.7.1	The ACCT_SRC_CD, ACCT_NBR and BRAND_CD designations described in **BU11.2.1** are maintained as foreign keys in the Data Warehouse to the following non-account tables: • TXN_HEADER • TXN_ITEM • REVIEW • CSC_INCIDENT • SITE_VISIT • ASSOCIATE
BU11.8	The system will create accounts based on a one-time migration of data from the MarketWorks Customer and Address tables utilizing the following business rules: • The account source code for MarketWorks data will be 'MW'. • The migration process will utilize the active address information for the Client as the account address. Non-active addresses will not be loaded into the system. • The MarketWorks customer is independent of brand. The migration process will identify each brand for which a MarketWorks customer has a transaction and create a separate account at each brand level for the Client. • The migration process will utilize the customer_id field as the account number.
BU11.8.1	The system will create accounts based on a one-time migration of data from the MarketWorks Customer and Address tables utilizing the following business rules: • The MarketWorks customer is independent of brand. The migration process identifies each brand for which a MarketWorks customer has a transaction and creates a separate account at each brand level for the Client. • The migration process will utilize the customer_id field as the account number.
BU11.8.1	When loading MarketWorks data into WRK_SRC_ACCOUNT the ACCT_SRC_CD is hardcoded with "MW"
BU11.8.2	The ANN_MIGRN_USER.CUSTOMER table is joined to ANN_MIGRN_USER.ADDRESS table on ACTIVE_ADDRESS_ID to select only active addresses associated with a MW customer
BU11.8.3	The MarketWorks customer number is loaded through its association to one or more brands as part of the WRK_SRC_ACCOUNT table load process. The BRAND_CD is derived through the logic defined the **Stage to Work Account mapping** document

BU11.8.4	The system populates the WRK_SRC_ACCOUNT.ACCT_NBR with ANN_MIGRN_USER.CUSTOMER. CUSTOMER_ID
BU11.9	ATG profile accounts - There are multiple addresses that can be stored on the profile. There is no indication of a preferred address. Epsilon will move the address with the most elements provided as the address associated to this account. This address is independent of the bill-to and ship-to addresses found on the transactions. This address will be able to relate back to Bazaarvoice through the profile_id.
BU11.9.1	The ATG Profile feed contains one or more Billing Address nodes. The process cycles through these for each STG_ATG_PROFBILL_ADDRESS in sequential order until one is found for selection that meets the following requirements: • Address1 and ((City and State) or Postal Code) Note the PROFILE hierarchy is defined as follows: 1. STG_ATG_PROFILE a. STG_ATG_PROFBILL_ADDRESS b. STG_ATG_LOCAL_STORE An account has a single active address at any point in time. The ATG profile accounts are being added just for BV linkages (the reviews only have a profile_id). ATG does not provide an address for a profile_id. Rather it has a series of credit cards each with a billing address. The process needs to pick one as the address for the account.
BU11.9.2	If no PROFILE ADDRESS is found with the criteria above the Account is created from: • First Name, Last Name and Email Address Note: A profile address without a physical address occurs when the person signs up with an email address and gives their name but has not purchased anything and thus ATG has no credit cards defined with a billing address.
BU11.9.3	Similarly, the Transaction BILL TO account may include multiple Billing Address nodes. The process cycles through these for each STG_ATG_BILLING_ADDRESS in sequential order until one is found for selection that meets the following requirements: • Address1 and ((City and State) or Postal Code) Note the TRANSACTION hierarchy is defined as follows: 1. STG_ATG_ORDER a. STG_ATG_RELATED_ORDERS b. STG_ATG_PRICE_INFO i. STG_ATG_ORDER_PRICE_INFO c. STG_ATG_COMMERCE_ITEM i. STG_ATG_PRICE_INFO 1. STG_ATG_ITEM_PRICE_INFO 2. STG_ATG_DTL_ITM_PRICE_INFO d. STG_ATG_PAYMENT_GROUP i. STG_ATG_CREDIT_CARD 1. STG_ATG_BILLING_ADDRESS e. STG_ATG_SHIPPING_GROUP i. STG_ATG_HARDGOOD_SHIP_GRP 1. STG_ATG_SHIPPING_ADDRESS
BU11.9.4	Similarly, the Transaction SHIP TO account may include multiple Billing Address nodes. The process cycles through these for each STG_ATG_SHIPPING_ADDRESS in sequential order until one is found for selection that meets the following requirements:

	• Address1 and ((City and State) or Postal Code) Note the TRANSACTION hierarchy is defined as follows: 2. STG_ATG_ORDER a. STG_ATG_RELATED_ORDERS b. STG_ATG_PRICE_INFO i. STG_ATG_ORDER_PRICE_INFO c. STG_ATG_COMMERCE_ITEM i. STG_ATG_PRICE_INFO 1. STG_ATG_ITEM_PRICE_INFO 2. STG_ATG_DTL_ITM_PRICE_INFO d. STG_ATG_PAYMENT_GROUP i. STG_ATG_CREDIT_CARD 1. STG_ATG_BILLING_ADDRESS e. STG_ATG_SHIPPING_GROUP i. STG_ATG_HARDGOOD_SHIP_GRP 1. STG_ATG_SHIPPING_ADDRESS
BU11.9.5	Conversely, the Catalog Requestor account has a single address generated for all requests Note the CATALOG REQUEST hierarchy is defined as follows: 1. STG_ATG_CATALOG_REQUEST a. STG_ATG_CATALOG_PREF
BU11.10	If the contact information provided for an account contains sufficient name and address attributes, the system will track the relationship of an account to a single deduplicated Postal Contact record containing hygiene and standardized information for an individual at a particular mailing address (and by implication the household id and global address id assigned to that postal contact). Accounts without sufficient information to assign a postal contact will have a postal contact id of 0.
BU11.10.1	Once all ACCOUNT data is processed through AGILITY, the system creates Postal Contacts that related back to one or more source Accounts. In the event that an address could not be coded by AGILITY, no Postal Contact is created and the ACCOUNT.POSTAL_CONTACT_ID is set to zero.
BU11.11	If sufficient contact information is available the system will track the individual id assigned to an account via the CDI process. Accounts without requisite contact information will have an individual id of 0.
BU11.11.1	Similarly, the system creates an INDIVIDUAL when there is a codable name/postal address or name/hard key identified through AGILITY. Where sufficient information is not available from the source, the system will not create an INDIVIDUAL and the indiv_id is set to zero.
BU11.12	The system will track the date that information was last provided for the account (activity date).
BU11.12.1	Each feed includes an ACTIVITY_DT. When the PK of an incoming record already exists in the Data Warehouse, the ACTIVITY_DT of the incremental feed is compared to that stored in the Data Warehouse. If the inbound date is greater than or equal to the DW date, the system updates the record.
BU11.12.2	In the event that the incremental data ACTIVITY_DT is older than the stores in the Data Warehouse the inbound record is considered out of date and ignored as part of the update process. The system provides counts of ignored records as part of the load statistics.

BU11.12.3 When a date representing the recency of the source data is not available, the system utilizes the current load date as the activity date.

The Activity Dates used for each feed are defined as follows:

Source	Feed	Activity Date
MW	Address	sysdate
MW	Class	sysdate
MW	Color	sysdate
MW	Customer	sysdate
MW	Customer_Alternate_Key	sysdate
MW	Customer_Xref	sysdate
MW	Department	sysdate
MW	Division	sysdate
MW	Product_Xref	sysdate
MW	Style	sysdate
MW	Style_Xref	sysdate
MW	Tender	sysdate
MW	Tender_Mapping	sysdate
MW	TI_Transaction_Detail_Error	sysdate
MW	TI_Transaction_Header_Error	sysdate
MW	TI_Transaction_Tender_Error	sysdate
MW	Transaction_Coupon	sysdate
MW	Transaction_Detail	sysdate
MW	Transaction_Header	sysdate
MW	Attribute_Grouping	sysdate
MW	Attribute_List	sysdate
MW	Customer_Attribute	sysdate
MW	Email_POS_Arch	sysdate
MW	Transaction_Detail_XREF	sysdate
ADS	Customer Daily PLCC	sysdate
ADS	Customer Daily CO Brand	sysdate
ADS	Customer Xref Daily PLCC	sysdate
ADS	Customer Xref Daily COBRAND	sysdate
ADS	Customer Monthly PLCC (type 0)	sysdate
ADS	Customer Monthly PLCC (type 7)	sysdate
ADS	Customer Monthly PLCC (type 9)	sysdate
ADS	Customer Monthly COBRAND (type 0)	sysdate
ADS	Customer Monthly COBRAND (type 7)	sysdate
ADS	Customer Monthly COBRAND (type 9)	sysdate
RIGHTNOW	Call Center Incident/Customer	CREATED
BV	Product Review ANN Taylor	CRDATE
BV	Product Review LOFT	CRDATE
SDL	IID Keys	sysdate
SDL	Bulk Mail Send Events	TIMESTAMP
SDL	Event Based Mail Send Events	TIMESTAMP
SDL	Open Events	TIMESTAMP
SDL	Click Events	TIMESTAMP
SDL	Unsubscribe Events	TIMESTAMP
SDL	Transaction Events	TIMESTAMP
SDL	Subscriptions	DATETIME_SUB
SDL	Un-Subscriptions	DATETIME_UNSUB
SDL	Change of Demographics	DATE_CHANGED
CM	Loader Exception	sysdate
SDL	Alterian Aggregate	sysdate

BU11.12.4

SAP	Location Hierarchy (Store)	DATE_TIME	
SAP	Associate Reference	DATE_TIME	
SAP	Vendor	DATE_TIME	
SAP	Style Attributes	DATE_TIME	
SAP	Product Hierarchy	DATE_TIME	
SAP	Promotion	DATE_TIME	
OMNI	Visits ANN Taylor	VISIT_DATE	
OMNI	Visits ANN Loft	VISIT_DATE	
OMNI	Visit by Product Category ANN Taylor	VISIT_DATE	
OMNI	Visit by Product Category Loft	VISIT_DATE	
OMNI	SAINT File ANN Talyor	sysdate	
OMNI	SAINT File Loft	sysdate	
ATG	ECOM Customer Profile	sysdate	
ATG	ECOM Transaction - Orders	LASTMODIFIEDDATE	
ATG	ECOM Transaction - Returns	CREATED_DATE	
ATG	ECOM Catalog Requests	REQUESTED_DATE	
AW	POS Trasaction Header (Type H)	TRANSACTION_DATE	
AW	POS Trasaction Line (Type L)	TRANSACTION_DATE	
AW	POS Merchandise Detail (Type M)	TRANSACTION_DATE	
AW	POS Discount Detail (Type D)	TRANSACTION_DATE	
AW	POS Return Detail (Type R)	TRANSACTION_DATE	
AW	POS Authorization Detail (Type A)	TRANSACTION_DATE	
AW	POS Customer Detail (Type C)	TRANSACTION_DATE	
AW	POS Expanded Customer Detail (Type E)	TRANSACTION_DATE	
AW	POS Line Notes (Type N)	TRANSACTION_DATE	
KBM	Demographic Append (migration)	sysdate	
KBM	Demographic Append (ongoing)	sysdate	
EXP	Reverse Email Append	sysdate	
EXP	Name/Zip Address Append	sysdate	
EXP	Trade Area Append	sysdate	
EXP	Customer Segment Append (migration)	sysdate	
EXP	Customer Segment Append (ongoing)	sysdate	
ANN	Fiscal Calendar	FISCAL_DATE	
CRM	Promotion History	PROMO_DATE	
CRM	Taxonomy	PROMO_DATE	

BU11.13	The system will track which hard keys (email address, ADS account id) were provided to the Agility CDI process for the account
BU11.13.1	AGILITY Hard Keys are retained in the Data Warehouse in the ACCOUNT_HARD_KEY table.
BU11.13.2	The Hard Key value is stored in HARD_KEY_VALUE. The type of hard key is stored in HARD_KEY_TYPE as • A = ADS Account ID • E = Email Address

12. Account Email

#	Design
BU12.1	The system will track the relationship of one or more email addresses over time for an account. This is referred to as the account email level.
BU12.1.1	Email addresses from all sources are stored in the EMAIL table. The Email to Account (and by association, Postal Contact/Individual) relationship is defined through the ACCOUNT_EMAIL table in the Data Warehouse.
BU12.1.2	When a new Account is loaded that includes one or more email addresses, the system stores a link to the associated hygiened email addresses in the ACCOUNT_EMAIL table.
BU12.1.3	When an existing Account is loaded that includes a new email address, the system creates the relationship to the new email address and the maintains the relationship to any previous email address as well. The solution provides for the ability to identify the most current email address associated with an account through the ELT tracking fields.
BU12.1.4	When the system load a new Account that references an Email address that had previously been loaded by another Account, the existing email ID is then linked to the new Account while the relationship to other accounts is retained.
BU12.1.5	In the case that an email address was changed due to Email Hygiene, the Account always references the update/corrected Email ID
BU12.2	The system will provide the facility to track current email permissions as provided by the source systems at the account email level.
BU12.2.1	The ACCOUNT_EMAIL table maintains the following two attributes based on variable transformation rules mapped into the WRK_SRC_ACCOUNT.EMAIL_OPTOUT_IND field as described in the **Account Stage to Work Source to Target** document • ACCT_EMAIL_OPTOUT_IND • ACCT_EMAIL_OPTOUT_CHG_DT This stores the most recent email_optout_ind for an email associated with an account (for referential purposes only) and the date that it last changed
BU12.2.2	The historical audit table used to track anytime an Account email preference changes is stored in ACCOUNT_EMAIL_HIST. A record is only inserted into ACCOUNT_EMAIL_HIST when a change is made to the preference. Initial preferences are not stored in this table until that preference has been changed.
BU12.3	The system will track the date an email address was last encountered for a particular account (activity date).
BU12.3.1	The ACCOUNT activity date from which a email address was derived is retained on the EMAIL table
BU12.4	The system will retain the file source for each email address. The sources that contain email address are: • CheetahMail • ADS • MW • RightNow • BazaarVoice • ATG • AW
BU12.4.1	The ACCOUNT EMAIL table includes the ACCT_SRC_CD indicating all of the source(s) an email address came from: • SDL • ADS • MW • RightNow

	• BazaarVoice • ATG • AW
BU12.4.2	Email Addresses that do not have an ACCOUNT_EMAIL relationship are email address only records that were not eligible for the creation of account. These will have a relationship to BRAND_EMAIL and, as such, are identified as CheetahMail/SDL sourced. This means that email addresses with a PID are stored in brand_email from CM where true permissions are stored. The Account Email table is there to just to store the relationship of email addresses to other account information.

13. Account Phone

#	Design
BU13.1	The system will track the relationship of one or more phone numbers to an account over time by phone type. This is referred to as the account phone level.
BU13.1.1	The system stores the unique occurrence of a given phone number from any source in the PHONE table. The Phone to Account (and by association, Postal Contact/Individual) relationship is defined through the ACCOUNT_PHONE table in the Data Warehouse.
BU13.1.2	When a new Account is loaded that includes a phone number the associated phone number is also loaded and linked to the Account.
BU13.1.3	When an existing Account is loaded that includes a phone number the relationship to the new phone number is provided and the old phone number is maintained. The solution provides for the ability to identify the most current phone number associated with an account through the ELT tracking fields.
BU13.1.4	When a new Account is loaded that references a phone number that had already been loaded by another Account, the existing phone ID is then linked to the new Account while the existing link is retained.
BU13.2	The system will retain phone source and types. The sources that contain phone numbers are: • ADS – Home, Work, Mobile • MW – Home, Work • RightNow – Home, Work, Mobile • CheetahMail - Home • ATG – Home, Mobile • AW – Home, Work
BU13.2.1	The ACCOUNT PHONE table includes the ACCT_SRC_CD indicating all of the source(s) an email address came from. ADS – Home, Work, Mobile • MW – Home, Work • RightNow – Home, Work, Mobile • CheetahMail - Home • ATG – Home, Mobile • AW – Home, Work
BU13.2.2	The ACCOUNT_PHONE . PHONE_TYPE_CD is used to store the type. Phone types are standardized as: • H = Home • W = Work • M = Mobile
BU12.2.1	The ACCOUNT_PHONE table maintains the following attribute • ACCT_PHONE_OPTOUT_IND

	This is based on variable transformation rules mapped into the WRK_SRC_ACCOUNT fields: • HOME_PHONE_OPTOUT_IND • WORK_PHONE_OPTOUT_IND • MOBILE_PHONE_OPTOUT_IND The specifics of these transformations are described in the **Account Stage to Work Source to Target** document
BU12.2.2	Historical preference changes for Phone are not tracked in the database
BU12.2.3	The Data Mart resolved phone preferences by type across multiple accounts such that if any account opted out to a particular phone (by brand), the individual is opted out for that phone type (by brand)
BU13.3	The system will provide the facility to track current call permissions as provided by the source systems at the account phone level
BU13.3.1	The system maintains the most current preference for a phone number in ACCOUNT PHONE. The system will keep the most recently received explicit preference. An undefined preference will never override an explicit value.

14. Individual

#	Design
BU14.1	The INDIVIDUAL table contains a record for every uniquely identified person on the database.
BU14.1.1	The system creates a record for each individual_id returned by the Agility process.
BU14.2	The primary key of this table is individual ID. This is the individual persistent ID assigned by Agility.
BU14.2.1	The indiv_id is the AG_INDIV_ID on the wrk_from_agility return feed. It exists as a foreign key to POSTAL_CONTACT and ACCOUNT
BU14.2.2	In the Data Mart, the Individual Summary joins indiv_id to postal_contact and gathers all underlying account data to establish the relationships to fact table on which rollups are performed.
BU14.3	An Individual can be indentified through Agility by using groups of fields like a Name and Address, a First Name and ADS Account ID or a Name and Email Address.
BU14.3.1	The system scans the consolidated contact data stored in the wrk_src_account table for records containing the minimum data required for Agility to perform key assignment. This minimum data includes last name in combination with street address and ither postal code or city/state; first name and ADS account id; or first name, last name, and email address. The system will only pass to Agility records containing this information.
BU14.4	If and when individuals are consolidated, the old individual record will be set to inactive and the current individual record will reflect the active individual ID.
BU14.4.1	As hard key matches in combination with postal address information and new change of address information is made available, the Agility process may determine that what was previously two separate individuals are in reality the same person. In those cases, Agility will retain one of the two individual keys going forward to identify that individual. Agility provides the CRM system a mapping file that indicates which previously utilized individual ids should now point to a new identifier. The system takes this information and updates the INDIVIDUAL table setting old records inactive and pointing them to the new individual id. In addition, the system updates the ACCOUNT and POSTAL CONTACT tables to contain utilize the new consolidated id. In the case of POSTAL CONTACT, this process

	generates a POSTAL_CONTACT_ARCHIVE record containing a history of the original keys. Finally, if the change in individual id results in two Postal Contacts having the same combination of individual id and global address id combination, the system consolidates those two records and update the active postal contact id on the ACCOUNT table accordingly.

15. Postal Contact

#	Design
BU15.1	The system will maintain a single master record for an Individual at particular postal address referred to as a Postal Contact.
BU15.1.1	For each combination of individual id and global address id returned by Agility, the system creates and maintains a unique POSTAL CONTACT record.
BU15.2	The system will maintain the cleansed and standardized name and address components for a Postal Contact. The address on a Postal Contact represents the “best” address after application of hygiene, standardization, address correction, and change of address processing during the update.
BU15.2.1	The system stores the standardized name and best address information returned from AGILITY in the POSTAL CONTACT table. There is one active POSTAL CONTACT record per individual Id and global address id combination.
BU13.2.2	The standardized address returned from AGILITY is selected amongst several addresses that are identified with a source address throughout the AGILITY process. The address that is ultimate selected is based on the following hierarchy: **1.** NCOA/CNCOA Address **2.** PCOA Address **3.** LACSLink Address **4.** CASS/SERP Standardized Address **5.** Original Address The address returned from this process is considered to be the “best” address.
BU15.3	The system will track the relationship of a Postal Contact to the multiple accounts providing the source name and address information that resolved to the data on the Postal Contact.
BU15.3.1	The ACCOUNT table maintains all source account name and address data elements. The relationship to POSTAL_CONTACT is many-to-one such that multiple account sources can point to the same individual at a specific address.
BU15.4	The system will populate the Postal Contact with the best name and address information from one of its related accounts (after Agility processing has been applied) based on the following ranking criteria: • Deliverability of best address • Ranking of the account source • Recency of account activity • Account number The system will track which account was the initial source of information used to populate the Postal Contact.
BU15.4.1	The first level of the Postal Contact selection hierarchy is determined by the deliverability of the AGILITY MAIL_SCORE sorted in ascending order: Mailability Score **1** DPV match (non-default) **2** DPV match (default), Secondary number incorrect

	<table><tr><td>3</td><td>DPV match (default), Secondary number missing</td></tr><tr><td colspan="2">Likely Deliverable (1-3)</td></tr><tr><td>4</td><td>DPV no match, street and primary range matched</td></tr><tr><td>5</td><td>DPV no match, address info incorrect/missing</td></tr><tr><td>6</td><td>Multiple match, missing qualifier (directional/suffix)</td></tr><tr><td colspan="2">Potentially Undeliverable (4-6)</td></tr><tr><td>7</td><td>Primary number does not exist</td></tr><tr><td>8</td><td>Primary number missing</td></tr><tr><td>9</td><td>Street name not found in city or ZIP</td></tr><tr><td>X</td><td>Undeliverable ZIP+4</td></tr><tr><td colspan="2">Likely Undeliverable (7-9 and X)</td></tr></table>
BU15.4.2	The second level of the Postal Contact selection hierarchy is determined by the Account Source ranking sorted in ascending order as defined in **BU11.5.1**
BU15.4.3	The third level of the Postal Contact selection hierarchy is determined by the recency of account activity sorted in descending order as defined through the ACTIVITY_DT mapping described in **BU11.12.4**
BU15.4.4	The fourth and final level of the Postal Contact selection hierarchy is determined by the ACCT_NBR value sorted in descending order. This would use the most recently loaded account number a tie-breaker.
	Where multiple ACCOUNT records were processed through AGILITY that resolved to the same Postal Contact ID, a single Postal Contact survives
BU15.5	The system will maintain the following hygiene attributes and indicators at the Postal Contact level as part of the Agility process: • Change of address status, move date, applied type, move type, return codes, and indicators for NCOA vs PCOA • CASS deliverability indicator • DPV (Delivery Point Validation) indicators including footnotes • PAC (Proprietary Address Correction) indicators • LACS indicators • LOT numbering and sequence • Implements indicators, including deceased individuals, prisons and residential/ business addresses, foreign, military address or US possession and DMA Pander • Vacant and seasonal address indicators • Dwelling/ Unit type indicators • Latitude and longitude of the address for US addresses (“Rooftop” Precision) • Census tract • Census blockgroup • Geocode - (indicating the precision of the latitude and longitude assignment) o 0 = Matched in address level o 1 = 9-digit match in Centroid o 4 = 7-digit match in Centroid o 5 = 5-digit match in Centroid o 7 = No match in Centroid o 8 = Not matched in Address level o 9 = Both options tried, but no match in either o blank = Not tried • An indicator if name components contain vulgar words • Zip4 Coding - includes address type code (rural route, street, p.o. box, high rise, general delivery, firm, unique, military) • Carrier route coding

- Mail score indicating deliverability of the address
- Address hygiene return codes
- DSF2 coding
- FIPS state and county
- Use store table to append closest store and distance by brand for US addresses
- SERP (Software Evaluation and Recognition Program) certification for Canadian addresses will indicate valid or invalid Canadian addresses
- Nursing home indicator for Canadian addresses
- Do not call indicator for Canadian addresses
- Do not fax indicator for Canadian addresses
- Canadian language preference flag
- Mail score indicating deliverability of the address

BU15.5.1

The data feed returned from the AGILITY process contains the required indicators as follows:

Field	DataType	Precision	Scale
EXTRACT_JOB_ID	VARCHAR2	20	0
EXTRACT_RECORD_ID	VARCHAR2	30	0
STD_AGILITY_INDID	NUMBER	13	0
STD_AGILITY_HHID	NUMBER	13	0
STD_AGILITY_CAID	NUMBER	13	0
STD_AGILITY_GAID	VARCHAR2	32	0
BEST_AGILITY_INDID	NUMBER	13	0
BEST_AGILITY_HHID	NUMBER	13	0
BEST_AGILITY_CAID	NUMBER	13	0
BEST_AGILITY_GAID	VARCHAR2	32	0
VULGAR_IND	VARCHAR2	1	0
STD_PREFIX	VARCHAR2	20	0
STD_FIRST_NM	VARCHAR2	50	0
STD_MIDDLE_NM	VARCHAR2	50	0
STD_LAST_NM	VARCHAR2	50	0
STD_MATURITY_SUFFIX	VARCHAR2	20	0
STD_PROFESSIONAL_SUFFIX	VARCHAR2	20	0
STD_FIRM_NM	VARCHAR2	50	0
STD_GENDER_CD	VARCHAR2	1	0
STD_ADDR_LINE_1	VARCHAR2	65	0
STD_ADDR_LINE_2	VARCHAR2	65	0
STD_ADDR_LINE_3	VARCHAR2	65	0
STD_ADDR_LINE_4	VARCHAR2	65	0
STD_CITY	VARCHAR2	50	0
STD_STATE	VARCHAR2	40	0
STD_COUNTRY_CD	VARCHAR2	3	0
STD_POSTAL_CD	VARCHAR2	10	0
STD_ZIP4	VARCHAR2	4	0
STD_URB_NAME	VARCHAR2	65	0
STD_CARRIER_ROUTE_CD	VARCHAR2	4	0
STD_DELIVERY_PT	VARCHAR2	2	0
STD_DELIVERY_CKDG	VARCHAR2	1	0
STD_LOT_NBR	VARCHAR2	4	0
STD_LOT_SEQ	VARCHAR2	1	0
STD_ACE_LATITUDE	VARCHAR2	10	0
STD_ACE_LONGITUDE	VARCHAR2	11	0

		STD_ACE_REC_TYPE	VARCHAR2	2	0	
		STD_ACE_ERROR_CD	VARCHAR2	30	0	
		STD_ACE_ERROR_STATUS	VARCHAR2	30	0	
		STD_ACE_DPV_STATUS	VARCHAR2	1	0	
		STD_DPV_FTNOTE	VARCHAR2	12	0	
		STD_ACE_FOREIGN_CD	VARCHAR2	1	0	
		STD_BUSINESS_IND	VARCHAR2	1	0	
		STD_FOREIGN_IND	VARCHAR2	1	0	
		STD_DSF2_MATCH_LVL	VARCHAR2	1	0	
		STD_DSF2_VACANT	VARCHAR2	1	0	
		STD_DSF2_SEASON	VARCHAR2	1	0	
		STD_DSF2_RESBUS	VARCHAR2	1	0	
		STD_DSF2_DELTYPE	VARCHAR2	1	0	
		STD_DSF2_DELPTDRP	VARCHAR2	1	0	
		STD_MAIL_SCORE	VARCHAR2	1	0	
		STD_CASS_DELIVERABLE_IND	VARCHAR2	1	0	
		STD_PRISON_IND	VARCHAR2	1	0	
		STD_PANDER_IND	VARCHAR2	1	0	
		STD_DECEASED_IND	VARCHAR2	1	0	
		STD_APO_FPO_IND	VARCHAR2	1	0	
		STD_US_POSS_IND	VARCHAR2	1	0	
		STD_DPV_CMRA_IND	VARCHAR2	1	0	
		STD_DPV_FP_IND	VARCHAR2	1	0	
		STD_DPV_THROWBACK_IND	VARCHAR2	1	0	
		BEST_ADDR_LINE_1	VARCHAR2	65	0	
		BEST_ADDR_LINE_2	VARCHAR2	65	0	
		BEST_ADDR_LINE_3	VARCHAR2	65	0	
		BEST_ADDR_LINE_4	VARCHAR2	65	0	
		BEST_CITY	VARCHAR2	50	0	
		BEST_STATE	VARCHAR2	40	0	
		BEST_COUNTRY_CD	VARCHAR2	3	0	
		BEST_POSTAL_CD	VARCHAR2	10	0	
		BEST_ZIP4	VARCHAR2	4	0	
		BEST_URB_NAME	VARCHAR2	65	0	
		BEST_CARRIER_ROUTE_CD	VARCHAR2	4	0	
		BEST_DELIVERY_PT	VARCHAR2	2	0	
		BEST_DELIVERY_CKDG	VARCHAR2	1	0	
		BEST_LOT_NBR	VARCHAR2	4	0	
		BEST_LOT_SEQ	VARCHAR2	1	0	
		BEST_ACE_LATITUDE	VARCHAR2	10	0	
		BEST_ACE_LONGITUDE	VARCHAR2	11	0	
		BEST_ACE_REC_TYPE	VARCHAR2	2	0	
		BEST_ACE_ERROR_CD	VARCHAR2	30	0	
		BEST_ACE_ERROR_STATUS	VARCHAR2	30	0	
		BEST_ACE_DPV_STATUS	VARCHAR2	1	0	
		BEST_DPV_FTNOTE	VARCHAR2	12	0	
		BEST_ACE_FOREIGN_CD	VARCHAR2	1	0	
		BEST_BUSINESS_IND	VARCHAR2	1	0	
		BEST_FOREIGN_IND	VARCHAR2	1	0	
		BEST_DSF2_MATCH_LVL	VARCHAR2	1	0	
		BEST_DSF2_VACANT	VARCHAR2	1	0	
		BEST_DSF2_SEASON	VARCHAR2	1	0	
		BEST_DSF2_RESBUS	VARCHAR2	1	0	
		BEST_DSF2_DELTYPE	VARCHAR2	1	0	

BEST_DSF2_DELPTDRP	VARCHAR2	1	0	
BEST_MAIL_SCORE	VARCHAR2	1	0	
BEST_CASS_DELIVERABLE_IND	VARCHAR2	1	0	
BEST_PRISON_IND	VARCHAR2	1	0	
BEST_PANDER_IND	VARCHAR2	1	0	
BEST_DECEASED_IND	VARCHAR2	1	0	
BEST_APO_FPO_IND	VARCHAR2	1	0	
BEST_US_POSS_IND	VARCHAR2	1	0	
BEST_DPV_CMRA_IND	VARCHAR2	1	0	
BEST_DPV_FP_IND	VARCHAR2	1	0	
BEST_DPV_THROWBACK_IND	VARCHAR2	1	0	
BEST_FIPS_STATE	VARCHAR2	2	0	
BEST_FIPS_COUNTY	NUMBER	3	0	
NCOA_IND	VARCHAR2	1	0	
PCOA_IND	VARCHAR2	1	0	
LACS_IND	VARCHAR2	1	0	
EPAC_IND	VARCHAR2	1	0	
BEST_MOVE_DT	DATE	0	0	
BEST_RETURN_CD	VARCHAR2	2	0	
BEST_MOVE_TYPE	VARCHAR2	1	0	
BEST_PAC_ACTION_CD	VARCHAR2	1	0	
BEST_PAC_FOOTNOTE	VARCHAR2	15	0	
CENSUS_TRACT	VARCHAR2	6	0	
CENSUS_BLOCKGROUP	NUMBER	4	0	
GEOCODE	VARCHAR2	1	0	
OCCUPANCY_SCORE	VARCHAR2	1	0	
DNC_FLAG	VARCHAR2	1	0	
DNF_FLAG	VARCHAR2	1	0	
DNM_FLAG	VARCHAR2	1	0	
NURSING_HOME_FLAG	VARCHAR2	1	0	
CAN_LANG_PREF	VARCHAR2	1	0	
CLOSEST_STORE_NBR	NUMBER	10	0	
CLOSEST_STORE_DIST	NUMBER	10	2	
CLOSEST_AT_STORE_NBR	NUMBER	10	0	
CLOSEST_AT_STORE_DIST	NUMBER	10	2	
CLOSEST_LOFT_STORE_NBR	NUMBER	10	0	
CLOSEST_LOFT_STORE_DIST	NUMBER	10	2	
CLOSEST_ATFS_STORE_NBR	NUMBER	10	0	
CLOSEST_ATFS_STORE_DIST	NUMBER	10	2	
CLOSEST_LOS_STORE_NBR	NUMBER	10	0	
CLOSEST_LOS_STORE_DIST	NUMBER	10	2	

BU15.5.2

The following attributes are ultimately posted to the POSTAL CONTACT table:

Field	DataType	Precision	Scale
POSTAL_CONTACT_ID	NUMBER	10	0
POSTAL_CONTACT_TYPE_CD	VARCHAR2	1	0
POSTAL_CONTACT_STATUS_CD	VARCHAR2	1	0
ACTIVE_POSTAL_CONTACT_ID	NUMBER	10	0
DEDUPE_DT	DATE	0	0
INDIV_ID	NUMBER	13	0
HH_ID	NUMBER	13	0
GAID	NUMBER	32	0
BRAND_CD	VARCHAR2	4	0
ACCT_SOURCE_CD	VARCHAR2	4	0

ACCT_NBR	VARCHAR2	20	0	
ACCT_SOURCE_RANK	NUMBER	3	0	
VULGAR_IND	NUMBER	1	0	
PREFIX	VARCHAR2	20	0	
FIRST_NM	VARCHAR2	50	0	
MIDDLE_NM	VARCHAR2	50	0	
LAST_NM	VARCHAR2	50	0	
MATURITY_SUFFIX	VARCHAR2	20	0	
PROFESSIONAL_SUFFIX	VARCHAR2	20	0	
FIRM_NM	VARCHAR2	50	0	
SRC_GENDER_CD	VARCHAR2	1	0	
STD_GENDER_CD	VARCHAR2	1	0	
ADDR_LINE_1	VARCHAR2	65	0	
ADDR_LINE_2	VARCHAR2	65	0	
ADDR_LINE_3	VARCHAR2	65	0	
ADDR_LINE_4	VARCHAR2	65	0	
CITY	VARCHAR2	50	0	
URBANIZATION_NM	VARCHAR2	65	0	
STATE	VARCHAR2	40	0	
COUNTRY_CD	VARCHAR2	3	0	
POSTAL_CD	VARCHAR2	10	0	
ZIP4	VARCHAR2	4	0	
CARRIER_ROUTE_CD	VARCHAR2	4	0	
DELIVERY_POINT	VARCHAR2	2	0	
DELIVERY_POINT_CHK	VARCHAR2	1	0	
LOT_NBR	VARCHAR2	4	0	
LOT_SEQ	VARCHAR2	1	0	
BUSINESS_IND	NUMBER	1	0	
FOREIGN_IND	NUMBER	1	0	
MAIL_SCORE	VARCHAR2	1	0	
CASS_DELIVERABLE_IND	NUMBER	1	0	
PRISON_IND	NUMBER	1	0	
PANDER_IND	NUMBER	1	0	
DECEASED_IND	NUMBER	1	0	
APO_FPO_IND	NUMBER	1	0	
US_POSSESSION_IND	NUMBER	1	0	
DPV_CMRA_IND	NUMBER	1	0	
DPV_FALSE_POSITIVE_IND	NUMBER	1	0	
DPV_THROWBACK_IND	NUMBER	1	0	
PAC_ACTION_CD	VARCHAR2	1	0	
PAC_FOOTNOTE	VARCHAR2	15	0	
ACE_REC_TYPE	VARCHAR2	2	0	
ACE_ERROR_CD	VARCHAR2	30	0	
ACE_ERROR_STATUS	VARCHAR2	30	0	
ACE_DPV_STATUS	VARCHAR2	1	0	
ACE_DPV_FOOTNOTE	VARCHAR2	8	0	
ACE_FOREIGN_CD	VARCHAR2	1	0	
ACE_LATITUDE	VARCHAR2	10	0	
ACE_LONGITUDE	VARCHAR2	11	0	
DSF2_MATCH_LEVEL	VARCHAR2	1	0	
DSF2_VACANT_IND	NUMBER	1	0	
DSF2_SEASON_IND	NUMBER	1	0	
DSF2_RES_BUS_CD	VARCHAR2	1	0	
DSF2_DELIVERY_TYPE_CD	VARCHAR2	1	0	
DSF2_DEL_PT_DROP_IND	NUMBER	1	0	

AGILITY_DT	DATE	0	0	
COA_STATUS_CD	VARCHAR2	1	0	
COA_APPLIED_DT	DATE	0	0	
COA_MOVE_DT	DATE	0	0	
NCOA_MOVE_TYPE	VARCHAR2	1	0	
NCOA_RETURN_CD	VARCHAR2	2	0	
NCOA_IND	NUMBER	1	0	
PCOA_IND	NUMBER	1	0	
EPAC_IND	NUMBER	1	0	
LACS_IND	NUMBER	1	0	
FIPS_STATE	VARCHAR2	2	0	
FIPS_COUNTY	VARCHAR2	3	0	
CENSUS_TRACT	VARCHAR2	6	0	
CENSUS_BLOCKGROUP	VARCHAR2	4	0	
GEOCODE	VARCHAR2	1	0	
OCCUPANCY_SCORE	NUMBER	1	0	
DNC_IND	NUMBER	1	0	
DNF_IND	NUMBER	1	0	
DNM_IND	NUMBER	1	0	
NURSING_HOME_IND	NUMBER	1	0	
CAN_LANG_PREF	VARCHAR2	1	0	
ACTIVITY_DT	DATE	0	0	
CREATED_JOB_ID	NUMBER	10	0	
CREATED_DT	DATE	0	0	
MODIFIED_JOB_ID	NUMBER	10	0	
MODIFIED_DT	DATE	0	0	
ALTERNATE_KEY	NUMBER	20	10	
ORIG_ALTERNATE_KEY	NUMBER	20	10	

BU15.5.3

Store attributes are posted to the POSTAL_CONTACT_STORE table for each POSTAL_CONTACT for which one or more stores could be found:

Field	DataType	Precision	Scale
POSTAL_CONTACT_ID	NUMBER	10	0
CLOSEST_STORE_NBR	NUMBER	10	0
CLOSEST_STORE_DIST	NUMBER	10	2
CLOSEST_AT_STORE_NBR	NUMBER	10	0
CLOSEST_AT_STORE_DIST	NUMBER	10	2
CLOSEST_LOFT_STORE_NBR	NUMBER	10	0
CLOSEST_LOFT_STORE_DIST	NUMBER	10	2
CLOSEST_ATFS_STORE_NBR	NUMBER	10	0
CLOSEST_ATFS_STORE_DIST	NUMBER	10	2
CLOSEST_LOS_STORE_NBR	NUMBER	10	0
CLOSEST_LOS_STORE_DIST	NUMBER	10	2
ACTIVITY_DT	DATE	0	0
CREATED_JOB_ID	NUMBER	10	0
CREATED_DT	DATE	0	0
MODIFIED_JOB_ID	NUMBER	10	0
MODIFIED_DT	DATE	0	0
ALTERNATE_KEY	NUMBER	20	10
ORIG_ALTERNATE_KEY	NUMBER	20	10

BU15.6 The system will provide a mechanism to track consolidation of Postal Contacts and their associated ids as data changes either through CDI key consolidation or change of address processing resulting in duplicate postal contacts.

BU15.6.1 Each POSTAL CONTACT in the system is identified by a unique postal contact id. The system only allows one active POSTAL CONTACT to have a particular combination of

	individual id and global address id. As part of the update process, the system applies individual id consolidations as provided by Agility via a cross reference. Additionally, the quarterly COA process passes all active POSTAL CONTACT records through Agility. Either of these processes may result in a change of individual_id for a given POSTAL CONTACT. The system generates a POSTAL CONTACT ARCHIVE record for any POSTAL CONTACT that has a change to their individual, household, or address identifiers. If, after the application of updates, two active POSTAL CONTACTS share the same individual id and address id, the system retains the record with the highest postal contact id. The other record is marked as inactive and a pointer the active POSTAL CONTACT record is set. Additionally, the system scans all other inactive POSTAL CONTACTs that were pointing to the newly inactivated record and sets their pointer to the active postal contact id as well. Finally, all ACCOUNTs with a newly inactivated POSTAL CONTACT foreign key are redirected to the new active postal contact id as well.
BU15.6.2	The system uses a similar process to 15.6.1 to maintain individual id integrity within the INDIVIDUAL table. See 14.4.4
BU15.7	If any of the Agility CDI keys change for a Postal Contact, the system will create an archive record of the Postal Contact name and address information before applying the change.
BU15.7.1	The system evaluates key changes on the POSTAL CONTACT table due to either the quarterly COA processing or individual_id consolidation as part of the ongoing update. If any of the keys are determined to be different, the existing Postal Contact is rolled off and stored in the POSTAL_CONTACT_ARCHIVE table for auditing purposes and the new Postal Contact data and keys are loaded.
BU15.8	The system will track the closest store number and distance for a Postal Contact based on latitude/longitude calculations for the following categories: overall store, Ann Taylor Store, Ann Taylor Factory Store, Loft Store, and Loft Outlet Store.
BU15.8.1	Upon initial load and with each incremental load, the SAP Store data is extracted from the Data Warehouse and run through AGILITY. The process standardizes the addresses and assigns geographic Latitude and Longitude to each store's address.
BU15.8.2	As each incremental feed is processed through AGILITY and has its own address standardized and Latitude and Longitude assigned, "straight-line" proximity is determine between each individual's address and the nearest store for each brand identified on the Store Feed
BU15.8.3	The closest store number of each branded store and the distinct in miles (to the hundredth of a mile) is returned from the AGILITY process and ultimately loaded to the POSTAL_CONTACT_STORE table.
BU15.9	The system will recalculate store distance calculations for all records run through Agility during each update.
BU15.9.1	Each record processed through AGILTY has it's Lat/Long compared to the AGILITY Store reference table for each branded store to calculate proximity in hundredths of a mile. Should the Contact Already exist in the Data Warehouse, the nearest store attributes are updated for the existing postal contact if any of the store distances are different than what is currently in the Data Warehouse for the POSTAL CONTACT.
BU15.10	Store distance calculations will be recomputed for all Postal Contacts as part of the quarterly NCOA process. Additions or closure of stores in between runs could change the distance calculations for individual records. These changes will not be reflected on Postal Contacts until the next NCOA run unless the record came across as part of a daily update process.
BU15.10.1	The AGILTY weekly process is provided with a full refresh of all open stores where have Lat/Long recalculated and stored as a lookup table within AGILITY.
BU15.10.2	When the full Postal Contact extract is re-processed through AGILITY, the store proximity calculations are recalculated and updated on the POSTAL_CONTACT_STORE table upon the return of the feed and processing through the Data Warehouse update
BU15.11	The system will flag the postal contact as being an ANN INC. employee as long as one of the account records has a source of 'EMP'.

BU15.11.1	When the SAP Associate feed is processed through Customer update process through it's mapping into WRK_SRC_ACCOUNT, the brand is hard-coded as "ENT"
BU15.11.2	These ultimately survive in the Data Warehouse ACCOUNT table. Any EXACT Name/Address combination not found in the current full refresh update feed is indicated as INACTIVE. This is done by flagging ACCOUNT records that have ACCT_SRC_CD = "EMP" and LST_UPDATE_DT <> current update date

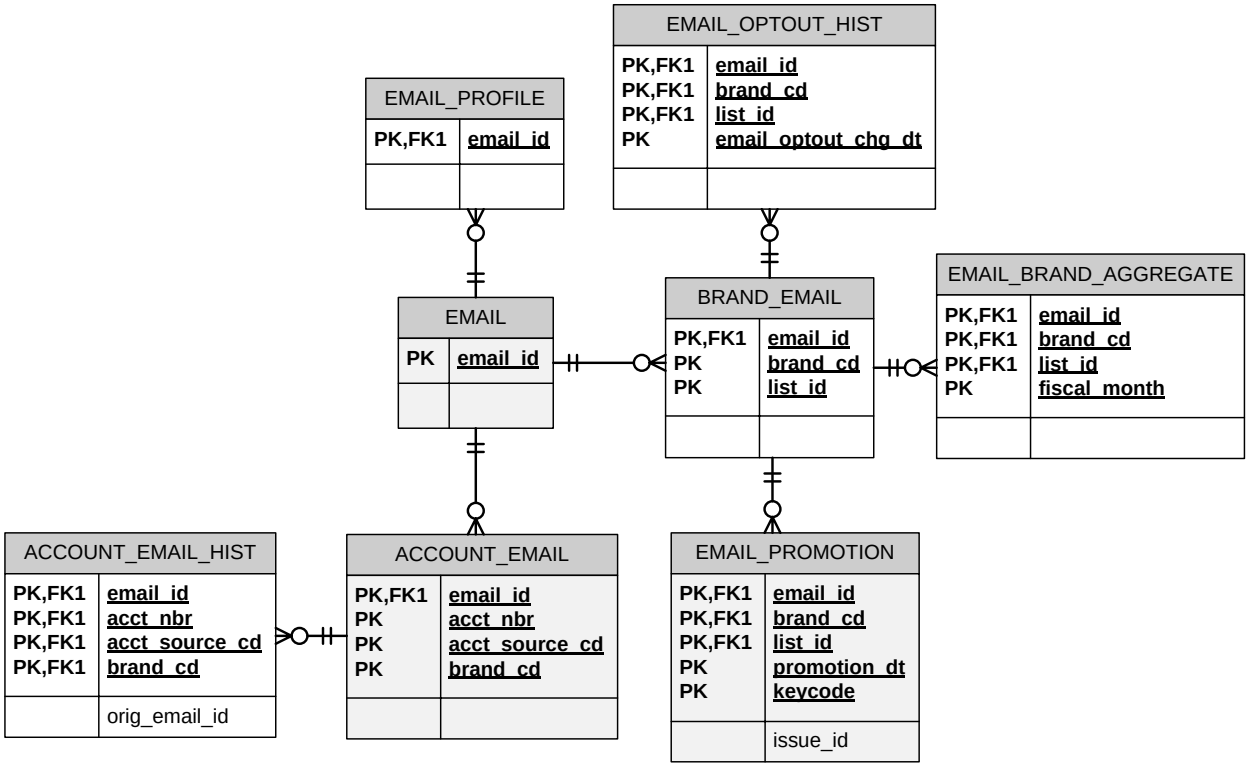
16. Phone

Phone numbers from ANN's sources will be gathered up and de-duplicated (exact matches only) and processed through a basic hygiene/ validation routine.

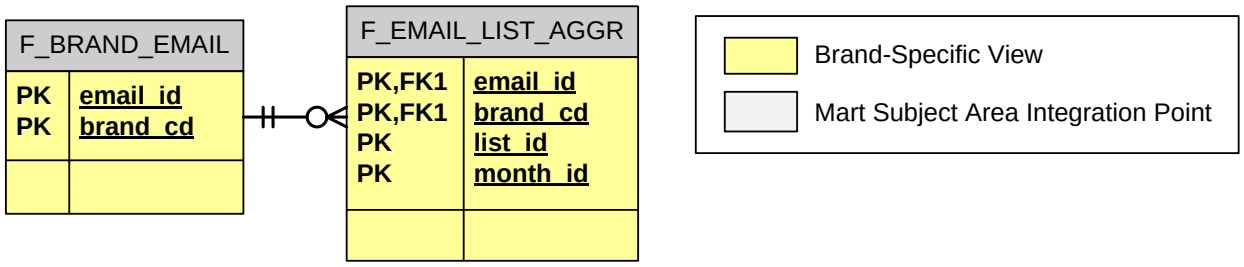
#	Design
BU16.1	The system will maintain one record for every unique phone number that is received on any of the ANN sources.
BU16.1.1	Phone numbers are stored on the PHONE table in the Data Warehouse unique on formatted PHONE_NBR and PHONE_COUNTRY_CD with a surrogate key called PHONE_ID
BU16.1.2	This table also has attributes of • Country • Valid Indicator • Nuclear Opt Out
BU16.1.3	Each phone number relates to one or more ACCOUNTs through the ACCOUNT_PHONE table. This table has additional for each Account Number, Phone Number, Brand, and Source attributes of: • Phone Type (Home, Office, Mobile) • Account Phone Preference
BU16.2	The system will perform basic hygiene by removing non-numeric data from the phone number.
BU16.2.1	Non-numeric data is stripped from all phone numbers as described in **BU6.5.1**
BU16.3	The system will indicate if the phone number is valid or invalid for Canadian and US addresses by the following hygiene rules; • Remove all non-numeric data and make sure there is a minimum of 10 digits • Is not all 0's, 1's, 2's, 3's, 4's, 5's, 6's, 7's, 8's, or 9's • Does not have a "bad" area code 1. Does not start with 1 or 0 (100xxxxxx, 101xxxxxxx, 001xxxxxxx,etc) 2. Does not have '11' in 2nd/3rd digits (411xxxxxxx, 911xxxxxxx, etc) 3. Is not greater >= 990 (990xxxxxxx,993xxxxxxx, etc) 4. Is not a free/pay line (800x,900x, 866x,877x, etc) 5. Validate the area code is in the public NPA database as assigned and in-service • Is not found in the STORE table • 4th position of the exchange is 2-9
BU16.3.1	The country assigned to a phone number is provided through the ADDRESS on the Account mapping. Where there is no address, the country is defaulted to "USA"
BU16.3.2	All US and CA phone numbers are validated through the following logic: CASE WHEN phone_country_cd IN ('USA','CAN')

	AND REGEXP_INSTR (phone_nbr,'^\(?([2-9][0-8][0-9])\)?[-.]?([2-9][0-9]{2})[-.]?([0-9]{4})$') = 0 THEN 0 ELSE 1 END AS valid_phone_ind
BU16.3.3	Phone numbers resulting from the above statement are formatted as (999) 999-9999
BU16.4	Any phone number with more than 10 digits after non-numeric data is removed, could be an international phone number and will be flagged as such.
BU16.4.1	Any phone number not linked to either a US or Canadian source (non-standardized) address is identified as invalid. Phone numbers with an undefined country code are defaulted to USA and eligible for further validation.
BU16.4.2	In order to be identified as valid, the phone must have exactly 10 digits. This excludes country code and/or extension inclusion on domestic or Canadian phone numbers from being identified as valid.
BU16.5	The system will indicate a preference at the phone level (regardless of brand or account associations) by assigning a flag with values for opt-in/ opt-out.
BU16.5.1	The brand agnostic phone preference is stored on the PHONE table through the NUCLEAR_OPTOUT_IND. In the event that multiple account sources have varying nuclear opt out preference, the most recently loaded is used to define the global phone preference. An undefined value for this preference will never override an explicit value.
BU16.5.2	The account phone preferences are stored on ACCOUNT_PHONE for each phone type and linked brand account.
BU16.6	The system will leverage 3rd party data to correctly identify the phone type (e.g. POTS, Wireless, etc) Epsilon to research capabilities/ cost etc….
BU16.6.1	The initial implementation does not leverage 3rd party phone type identification or validation. This may be included as a **future enhancement** to the solution.

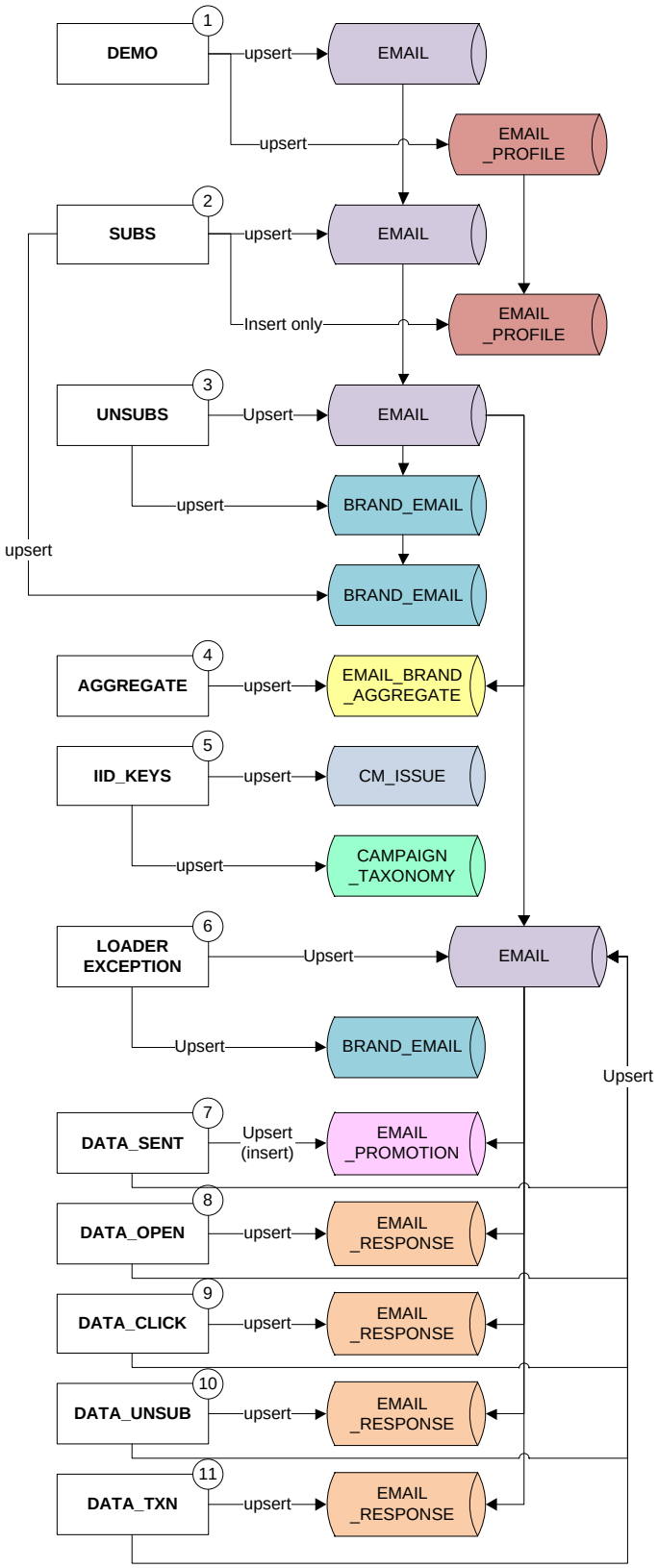
Email Entity Key Relationships – Data Warehouse



Email Entity Key Relationships – Data Mart



Email Data Warehouse Update Process



Email Data Warehouse Update Process Annotations

1a. Email Addresses found on the DEMOGRAPHIC feed are upserted into the Data Warehouse EMAIL table. Any email address not found is inserted and a new Email ID is generated.

1b. Demographic data for the Email ID referenced in step 1 is upserted into the EMAIL_PROFILE table.

2a. Email Addresses found on the SUBS feed are upserted into the Data Warehouse EMAIL table. Any email address not found is inserted and a new Email ID is generated.

2b. For new inserts only in step 3, the demographic data found on the SUBS feed is inserted into EMAIL_PROFILE.

3a. UNSUBS are used to update the NUCLEAR_OPTOUT_IND on the EMAIL table where the unsub reason code (for any list/PID) is either "A" (Abuse Complaint), "O" (Global Suppression Opt-Out Page), or "S" (Global Suppression Upload). Any email address in the feed that is not found is inserted and a new email_id is generated for the email table.

The identification of NUCLEAR OPTED OUT email addresses is intended to sync the CRM database with the stop listed email addresses maintained at CheetahMail.

3b. The UNSUBS feed is used to upsert the BRAND_EMAIL table, setting the OPTOUT_IND and OPTOUT_DT for each Email ID and List/PID, and setting the BRAND_CD based on PID, and unrecognized PID is defaulted to "ENT". When an existing Email ID was already subscribed, and entry is made into EMAIL_OPTOUT_HIST to track the CHANGE in preference. Any email address in the UNSUBS feed not already found in the EMAIL table (because it was previously loaded as a SUBS to DEMOGRAPHIC change) is added.

3c. Similarly, the SUBS feed is used to upsert the BRAND_EMAIL table in the opposite manner such that and UNSUBBED email address on a particular address are re-subscribed and the change is logged in the history table.

4. The AGGREGATES feed is upserted into EMAIL and EMAIL_BRAND_AGGREGATE based on an EMAIL_ID lookup, List ID/PID, decoded brand code, and rollup fiscal month obtained from the source data

5a. The IID_KEYS is upserted into the CM_ISSUE table to facilitate the tracking of email campaigns by CheetahMail key.

5b. The IID_KEYS is loaded into the CAMPAIGN_TAXONOMY table to provide referential integrity in that the CheetahMail issue_id is used a surrogate for campaign key code until such time that Email campaigns are extracted from the CRM solution.

6. The LOADER EXCEPTION feed is used to update the EMAIL table, setting the UNDELIVERABLE_IND = 1 for any email addresses rejected by the CheetahMail load process. Any Email Address in the source table that doesn't already exist in EMAIL is added

7. The Send Events are upserted into EMAIL and EMAIL_PROMOTION. Any email address that is not found in the EMAIL table results in a reject of the activity record.

8. The Open Events are upserted into EMAIL and EMAIL_RESPONSE. Any email address that is not found in the EMAIL table results in a reject of the activity record.

9. The Click Events are upserted into EMAIL and EMAIL_RESPONSE. Any email address that is not found in the EMAIL table results in a reject of the activity record.

10. The Unsub Events are upserted into EMAIL and EMAIL_RESPONSE. Any email address that is not found in the EMAIL table results in a reject of the activity record.

11a. The Transaction Events are upserted into EMAIL and EMAIL_RESPONSE. Any email address that is not found in the EMAIL table results in a reject of the activity record. The Transaction Events include the transaction amount as provided by SDL, however the transaction ID provided on the email event is not the same as provided by ATG for E-Commerce orders, as such the direct link from response to F_TXN_HEADER cannot be established. T2P aggregates are performed solely on F_EMAIL_RESPONSE data.

17. Email

Email addresses from ANN's sources will be gathered up and de-duplicated (exact matches only) and processed through a hygiene routine. Hygiene will include validating that the email has the basic components of an email address, domain correction, and limited character replacement/removal.

#	Design
BU17.1	The system will maintain one record for every unique email.
BU17.1.1	The EMAIL table contains one row for each EMAIL_ADDR and a unique surrogate called EMAIL_ID
BU17.1.2	In the event that an email was corrected both values are present in the table and thus have different EMAIL_IDs. Each corrected email address is referenced by the incorrect email addresses ACTIVE_EMAIL_ID
BU17.1.3	When assigning EMAIL_ID foreign key relationships based on an Email Address value the following steps are performed • An UPSERT into the EMAIL table is performed to update/insert the Email table • Join on Source to EMAIL.EMAIL_ADDR • Select ACTIVE_EMAIL_ID In this manner, foreign key references to corrected email address are re-pointed to the active counterpart.
BU17.2	The system will indicate if the email address has a valid email format and does not contain any invalid characters.
BU17.2.1	The EMAIL.valid_ind is set based on the Epsilon Email Hygiene rules defined in **EH1.3.1**
BU17.3	The system will indicate if the email has been deemed undeliverable by CheetahMail.
BU17.3.1	The EMAIL.UNDELIVERABLE_IND_ind is set based on CheetahMail LOADER EXCEPTION data is used to set the valid indicator as defined in **EH1.3.2**
BU17.4	The system will indicate a preference at the email level (regardless of brand or account associations).
BU17.4.1	The brand agnostic email preference is stored on the EMAIL table through the NUCLEAR_OPTOUT_IND. In the event that multiple account sources have varying nuclear opt out preference, the most recently loaded is used to define the global email preference. This indicator is intended to reflect the stop-listed email IDs maintained at CheetahMail. Typically it is used as a suppression for Email Campaigns such that email addresses are not selected for campaigns that will later be rejected by the ESP.
BU17.4.2	The account email preferences are stored on ACCOUNT_EMAIL for each email address and linked brand account.
BU17.5	The system provides functionally to track promotion and response activity for email campaigns originating both from SAS CI solution as well as directly from CheetahMail. • Email campaigns will not be generated out of SAS CI initially after implementation

	of the CRM is completed. Time frame will be discussed and at that time the 'keycode' field from the XDB process from CheetahMail needs to be in place for the tracking of response activity back to a SAS CI generated email promotion.
BU17.5.1	The SDL/CheetahMail update process is designed to support future extraction of Email Campaigns directly from the CRM solution. 1. the SAS campaign process would generate a campaign file in a predetermined naming convention in a staging area while the campaign the being validated by the ANN marketing team. The campaign file would contain the following attributes: a. Email Address b. Campaign Keycode c. Any elements required to define the dynamic or static creative content for CheetahMail's deployment process 2. An Email Promotion feed is generated with the following attributes: a. Email ID b. Campaign Keycode c. Brand Code (may be derived from keycode) d. Mail and Control/Holdout indicator e. Any elements to freeze at the time of campaign extraction to be defined as a **Future Enhancement** 3. A corresponding Campaign Taxonomy record is generated to supply required campaign meta-data 4. Upon approval, the Campaign Feed is deployed by CheetahMail 5. The Email Promotion table is loaded in the DW with the following initialized fields a. LIST_ID (PID) = null b. PROMOTION_DT = null c. ISSUE_ID = null d. DELIVERED_IND = 0 6. CheetahMail returns the SEND EVENT data. Based on the returned Email Address and Keycode the elements identified in Step 5 are updated. Note: The DELIVERED_IND field is only updated where Result Code = "R". This excludes emails that were successfully loaded but soft-bounced on the campaign. 7. CheetahMail also sends the IID_KEYS feed to provide descriptive campaign header information related to the campaign. 8. Over the subsequent days/weeks the remaining send events are provided and incrementally updated. These are loaded into EMAIL_RESPONSE and keyed on email address (email_id), ISSUE_ID, event type, and event date a. Open Events b. Click Events c. Unsub Events d. Transaction Events (including TXN_AMT) 9. In this manner, send events can be linked to response activity by way of the CheetahMail ISSUE_ID and EMAIL_ID 10. Note: The ORDER_ID on transaction events may not be the same as provided by ATG, so a directly link to F_TXN_HEADER on this attribute is not possible. All T2P aggregates are performed on the F_EMAIL_RESPONSE data
BU17.5.2	Email Promotion and Activity data tracked from a CheetahMail soured campaign works much the same way except that:

	• Lacking a CRM sourced campaign keycode, the ISSUE_ID is used as a surrogate. This forces a 1:1 relationship between message and segmentation. • A taxonomy record is generated for each ISSUE_ID received in the IID_KEYS feed • Lacking a true promotion history feed, the concept of control/holdout segments is not supported
BU17.5.3	The Email Response activity is keyed on EMAIL_ID, ISSUE_ID, EMAIL_RESPONSE_DT and EMAIL_RESPONSE_TYPE_CD. Email Response Date is stores as a DATE field which down to the precision of a 'second'. In the event that multiple responses are received within a second for the same email address and type, these are de-duped and loaded distinctly. This may cause a cause a discrepancy when comparing total responses in the CRM database to that aggregated in the CheetahMail campaign reports.

18. Brand Email

<table>
<tr><th>#</th><th>Design</th></tr>
<tr><td>BU18.1</td><td>The system will maintain one record for every unique email, brand, list combination. This table supports email data that is not associated with an account (for example - from ATG).</td></tr>
<tr><td>BU18.1.1</td><td>The BRAND_EMAIL table is used to stored Email Preferences sourced from the CheetahMail/SDL SUBS and UNSUBS feeds</td></tr>
<tr><td>BU18.1.2</td><td>The subscriber feeds are processed such that the last preference, based on SUB/UNSUB timestamp recency is identified as current within the database.</td></tr>
<tr><td>BU18.1.3</td><td>The BRAND is decoded from the LIST_ID (PID) by the following table:
<table>
<tr><th>PID</th><th>Description</th><th>Brand</th></tr>
<tr><td>1617026943</td><td>Ann Taylor Stores</td><td>AT</td></tr>
<tr><td>1617027553</td><td>LOFT</td><td>LOFT</td></tr>
<tr><td>1617027907</td><td>Ann Taylor Factory</td><td>ATF</td></tr>
<tr><td>2071576052</td><td>LOFT Outlet Stores</td><td>LOS</td></tr>
<tr><td>2089582708</td><td>Ann Taylor Canada</td><td>AT</td></tr>
<tr><td>2089600494</td><td>LOFT Canada</td><td>LOFT</td></tr>
<tr><td>2068023233</td><td>Celebrations Bride Outreach</td><td>AT</td></tr>
<tr><td>2072079322</td><td>LOFT Loves Teachers</td><td>LOFT</td></tr>
<tr><td>2078200572</td><td>LiveLoveLOFT</td><td>LOFT</td></tr>
<tr><td>2072079327</td><td>Network for Style (can be completed ignored; 52 people a long time ago)</td><td>ENT</td></tr>
<tr><td>1950820576</td><td>Temp List (probably unnecessary to have on Epsilon?)</td><td>ENT</td></tr>
<tr><td>2088200651</td><td>one of the credit card lists (LOFT PLCC, LOFT MC, AT PLCC, or AT MC)</td><td>ENT</td></tr>
<tr><td>2088200983</td><td>one of the credit card lists (LOFT PLCC, LOFT MC, AT PLCC, or AT MC)</td><td>ENT</td></tr>
<tr><td>2088200991</td><td>one of the credit card lists (LOFT PLCC, LOFT MC, AT PLCC, or AT MC)</td><td>ENT</td></tr>
<tr><td>2088200995</td><td>one of the credit card lists (LOFT PLCC, LOFT MC, AT PLCC, or AT MC)</td><td>ENT</td></tr>
</table></td></tr>
<tr><td>BU18.1.4</td><td>Any new list encountered that is not in the above table is assigned the "ENT" (enterprise) brand. The integration of an additional list requires a small change request for to identify the additional reference</td></tr>
<tr><td>BU18.1.5</td><td>The EMAIL_OPTOUT_HIST table in the Data Warehouse tracks the date each time an email preference changes. This is used to audit the process as an email address is opted in and out over time.</td></tr>
<tr><td>BU18.1.6</td><td>In the Data Mart, the F_BRAND_EMAIL email table is unique on EMAIL_ID and BRAND. Individual list preferences are "flattened out" to this level such that each email address has up to NINE (9) subscriber list designations for each brand.

In the warehouse, permissions are stored for each brand and pid (list). In the mart, there</td></tr>
</table>

<table>
<tr><td></td><td>will be one record per email address and brand. If a brand has multiple pids, then their block of permissions has to be flattened out into one of up to nine sets of list opt-outs for the brand.

They are defined as follows;
<table>
<tr><th>Brand</th><th>List1 Field Name</th><th>List2 Field Name</th><th>List3 Field Name</th></tr>
<tr><td>AT</td><td>PRIMARY_OPTOUT_IND</td><td>BRIDE_OPTOUT_IND</td><td>CA_OPTOUT_IND</td></tr>
<tr><td>LOFT</td><td>PRIMARY_OPTOUT_IND</td><td>TEACHER_OPTOUT_IND</td><td>CA_OPTOUT_IND</td></tr>
<tr><td>ATF</td><td>PRIMARY_OPTOUT_IND</td><td>*future use</td><td>*future use</td></tr>
<tr><td>LOS</td><td>PRIMARY_OPTOUT_IND</td><td>*future use</td><td>*future use</td></tr>
<tr><td>ENT</td><td>NETWRK_STYLE_OPTOUT_IND</td><td>TEMPLIST_OPTOUT_IND</td><td>CC1_OPTOUT_IND</td></tr>
</table>
<table>
<tr><th>Brand</th><th>List4 Field Name</th><th>List5 Field Name</th><th>List6 Field Name</th></tr>
<tr><td>AT</td><td>MC_OPTOUT_IND</td><td>*future use</td><td>*future use</td></tr>
<tr><td>LOFT</td><td>LIVELOVE_OPTOUT_IND</td><td>*future use</td><td>*future use</td></tr>
<tr><td>ATF</td><td>*future use</td><td>*future use</td><td>*future use</td></tr>
<tr><td>LOS</td><td>*future use</td><td>*future use</td><td>*future use</td></tr>
<tr><td>ENT</td><td>CC2_OPTOUT_IND</td><td>CC3_OPTOUT_IND</td><td>CC4_OPTOUT_IND</td></tr>
</table>
<table>
<tr><th>Brand</th><th>List7 Field Name</th><th>List8 Field Name</th><th>List9 Field Name</th></tr>
<tr><td>AT</td><td>*future use</td><td>*future use</td><td>*future use</td></tr>
<tr><td>LOFT</td><td>*future use</td><td>*future use</td><td>*future use</td></tr>
<tr><td>ATF</td><td>*future use</td><td>*future use</td><td>*future use</td></tr>
<tr><td>LOS</td><td>*future use</td><td>*future use</td><td>*future use</td></tr>
<tr><td>ENT</td><td>*future use</td><td>*future use</td><td>*future use</td></tr>
</table></td></tr>
<tr><td>BU18.1.7</td><td>Each of the above List OPTOUT_IND attributes has an accompanying LIST_ID (for reference), last OPTOUT_DT, and OPTOUT_REASON_CD</td></tr>
<tr><td>BU18.1.8</td><td>As CheetahMail/SDL is the sole-source of truth on email preferences, ACCOUNT based email preferences are stored on ACCOUNT_EMAIL for reference only, these are not made visible to the end user in the Data Mart</td></tr>
<tr><td>BU18.2</td><td>The system will indicate an email preference at the brand level (regardless of account associations).</td></tr>
<tr><td>BU18.2.1</td><td>The BRAND_EMAIL table stores list preferences completely independently of Account source preferences. Each preference is stored at a list level and through association at the brand level.</td></tr>
</table>

19. Email Permissions/ Preferences

#	Design
BU19.1	The system will support email global opt-out permissions. • Abuse records for the email level will be sent to Epsilon on the loader exception file. Any abuse at this level will opt out the email address from all list level mailings. This is a global opt-out.
BU19.1.1	The NUCLEAR_OPTOUT_IND is set on the EMAIL table for any email address on any subscriber list that is loaded in the UNSUBS feed with one of the following reason codes: • A – Abuse complaint • O – Global Suppression Opt-Out Page • S – Global Suppression Upload
BU19.1.2	Each LIST optout includes the granular optout reason codes as defined on the UNSUBS feed in the warehouse. In the mart these are decoded to: • A = Abuse • B = Bounce

	• U = Other unsub reasons
BU19.1.2	Once the NUCLEAR_OPTOUT_IND is set to 1, there is no automated mechanism for unsetting it. Doing so will require a written Change Request.
BU19.2	The system will support email subscriber list level permissions. The current configuration with CheetahMail supports several Program IDs. The program ID represents the ANN subscriber lists by brand; • (Ann Taylor, LOFT, Ann Taylor Factory, LOFT Outlet, LOFT Teachers, Ann Taylor Celebrations Bride, Ann Taylor Canada, LOFT Canada). Only data associated with these lists will be in the system. • Opt-in/ Opt-out permissions will be sent to Epsilon on the Subscription and Unsubs files. • The migration will be a one-time load to the system to indicate current opt-in/ opt-out statuses for each email address by subscriber list. • Ongoing the system will retain email preferences and retain history, including the date of each time the preference setting of an email address /subscriber list changed.
BU19.2.1	Email Preference data is supported as described in **BU18.1**
BU19.3	The system will retain email preference data from CheetahMail from the Changed Demo file, in preference fields.
BU19.3.1	Email preference data is set completely based on the SUBS and UNSUBS feeds
BU19.3.2	When a demographic attribute change is made at CheetahMail, this triggers an inclusion of that email address in the Change of Demographics feed, even if not other demographics attribute has been modified.
BU19.3.3	The EMAIL_PROFILE table is loaded as follows: 1. Upsert EMAIL from STG_SDL_DEMOGRAPHICS 2. Upsert EMAIL_PROFILE from STG_SDL_DEMOGRAPHICS 3. Upsert EMAIL from STG_SDL_SUBS 4. Insert ONLY EMAIL_PROFILE from STG_SDL_SUBS This states that the EMAIL table is loaded to generate an EMAIL ID as a dependency for the load of the email profile (demographics) from the DEMO and SUBS feed. It also states that only New to File email addresses are loaded into demographics from the SUBS feed. Though theoretically, it should not matter because demographics are stored at the email address level at CM so there is no harm in applying demographics feed from the SUBS (except that we already have it loaded based on the DEMO feed). Merely a comment on process dependency enforcement and load efficiency.
BU19.3.4	The BRAND_EMAIL table subsequently loaded as follows: 1. Upsert BRAND_EMAIL from STG_SDL_UNSUBS 2. Upsert BRAND_EMAIL from STG_SDL_SUBS 3. Upsert BRAND_EMAIL from STG_CM_LOADER (albeit as undeliverable, unless the email address can be corrected and confirmed as OPTED IN through the CheetahMail process)
BU19.4	The system does not get email permissions from RightNow located in the Category ID field If there is a value of email-subscribe or email-unsubscribe, these are just identify the category or the reason the customer called the call center. These are not to be used to set any email permissions.
BU19.4.1	The only access provided to end users on email-specific preferences of any kind are those defined through the CheetahMail SUBS and UNSUBS feeds.

BU19.4.2	The NUCLEAR OPT OUT IND is set globally by the Account update process and cascaded down to any emails linked to an account with this account attribute.
BU19.5	The system will store segments identified by ANN from the MarketWorks Customer Attribute, in email preference fields. **See Appendix F**
BU19.5.1	Customer segments are stored in a reference table as a relationship to ACCOUNT. As email preferences, these are only stored for reference and only within the Data Warehouse. Email preferences in the mart are driven exclusively off the SDL subs/unsubs feeds.
BU19.6	The system will retain only the most current demographic attributes for each email address provided by CheetahMail as an existing email address or as a new email address. The list of included demographic attributes are described in the interface agreement on the Change of Demographic file for existing email addresses and on the subscription file for new email addresses – see **Appendix I**
BU19.6.1	The DEMOGRAPHICS data is upserted with each update cycle in the following sequence: • STG_SDL_DEMOGRAPHICS • STG_SDL_SUBS (new inserts only)

20. Direct Mail/ Telemarketing/ Rental Permissions

#	Design
BU20.1	The system will store segments identified by ANN from the MarketWork's Customer Attribute, in direct mail/ telemarketing/ rental permission fields. **See Appendix F**
BU20.1.1	As part of the WRK_SRC_ACCOUNT mapping the MarketWorks MAIL_OPTOUT_IND (Direct Mail Preference) is set as described in the **Source to Work Account** mapping documents
BU20.1.2	The MarketWorks telemarketing preference for HOME phone is set as described in the Source to Work Account mapping documents
BU20.1.3	The MarketWorks telemarketing preference for SMS is set as described in the Source to Work Account mapping documents
BU20.1.4	There are no attributes available for setting MarketWorks telemarketing preferences for: • Work Phone • Mobile Phone
BU20.1.5	The MarketWorks RENT_OPTOUT_IND is set as described in the Source to Work Account mapping documents
BU20.1.6	In the Data Mart, Account mail, telemarketing and rental preferences are resolved across multiple account sources such that when an individual/household is opted out from one or more accounts it is opted out at the individual/household level

BU20.2	The system will store direct mail permissions supplied by the following sources:
	• ATG to provide direct mail opt-in/ opt-out on the Catalog Request file
	• MW historical customer table to provide direct mail opt-in/ opt-out in the following fields;
	  ○ Segmentation Flag A – ATS Do not mail (no date associated with flag)
	  ○ Segmentation Flag B – LOFT Do not mail (no date associated with flag)
	  ○ Segmentation Flag D – AT Factory Do not mail (no date associated with flag)
	  ○ Segmentation Flag F – LOS Do not mail (no date associated with flag)
	• ADS to provide direct mail and do not rent opt-in/ opt-out on the Daily and monthly customer files
	• RightNow direct mail permissions located in the Category ID field mailer-subscribe, mailer-unsubscribe).

BU20.2.1	The ATG Catalog Requestor feed does not include an attribute that can be leveraged to set the MAIL_OPTOUT_IND
BU20.2.2	The MarketWorks MAIL_OPTOUT_IND is set as described in **BU20.1.1**
BU20.2.3	The ADS Daily and Monthly feeds do not include any attribute that can be leveraged to set the MAIL_OPTOUT_IND
BU20.2.4	The ADS Daily and Monthly feeds do not include any attributes that can leveraged to set the RENT_OPTOUT_IND
BU20.2.5	The RightNow Call Center feed does not include any attributes that can leveraged to set the MAIL_OPTOUT_IND
BU20.2.6	In the Data Mart, Account mail and rental preferences are resolved across multiple account sources such that when an individual/household is opted out from one or more accounts it is opted out at the individual/household level
BU20.3	The system will store telemarketing opt -In/ opt-out, supplied by ADS on the daily and monthly customer files.
BU20.3.1	The ADS Telemarketing indicators are set based on the Monthly and Daily customer attribute as follows: DECODE(a.TM_OPTOUT_CD, 'X', 1, 0) This applies to the following preferences: • HOME_PHONE_OPTOUT_IND • WORK_PHONE_OPTOUT_IND • MOBILE_PHONE_OPTOUT_IND
BU20.3.2	In the Data Mart, Account telemarketing preferences are resolved across multiple account sources such that when an individual/household is opted out from one or more accounts it is opted out at the individual/household level

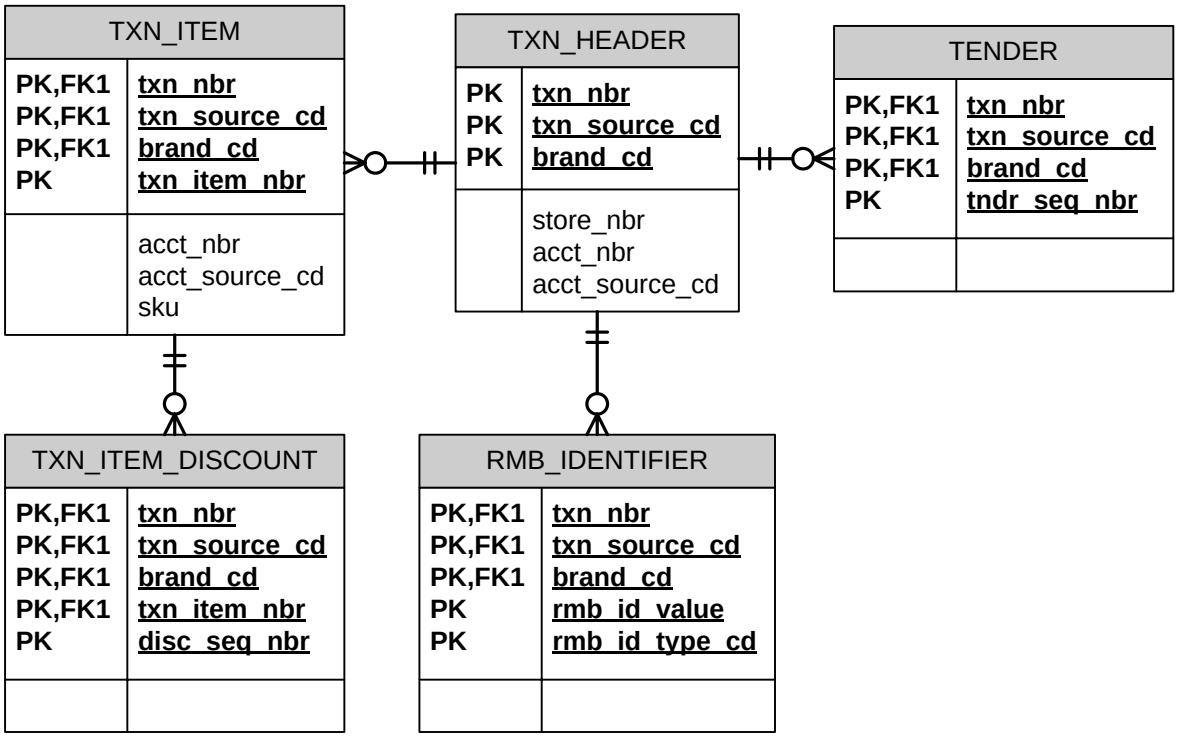
21. Overlays

#	**Design**
BU21.1	MOSAIC: The system will extract a feed to be sent to Experian for overlay of MOSAIC clusters to addresses found on the database. All unique US addresses associated with active, unique postal contacts will be pulled if they have conducted a transaction in the past 36 months. New and changed records will be extracted quarterly basis with an annual refresh of the past 36 month clients Licensing of is for 12 month periods. Therefore, cleanup will be necessary on records no longer part of the past 36M client append records.
BU21.1.1	To perform the initial full and refresh extract, The POSTAL_CONTACT table is inner joined to TXN_HEADER through ACCOUNT (bill to) to determine activity. The order date is referenced on the fiscal calendar. If the fiscal month of the transaction is >= current fiscal month – 36 the POSTAL CONTACT is included for extraction
BU21.1.2	To perform the quarterly extract the same join is used to determine 36-month activity, however the universe is further limited by only including postal contacts that have been created/updated since the latest overlay process executed. This is done by referencing the FILE_DETAIL table where SOURCE_CD = 'EXP' and FILE_TYPE_CD = 'MOSAIC' and extracting only where the Postal Contact DW Create/Update date > File Detail FEED_DT
BU21.1.3	MOSAIC data is presented in the Data Mart layer as the overlay data related to the BEST postal contact
BU21.2	Demographic append: The system will extract a feed to be sent to KBM Group for overlay of demographic information at the individual/household/address/geographic level. Address and

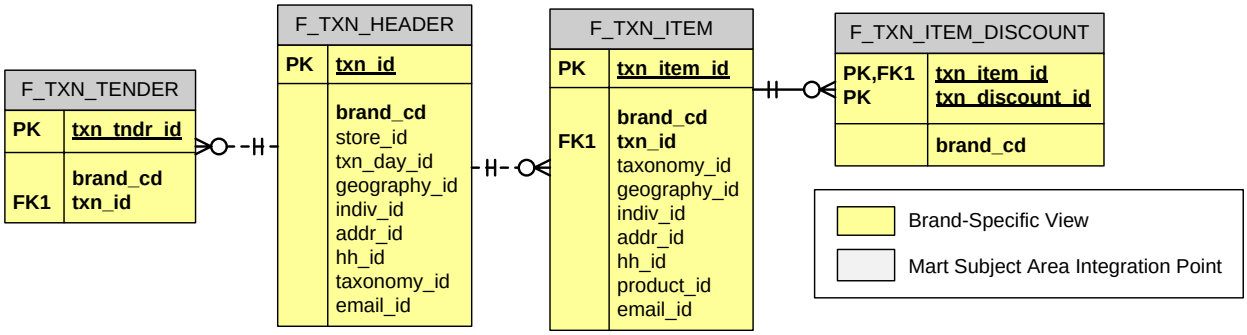
	geographic level attributes can be appended consistently to each member of that household or address. Unique clients making a transaction in the past 36 months will be selected for append. New and changed records will be extracted quarterly with an annual refresh of the past 36 month clients. Licensing of is for 12 month periods. Therefore, cleanup will be necessary on records no longer part of the past36M client append records.
BU21.2.1	To perform the initial full and refresh extract, The POSTAL_CONTACT table is inner joined to TXN_HEADER through ACCOUNT (bill to) to determine activity. If the calendar month of the transaction is >= current calendar month – 36 the POSTAL CONTACT is included for extraction
BU21.2.2	To perform the quarterly extract the same join is used to determine 36-month activity, however the universe is further limited by only including postal contacts that have been created/updated since the latest overlay process executed. This is done by referencing the FILE_DETAIL table where SOURCE_CD = 'KBM' and FILE_TYPE_CD = 'DEMOGRAPHIC' and extracting only where the Postal Contact DW Create/Update date > File Detail FEED_DT
	DEMOGRAPHIC data is presented in the Data Mart layer as the overlay data related to the BEST postal contact
BU21.3	Email Aggregate Overlay: The system will retain the aggregate fields that are provided by CheetahMail for each email address at the subscriber list level to summarize latest promotion/activity data.
BU21.3.1	The Email Aggregates are performed by SDL and upserted directly into the EMAIL_LIST_AGGREGATE table by email ID, List ID (PID), Brand Code (decoded PID) and Fiscal Month.
BU21.3.2	The granular aggregates are provided in the Data Mart as a dimension of BRAND_EMAIL called LU_EMAIL_LIST_AGGR

Transaction Subject Area

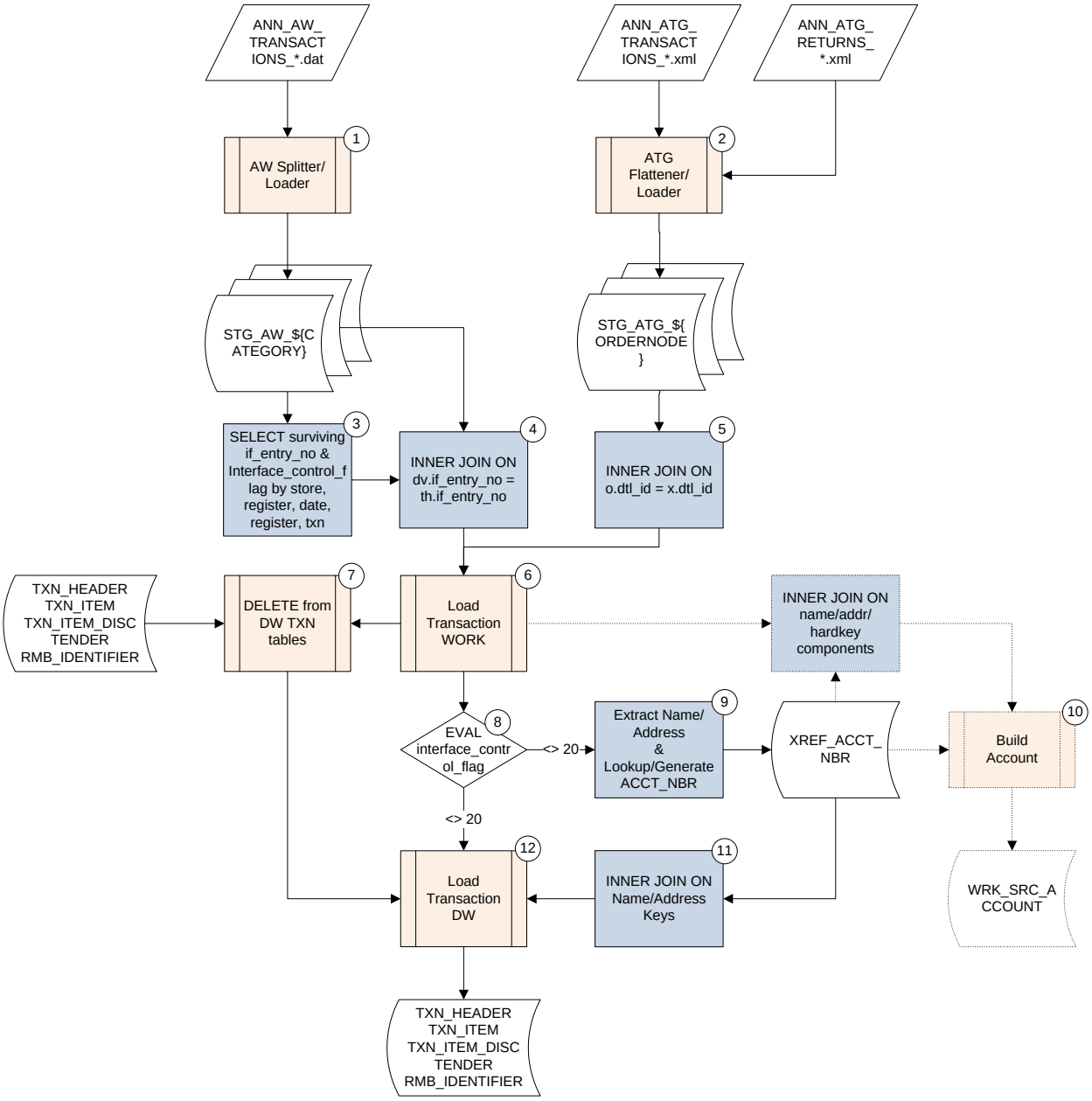
Transaction Entity Key Relationships – Data Warehouse



Transaction Entity Key Relationships – Data Mart



Transaction Data Warehouse Update Process Flow



Transaction Data Warehouse Update Process Annotations

1. The AuditWorks load process receives and splits the feed based on the RECORD_TYPE. Each file is then staged independently:
    a. A = Authorization detail = STG_AW_AUTH_DTL
    b. C = Customer detail = STG_AW_CUST_DTL
    c. D = Discount detail = STG_AW_DISC_DTL
    d. E = Customer Extra Info detail = STG_AW_EXPCUST_DTL
    e. H = Transaction Header = STG_AW_TXN_HEADER
    f. L = Transaction Detail = STG_AW_TXN_LINE
    g. M = Merchandise detail = STG_AW_MERCH_DTL
    h. N = Line notes = STG_AW_LINE_NOTES
    i. O = Special Order detail = not loaded
    j. P = Payroll detail = not loaded
    k. R = Return detail = STG_AW_RETURN_DTL
    l. S = Stock Control detail = not loaded
    m. T = Tax Override detail = not loaded
    n. V = Post-Void detail = not loaded
2. The ATG load process receives and flattens the XML feed creating delimited files including identifiers that are used to reassemble the hierarchy:
    a. STG_ATG_ORDER
        i. STG_ATG_RELATED_ORDERS
        ii. STG_ATG_PRICE_INFO
            1. STG_ATG_ORDER_PRICE_INFO
        iii. STG_ATG_COMMERCE_ITEM
            1. STG_ATG_PRICE_INFO
                a. STG_ATG_ITEM_PRICE_INFO
                b. STG_ATG_DTL_ITM_PRICE_INFO
        iv. STG_ATG_PAYMENT_GROUP
            1. STG_ATG_CREDIT_CARD
                a. STG_ATG_BILLING_ADDRESS
        v. STG_ATG_SHIPPING_GROUP
            1. STG_ATG_HARDGOOD_SHIP_GRP
                a. STG_ATG_SHIPPING_ADDRESS
    b. STG_ATG_RETURN_REQUEST
        i. STG_ATG_RETURN_ITEM
        ii. STG_ATG_REFUND_METHOD
3. For AuditWorks, we receive multiple versions of a transaction throughout its lifecycle. This may be within a single feed or updates to a transaction over time. Each load the

process creates a DRIVER table that is used to identify the final/most recent version of a transaction to apply in the update process. This is done by sorting the IF_ENTRY_NO within each Transaction ID (Store, Register, Date, AW Transaction ID) and selecting the IF_ENTRY_NO with the Maximum value. This table serves two purposes

   a. As we build out each transaction table – the surviving version is applied to each resulting structure's source data
      i. TXN_HEADER
      ii. TXN_ITEM
      iii. TXN_ITEM_DISCOUNT
      iv. TENDER
      v. RMB_IDENTIFIER
   b. The process uses the previously applied IF_ENTRY_NO for a transaction and verifies that the inbound value is greater than/more recent than that which is already loaded into the Data Warehouse. This insures that we do not overwrite more recent transaction data with older data that was inadvertently provided to the udpate
   c. The process uses the INTERFACE_CONTROL_FLAG on the surviving version to determine whether or not to load the transaction
      i. 10 – Initial – Load to DW
      ii. 20 – Reversal – Do NOT load to DW
      iii. 30 – Correction – Load to DW

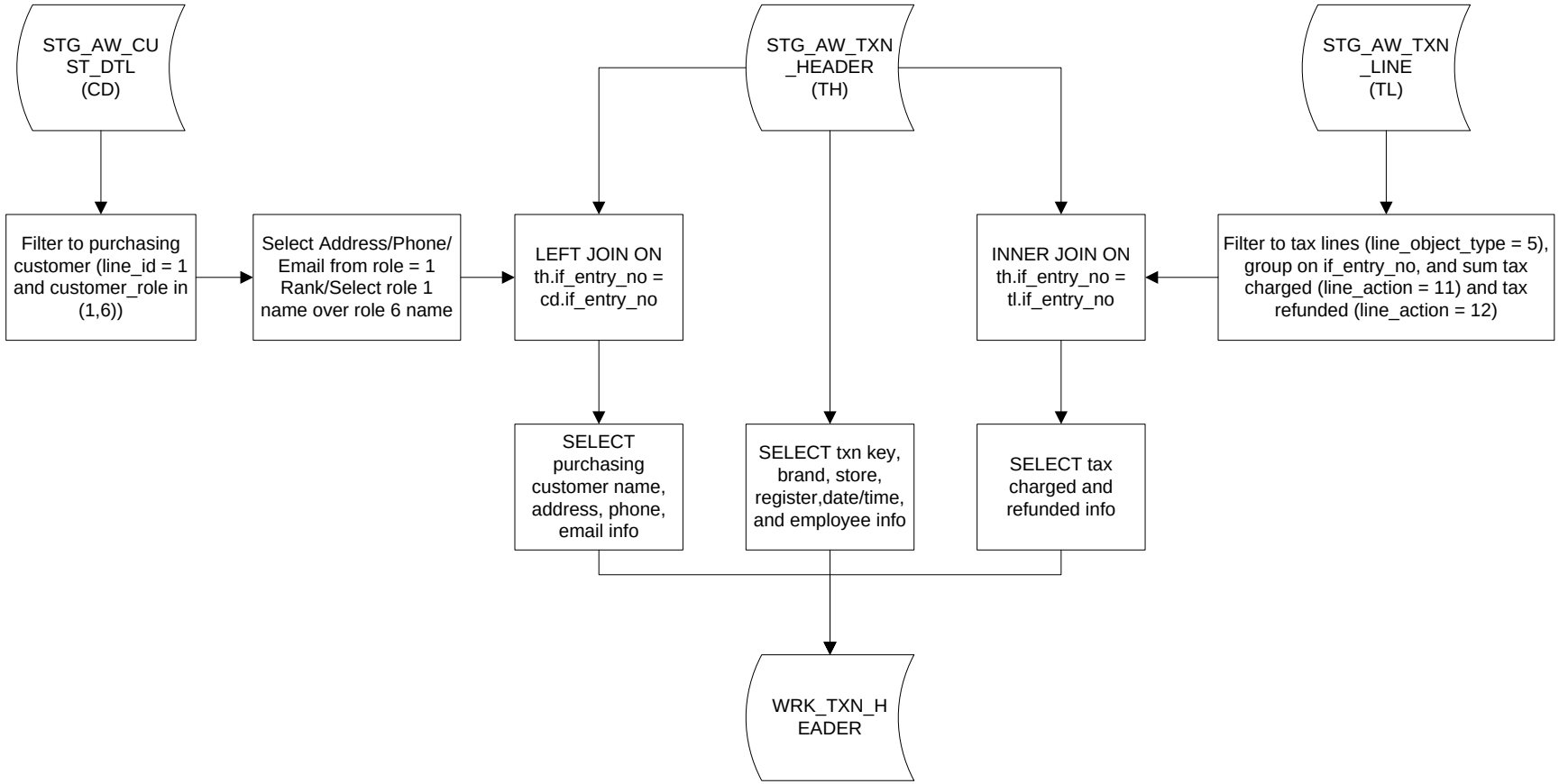
4. The original AuditWorks data is joined on the most surviving IF_ENTRY_NO for each transaction to load only the most recent transaction (as described in 3a above)
5. The ATG transaction nodes are reassembled through a series of hierarchical join keys to built the appropriate Transaction Source Data for load into the Data Warehouse. At this time the RETURNS data is also loaded to later apply updates to the TRANSACTIONS data.
6. A set of WRK tables mirroring the Data Warehouse structures is loaded to consolidate all incremental transactions to be loaded
   a. AuditWorks POS Orders – surviving version only
   b. ATG E-Commerce Orders – all transactions
   c. ATG E-Commerce Returns – all transactions
7. With all incremental transactions identified, the Data Warehouse is wiped of all transaction data for any inbound Transaction Header by order ID. This is applied throughout the transaction hierarchy described in Step 3a. Prior to wiping the transaction data from the Data Warehouse the ATG inbound activity_dt (submitted date) is compared to that in the target transaction header to insure the inbound data is more recent. (note: AuditWorks transactions address this in step 3b)
   a. ATG – Order ID
   b. Auditworks – Store, Register, Txn Date, Txn ID
8. For AuditWorks the surviving version's INTERFACE_CONTROL_FLAG is evaluated as described in Step 3c. All flags except 20 (reversal) are loaded/reloaded to the Data Warehouse. Effectively this performs a full refresh of each transaction header and associated items, tenders, discounts, and Retail Matchback identifiers. It also purges cancelled transactions from the database.

9. The Bill and Ship name and address data is extracted from the transaction and referenced in a cross-reference table by the exact unique combination of name, address, and hardkey components to determine if that name and address already exists in the solution as an account. If so the ACCT_NBR is returned and the transaction is updated with the existing account (and by associated to the Postal Contact). If not, the name and address components are loaded into the XREF and an ACCT_NBR is assigned and applied to the transaction. Where multiple names/addresses exist on a transaction (i.e. ATG), the XML hierarchy is traversed to identify the first node instance containing enough information to identify an Individual: Last Name, Address Line 1, and either Postal Code or City and State.
10. The same ACCT_NBR assigned to transactions is used to build WRK_SRC_ACCOUNT which is ultimately used in the AGILITY process to load the following tables:
    a. ACCOUNT
    b. ACCOUNT_EMAIL
    c. ACCOUNT_HARD_KEY
    d. ACCOUNT_PHONE
    e. BRAND_EMAIL
    f. EMAIL
    g. INDIVIDUAL
    h. PHONE
    i. POSTAL_CONTACT
    j. POSTAL_CONTACT_ARCHIVE
    k. POSTAL_CONTACT_STORE
11. The inbound transaction data is updated with the ACCT_NBR identified with the name and address components to define both the BILL TO and SHIP TO relationships
12. The incremental AW and ATG transaction data is inserted into the Data Warehouse transactions tables.

AuditWorks TXN_HEADER Data Warehouse Update Process Flow

Assume that the incoming file feed loaded to the initial staging tables has already limited transactions to:

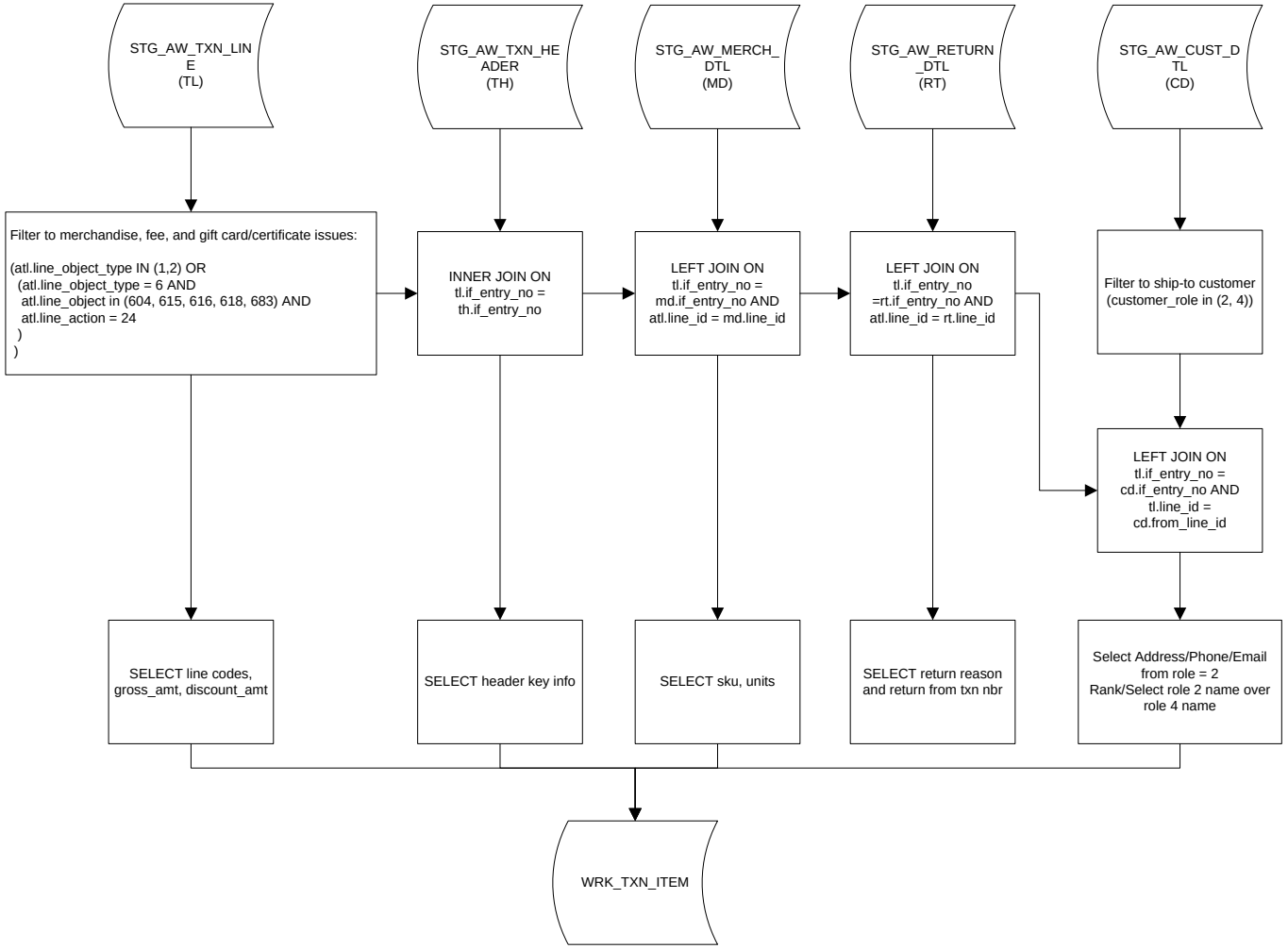
- category 1 (customer)
- non-voided transactions with an interface control flag of 10 (original transaction) or 30 (corrected transaction)
- maximum(IF_CONTROL_NO) within store, register, date, txn_id (most current version of transaction) has been selected
- retail store transactions only (filtering E-Commerce by store_no in ('611', '612', '618', '619'))



AuditWorks TXN_ITEM Data Warehouse Update Process Flow

Assume that the incoming file feed loaded to the initial staging tables has already limited transactions to:

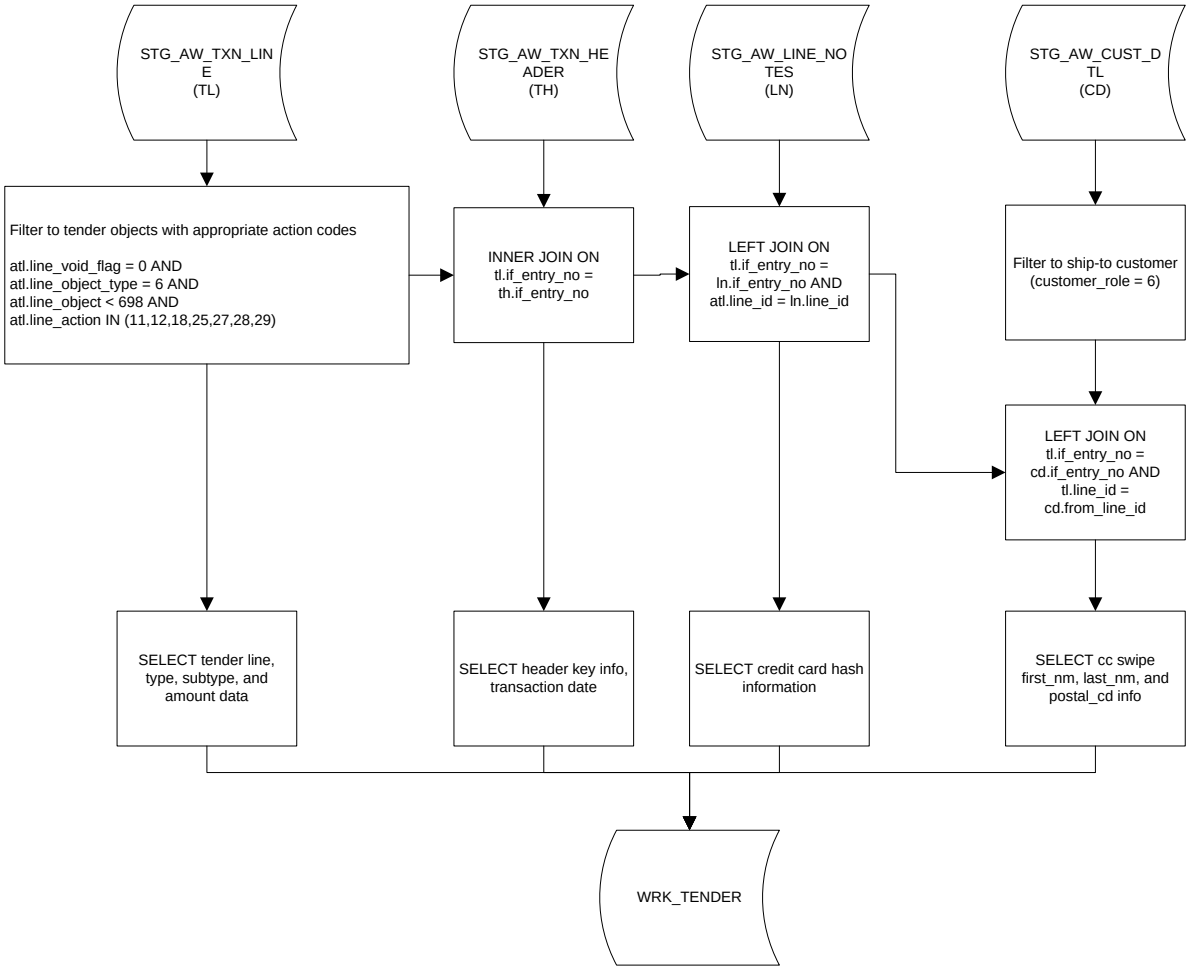
- category 1 (customer)
- non-voided transactions with an interface control flag of 10 (original transaction) or 30 (corrected transaction)
- maximum(IF_CONTROL_NO) within store, register, date, txn_id (most current version of transaction) has been selected
- retail store transactions only (filtering E-Commerce by store_no in ('611', '612', '618', '619'))



AuditWorks TENDER Data Warehouse Update Process Flow

Assume that the incoming file feed loaded to the initial staging tables has already limited transactions to:

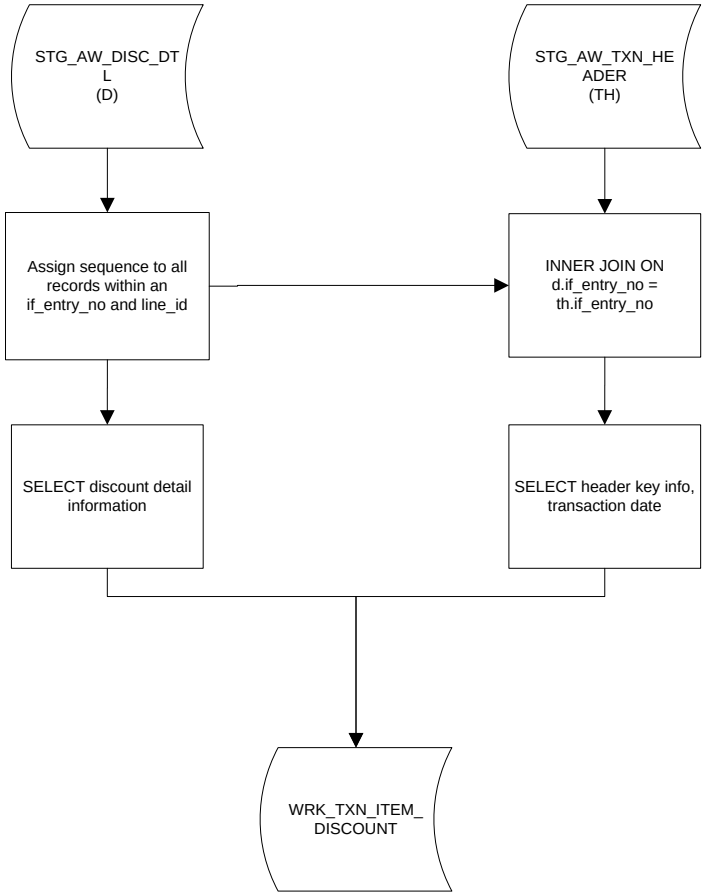
- category 1 (customer)
- non-voided transactions with an interface control flag of 10 (original transaction) or 30 (corrected transaction)
- maximum(IF_CONTROL_NO) within store, register, date, txn_id (most current version of transaction) has been selected
- retail store transactions only (filtering E-Commerce by store_no in ('611', '612', '618', '619'))



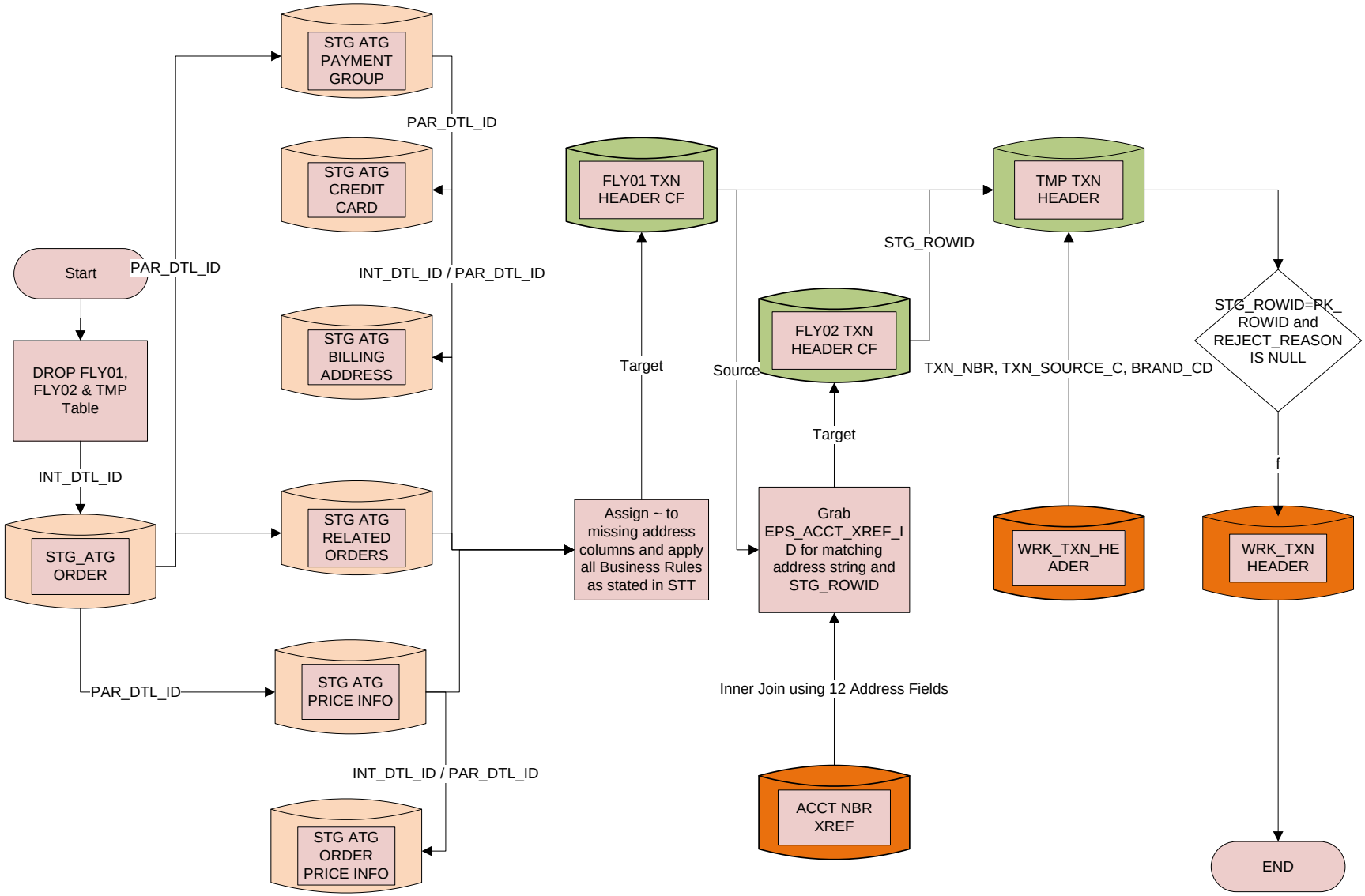
AuditWorks TXN_ITEM_DISCOUNT Data Warehouse Update Process Flow

Assume that the incoming file feed loaded to the initial staging tables has already limited transactions to:

- category 1 (customer)
- non-voided transactions with an interface control flag of 10 (original transaction) or 30 (corrected transaction)
- maximum(IF_CONTROL_NO) within store, register, date, txn_id (most current version of transaction) has been selected
- retail store transactions only (filtering E-Commerce by store_no in ('611', '612', '618', '619'))



ATG TXN_HEADER Data Warehouse Update Process Flow



22. Transaction

Transaction data is being collected from the AuditWorks POS system and ATG Ecommerce fulfillment system that supports call center and website.

#	Design
BU22.1	The system will track monetary interactions with ANN Inc regardless of data source as transactions.
BU22.1.1	Transactions from AuditWorks, ATG, and Marketworks source systems are loaded into standardized Data Warehouse structures • TXN_HEADER • TXN_ITEM • TXN_ITEM_DISCOUNT • TENDER
BU22.1.2	Values at the TXN_HEADER level are loaded when provided by the source and used only for validation purposes of the line level data, except where the source system only supports a particular value at the header level (i.e. ATG tax). These specific rules are defined in the Source to Target mapping documents.
BU22.1.3	Values at the TXN_ITEM, TENDER, and TXN_ITEM_DISCOUNT are derived to arrive at a consistent representation of varying source values. The specific formulas for these transformations are defined in the Source to Target Mapping documents.
BU22.2	The sources for transaction data are: • ATG (ongoing Ecommerce transactions and historical Ecommerce transactions back to 1/2009) • AuditWorks (ongoing retail transactions) • MarketWorks (historical Ecommerce transactions prior to 1/2009 and historical retail transactions including both identified and unidentified transactions)
BU22.2.1	Transactions can be associated with their originating system by in the TXN_SRC_CD • AW – Auditworks • ATG – ATG • MW - MarketWorks
BU22.2.2	The native unique transaction for each source is retained in the TXN_NBR field as follows • AW – TO_CHAR(STG_AW_TXN_HEADER.STORE_NO) || ':' || STG_AW_TXN_HEADER.REGISTER_NO || ':' || STG_AW_TXN_HEADER.TRANSACTION_DATE in YYYYMMDD format || ':' || STG_AW_TXN_HEADER.TRANSACTION_NO • ATG – STG_ATG_ORDER.ORDER_ID • MW – TRANSACTION_HEADER.TRANSACTION_ID • MW – TI_TRANSACTION_HEADER_ERROR.TRANSACTION_ID
BU22.3	The system will load the migration Ecommerce data by loading ATG (1/2009 to current) then loading MarketWorks Ecommerce data prior (1/2009).
BU22.3.1	ATG will not be providing any migration data. The data gap between MW and ATG (and AW) is closed through the load of incremental “catch-up” feeds.
BU22.4	The MarketWorks system stores Ecommerce transactions based on their final ship date. The ATG Ecommerce system will send historical transactions based on submit date. Thus, regardless of cutoff date, there will be an overlap of MarketWorks and historical ATG Ecommerce transactions. The migration process will attempt to identify as many of the overlapping transactions as possible and keep the version supplied by ATG.

BU22.4.1	ATG migration data has been determined to be out of scope.
BU22.5	The system assumes that all components of a transaction (header, lines, tender, discounts, and alternate identifiers) are received as a complete unit in an update. During an update, if the system encounters a transaction (based on the transaction key) that is already in the warehouse, then the warehouse version of the transaction and all related components will be deleted and replaced with the new version supplied in the update.
BU22.5.1	All transaction data based on the keys defined in **BU22.2.2** are completely deleted from the Data Warehouse prior to inbound incremental load: • TXN_HEADER • TXN_ITEM • TXN_ITME_DISCOUNT • TENDER • RMB_IDENTIFIER This means that if an order has 3 lines on day 1, and day 2 we receive the same order key with only two lines, the resulting transaction in the Data Warehouse has 2 lines.
BU22.6	All Canadian monetary values will be converted to US dollars by the source system extracts. The system assumes all US transactions are in US dollars and all Canadian transactions will be converted into US dollars.
BU22.6.1	All monetary values will be stored in their native currency in the data warehouse. As part of a future, post go-live enhancement to the Data Mart build, the system will join to a currency conversion table by currency code and order date to apply the appropriate conversion to US dollars. If the daily conversion rate is not found in the conversion table, the most recent conversion value-will be used to migrate data to the mart.
BU22.7	For use in the retail matchback process, the system will maintain the following alternate identifiers pertinent to a transaction if available on the source: • ADS Account ID • Credit card surrogate hash and first and last name for the primary credit card used on the transaction • Email address plus first and last name from the primary credit card • Coupon Barcode
BU22.7.1	The RMB_IDENTIFIER table stores the transaction used to account identification. These types are defined in the RMB_ID_TYPE_CD • ADS = ADS PLCC/Cobrand • CC = Non-ADS CC Hash • EMAIL = EMAIL – used for REA • ZIP = Name/Zip – NZA/TAA • BC = IndivID from Barcode
BU22.7.2	The value of the aforementioned RMB_ID_TYPE_CD is retain in RMB_ID_VALUE as defined in the STT documents
BU22.8	The system will load the migration tables Email_POS_Arch and Email_POS_Arch3 to the transaction_alternate_id table.
BU22.8.1	Epsilon performs a join on the EMAIL_POS_ARCH data to the TI_TRANSACTION_TENDER_ERROR feed. The result is manifested as additional fields on the TI_TRANSACTION_TENDER_ERROR feed. The query takes each tender for an unidentified transaction and attempts to link it to the table containing the name and zip code from the credit card swipe for each MW hashed credit card number. All tenders will be pulled regardless of whether or not a match if found. The inline view on the right hand side of the join chooses the most recent credit card swipe value to retrieve the name and zip when there are multiple records for a single hash. The

	results of this query, if run on the ANN side, will have to be run through a conversion where the hashed credit card number is decrypted back to native from and then rehashed using the common algorithm prior to sending to Epsilon. This query extends the tender error feed and eliminates having to decrypt/encrypt the large email_pos_arch table
BU22.9	The system will only process non-voided, customer transactions for retail stores from AuditWorks.
BU22.9.1	As described in the Transaction Data Warehouse Update Flow, all transactions are deleted and (re)inserted by transaction key.
BU22.9.2	In AuditWorks the load process identifies the most current version of the transaction by selecting the MAX(IF_ENTRY_NO) within Store, Register, Date, TxnID
BU22.9.3	If the INTERFACE_CONTROL_FLAG for that transaction version is equal to "20" (reversal), then the transaction data is NOT loaded into the Data Warehouse, otherwise the TXN_HEADER, TXN_ITEM, TXN_ITEM_DISCOUNT, TENDER, RMB_IDENTIFIER data is loaded based on the selected IF_ENTRY_NO (transaction version)
BU22.10	The system will only receive from AuditWorks record types (H,L,M,D,R,A,C,E,N)
	The audit works split process creates feeds based on the RECORD_TYPE. These are loaded and used in the STG to WRK update process to generate Transactions and Accounts: H = Transaction Header = STG_AW_TXN_HEADER L = Transaction Detail = STG_AW_TXN_LINE M = Merchandise detail = STG_AW_MERCH_DTL D = Discount detail = STG_AW_DISC_DTL R = Return detail = STG_AW_RETURN_DTL A = Authorization detail = STG_AW_AUTH_DTL C = Customer detail = STG_AW_CUST_DTL E = Customer Extra Info detail = STG_AW_EXPCUST_DTL N = Line notes = STG_AW_LINE_NOTES
BU22.10.2	These RECORD_TYPES are also split out, but are not loaded or included in the solution in any way: O = Special Order detail = not loaded P = Payroll detail = not loaded S = Stock Control detail = not loaded T = Tax Override detail = not loaded V = Post-Void detail = not loaded
BU22.11	The system will only process AuditWorks header records (record type H) to those with a transaction category value of 1 for "Customer", all others will be excluded as well the child line records.
BU22.11.1	The type 1 records contain customer data, which is typically a name and email address
BU22.11.2	It has been later determined that type 6 records contain the TENDER credit card swipe data containing a name and address data, these are also included as BILL TO account source names
BU22.11.3	To reconcile these multiple occurrences of viable name and address/email across these types they are ranked according to the following for name selection 1. If the CUSTOMER (1) first name is NOT NULL then select the CUSTOMER Name 2. Else, select TENDER (6) name

<table>
<tr><td>BU22.11.4</td><td>The postal address, email, and phone data is always selected from the CUSTOMER(1) type record</td></tr>
<tr><td>BU22.12</td><td>The system will count a transaction in frequency calculations when a Client interaction generates an overall positive dollar result. This will be the rule applies to all brands.</td></tr>
<tr><td>BU22.12.1</td><td>The Data Mart F_TXN_HEADER.TXN_TYPE_CD field contains the following values, that can be used to determine which records count for frequency
<table>
<tr><th>Value</th><th>Rule</th></tr>
<tr><td>PUR</td><td>TXN_HEADER.TXN_NET_AMT > 0</td></tr>
<tr><td>EXC</td><td>TXN_HEADER.TXN_NET_AMT = 0</td></tr>
<tr><td>RTN</td><td>TXN_HEADER.TXN_NET_AMT < 0</td></tr>
</table></td></tr>
<tr><td>BU22.12.3</td><td>The Individual and Household summary rollups will use this field for deriving these fields:
<table>
<tr><td>12 Month Frequency</td></tr>
<tr><td>13-24 Month Frequency</td></tr>
<tr><td>Lifetime Frequency</td></tr>
<tr><td>12 Month Frequency - RET</td></tr>
<tr><td>13-24 Month Frequency - RET</td></tr>
<tr><td>Lifetime Frequency - RET</td></tr>
<tr><td>12 Month Frequency - WEB</td></tr>
<tr><td>13-24 Month Frequency - WEB</td></tr>
<tr><td>Lifetime Frequency - WEB</td></tr>
</table></td></tr>
<tr><td>BU22.12.4</td><td>The outbound feeds do not perform any aggregated transaction rollups (only low-level transaction or LAST transaction rollups).
• ADS – Header & Detail
• SDL Profile – Last Transaction and dollar rollups
• SDL Transaction – Header & Detail by brand

All of these extracts include all transactions and are not limited by any “true order” indicator</td></tr>
<tr><td>BU22.13</td><td>The system will count transaction frequency based on raw frequency, multiple transactions within the same day/same channel will be counted as separate transactions</td></tr>
<tr><td>BU22.13.1</td><td>The Data Warehouse and Mart maintain all (non-reversed) transactions including Purchases, Exchanges, and Returns. All are eligible for order counting.</td></tr>
<tr><td>BU22.13.2</td><td>The transaction primary key for each source is defined that multiple transactions within a single channel would store multiple transactions:
• ATG – Order ID
• Auditworks – Store, Register, Txn Date, Txn ID
• MarketWorks – Transaction_ID</td></tr>
</table>

23. Transaction Header

<table>
<tr><th>#</th><th>Design</th></tr>
<tr><td>BU23.1</td><td>The system will maintain a transaction header for each transaction that contains information that is common for all elements of the transaction.</td></tr>
<tr><td>BU23.1.1</td><td>The AuditWorks TXN_HEADER table is derived from the following record types:
<table>
<tr><td>STG_AW_CUST_DTL</td></tr>
<tr><td>STG_AW_TXN_HEADER</td></tr>
<tr><td>STG_AW_TXN_LINE</td></tr>
</table></td></tr>
<tr><td>BU23.1.2</td><td>The ATG TXN_HEADER table is derived from the following XML nodes:
<table>
<tr><td>STG_ATG_BILLING_ADDRESS</td></tr>
<tr><td>STG_ATG_CREDIT_CARD</td></tr>
<tr><td>STG_ATG_ORDER</td></tr>
</table></td></tr>
</table>

	STG_ATG_ORDER_PRICE_INFO STG_ATG_PAYMENT_GROUP STG_ATG_PRICE_INFO
BU23.1.3	The MarketWorks TXN_HEADER table is derived from the following source tables: TI_TRANSACTION_HEADER_ERROR TRANSACTION_HEADER
BU23.1.4	The low-level fact data is made available in the Data Mart as F_TXN_HEADER
BU23.2	The system will track a standardized channel for each transaction header. Valid channel values are 'RET' for retail and 'WEB' for Ecommerce.
BU23.2.1	The AuditWorks TXN_HEADER.TXN_CHANNEL_CD is defined as The literal "WEB"
BU23.2.2	The ATG TXN_HEADER.TXN_CHANNEL_CD is defined as The literal "RET"
BU23.2.3	The MarketWorks TXN_HEADER.TXN_CHANNEL_CD is defined as 'if store in (611,612,618,619) then WEB else RET
BU23.3	The system will maintain a standardized transaction type at the transaction header level. Transaction types are defined as: • 'PUR' – Ecommerce orders and retail transactions with positive net transaction amount • 'RTN' – Ecommerce returns and retail transactions with negative net transaction amount • 'EXC' – Ecommerce exchange transactions and retail transactions with zero net transaction amount
BU23.3.1	The TXN_HEADER.TXN_TYPE_CD is derived as described in **BU22.12.1**
BU23.3.2	The dependant TXN_NET_AMT attribute varies by source • AW – STG_AW_TXN_HEADER.TENDER_TOTAL • ATG – STG_ATG_PRICE_INFO.AMOUNT • MW – TRANSACTION_HEADER.TOTAL_NET_RETAIL
BU23.4	A transaction header will be uniquely identified by the combination of brand and transaction number • The transaction number for retail transactions will consist of the combination of store number, register number, transaction date and a sequence number. Note: There may be duplicate transactions in AW, Epsilon will need to determine in development if all the sub line data is an exact duplicate and if so, then the duplicates can be ignored. If not all data is duplicated a business rule will need to be determined to keep one record over the other. • The transaction number for transactions loaded from ATG will be the ATG order id. • The transaction number for Ecommerce transactions loaded from MarketWorks prior to the ATG cutoff will be the MarketWorks transaction_id.
BU23.4.1	AuditWorks TXN_HEADER.TXN_NBR is defined as: TO_CHAR(AW_TXN_HEADER.STORE_NO) || ':' || AW_TXN_HEADER.REGISTER_NO || ':' || AW_TXN_HEADER.TRANSACTION_DATE in YYYYMMDD format || ':' || AW_TXN_HEADER.TRANSACTION_NO
BU23.4.2	Auditworks TXN_HEADER.BRAND_CD is defined as: need to link STG_AW_TXN_HEADER.STORE_NO to SAP_STORE.STORE_NO to get brand which is defined as: if STG_OWNER.STG_SAP_STORE.BRAND in ('02', '03'), then 'AT', if STG_OWNER.STG_SAP_STORE.BRAND = ('05', '06') then 'LOFT', if STG_OWNER.STG_SAP_STORE.BRAND = '04', then 'ATF',

	if STG_OWNER.STG_SAP_STORE.BRAND = '07', then 'LOS', else 'ENT'
BU23.4.3	Duplicate transaction are included in the feed the surviving version is selected by the AuditWorks the load process identifying the most current version of the transaction through selecting the MAX(IF_ENTRY_NO) within Store, Register, Date, TxnID
BU23.4.4	ATG TXN_HEADER.TXN_NBR is defined as STG_ATG_ORDER.ORDERID
BU23.4.5	ATG TXN_HEADER.BRAND_CD is defined as if siteId = 611 or 619 then AT else if 612 then LOFT else ENT
BU23.4.6	MarketWorks TXN_HEADER.TXN_NBR is defined as TRANSACTION_HEADER.TRANSACTION_ID
BU23.4.7	MarketWorks TXN_HEADER.BRAND_CD is defined as Join to STORE and DECODE(division_code,'10','AT','20','LOFT', '30','ATF','40','AT', '50','LOFT','60','LOS','ENT')
BU23.5	The system will link a transaction header to the contact information for the individual fiscally responsible for the transaction through a billing account key consisting of brand code, billing account source, and billing account number.
BU23.5.1	The BILL_ACCT_NBR and BILL_ACCT_SRC_CD are used to link to the same name and address/email data mapped into the WRK_SRC_ACCOUNT for eventual load to the Account and Postal Contact tables • AW – this is derived from the CUSTOMER_ROLE = 1 (bill to) and CUSTOMER_ROLE = 6 (tender), taking the address/phone/email from the bill to. And ranking 1 over 6 to select the best available name data • ATG – this is derived from the BILLINGADDRESS node by cycling through in sequential order and ranking/selecting the first contact with the minimum address requirements for Agility processing (with postal data selected over email only data) • MW – this is the MW customer_id
BU23.5.2	The mart would resolve each Account through the Agility-derived Postal Contact to define the INDIVIDUAL linked to transactions
BU23.5.2	In the event, that a transaction does not have enough identifiable information to accurately associate it with an Account the transaction is deemed unidentified and orphaned in the database. At this time it becomes eligible for the Retail Matchback process.
BU23.6	In the data warehouse, the system will only track monetary values at the transaction header level that are provided by all the sources directly at that level or exist only at that level (such as tax amounts). The mart process will roll up any required metrics that can be derived from the transaction lines. • The system will track tax amount and refunded tax amount at the transaction header level.
BU23.1.1	The Data Warehouse TXN_HEADER table maintains only native and derived fields that are available at the source system at that level (as described in the **STG to DW STT document**)
BU23.1.2	The Data Mart TXN_HEADER table maintains a number of fields derived from the TXN_ITEM level and representing the complete order comprehensively. These rules are described in the **DW to DM STT document.**
BU23.7	AuditWorks transactions potentially have multiple versions that track through the auditing process including an original version, a version that reverses all amounts on the original version, and a final version. The system will process these versions in such a manner that the final version of the transaction is reflected in the data warehouse.
BU23.7.1	A transaction version is defined by the IF_ENTRY_NO field. It is a non-consecutive numeric sequence that is used to sequence each revision to a transaction.
BU23.7.2	The INTERFACE_CONTROL_FLAG is used to determine type of revision applied to the version:

	• 10 = Initial. This is the initial transaction, it is inserted into the DW transaction tables • 20 = Reversal. This is a voided transaction. It is not loaded to the DW. Any previous transaction loaded is deleted from the DW. • 30 = Correction. This is the corrected version of the transaction. It replaces any previous version of the transaction.
BU23.7.3	A transaction may have multiple version's provided within a single feed. Including an initial and any number of reversals and corrections in any sequence. To address this, all daily feeds are staged and sorted by IF_ENTRY_NO within Store, Register, Date, and TxnID. The maximum value is processed based on its INTERFACE_CONTROL_FLAG. All other versions are rejected by the load process.
BU23.7.4	The assumption is that versions are always provided in sequential order such that AuditWorks will not provide an older (lesser IF_ENTRY_NO) on a subsequent feed than what had been previously loaded into the Data Warehouse.
BU23.8	If an account containing postal contact information is assigned to a transaction via the retail matchback process, the system will track the retail matchback method, vendor, and date of application for the transaction.
BU23.8.1	The retail matchback process includes both internal and external methods of matchback. An internal method is defined as taking an unidentified transaction and using the associated keys contained in the RMB_IDENTIFIER table look for other accounts and/or identified transactions that have the same key value. If a match occurs, the system will replace the native bill account information on the transaction with the matching account information. For external matchback methods such as reverse email append, name zip append, and trade area append, the system creates a new account with the information returned from the vendor via the ACCOUNT update process and that account information replaces the former bill-to account information on the transaction header. The system tracks the replaced account information in the RMB_POOL_MEMBER table in case the process ever needs to be reversed. On the transaction header, the system stores a code indicating the method used for matchback in the rm_method_cd, a code indicating which vendor provided the matchback information in the rm_vendor_cd, and the date that the matchback occurred in the rm_dt field. These fields are carried into the Data Mart so that ANN can generate matchback statistics.
BU23.8.2	The retail matchback method is maintained in the Data Warehouse and Data Mart TXN_HEADER.RM_METHOD_CD • ADS • CC • EM • CPN • NZA • REA • TAA
BU23.8.3	The retail matchback vendor is maintained in the Data Warehouse TXN_HEADER.RM_VENDOR_CD • EPS • EXP
BU23.8.4	The retail matchback application date is maintained in the Data Warehouse TXN_HEADER.RM_DT
BU23.9	The system will track the store number, register and cashier number within the store where the transaction occurred. These attributes will have default values for Ecommerce transactions.

<table>
<tr><td>BU23.9.1</td><td>AuditWorks attributes are mapped into these field to retain the native keys<table><tr><td>STORE_NBR</td></tr><tr><td>REGISTER_NBR</td></tr><tr><td>TXN_DT</td></tr><tr><td>TXN_SEQ_NBR</td></tr></table></td></tr>
<tr><td>BU23.9.2</td><td>The AuditWorks keys are also concatenated and mapped to the standardized TXN_NBR attribute which defines the transaction Primary Key for all sources and drives the EPS_TXN_HEADER_ID numeric surrogate key to enable fast joins in the Data Mart</td></tr>
<tr><td>BU23.9.3</td><td>ATG attributes are mapped into STORE_NBR (SITE_ID) and TXN_DT (SUBMITTEDDT).</td></tr>
<tr><td>BU23.9.4</td><td>The REGISTER_NBR and TXN_SEQ_NBR attributes are NULL for ATG</td></tr>
<tr><td>BU23.9.5</td><td>MarketWorks attributes are mapped from the source fields<table><tr><td>STORE_NO</td></tr><tr><td>REGISTER_NO</td></tr><tr><td>TRANSACTION_DATE</td></tr></table></td></tr>
<tr><td>BU23.9.6</td><td>The TXN_SEQ_NBR remains null for MarketWorks.</td></tr>
<tr><td>BU23.10</td><td>The system will track the date and time the transaction occurred. For Ecommerce transactions, this will correspond to the submitted date and time from the web site.</td></tr>
<tr><td>BU23.10.1</td><td>The TXN_DT field is defined as a DATE field in the Data Warehouse and Mart maintaining precision down to the second a transaction occurred (when available at the level within the source feed)<table><tr><th>Table</th><th>Source Field</th><th>Format</th></tr><tr><td>STG_AW_TXN_HEADER</td><td>ENTRY_DATETIME</td><td>YYYY-MM-DD</td></tr><tr><td>STG_ATG_ORDER</td><td>SUBMITTEDDATE</td><td>YYYY-MM-DD HH24:MI:SS</td></tr><tr><td>TRANSACTION_HEADER</td><td>TRANSACTION_DATE</td><td>YYYY-MM-DD</td></tr><tr><td>TI_TRANSACTION_HEADER_ERROR</td><td>TRANSACTION_DATE</td><td>YYYY-MM-DD</td></tr></table></td></tr>
<tr><td>BU23.10.2</td><td>The ATG completed date attribute is also stored in TXN_COMPLETE_DT</td></tr>
</table>

24. Transaction Item

#	Design
BU24.1	The system will maintain detailed line-items for a transaction that specify the products, fees or gift cards purchased by a Client and the financial amount associated to those items. This is referred to by the system as the transaction item level.
BU24.1.1	The AuditWorks TXN_ITEM table is derived from the following record types: STG_AW_CUST_DTL STG_AW_MERCH_DTL STG_AW_RETURN_DTL STG_AW_TXN_HEADER STG_AW_TXN_LINE
BU24.1.2	The ATG TXN_ITEM table is derived from the following XML nodes: STG_ATG_COMMERCE_ITEM STG_ATG_DTL_ITM_PRICE_INFO STG_ATG_HARDGOOD_SHIP_GRP STG_ATG_ITEM_PRICE_INFO STG_ATG_ORDER STG_ATG_SHIPPING_ADDRESS STG_ATG_SHIPPING_GROUP
BU24.1.3	The ATG TXN_HEADER table is also updated based on the following XML nodes: STG_ATG_RETURN_ITEM STG_ATG_RETURN_REQUEST
BU24.1.4	The MarketWorks TXN_HEADER table is derived from the following source tables: TI_TRANSACTION_DETAIL_ERROR TI_TRANSACTION_HEADER_ERROR TRANSACTION_COUPON TRANSACTION_DETAIL
BU24.1.5	The low-level fact data is made available in the Data Mart as F_TXN_ITEM
BU24.2	The system will maintain an identifier in conjunction with the transaction key that will uniquely identify a transaction item.
BU24.2.1	The physical primary key of the TXN_HEADER table is the TXN_NBR (whose source varies as described previously), BRAND_CD and TXN_SOURCE_CD (AW, ATG, MW). A unique numeric surrogate key is incremented for each combination of the above keys in the EPS_TXN_HEADER_ID field.
BU24.2.2	As transactions are wiped and reloaded (refreshed) at all levels prior to applying updates to them existing transactions retain their originally assigned EPS_TXN_HEADER_ID on subsequent updates.
BU24.2.3	The transaction identifier is inherited at all “lower” transaction levels with additional fields to define uniqueness at those levels • **TXN_HEADER** – BRAND_CD, TXN_SOURCE_CD, TXN_NBR • **TXN_ITEM** – BRAND_CD, TXN_SOURCE_CD, TXN_NBR, TXN_ITEM_NBR • **TXN_ITEM_DISCOUNT**– BRAND_CD, TXN_SOURCE_CD, TXN_NBR, TXN_ITEM_NBR, DISC_SEQ_NBR • **TENDER** – BRAND_CD, TXN_SOURCE_CD, TXN_NBR, TXN_SEQ_NBR • **RMB_IDENTIFIER** – BRAND_CD, TXN_SOURCE_CD, RMB_ID_TYPE_CD, RMB_ID_TYPE_VALUE
BU24.2.4	As the Data Mart fact tables are created the surrogate keys are applied at each level to enable single-key joins • F_TXN_HEADER ○ EPS_TXN_HEADER_ID (PK)

	• F_TXN_ITEM
	  ○ EPS_TXN_ITEM_ID (PK)
	  ○ EPS_TXN_HEADER_ID (FK)
	• F_TXN_ITEM_DISCOUNT
	  ○ EPS_TXN_ITEM_ID (FK)

BU24.3	Transaction items can pertain to specific products or non-inventoried items such as fees for services or gift cards. The system will maintain a linkage to a product master to identify a product referenced on a transaction item. In addition, the system will identify if a transaction item is for a fee or gift card.
BU24.3.1	The linkage to the PRODUCT table is performed from TXN_ITEM on the SKU field
BU24.3.2	For AuditWorks this is sourced from and joined to the SAP SKU field STG_AW_MERCH_DTL SKU_ID
BU24.3.3	For ATG this is sourced from and joined to the SAP SKU field STG_ATG_COMMERCE_ITEM CATALOGREFID
BU24.3.4	For MarketWorks the join is performed through a MW to SAP product XREF: Select * from ann_migrn_user.transaction_detail a inner join ann_migrn_user.transaction_detail_xref c on a.transaction_id = c.transaction_id and a.transaction_line_no = c.transaction_line_no inner join dw_owner.product d on c.product_code = d.sku
BU24.3.5	For MarketWorks transaction failing the join to SAP, they are defaulted back to a “SKU”s loaded into the PRODUCT table specifically for migration purposes (STYLE_ID || ":' || COLOR_CD || ":" || SIZE_DESCRIPTION) join to ANN_MIGRN_USER.PRODUCT to obtain SKU
BU24.3.5	Fees and Gift cards are identified through the LINE_OBJECT_TYPE field: • 1 = merchandise • 2 = fee • 4 = gift card sale • 5 = sales tax
BU24.3.6	ATG does not support Gift Card Sale or Fee identifiers (apart from Shipping/Handling fees)
BU24.3.7	MarketWorks does not support Gift Card Sale or Fee identifiers
BU24.4	The system will store the following non-voided line data from AuditWorks as transaction items: • Merchandise Lines • Fee Lines • Tender lines representing the issuance of gift certificates, electronic gift certificates, internet SVS gift cards, SVS gift cards, wardrobe gift certificates, and reward certificates. Note: Epsilon will not store gift card numbers
BU24.4.1	AuditWorks transaction line type identifiers are stored in the LINE_OBJECT_TYPE field as described in **BU24.3.5**
BU24.4.2	The system generates “SKU” values for each of the tender issuances supported by AuditWorks. These are loaded into the data warehouse product table and relate to transactions via the TXN_ITEM table.

BU24.4.3	Gift Card Numbers are provided in the Line Notes record. The note for Gift Card numbers is not mapped into the solution.
BU24.5	The system will track the original AuditWorks line object type, line object, and line action for the transaction. • For line level records Ann Inc. will limit Auditworks records to line_object_type values of 1-Merchandise, 2-Fee, 4-Gift Certificate Sale, 5-Sales Tax, 6-Tender
BU24.5.1	The AuditWorks line object type, object, and action attributes are retained in LINE_OBJECT_TYPE_CD LINE_OBJECT_CD LINE_ACTION_CD
BU24.5.2	Line Object Types other than 1, 2, 4, 5, or 6 are rejected as part of the transaction build process.
BU24.6	The system will link a transaction item to the contact information for the individual receiving the transaction item through a shipping account key consisting of brand code, ship account source, and ship account number if ship-to information is available on the source.
BU23.5.1	The SHIP_ACCT_NBR and SHIP_ACCT_SRC_CD are used to link to the same name and address/email data mapped into the WRK_SRC_ACCOUNT for eventual load to the Account and Postal Contact tables • AW – this is derived from the CUSTOMER_ROLE = 2 (ship to) and CUSTOMER_ROLE = 4 (gift recipient), taking the address/phone/email from the bill to. And ranking 2 over 4 to select the best available name data • ATG – this is derived from the SHIPPINGADDRESS node by cycling through in sequential order and ranking/selecting the first contact with the minimum address requirements for Agility processing (with postal data selected over email only data) • MW – this is NULL (not available within MW)
BU23.5.2	The mart would resolve each Account through the Agility-derived Postal Contact to define the INDIVIDUAL linked to transactions
BU24.7	The system will track the quantity of the products purchased on the transaction item as the total sold quantity.
BU24.7.1	The total sold quantity is retained in ITEM_SOLD_QTY sourced from the following fields: • STG_AW_MERCH_DTL.UNITS CASE WHEN STG_AW_TXN_LINE.LINE_OBJECT_TYPE = 1 AND STG_AW_TXN_LINE.LINE_ACTION_CD = 1 THEN STG_AW_MERCH_DTL.UNITS WHEN (STG_AW_TX_LINE.LINE_OBJECT_TYPE = 2 AND STG_AW_TX_LINE.LINE_ACTION = 11) OR (STG_AW_TX_LINE.LINE_OBJECT_TYPE = 6 AND STG_AW_TX_LINE.LINE_ACTION = 24 AND STG_AW_TX_LINE.LINE_OBJECT in ('604', '615', '616', '618', '693')) THEN 1 ELSE 0 END • STG_ATG_COMMERCE_ITEM.QUANTITY • TRANSACTION_DETAIL.QUANTITY if SALE_OR_RETURN_INDICATOR <> "R" • TI_TRANSACTION_DETAIL_ERROR.QUANTITY if SALE_OR_RETURN_INDICATOR <> "R"

BU24.7.2	The ITEM_SOLD_QTY is always stored as a positive number
BU24.7.3	Similarly, the total return quantity is retained in the ITEM_RET_QTY field sourced from the following fields: • STG_AW_MERCH_DTL.UNITS CASE WHEN STG_AW_TXN_LINE.LINE_OBJECT_TYPE = 1 AND STG_AW_TXN_LINE.LINE_ACTION_CD = 2 THEN STG_AW_MERCH_DTL.UNITS WHEN STG_AW_TXN_LINE.LINE_OBJECT_TYPE = 2 AND STG_AW_TXN_LINE.LINE_ACTION_CD = 12 THEN 1 ELSE 0 END • STG_ATG_RETURN_ITEM.QUANTITY_RECEIVED • TRANSACTION_DETAIL.QUANTITY if SALE_OR_RETURN_INDICATOR = "R" • TI_TRANSACTION_DETAIL_ERROR.QUANTITY if SALE_OR_RETURN_INDICATOR = "R"
BU24.7.4	The ITEM_RET_QTY is always stored as a positive dollar amount
BU24.8	The system will track the gross amount of the transaction item line which is defined as the total dollar value of the line prior to application of any discounts.
BU24.8.1	The total dollar value is retained in the ITEM_GROSS_AMT field sourced from the following fields: • STG_AW_TXN_LINE.GROSS_LINE_AMOUNT CASE WHEN (STG_AW_TXN_LINE.LINE_OBJECT_TYPE = 1 AND STG_AW_TXN_LINE.LINE_ACTION_CD = 1) OR (STG_AW_TX_LINE.LINE_OBJECT_TYPE = 2 AND STG_AW_TX_LINE.LINE_ACTION = 11) OR (STG_AW_TX_LINE.LINE_OBJECT_TYPE = 6 AND STG_AW_TX_LINE.LINE_ACTION = 24 AND STG_AW_TX_LINE.LINE_OBJECT in ('604', '615', '616', '618', '693')) THEN STG_AW_TXN_LINE.GROSS_LINE_AMOUNT ELSE 0 END • STG_ATG_ADJUSTMENT.TOTALADJUSTMENT • Marketworks: if SALE_OR_RETURN_INDICATOR <> "R" then (net retail/(1-markdown percent/100)) * quantity

BU24.8.2	The ITEM_GROSS_AMT is always stored as a positive dollar amount
BU24.8.3	The ITEM_GROSS_AMT is always stored as EXTENDED (line qty * item sold price)
BU24.9	The system will track the prorated total discount amount of the transaction item line. The discount amount will be represented as a positive value. Thus, from a calculation perspective, the gross amount minus the discount amount will give the net total amount actually charged the Client for the transaction item line prior to taxes.
BU24.9.1	The prorated total discount amount is retained in the ITEM_DISC_AMT field sourced from the following fields: • STG_AW_TXN_LINE.POS_DISCOUNT_AMOUNT • STG_ATG_DTL_ITM_PRICE_INFO.ORDERDISCOUNTSHARE • Marketworks: ○ net retail/(1-markdown percent/100)-net retail
BU24.9.2	The ITEM_DISC_AMT is always stored as a positive dollar amount
BU24.10	If the line represents the return of products, the system will track the return quantity and return amount as separate fields on the transaction item. These will be represented as positive values. The return amount is an amount net of any discounts.

BU24.10.1	AuditWorks returns are identified as the following
	• Merchandise Returned =
	  ○ STG_AW_TXN_LINE.LINE_OBJECT_TYPE = 1 AND STG_AW_TXN_LINE.LINE_ACTION_CD = 2
	• Fee Refunded =
	  ○ STG_AW_TXN_LINE.LINE_OBJECT_TYPE = 2 AND STG_AW_TXN_LINE.LINE_ACTION_CD = 12

BU24.10.2	ATG returns are provided via a separate feed. These are loaded into the DW as separate/independent orders. They refer to their returned order via the ATG order ID posted into ITEM_RET_FROM_TXN_NBR
BU24.10.3	Marketworks returns are identified through the SALE_OR_RETURN_INDICATOR = “R” quantity and sales are mapped alternately according to this flag setting on the TRANSACTION_DETAIL table
BU24.10.4	The Return Quantity is stored in the ITEM_RET_QTY field as described in **BU24.7.3**

BU24.10.5	The Return Amount is stored in the ITEM_REF_AMT field as sourced from the following fields:
	• STG_AW_TXN_LINE.GROSS_LINE_AMOUNT
	
	CASE
	WHEN (STG_AW_TXN_LINE.LINE_OBJECT_TYPE = 1 AND STG_AW_TXN_LINE.LINE_ACTION_CD = 2) OR (STG_AW_TXN_LINE.LINE_OBJECT_TYPE = 2 AND STG_AW_TXN_LINE.LINE_ACTION_CD = 12) THEN STG_AW_TXN_LINE.GROSS_LINE_AMOUNT
	ELSE 0
	END
	
	• STG_ATG_RETURN_ITEM.REFUNDAMOUNT
	• MarketWorks:
	  ○ TRANSACTION_DETAIL.NET_RETAIL
	  ▪ if SALE_OR_RETURN_INDICATOR = "R"

	○ TI_TRANSACTION_DETAIL_ERROR.NET_RETAIL ▪ if SALE_OR_RETURN_INDICATOR = "R"
BU24.11	The system will track a return reason for the transaction item if available.
BU24.11.1	The Return Reason Code for the transaction is stored in the ITEM_RETURN_CD field sourced as follows: • AuditWorks ○ STG_AW_RETURN_DTL.RETURN_REASON_CODE • ATG – not available • MarketWorks: ○ TRANSACTION_DETAIL..SALE_OR_RETURN_INDICATOR ○ TI_TRANSACTION_DETAIL_ERROR.SALE_OR_RETURN_INDICATOR
BU24.12	The system will maintain a link back to the originating transaction for a returned item if available.
BU24.12.1	The returned order link is stored in the ITEM_RET_FROM_TXN_NBR field sourced as follows: • AuditWorks: TO_CHAR(STG_AW_RETURN_DTL.RETURN_FROM_STORE) || ':' || TO_CHAR(STG_AW_RETURN_DTL.RETURN_FROM_REGISTER) || ':' || STG_AW_RETURN_DTL.TRANSACTION_DATE in 'YYYYMMDD' format || ':' || TO_CHAR(STG_AW_RETURN_DTL.RETURN_FROM_TRANSACTION_NO) • ATG ○ STG_ATG_RETURN_REQUEST.REPLACEMENTORDERID • MarketWorks – Not Available

25. Tender

#	Design
BU25.1	For each transaction, the system will track details for each payment method utilized. If available from the source. This level of data is referred to as tender.
BU25.1.1	The AuditWorks TENDER table is derived from the following record types: STG_AW_CUST_DTL STG_AW_LINE_NOTES STG_AW_TXN_HEADER STG_AW_TXN_LINE
BU25.1.2	The ATG TENDER table is derived from the following XML nodes: STG_ATG_CREDIT_CARD STG_ATG_ORDER STG_ATG_PAYMENT_GROUP
BU25.1.3	The MarketWorks TENDER table is derived from the following source tables: TENDER
BU25.1.4	MarketWorks Identified transactions to not retain TENDER data, for these transactions a TENDER level is derived from TRANSACTION_HEADER
BU25.1.5	The low-level fact data is NOT made available in the Data Mart. This data is rolled up to F_TXN_HEADER to include

<table>
<tr><td></td><td>

SUM() of all tender amounts
The LAST tender date is made available
The COUNT of tenders
BEST Tender Type/SubType according to the following ranking:
Currency Code is not included
If the BEST Tender Type = “CC” the last 4 CC digits for that CC is shown

</td></tr>
<tr><td>BU25.1.5</td><td>The BEST Tender Type & Sub Type is determined by the tender type/subtype with the greatest dollar value. In the event of a tie a tender type is selected at random as the BEST</td></tr>
<tr><td>BU25.2</td><td>The tender detail is lost for identified transactions in MarketWorks. A single tender type is available at the transaction header level. In those cases, the system will take that tender type and apply the total dollar value of the transaction for purposes of creating a tender record.</td></tr>
<tr><td>BU25.2.1</td><td>The MarketWorks identified transaction tender is derived by creating a single TENDER row from the following TRANSACTION_HEADER attributes

TNDR_SUBTYPE_CD = TENDER_TYPE
TNDR_DT = TRANSACTION_DATE
TNDR_AMT = TOTAL_NET_RETAIL
CURRENCY_CD = CURRENCY_CODE
RSA_KEY_CLASS = STG_MW_CUST_ALT_KEY.CC_LAST_4_DIGITS

</td></tr>
<tr><td>BU25.3</td><td>The system will convert the codes from the various sources to a standard tender type with the following values. These values must be mapped to match the TENDER_MAPPING table in MarketWorks. Specific details to be defined during the Design Phase and captured in the Solution Design Document.

CASH – cash payment
CHK – check
CERT – certificate(gift certificate, rewards certificate, etc)
CC – bank credit card
GC – gift card
MC – merchandise credit
DC – debit card
UNK – unknown
Note: Epsilon will not store gift card numbers

</td></tr>
<tr><td>BU25.3.1</td><td>The AuditWorks tender is defined based on the LINE_OBJECT attribute as follows:
<table>
<tr><td>STG_AW_TRAN_LINE</td><td>DW.TENDER</td><td>DW.TENDER</td><td></td></tr>
<tr><td>LINE_OBJECT</td><td>TNDR_TYPE_CD</td><td>TNDR_SUBTYPE_CD</td><td></td></tr>
<tr><td>600</td><td>CASH</td><td>CASH</td><td></td></tr>
<tr><td>601</td><td>CHK</td><td>CHK</td><td></td></tr>
<tr><td>602</td><td>CHK</td><td>TRVL</td><td></td></tr>
<tr><td>604</td><td>CERT</td><td>GIFT</td><td></td></tr>
<tr><td>606</td><td>CHK</td><td>SEND</td><td></td></tr>
<tr><td>610</td><td>CC</td><td>PLCC</td><td></td></tr>
<tr><td>611</td><td>CC</td><td>MC</td><td></td></tr>
<tr><td>612</td><td>CC</td><td>AMEX</td><td></td></tr>
<tr><td>613</td><td>CC</td><td>VISA</td><td></td></tr>
<tr><td>614</td><td>CC</td><td>JCB</td><td></td></tr>
<tr><td>615</td><td>CERT</td><td>EGIFT</td><td></td></tr>
<tr><td>616</td><td>GC</td><td>ESVS</td><td></td></tr>
<tr><td>618</td><td>GC</td><td>SVS</td><td></td></tr>
</table>
</td></tr>
</table>

619	CRED	SVS		
620	DEBIT	DEBIT		
630	GC	REWRD		
651	CRED	STORE		
652	CRED	STORE		
680	CASH	INVLDCC		
693	CERT	WARD		
694	CERT	REWRD		
695	CC	DNRS		
696	CC	DSCV		
698	CC	PLCCADJ		
699	CC	MCADJ		
700	CC	AMEXADJ		
701	CC	VISAADJ		
702	CC	JCB		
703	CC	DNRSADJ		
704	CC	DSCVADJ		
706	CERT	MALL		
708	CC	COBRAND		
709	REWRD	COBRAND		
710	REWRD	PLCC		

BU25.3.2

The ATG tender is defined based on the Payment Method and CC Type attributes as follows:

STG_ATG_PAYMENT_GROUP	STG_ATG_CREDIT_CARD	DW.TENDER	DW.TENDER
PAYMENT_METHOD	CREDITCARDTYPE	TNDR_TYPE_CD	TNDR_SUBTYPE_CD
CREDITCARD	MASTERCARD	CC	MC
CREDITCARD	AMERICANEXPRESS	CC	AMEX
CREDITCARD	ANNTAYLORLOFTMASTERCARD	CC	COBRAND
CREDITCARD	JCB	CC	JCB
CREDITCARD	VISA	CC	VISA
CREDITCARD	ANNTAYLORLOFTCARD	CC	PLCC
CREDITCARD	DISCOVER	CC	DSCV
CREDITCARD	DINERSCLUB	CC	DNRS
GIFTCERTIFICATE	n/a	CERT	GIFT
STORECREDIT	n/a	CRED	STORE
GIFTCARD	n/a	GC	GC

BU25.3.3

The MarketWorks tender is defined based on the TENDER_TYPE attribute as follows:

TRANSACTION_HEADER	DW.TENDER	DW.TENDER
TENDER_TYPE	TNDR_TYPE_CD	TNDR_SUBTYPE_CD
AMEX	CC	AMEX
CASH	CASH	CASH
CHEK	CHK	CHK
COMC	CC	COBRAND
CORC	REWRD	COBRAND
DBT	DEBIT	DEBIT
DISC	CC	DSCV
DNRS	CC	DNRS
DSCV	CC	DSCV
GFCD	GC	GC

GIFT	CERT	GIFT	
HOUS	CC	PLCC	
JCB	CC	JCB	
MAST	CC	MC	
MCCR	CRED	STORE	
PLRC	REWRD	PLCC	
VISA	CC	VISA	

BU25.3.4

The Decode values are above are maintained two lookup tables leveraged in the DW process.

TENDER_TYPE

Field Name	Data Type	PK
TXN_SOURCE_CD	VARCHAR(4)	Y
TXN_SOURCE_TNDR_TYPE_CD	VARCHAR(30)	Y
TNDR_TYPE_CD	VARCHAR(4)	

TENDER_SUBTYPE

Field Name	Data Type	PK
TXN_SOURCE_CD	VARCHAR(4)	Y
TXN_SOURCE_TNDR_SUBTYPE_CD	VARCHAR(30)	Y
TNDR_TYPE_CD	VARCHAR(4)	

BU25.3.4

Any value received from AW, ATG, or MW encountered in the source field that is not in one of the decode tables above result

- sets the value on the TENDER table “UNKN"
- inserts the unknown source value (and transaction source system ID – AW/ATG) into the appropriate table above
- initializes the standardized value to NULL
- Notifies the support team of any values that failed the decode
- The support team escalates with ANN INC to determine the appropriate decode and updates the appropriate decode table
- The decode rule is then employed on an ongoing basis. If there has been only a single UNKN decode encountered before the translation was applied, the UNKN values can be appropriately reassigned. Otherwise, there is no attempt made to recode the “UNKN” value as the source data on the transactions is no longer available, unless a Change Request is issued to perform a transaction reload.

BU25.4

The system will track a tender subtype that further refines the tender type. For example, distinguishing a CC (credit card) tender type;

- VISA
- MasterCard
- American Express
- Diners Club
- Discover
- Cobrand AT
- Cobrand LOFT
- PLCC

To distinguish a GC (gift card) tender type;

- EGC – electronic gift card

These values must be mapped to match the TENDER_MAPPING table in MarketWorks.

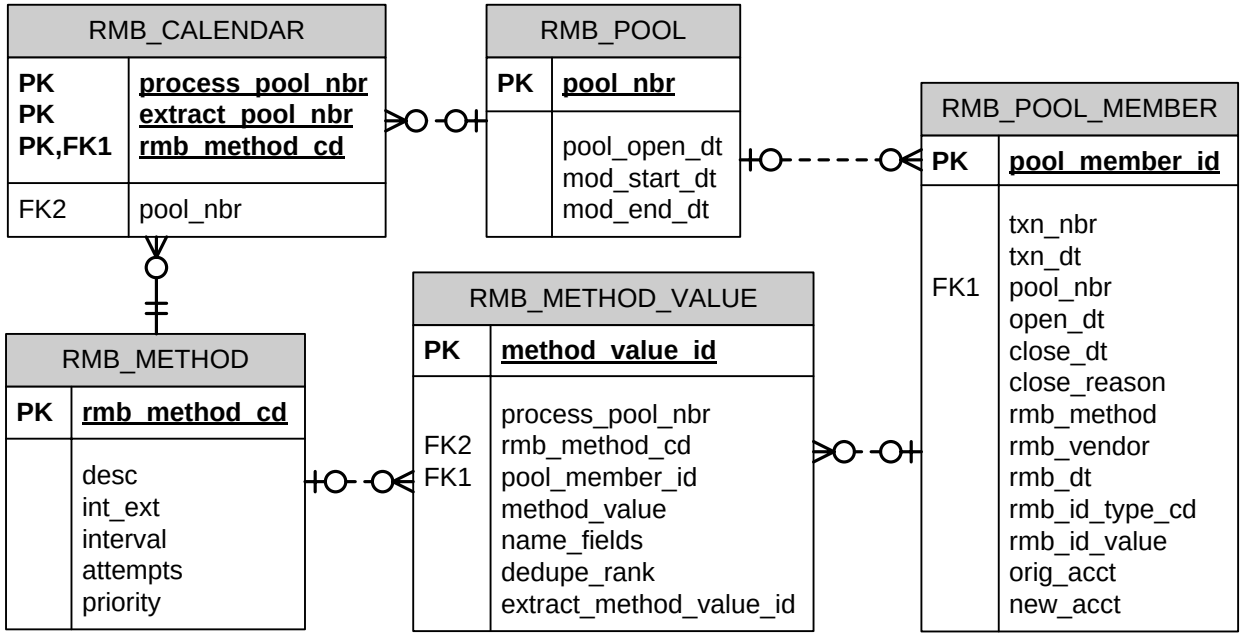
	Specific details to be defined during the Design Phase and captured in the Solution Design Document.
BU25.4.1	Tender subtype design is addressed as described in **BU25.3**
BU25.5	The system will track the date that the tender was applied (typically the same as the transaction date) and the tender amount.
BU25.5.1	The date the tender was applied is stored in the TNDR_DT field sourced as follows: • STG_AW_TXN_HEADER.TRANSACTION_DATE • STG_ATG_PAYMENT_GROUP.SUBMITTEDDATE • TI_TRANSACTION_HEADER_ERROR.TRANSACTION_DATE • TRANSACTION_HEADER.TRANSACTION_DATE

26. Transaction Item Discount

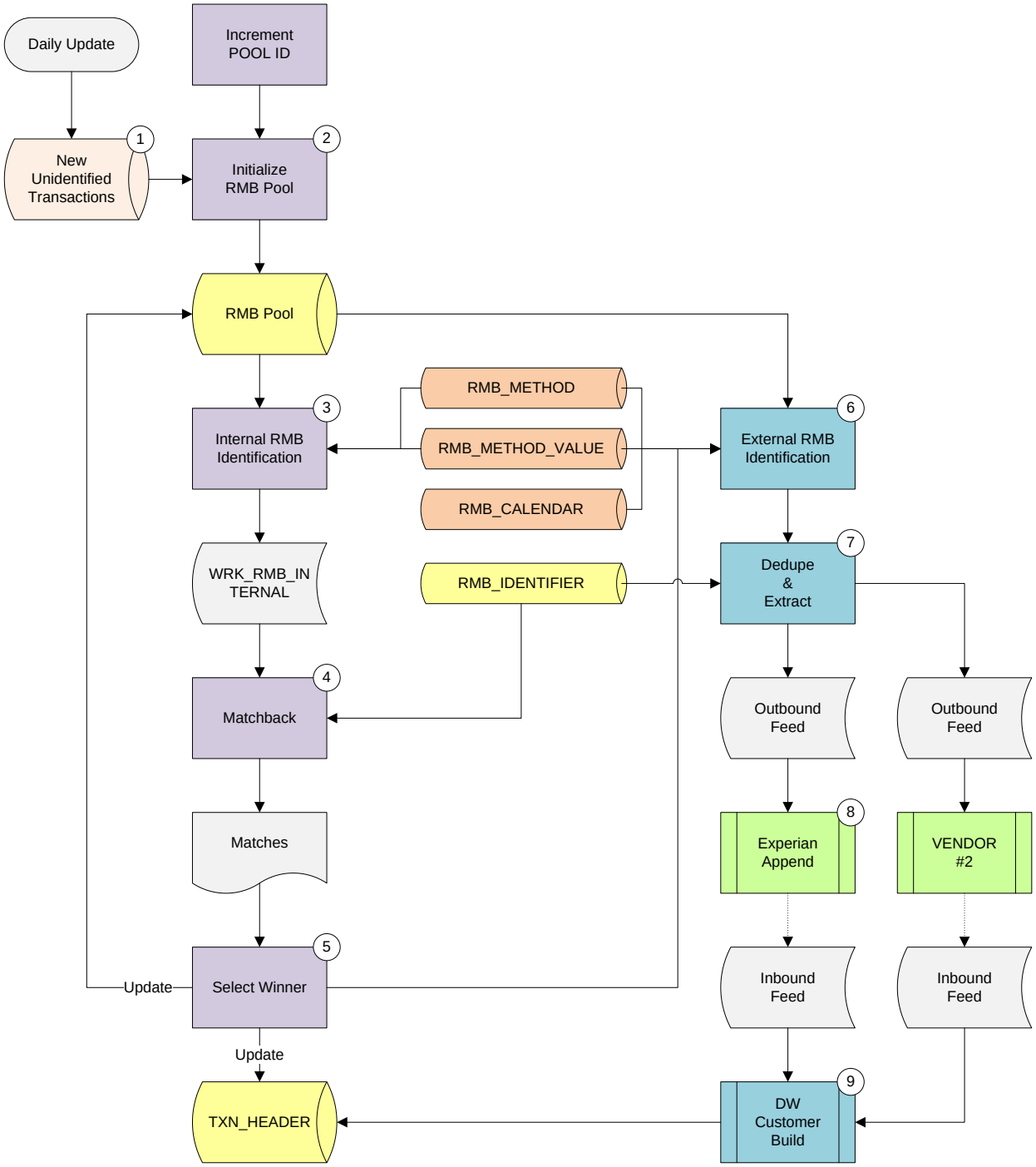
#	Design
BU26.1	For each transaction item the system will track the details of the discounts applied to the item.
BU26.1.1	The AuditWorks TXN_ITEM_DISCOUNT table is derived from the following record types: STG_AW_DISC_DTL
BU26.1.2	The ATG TXN_ITEM_DISCOUNT table is derived from the following XML nodes: STG_ATG_ADJUSTMENT STG_ATG_COMMERCE_ITEM STG_ATG_ORDER STG_ATG_PRICE_INFO
BU26.1.3	The MarketWorks TXN_ITEM_DISCOUNT table is derived from the following source tables: TI_TRANSACTION_COUPON_ERROR TRANSACTION_COUPON
BU26.1.3	The low-level fact data is made available in the Data Mart as F_TXN_ITEM_DISCOUNT
BU26.2	The system will store the type of discount applied to the item. Specific details to be defined during the Design Phase and captured in the Solution Design Document. For example: coupon, markdown….
BU26.2.1	The Discount Type is stored in the DISC_TYPE_CD field sourced as follows: • STG_AW_DISC_DTL.POS_DISCOUNT_TYPE • STG_ATG_ADJUSTMENT.ADJUSTMENTDESCRIPTION • MarketWorks – not available
BU26.2.2	The ATG & AW native discount types are retained in the Data Warehouse. The discount types are standardized in the Data Warehouse.
BU26.3	The system will track the amount of the discount applied to the item for the specific discount type.
BU26.3.1	The Discount Amount is stored in the DISC_AMT field sourced as follows: • STG_AW_DISC_DTL.POS_DISCOUNT_AMOUNT • STG_ATG_DTL_ITM_PRICE_INFO.ORDERDISCOUNTSHARE • Marketworks: ○ net retail/(1-markdown percent/100)-net retail net retail/(1-markdown percent/100)-net retail

BU26.4	The system will track a specific coupon code captured by the source system and applied to the transaction item.
BU26.4.1	The Coupon code is stored in DISC_SERIAL_NBR sourced as follows: • STG_AW_DISC_DTL.POS_DISCOUNT_SERIAL_NO • STG_ATG_ORDER.COUPONCODE • TRANSACTION_COUPON.COUPON_CODE • STG_TI_TRANSACTION_COUPON_ERROR.COUPON_CODE
BU26.4.2	There are potentially multiple coupons applied to a single line there are made unique within a TXN_ITEM by incrementing a sequence value DISC_SEQ_NBR

Retail Matchback Entity Key Relationships – Data Warehouse



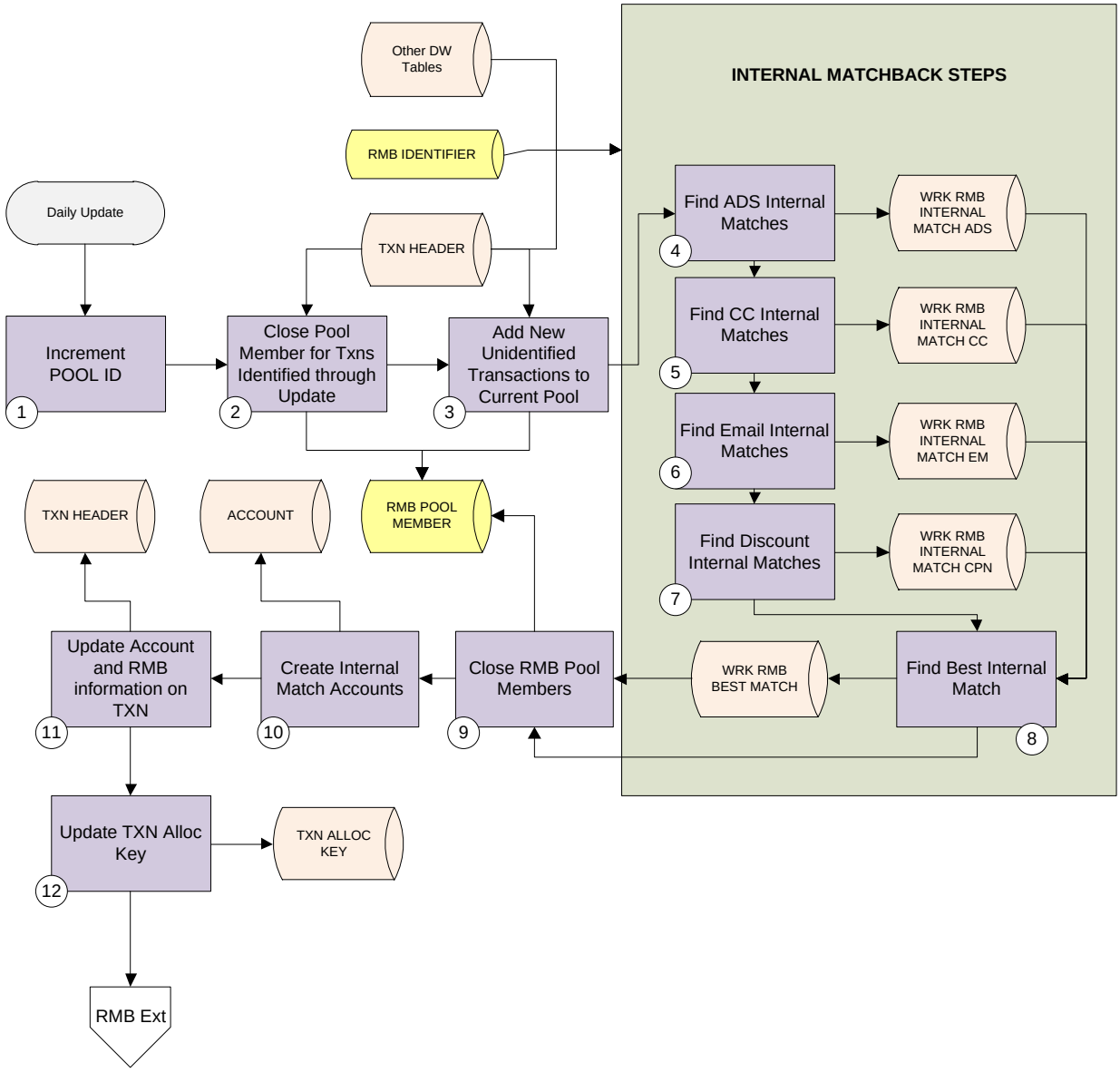
Retail Matchback Update Process Flow



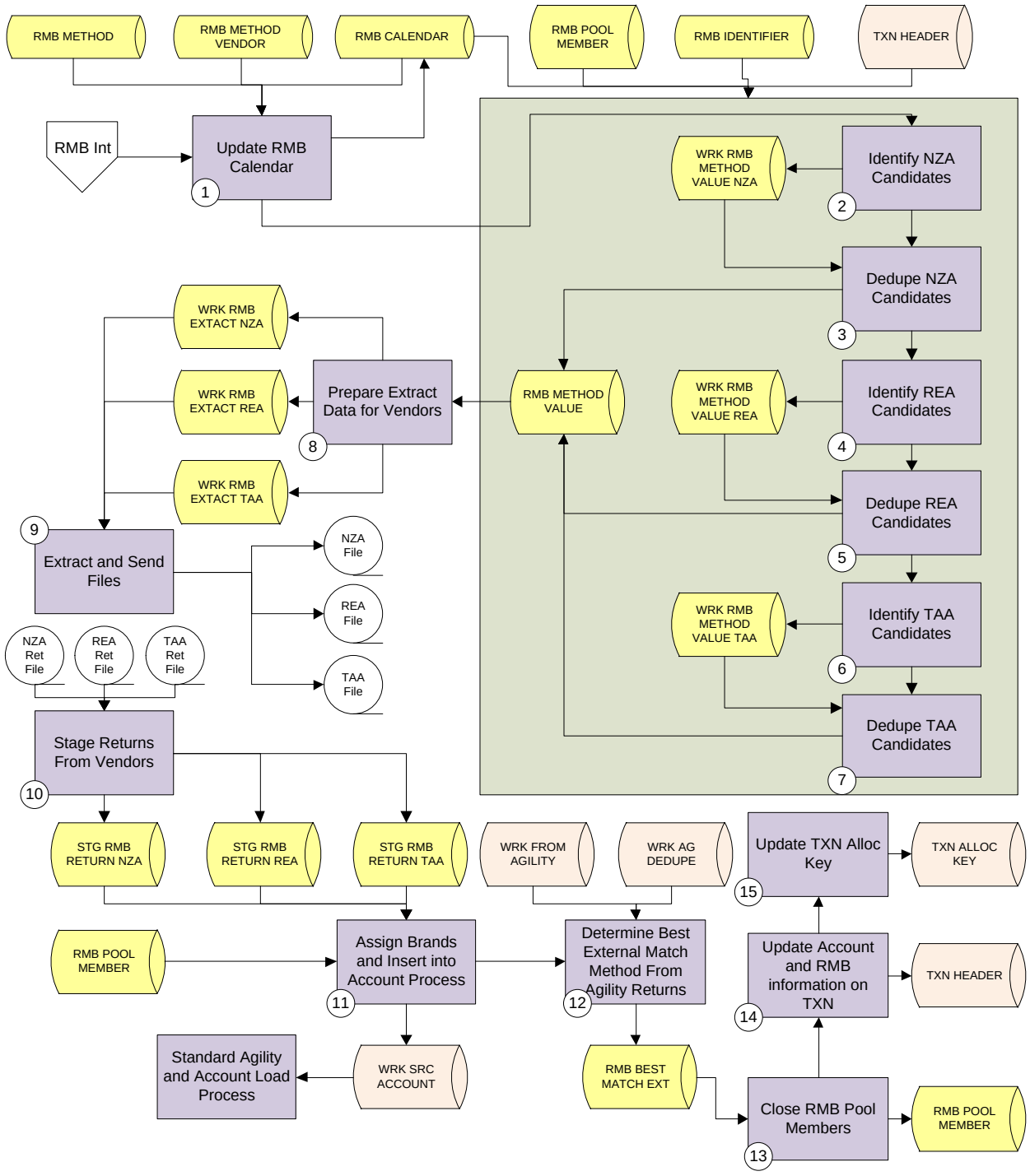
Retail Matchback Update Process Annotations

1) Transactions are loaded as part of the daily update from Auditworks POS and ATG ECOM. Many of these do not have sufficient information included on the transaction to identify an ANN INC customer. These are referred to as Unidentified Transactions.
2) The Unidentified transactions are loaded into a Retail Matchback Pool with a POOL_ID incremented to represent this update cycle's batch of new unidentified transactions. In the event that one or more update cycles were missed the pool increment would represent multiple days worth of transactions. The RMB_POOL_MEMBER is loaded with the Transaction IDs and POOD IDs for each transaction. All RMB identifier fields (method, vendor, date, type, value) are initialized. The Orig_Acct is set to the account number from the transaction load (if available).
3) The Internal Matchback process is invoked. It references the RMB_METHOD and RMB_METHOD_VALUE to select all eligible internal matchback transactions (method interval = DAILY and txn_dt >= max age for each internal matchback method and close_dt is null)
4) The Internal Matchback process runs by referencing the data warehouse transaction history table of attributes used for Retail Matchback identification (RMB_IDENTIFIER). The unmatched transactions compare their attributes to those in the warehouse in an attempt to find a match. The sequence is performed as follows:
    a. Credit Card Hash decoded as ADS Account Number
    b. Credit Card Hash & Full Name
    c. Email Address & Full Name
    d. Coupon Barcode
5) In the event that multiple methods result in a match, one is selected based on the hierarchy above and recency. The RMB Pool Member table is updated with the method, vendor (EPS), date, type, and value. The close date and reason are also updated. Lastly, the data warehouse TXN_HEADER table is updated with the assigned ACCT_NBR for linkage to and Individual and inclusion in transaction rollups.
6) The External Matchback process is invoked. It referenced the RMB_METHOD, RMB_METHOD_VALUE, and RMB_CALENDAR to select all eligible external matchback transactions. These are determined through the batching of several days worth of unidentified transactions and performing external process "retries" based on the vendor and append type defined in the calendar.
7) The appropriate attributes are pulled in from the data warehouse for the selected transactions and deduped by attributes and type. The resulting datasets are distinct overlay input records create for each vendor:
    a. NZA – Full Name and Postal Code
    b. REA – Full Name and Email Address
    c. TAA – Full Name and Store Postal Code
8) The feeds are sent via SFTP to the appropriate vendor where overlay processing is performed. The feeds are returned once the external process completes (which may be one or more daily database update cycles later). They are collected and staged to the CRM database.
9) The overlay data is mapped into WRK_SRC_ACCOUNT where it follows the entire customer build process including account number generation and key assignment through AGILITY. With the account number assigned the, the transaction is updated and linked to the AGILITY keys. The transaction can then be related to other accounts through an individual for rollup inclusion.

Retail Matchback Detailed Update Process Flow (Internal)



Retail Matchback Detailed Update Process Flow (External)



27. Retail Matchback Process (Internal and External)

<table>
<tr><th>#</th><th>Design</th></tr>
<tr><td>BU27.1</td><td>For transactions that do not have a billing account with a valid address the system will utilize a process called retail matchback to attempt to link the transaction to an account with a valid address.</td></tr>
<tr><td>BU27.1.1</td><td>Transactions that do not have a billing address are NULL on the TXN_HEADER table. They are also identified in the RMB_POOL_MEMBER with is loaded and processed daily in an attempt to use all methods available to assign or reattempt assignment of unassigned transactions. In the POOL unassigned transactions are identified as having a CLOSE_DT = null</td></tr>
<tr><td>BU27.2</td><td>The system will process the transaction error tables from MarketWorks in the internal retail matchback process to attempt to link the transaction to an account via the hashed credit card number.</td></tr>
<tr><td>BU27.2.1</td><td>The RMB_IDENTIFIER is loaded from MarketWorks as follows:
<table>
<tr><th>MW TXN Type</th><th>Method</th><th>Source Table</th><th>Field(s)</th></tr>
<tr><td>Identified</td><td>ADS</td><td>MW_CUSTOMER_ALTERNATE_KEY</td><td>ALTERNATE_KEY & FIRST_NAME & LAST_NAME</td></tr>
<tr><td>Identified</td><td>CC</td><td>MW_CUSTOMER_ALTERNATE_KEY</td><td>ALTERNATE_KEY & FIRST_NAME & LAST_NAME</td></tr>
<tr><td>Identified</td><td>EM</td><td>CUSTOMER</td><td>EMAIL_ADDRESS & FIRST_NAME & LAST_NAME</td></tr>
<tr><td>UnID</td><td>CC</td><td>STG_MW_TRAN_TEND_ERR</td><td>TENDER_IDENTIFIER & FIRST_NAME & LAST_NAME</td></tr>
<tr><td>UnID</td><td>CC</td><td>STG_MW_EMAIL_POS_ARCH</td><td>CC_NO</td></tr>
<tr><td>UnID</td><td>EM</td><td>STG_MW_EMAIL_POS_ARCH</td><td>EMAIL_ADDRESS & FIRST_NAME & LAST_NAME</td></tr>
<tr><td>UnID</td><td>ZIP</td><td>STG_MW_EMAIL_POS_ARCH</td><td>FIRST_NAME & LAST_NAME & ZIPCODE</td></tr>
</table></td></tr>
<tr><td>BU27.2.2</td><td>Identified transactions are backloaded with all combinations available at the customer level because they are no longer available at the transaction level at the source. This, however, does make additional matches possible for future ATG & AW transactions.</td></tr>
<tr><td>BU27.3</td><td>For each transaction successfully processed through retail matchback, the system will keep a history of the original billing account key (if any), the retail matchback method utilized, the retail matchback vendor, and the date that the matchback was performed.</td></tr>
<tr><td>BU27.3.1</td><td>The RMB_POOL_MEMBER table retains Retail Matchback attributes:
ORIG_ACCT – original billing account key
NEW_ACCT – account assigned via RMB processing
RMB_METHOD – method used to assign account:
ADS – ADS Account (internal)
CC – Credit Card (internal)
EM – Email & Name match (internal)
BC – Coupon/Barcode (internal)
NZA – Name/Zip Append
REA – Reverse Email Append
TAA – Trade Area Append</td></tr>
</table>

	• RMB_VENDOR – External Append Vendor
	  ○ EXP – Experian
	  ○ ??? – Vendor #2
	• RMB_DT – Data match-back account was applied

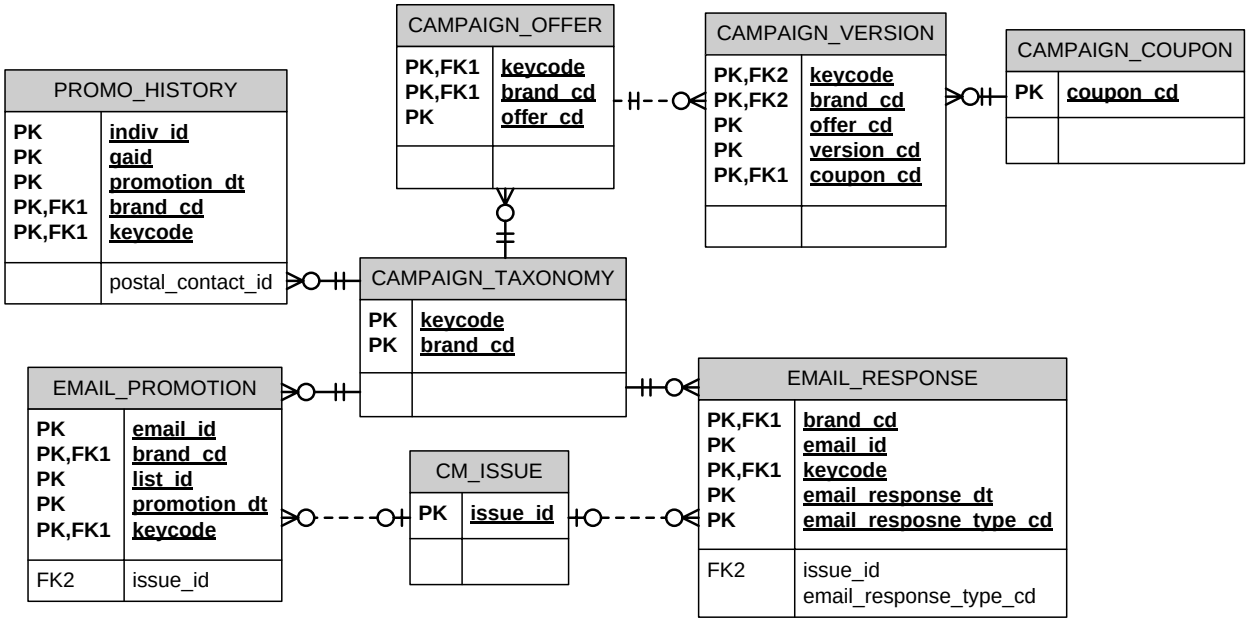
BU27.4	Within each update, the system will perform the process of **Internal Retail Matchback**, which will attempt to match unidentified transactions to a valid account utilizing data contained within the marketing database. The following Internal Retail Matchback methods will be attempted in the sequence indicated:
	1. ADS Account Match – the system will attempt to match a hashed ADS credit card number from the unidentified transaction to an ADS account number (via the supplied cross reference) and assign that ADS account to the transaction.
	
	2. Credit Card Match – the system will attempt to match the primary hashed bank credit card number on the unidentified transaction to other transactions utilizing the same credit card and also having a billing account with a valid address.
	  a. The primary credit card for transactions used in the matching process is defined as the credit card with the highest dollar value with ties being resolved by a sort on the credit card hash.
	  b. The system will attempt the credit card match process for all unidentified transactions containing a credit card each update regardless of age of the unidentified transaction.
	  c. The credit card match process will consider candidate matching transactions regardless of age of the candidate transaction.
	  d. The system will further evaluate candidate matches by comparing the soundex of the first and last name of the credit card swipe with the first and last name on the candidate account. Only soundex matches on both fields will be considered. This implies that single first initial to full first name will not be considered a match and records where the first name matches but the last name is different (as occurs in marriage name changes) will not be considered a match.
	  e. Of all records in the matched result, the system will choose the account from the most recent candidate transaction to assign to the unidentified transaction.
	  f. Transactions placed in stores located in the states of CA, NJ and MA will not have their zip codes captured at POS.
	
	3. Email/Name Match – if an unidentified transaction contains an email address first name, and last name the system will search for an account having a valid address and the same email address and name.
	  a. The name components for the email match will come from the account name if populated followed by the swipe name from the primary credit card.
	  b. The system will attempt the email/name match process for all unidentified transactions containing a credit card each update regardless of age of the unidentified transaction.
	  c. The email/name match process will consider candidate matching transactions regardless of age of the candidate transaction.
	  d. The system will further evaluate candidate matches by comparing the

	soundex of the first and last name of the unidentified transaction with the first and last name on the candidate account. Only soundex matches on both fields will be considered. This implies that single first initial to full first name will not be considered a match and records where the first name matches but the last name is different (as occurs in marriage name changes) will not be considered a match. e. Of all records in the matched result, the system will assign the account with the most recent activity date to the unidentified transaction. 4. Coupon Barcode Match – if an unidentified transaction contains a coupon barcode the system will search for an account for the individual id contained in the barcode having a valid address. a. If multiple coupons are redeemed on a transaction, the system will use the distinct set of individual ids from the coupons for the attempted matches. b. The system will attempt the coupon barcode match process for all unidentified transactions containing a barcode each update regardless of age of the unidentified transaction. c. The system will resolve the individual id contained on the barcode to the current individual id in the case of merges. d. The system will consider all account sources, including those originally generated from External Retail Matchback sources such as name zip append as candidates for a coupon barcode match. e. From the set of matching accounts from a barcode match, the system will assign the account with the latest activity date to the unidentified transaction.
BU27.4.1	The ADS account lookup is done through the ACCOUNT_HARD_KEY table where type = "ADS" by referencing the hashed credit card number. If a match is found the ADS ACCT_NBR is assigned.
BU27.4.2	The Credit Card match is perform with the detailed logic described in the requirement
BU27.4.3	The Email/Name match is perform with the detailed logic described in the requirement
BU27.4.4	The Barcode match is performed through obtaining the ending 8-bytes of the barcode. This is the hashed individual ID extracted from the SUM_INDIVIDUAL/SUM_HOUSEHOLD table. This is matched to the INDIVIDUAL table to identify the individual_ID making the purchase
BU27.5	If an unidentified transaction containing name and zip information is not matched via the Internal Retail Matchback process. The system will schedule sending the record to a third party data provider for an address append. This is known as **External Retail Matchback**. 1. The system will obtain the name and zip information for External Retail Matchback from the primary credit card associated with the unidentified transaction. 2. Transactions placed in store located in the states of CA, NJ and MA will not have their zip codes captured at POS. 3. The system will first send the full name and zip code for transactions not identified through Internal Retail Matchback process to an external vendor for an append of address. This process is referred to as External Name Zip Append (NZA). a. KBM Group or Experian will be the primary vendor for the 1st NZA attempt b. If the transaction is still unidentified after 60 days from the first NZA

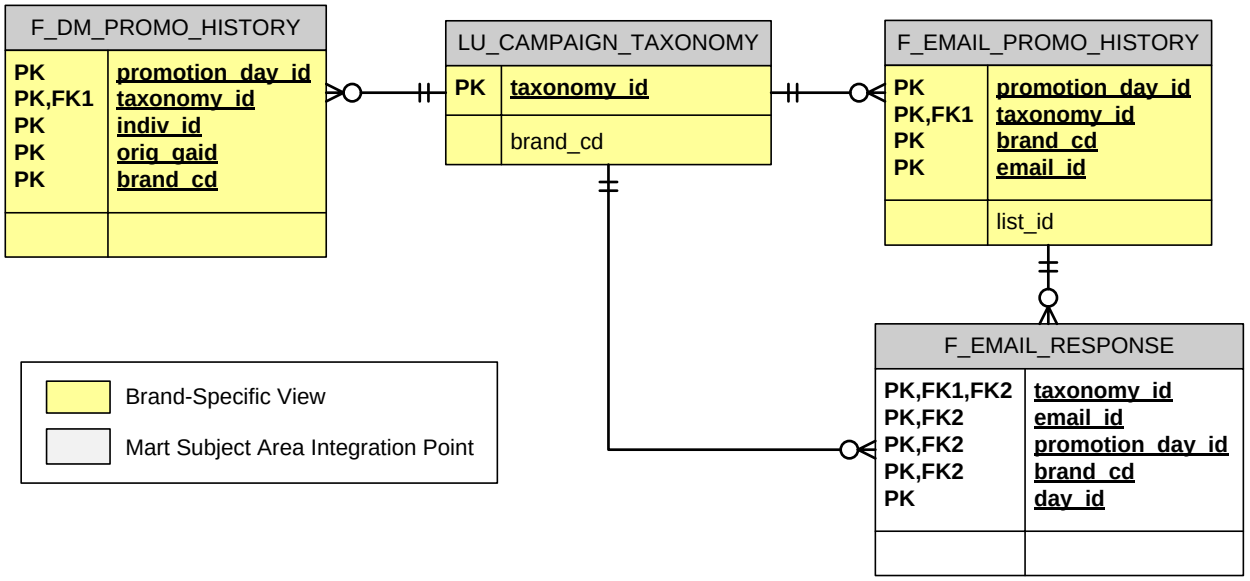
	attempt, the system will send the information to a secondary service provider for a second and final NZA attempt. Client will determine the vendor during the development process. c. The system will create an account for records successfully appended through the NZA process with an account source of 'NZA' and specify that as the billing account for the transaction. d. The system will extract files for NZA processing to go to KBM or Experian daily. The transactions processed on any given day will be different. A specific transaction will only be processed a maximum of 2 times. 4. For transactions that are still unidentified after the first NZA attempt, the system will send the full name and email address to KBM Group or Experian for an address append. This process is referred to as External Reverse Email Append (REA). a. The system will only make one REA attempt for a given transaction. b. The system will collect records qualifying for an REA attempt into a combined file that will be sent to the vendor weekly. c. The system will create an account for records successfully appended through the REA process with an account source of 'REA' and specify that as the billing account for the transaction. 5. For transactions that are still unidentified after the REA attempt, the system will send the full name and store number from the transaction to KBM Group or Experian for an address append. This process is referred to as External Trade Area Append (TAA). a. The system will only make one TAA attempt for a given transaction. b. The system will collect records qualifying for a TAA attempt into a combined file that will be sent to the vendor weekly. c. The system will create an account for records successfully appended through the TAA process with an account source of 'TAA' and specify that as the billing account for the transaction. As more vendors are used for TAA, REA and NZA there will be a need to add more account sources.
BU27.5.1	The Name/Zip Append process is performed through the detail process described in the requirement
BU27.5.2	The Reverse Email Append process is performed through the detail process described in the requirement
BU27.5.3	The Trade Area Append process is performed through the detail process described in the requirement

Promotion Subject Area

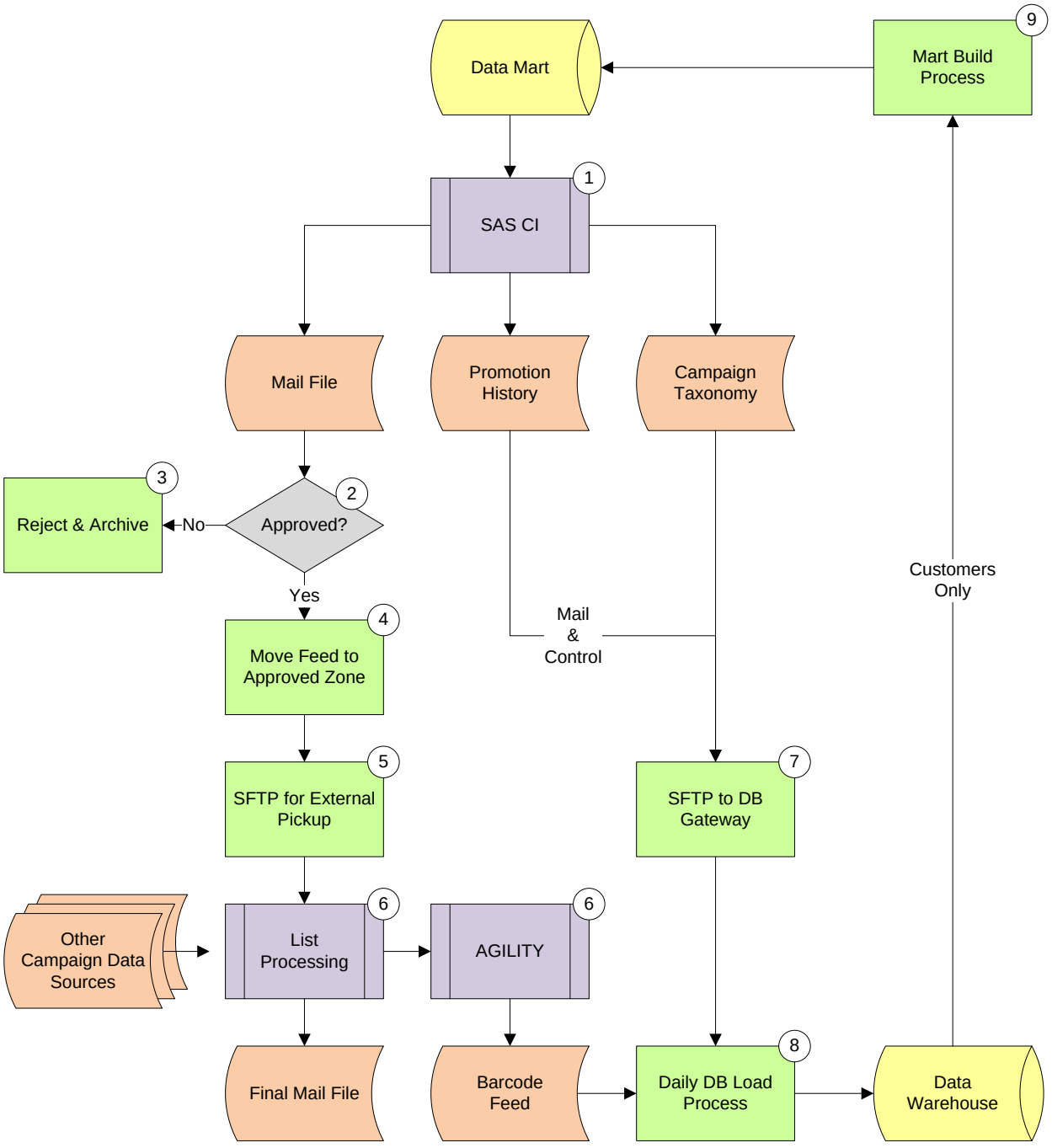
Promotion Entity Key Relationships – Data Warehouse



Promotion Entity Key Relationships – Data Mart



Direct Mail Promotion Data Warehouse Update Process Flow



Direct Mail Promotion Data Warehouse Update Process Annotations

1. The ANN INC CRM team uses SAS CI to access the Data Mart to select and segment and extract a campaign from the database. Three feeds are created as part of this process:

    a. Mail Feed – the extracted customer file with an intelligent keycode used to identify the campaign, segment, and version associated with each marketed individual our household.

    b. Promotion History Feed – a corresponding feed used to inform the database of the marketing activity, the mail records are included in promotion history along with "freeze field" that identify attributes of a customer as they were at the time the campaign was extracted. The Promotion History feed also includes Control records who are customer that look like those that were marketed to as part of the campaign, but did not receive marketing materials. This is used measure effectiveness of the marketing effort.

    c. Campaign Taxonomy Feed – a taxonomy feed is generated that describes each campaign keycode for later reference.

2. Once the campaign has been audited by the CRM team and is approved, it is moved into the /outbound folder on the SAS server.

3. Unapproved campaign files remain in the in the /pending folder where they are deleted after 30 days

4. Approved campaign files are pushed out to the DB Gateway server.

5. The list processing platform is provided with the extracted mail file.

6. The list processing process merges in additional data sources and performs final list hygiene. The coupon barcode is assigned at this time. The final mail file is generated and provided to the print shop for processing and deployment. A barcode update file is provided for update back into the database. This feed contains additional names and addresses for other sources. The mail file is referenced to determine if the data was soured from the CRM solution, if so it is loaded, otherwise the data is run through AGILITY and keys are identified and loaded into the Data Warehouse

7. The promotion history feed is SFTP'd from the SAS server.

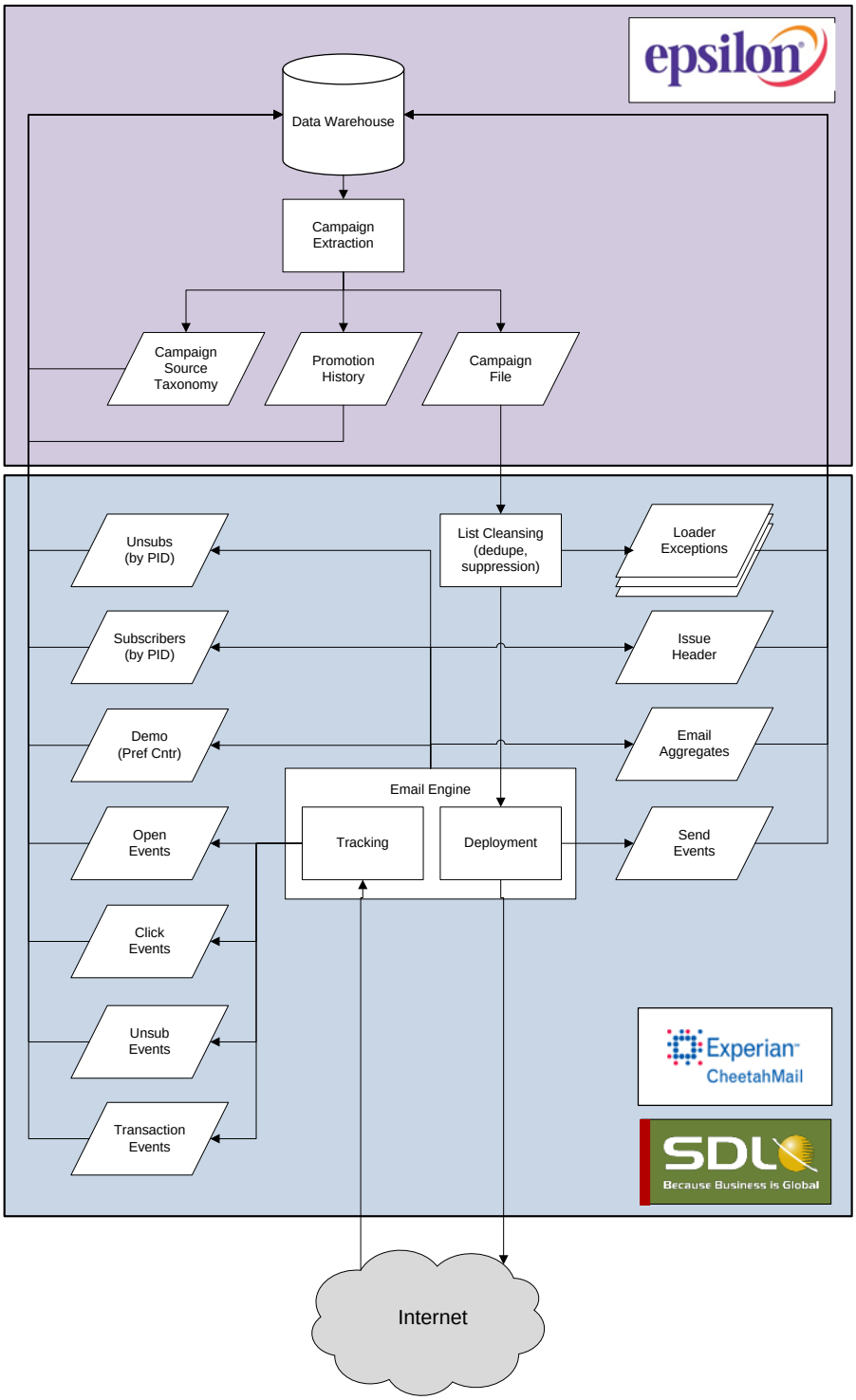
8. Promotion history for all key codes for which a barcode feed was found (indicating approved campaigns) is loaded into the database with the descriptive campaign taxonomy attributes.

9. The data mart is updated with the refreshed promotion history for those with AGILITY keys linking them to ANN INC customers and taxonomy tables. The individual ID on promotion history is used to perform ad-hoc campaign order attribution analysis.

Email Promotion Data Warehouse Update Process Flow

This diagram illustrates the future state for Email Campaign deployment from the CRM solution. Until this is employed the following structures are substituted for load into the Database using the CheetahMail campaign identifier ISSUE_ID as a surrogate for CRM Solution KEYCODE:

- Promotion History = Send Events
- Campaign Source Taxonomy = IID Keys



28. Promotion Tracking – Direct Mail

The system will contain table structures for the purpose of capturing direct mail promotion history and campaign metadata information to be used in the data mart for reporting, campaign analysis, future campaign selection, segmentation and suppression.

<table>
<tr><th>#</th><th>Design</th></tr>
<tr><td>BU28.1</td><td>The system will load the ANN Inc. direct mail promotion files for the migration load. The direct mail promotions have been reformatted to the same file layout. The data goes back 24 months and will need to be realigned to the current customer id using the cross reference supplied by MarketWorks.</td></tr>
<tr><td>BU28.1.1</td><td>Historical mail files are being provided to Epsilon in a variety of formats for feed standardization as described in SOW. The data included is up to 24 months old and will be loaded from most recent mail dates and back until the 250 hours of allocated time included in the Consolidation of Historical AnnTaylor Direct Mail Files SOW is reached.</td></tr>
<tr><td>BU28.1.2</td><td>The standardized format which is TBD as outcome of the campaign taxonomy design, is described below:
<table>
<tr><th>Field</th><th>Required</th><th>Data Type</th><th>Precision</th><th>Description</th><th>Value/Format</th></tr>
<tr><td>SRC_PREFIX</td><td></td><td>VARCHAR2</td><td>20</td><td>Either parsed name fields or FULL name</td><td></td></tr>
<tr><td>SRC_FIRST_NM</td><td></td><td>VARCHAR2</td><td>50</td><td></td><td></td></tr>
<tr><td>SRC_MIDDLE_NM</td><td></td><td>VARCHAR2</td><td>50</td><td></td><td></td></tr>
<tr><td>SRC_LAST_NM</td><td></td><td>VARCHAR2</td><td>50</td><td></td><td></td></tr>
<tr><td>SRC_SUFFIX</td><td></td><td>VARCHAR2</td><td>20</td><td></td><td></td></tr>
<tr><td>SRC_FULL_NM</td><td>Y</td><td>VARCHAR2</td><td>80</td><td>Either parsed name fields or FULL name</td><td></td></tr>
<tr><td>SRC_ADDR_LINE_1</td><td>Y</td><td>VARCHAR2</td><td>65</td><td>Promoted address data</td><td></td></tr>
<tr><td>SRC_ADDR_LINE_2</td><td></td><td>VARCHAR2</td><td>65</td><td></td><td></td></tr>
<tr><td>SRC_ADDR_LINE_3</td><td></td><td>VARCHAR2</td><td>65</td><td></td><td></td></tr>
<tr><td>SRC_ADDR_LINE_4</td><td></td><td>VARCHAR2</td><td>65</td><td></td><td></td></tr>
<tr><td>SRC_CITY</td><td>Y</td><td>VARCHAR2</td><td>50</td><td></td><td></td></tr>
<tr><td>SRC_STATE_CD</td><td>Y</td><td>VARCHAR2</td><td>40</td><td></td><td></td></tr>
<tr><td>SRC_POSTAL_CD</td><td>Y</td><td>VARCHAR2</td><td>10</td><td></td><td></td></tr>
<tr><td>SRC_COUNTRY_NM</td><td></td><td>VARCHAR2</td><td>40</td><td>Country Name</td><td></td></tr>
<tr><td>SRC_COUNTRY_CD</td><td></td><td>VARCHAR2</td><td>3</td><td>Country Code</td><td></td></tr>
<tr><td>SRC_LANG_PREF_FLG</td><td></td><td>VARCHAR2</td><td>1</td><td>Language Preference - if available</td><td></td></tr>
<tr><td>SRC_EMAIL_ADDR</td><td></td><td>VARCHAR2</td><td>100</td><td>Email address - likely not available</td><td></td></tr>
<tr><td>KEYCODE</td><td>Y</td><td>VARCHAR2</td><td>20</td><td></td><td></td></tr>
<tr><td>BRAND_CD</td><td>Y</td><td>VARCHAR2</td><td>4</td><td>Brand associated with the campaign</td><td>AT = Ann Taylor
LOFT = Loft
ATF = Ann Taylor Factory</td></tr>
</table>
</td></tr>
</table>

						LOS = Loft Outlet ENT = Unknown
	PROMOTION_DT	Y	DATE	8	Date the campaign was deployed	YYYYMMDD
	CUSTOMER_NO	Y	NUMBER	20	This was used to key historical transactions - should be available on all files	
	CUSTOMER_ID		NUMBER	20	ANN (MarketWorks) has a historical CUSTOMER_NO, and current CUSTOMER_ID - if both are available, we'll take both	
	HOLDOUT_IND	Y	NUMBER	1	Holdout/Control group indicator	

BU28.1.3	The feeds are developed using the TSSUtil conversion tool with the following conventions:
	Some other conventions:
	• Record Format
	  ○ Field Delimiter = TAB
	  ○ Line Delimiter = LF
	• Naming Convention = ANN_MW_PROMO_$yyyymmdd_$seq.dat
	  ○ yyyymmdd = date the promotion history files was converted (alternately could use promo date if we're getting 1 file/campaign)
	  ○ seq = if a single file needs to get converted in pieces, this can be used to sequence them
	• Extract one file per campaign
	• Control File - We'd like a control file created that would indicate the number of records in each file. (control file paired with each data file as "ANN_MW_PROMO_$yyyymmdd_$seq.ctl"
	  ○ Delimiter = pipe
	  ○ Field1 = file name
	  ○ Field2 = uncompressed file size
	  ○ Field3 = record count
	• Compression = GZIP

BU28.1.4	All files are then loaded into RAW_MW_PROMO with data types cast in STG_MW_PROMO. Any record-level rejections are reported to ANN INC for each campaign feed.
BU28.1.5	Any PROSPECT records are excluded from or filtered as part of the conversion process.
BU28.1.6	After the final MarketWorks migration data has been provided and successfully loaded the data is loaded into the Data Warehouse PROMO_HISTORY table. As part of this, a join is performed through the ANN_MIGR_USER.CUSTOMER_XREF on OLD_CUSTOMER_NO to POSTAL_CONTACT to obtain the AGILITY Individual and Address keys and Postal Contact IDs for existing accounts: select a.CUSTOMER_NO, b.NEW_CUSTOMER_NO, c.POSTAL_CONTACT_ID, c.INDIV_ID, c.GAID from STG_OWNER.STG_MW_PROMO a inner join ANN_MIGRN_USER.CUSTOMER_XREF b on a.CUSTOMER_NO = b.OLD_CUSTOMER_NO inner join DW_OWNER.POSTAL_CONTACT c on b.NEW_CUSTOMER_NO = c.ACCT_NBR and 'MW' = c.ACCT_SOURCE_CD;
BU28.1.7	Promotion History records failing the above join are reformatted into the WRK_TO_AGILITY layout and processed through AGILITY as described in **Agility** The

<table>
<tr><td></td><td>AGILITY Individual and Address IDs are loaded into PROMO_HISTORY and the POSTAL_CONTACT_ID for these records is set to 0 (zero).</td></tr>
<tr><td>BU28.2</td><td>Migration campaign taxonomy will be loaded manually to the system per the design ANN Inc. and Epsilon developed.</td></tr>
<tr><td>BU28.2.1</td><td>The PROMOTION identifier and SEGMENT KEYCODE is defined for each campaign as part of the Promotion History conversion project and loaded into the following taxonomy structures through an ad-hoc process:
CAMPAIGN_TAXNOMY
CAMPAIGN_OFFER
CAMPAIGN_VERSION
CAMPAIGN_COUPON</td></tr>
<tr><td>BU28.2.2</td><td>The PROMOTION identifier (CAMPAIGN_CD) is defined through the combination of the following components: (still working on final promotion identifier).

<table>
<tr><th>Component</th><th>Length</th><th>Values</th></tr>
<tr><td>In Home Month/Year</td><td>4</td><td>YYMM</td></tr>
<tr><td>Brand Code</td><td>1</td><td>1 = AT, 2 = ATF, 3 = LOFT,4 = LOS</td></tr>
<tr><td>Campaign Name</td><td>10</td><td></td></tr>
</table></td></tr>
<tr><td>BU28.2.3</td><td>The segment KEYCODE identifier is defined through the combination of the following components:

<table>
<tr><th>Component</th><th>Length</th><th>Values</th></tr>
<tr><td>Client Type Code</td><td>3</td><td>COR 0-12 months
REA 25+ and 0-12 months
NEW 0-12 months and no previous purchase
ATT 13-24 months and not 0-12 months
INA 25+ month
XXX No Recency
2YR 0-24 months
SED Seed
PCG Perm CG
PRO Prospecting
SPE Special Requestors
RFM RFM</td></tr>
<tr><td>Card Member</td><td>2</td><td>CM = Card Member, NM = Non Card Member, BT = Both</td></tr>
<tr><td>Segment Number</td><td>3</td><td></td></tr>
</table>

<table>
<tr><th>Component</th><th>Length</th><th>Values</th></tr>
<tr><td>Source Code</td><td>3</td><td>SEeD, Perm CG, RANdom nth, ADS, PROspecting, MODdelled, SPEcial requestor, INTernal database</td></tr>
<tr><td>Client Type Code</td><td>3</td><td>CORe, REActivation, NEW, ATTrition, INActive, XXX, 2YR, SEeD, Perm CG, PROspecting, SPEcial requestor, RFM</td></tr>
</table></td></tr>
</table>

Card Member	2	CM, NM, BT
Mail/Control Indicator	1	M, C
Segment Number	3	

BU28.3	At a minimum, the system will include data structures to track historical campaign taxonomy at the following levels; • Campaign • Segment • Keycode • Offer • Version • Coupon
BU28.3.1	Campaign is defined on the CAMPAIGN_TAXONOMY table
BU28.3.2	Segment is defined on the CAMPAIGN_TAXONOMY table
BU28.3.3	Keycode is defined on the CAMPAIGN_TAXONOMY table
BU28.3.4	Offer is defined on the CAMPAIGN_OFFER table
BU28.3.5	Version is defined on the CAMPAIGN_VERSION table
BU28.3.6	Coupon is defined on the CAMPAIGN_COUPON table
BU28.4	Ongoing the system will receive data from SAS CI this subject area. 1. SAS CI. – Direct mail marketing promotion history. 2. Campaign taxonomy will be extracted from SAS CI.
BU28.4.1	Campaign taxonomy will be extracted from a table that will be the output of a SAS CI campaign, design still TBD. A backend database process will read the table and pull the data needed to load to the campaign taxonomy tables. Other data elements will be part of the promotion history load process.
BU28.5	The system will retain a rolling 60 months worth of direct mail promotion history data. Records older than 60 months will be archived off the system every month
BU28.5.1	Promotion History is performed after a full database backup and archiving completes
BU28.5.2	Promotion History roll-off occurs during as part of the Monthly process the on the first week of the month
BU28.5.3	Data is deleted from the PROMO_HISTORY table based on PROMOTION_DT that has aged >= 60 months past the current date.
BU28.6	The system will maintain the direct mail promotion history table based on individual id, with the ability to do analysis at household id as well as address id, both on the “as was” values and current values.
BU28.6.1	The PROMO_HISTORY table includes the following attributes: • POSTAL_CONTACT_ID • INDIV_ID • GAID (Address ID)
BU28.6.2	The POSTAL_CONTACT_ID is used to reference BEST Postal Contact of the Individual at the time the campaign was extracted table.
BU28.6.3	The INDIV_ID/GAID is used to reference the individual / address combination at the time the promotion was executed. When individuals are consolidated through COA processing, they are potentially orphaned from a mart perspective they are re-pointed to the current individual (best postal contact) with a reference to the original postal contact to keep promotion counts accurate as individuals are consolidated.
BU28.7	The system will maintain direct mail promotion history freeze fields to be stored in the data warehouse to retain as-was history; (Exact fields to be determined during design) • IDs • Campaign Data • Model Score, model score predictive data • Zip Code

BU28.7.1	The Data Warehouse maintains the following IDs defined at the time they were loaded (incremental and migration) • POSTAL_CONTACT_ID • INDIV_ID • GAID (Address ID)
BU28.7.2	Campaign Data is persisted within the Campaign Taxonomy structures
BU28.7.3	Model Score Data retained on promotion history is the Model ID, Version, Segment, and Score that was for the specified campaign
BU28.7.4	The ZIP CODE at the time of promotion is loaded based on the BEST postal contact being promoted

29. Promotion Tracking – Email

The system will contain table structures for the purpose of capturing email promotion history and campaign metadata information to be used in the data mart for reporting, campaign analysis, future campaign selection, segmentation and suppression.

#	Design
BU29.1	The system will load CheetahMail send events for the initial migration load of email promotion history. This data goes back to 11/2009.
BU29.1.1	STG_CM_DATA_SENT data is staged and loaded into the Data Warehouse for all historical feeds provided by SDL back to Nov 2009.
BU29.1.2	Email Response data for those same campaigns are also loaded at this time to maintain historical reporting.
BU29.1.3	The ISSUE_ID is used as the campaign KEYCODE for historical (and incremental) campaigns.
BU29.2	Migration campaign taxonomy will be loaded to the system per the design ANN Inc. and Epsilon developed. A cross-reference table from ANN Inc. will be provided to align historical campaigns by issue_id to the associated keycode.
	CAMPAIGN TAXONOMY is loaded for each ISSUE_ID in the IID_KEYS feed with all additional campaign meta-data provided through CheetahMail/SDL
BU29.3	The system will include data structures to track historical campaign taxonomy at the following levels; • Campaign • Segment • Keycode • Offer • Version • Coupon
BU29.3.1	Eventual email campaigns sourced from the CRM solution have full campaign hierarchy as defined for Direct Mail campaigns in **BU28.3**
BU29.3.2	Migration and Incremental taxonomy data is provided in limited format based on what is available in the IID KEYS feed loaded into the CM_ISSUE table. In the mart, the ISSUE_ID is generated as the campaign keycode when the campaign only comes from SDL/CM. For these campaigns only the following attributes are available: • Keycode (ISSUE_ID) • Issue Name (campaign description) • Mailing Name (campaign name) • Brand Code (decoded from promotion PID) • Mail Date (derived from sent date) • Channel (hardcoded)

	• Source (hardcoded)
BU29.4	Ongoing the system will receive daily send events for CheetahMail to load to the email promotion history table.
BU29.5	The system will retain a rolling 60 months worth of email promotion history data. Records older than 60 months will be archived off the system every month
BU29.5.1	Promotion History is performed after a full database backup and archiving completes
BU29.5.2	Promotion History roll-off occurs during as part of the Monthly process the on the first week of the month
BU29.5.3	EMAIL_RESPOSNE data is deleted where RESPONSE_DT has aged past 60 months relative the current date
BU29.5.3	EMAIL_RESPONSE data is deleted where PROMOTION_DT of the promotion data has aged past 60 months relative the current date
BU29.5.4	EMAIL_PROMOTION data is deleted where PROMOTION_DT of the promotion data has aged past 60 months relative the current date
BU29.6	The system will maintain the email promotion history table based on email address.
BU29.6.1	The following tables are loaded and keyed on EMAIL_ID. This assumes that the email address on these feeds was already loaded as part of the current or prior SDL SUBS or DEMO feeds (or, by originally loading through one of the other ACCOUNT sources). • EMAIL • BRAND_EMAIL • EMAIL_PROMOTION • EMAIL_RESPONSE • EMAIL_BRAND_AGGREGATE
BU29.6.2	In the event that an email address was not found in EMAIL (promo, response, aggregate, unsubs), the inbound record is rejected on the error to decode EMAIL_ID
BU29.7	The system will maintain email promotion history freeze fields to be stored in the data warehouse to retain as-was history; (Exact fields to be determined during design) • IDs • Campaign Data • Model Score, model score predictive data • Personalization fields
BU29.7.1	The following IDs are retained on EMAIL_PROMOTION as attributes of an email address at the time of promotion: • EMAIL_ID • INDIV_ID – where an email address has a direct relationship to a physical address (future/optional use) • ISSUE_ID – to support eventual possible multi subject-line testing within a single key code
BU29.7.2	Campaign Data is persisted in the Campaign Taxonomy structures
BU28.7.3	Support for the scoring of Email Addresses is reserved as a **Future Enhancement**
BU28.7.4	Personalization “freeze” Fields are to be determined once email campaigns are sourced directly from the CRM solution as a **Future Enhancement**
BU29.8	The system will retain campaign identifiers internal to CheetahMail.
BU29.8.1	The Data Warehouse and Data Marts retain the following CM Identifiers • ISSUE_ID on CM_ISSUE, EMAIL_PROMOTION, and EMAIL_RESPONSE • PID on BRAND_EMAIL
BU29.9	The system will retain both bulk and event based email promotions from CheetahMail.
BU29.9.1	The Bulk Send Events are identified as Event Type = 1 in the events feed and loaded into

	RAW_SDL_DATA
BU29.9.2	The Event Based Send Events are identified as Event Type = 2 in the events feed and loaded into RAW_SDL_DATA_SENT
BU29.9.3	Both Email promotion sources are loaded into EMAIL_PROMOTION and are identified through their CheetahMail ISSUE_IDs

30. Email Response

The system will capture email response data to be used in the data mart for reporting, campaign analysis, future campaign selection, segmentation and suppression.

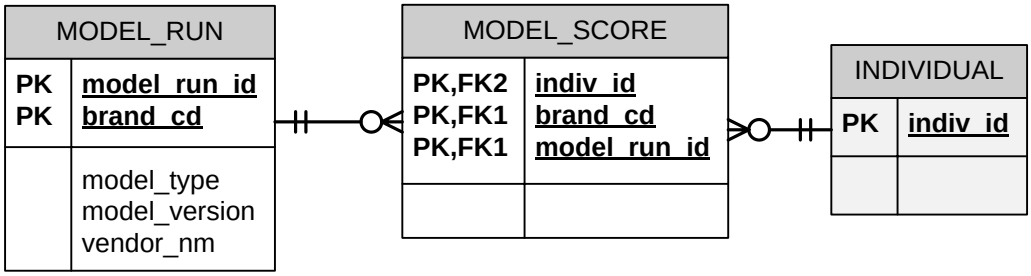
All response data collected by CheetahMail/SDL is retained in the CRM database for each campaign for as long as they are tracked within those systems. Epsilon does not filter upon load any activity data based on relative campaign date parameters.

#	Design
BU30.1	CheetahMail will provide all the email response data for the initial migration load as well as the on-going data. Migration data goes back to 11/2009.
BU30.1.1	SDL provides historical event extracts that are provided for initial load. The expectation is that every email address occurring in the send/response event feeds is found in at least one of the historical SUBS/UNSUBS feeds., however in the event that a campaign is deployed to an email address that was never subscribed, the SENDS are included as an email source (as are the responses). This will insure that any email address we encounter has an ID field already defined in the EMAIL table
BU30.1.2	The feeds are split and loaded in exactly the same manner as the incremental loads.
BU30.2	The keycode field on all CheetahMail event feeds will be used to link back to the promotion history created from SAS CI, initially this keycode field will contain nulls
BU30.2.1	The keycode field contains NULL in the inbound feeds until such time as email campaigns are sourced from the CRM solution
BU30.2.2	In the absence of a campaign keycode, the CheetahMail campaign identifier (ISSUE_ID) is used as a substitute to tie email promotion response data
BU30.3	Email activity should be available each day for the preceding day's events occurring through the data cut-off time of TBD. Specific details to be defined during the Design Phase and captured in the Solution Design Document.
BU30.3.1	Email Activity is collected at Cheetah Mail in real time. It is extracted for load into SDL on the following day. This data is then made available to the CRM system for loading on the day after receipt. This means that the data is not available to the end user in the mart for up to 3 days after the activity occurs.
BU30.4	The system will relate the email response activity (opens, bounces, clicks, unsubs and transactions), back to an email promotion record and associated promotional hierarchy within the data mart for reporting purposes.
BU30.4.1	The relationship between EMAIL_PROMOTION and EMAIL_RESPONSE is KEYCODE (issue_id) and EMAIL_ID (email address). Both fields are part of the Primary Key of these tables.
BU30.5	The system will receive hard bounce rejects, the identification of soft bounce rejects are out of scope within the solution.
BU30.5.1	Hard Bounce rejects are included in the update of the EMAIL.UNDELIVERABLE_IND. There is not an identifier retained in the solution specific to hard bounces at the email level.
BU30.5.2	Hard Bounce rejects are identified in the EMAIL_PROMTION table as having an ISSUE_ID that is NOT NULL, but a DELIVERED_IND = 0 (not resultcode = "S")
BU30.6	The system will retain the clicked URL that will be provided on the event feed for a clicked response type.
BU30.6.1	The EMAIL_RESPONSE table retains the 4000 byte URL provided on the CLICK events
BU30.7	The system will receive transaction response events which will be used to directly attribute

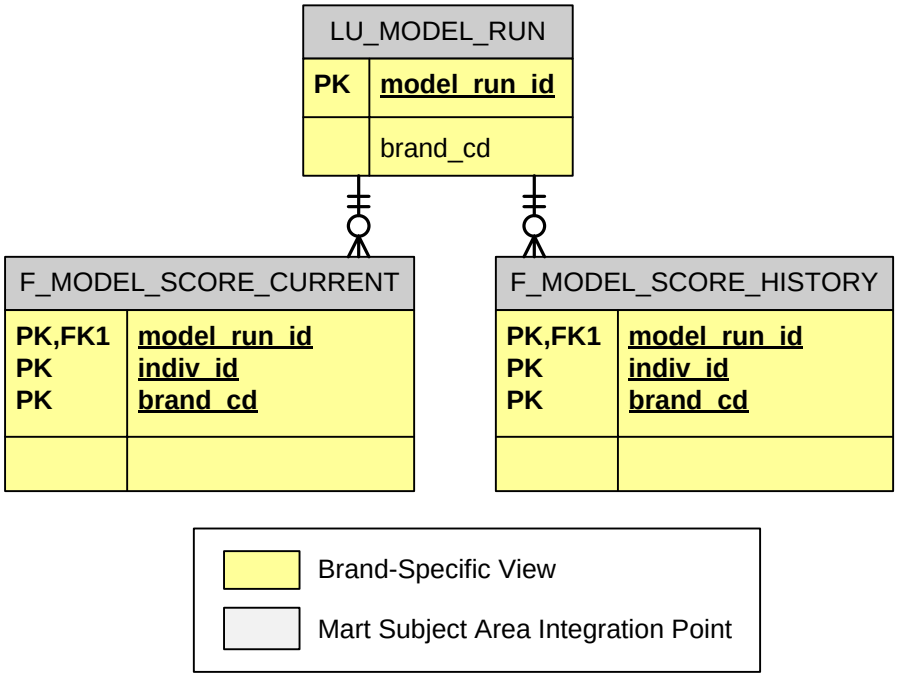
	order data provided by ATG or AuditWorks through a transaction identifier provided to CheetahMail.
BU30.7.1	The transaction ID that is provided on the Transaction Event feed is a sequencing number that does not join directly to ATG transaction IDs. The transaction events are retained on F_EMAIL_RESPONSE for each campaign. The TXN_ID can be used to aggregate frequency; the TXN_AMT field can be used to aggregate monetary values. There is currently no mechanism by which a direct link to F_TXN_HEADER can be established.

Model Score Subject Area

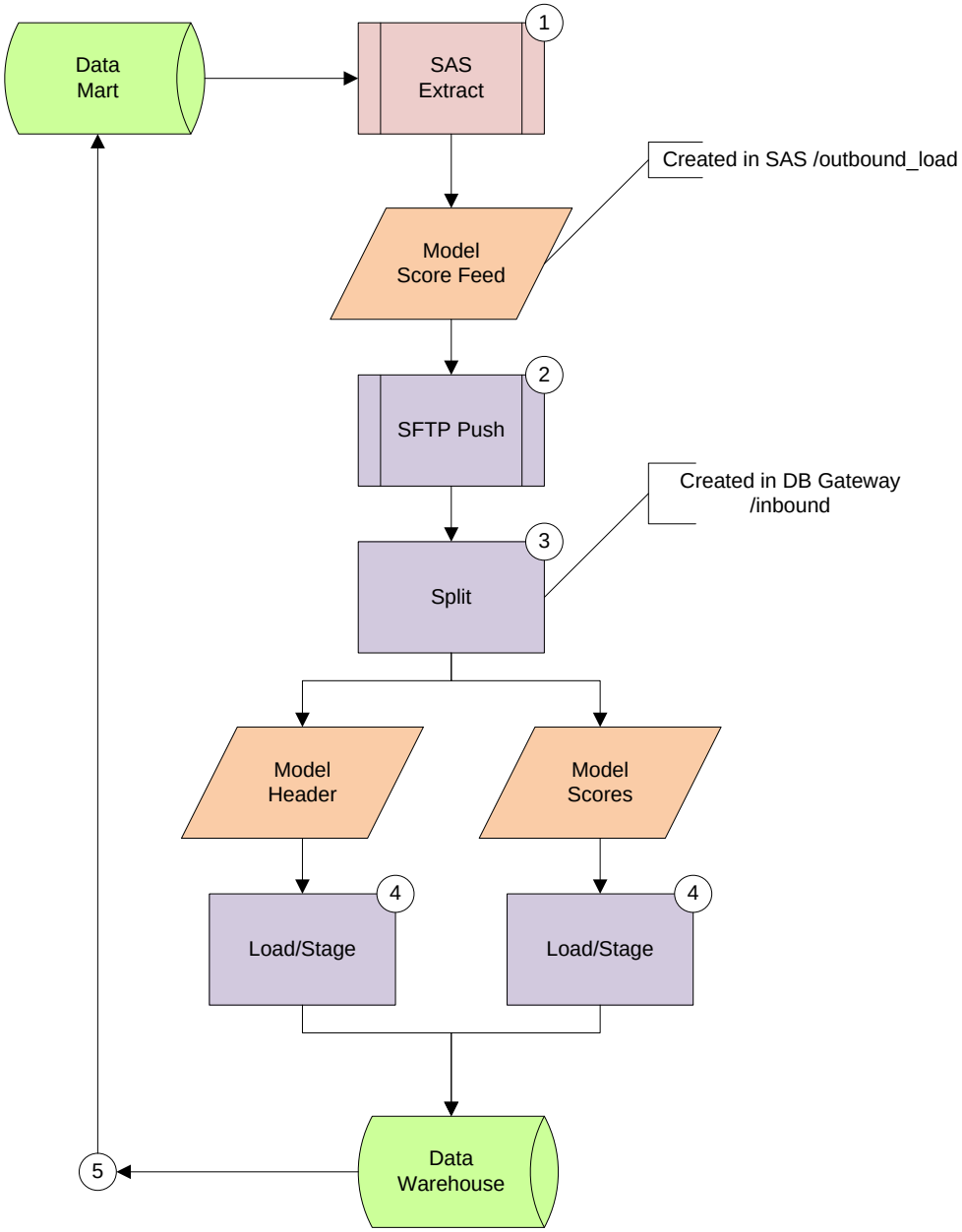
Model Score Entity Key Relationships – Data Warehouse



Model Score Entity Key Relationships – Data Mart



Model Score Data Warehouse Update Process Flow



Model Score Data Warehouse Update Process Annotations

1. The model score algorithm is created in SAS to score INDIVIDUALS (or score households/addresses and inherit scores at the Individual level). The extract contains both Model Score meta-data (identifiers) as well as the scores/segmentations
2. The feed is placed in the /outbound_load. A process polls for new files and PUSHs then via SFTP to the database loading area. The feed is picked up during the next update cycle and identified by naming convention.
3. The meta-data fields are split off from the scored fields.
4. The data is staged and loaded into the Data Warehouse.

a. The meta-data table is populated

b. The CURRENT model score table is deleted for the most recent valid version of the model

c. The HISTORY and CURRENT model score tables are loaded with the new version

5. The Data Mart is refreshed and the scored data is made available to the end user.

31. Model Score

The system stores model score data at the Client (individual) level. It includes the model number, date the model was scored, model segment, decile, and rank data.

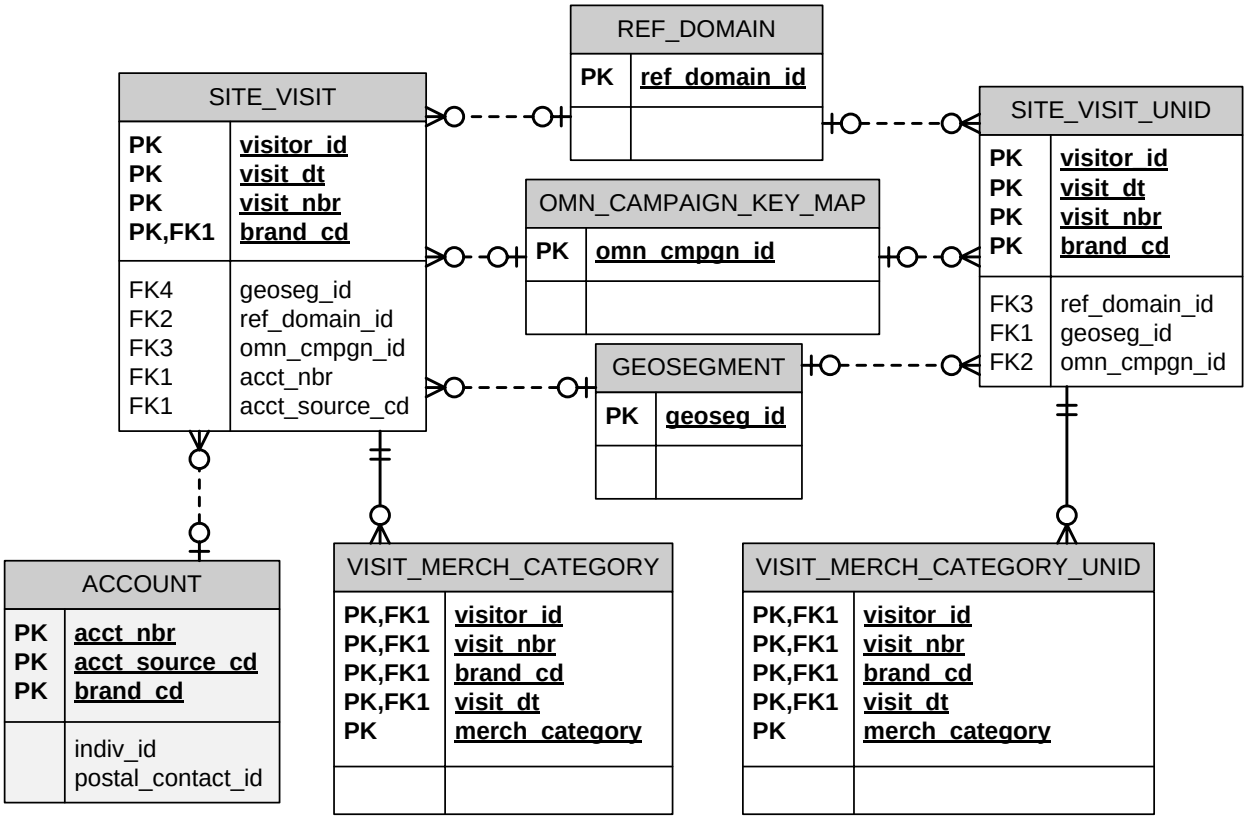
<table>
<tr><th>#</th><th>Design</th></tr>
<tr><td>BU31.1</td><td>The system has an automated process for loading model scores to the model score table.</td></tr>
<tr><td>BU31.1.1</td><td>A feed with the following attributes are extracted from the SAS modeling tool.
<table>
<tr><th>COLUMN NAME</th><th>TYPE NAME</th><th>COLUMN SIZE</th></tr>
<tr><td>MODEL_RUN_ID</td><td>NUMBER</td><td>10</td></tr>
<tr><td>BRAND_CD</td><td>VARCHAR2</td><td>16</td></tr>
<tr><td>MODEL_TYPE</td><td>VARCHAR2</td><td>40</td></tr>
<tr><td>MODEL_NM</td><td>VARCHAR2</td><td>200</td></tr>
<tr><td>MODEL_VERSION</td><td>VARCHAR2</td><td>40</td></tr>
<tr><td>MODEL_RUN_DT</td><td>DATE</td><td>7</td></tr>
<tr><td>VENDOR_CD</td><td>VARCHAR2</td><td>20</td></tr>
<tr><th>COLUMN NAME</th><th>TYPE NAME</th><th>COLUMN SIZE</th></tr>
<tr><td>INDIV_ID</td><td>NUMBER</td><td>13</td></tr>
<tr><td>MODEL_RUN_ID</td><td>NUMBER</td><td>10</td></tr>
<tr><td>BRAND_CD</td><td>VARCHAR2</td><td>16</td></tr>
<tr><td>MODEL_SEGMENT</td><td>VARCHAR2</td><td>80</td></tr>
<tr><td>MODEL_SCORE</td><td>NUMBER</td><td>12</td></tr>
<tr><td>MODEL_DECILE</td><td>NUMBER</td><td>2</td></tr>
<tr><td>MODEL_RANK</td><td>NUMBER</td><td>10</td></tr>
</table></td></tr>
<tr><td>BU31.1.2</td><td>The feed is generated in TAB-delimited format with the following naming convention:
STG_CRM_MODEL_yyyymmdd_seq.dat</td></tr>
<tr><td>BU31.1.3</td><td>The feed is created in the SAS /outbound_load folder. The file_poller process scans for files recognizing new files. The file is SFTP’d and transfer is validated with a chksum. Upon successful execution, the file is then moved into the archive folder.</td></tr>
<tr><td>BU31.1.4</td><td>The feed is this split and loaded similar to any feed into the solution and subject to the updated rules defined in the Source to Target mapping documents</td></tr>
<tr><td>BU31.2</td><td>The system will retain a rolling TBD months worth of model score history data. Records older than TBD months will be archived off the system every month. Specific details to be defined during the Design Phase and captured in the Solution Design Document.</td></tr>
<tr><td>BU31.2.1</td><td>The data warehouse rolloff process executes monthly. Model Score identified as having a MODEL_RUN_DT greater than the specified retention period are archived through the</td></tr>
</table>

	standard Data Warehouse backup process and deleted from the table. During the subsequent update cycle rolled off models are no longer available in the Data Mart.
BU31.3	The system will run ANN's RFM model score algorithm and load the score to the model score table. Details for the algorithm to be provided during design phase.
BU31.3.1	The ANN RFM algorithm is executed as part of the monthly update process occurring on the first week of the calendar month after performing roll-offs
BU31.3.2	Decile all INDIVIDUALS (through their underlying ACCOUNT relationships to TXN_HEADER) where they have Transaction ORDER_DT occurring within the 12 CALENDAR months prior to the current update date: The following fields are pulled from F_TXN_HEADER in the Data Mart (rolled up from TXN_ITEM) • RECENCY – max(ORDER_DT) • FREQUENCY – Count ALL orders • MONETARY - TXN_NET_AMT
BU31.3.3	A weighted score is generated for these decile scores as follows: • RECENCY * 0.2 • FREQUENCY * 0.3 • MONETARY * 0.5
BU31.3.4	The resulting score is then deciled
BU31.3.5	The scores are loaded into MODEL_SCORE_CURRENT (and HISTORY) as follows: • MODEL_SEGMENT – the final decile • MODEL_SCORE – the weighted score • MODEL_DECILE – the final decile • MODEL_RANK – the final decile
BU31.3.6	The model meta-data is loaded as follows: • MODEL_RUN_ID – increment for each model/version in HISTORY • BRAND_CD – AT, LOFT, ATF, LOS (respectively) • MODEL_TYPE – "RFM" • MODEL_NM – "12M ENT WEIGHTED RFM 20/30/50" • MODEL_VERSION – "20120510" – this will increment if a Change Request is submitted to modify the algorithm • MODEL_RUN_DT – current date
BU31.4	The system will load specific segments identified by ANN from the MarketWork's Customer Attribute to the model score table. **See Appendix F**
BU31.4.6	The MarketWorks segmentation data is loaded into the MODEL_SCORE_HISTORY table as defined in the **Stage to Data Warehouse mapping document.**
BU31.5	The system will determine if there are multiple records for a client for the same model score, which one to keep in the model score table. This will be done by keeping the 'best' model score. The 'best' is determined by the lowest model score. 1=best,
BU31.5.1	As individuals are consolidated over time the Quarterly AGILITY XREF is used to CURRENT and HISTORICAL scored data
BU31.5.2	The MODEL_RANK is used to select the surviving individual with a random individual selected in the event of a tie. Select * From (Select indiv_id, model_run_id, (row_number() over (partition by indiv_id,

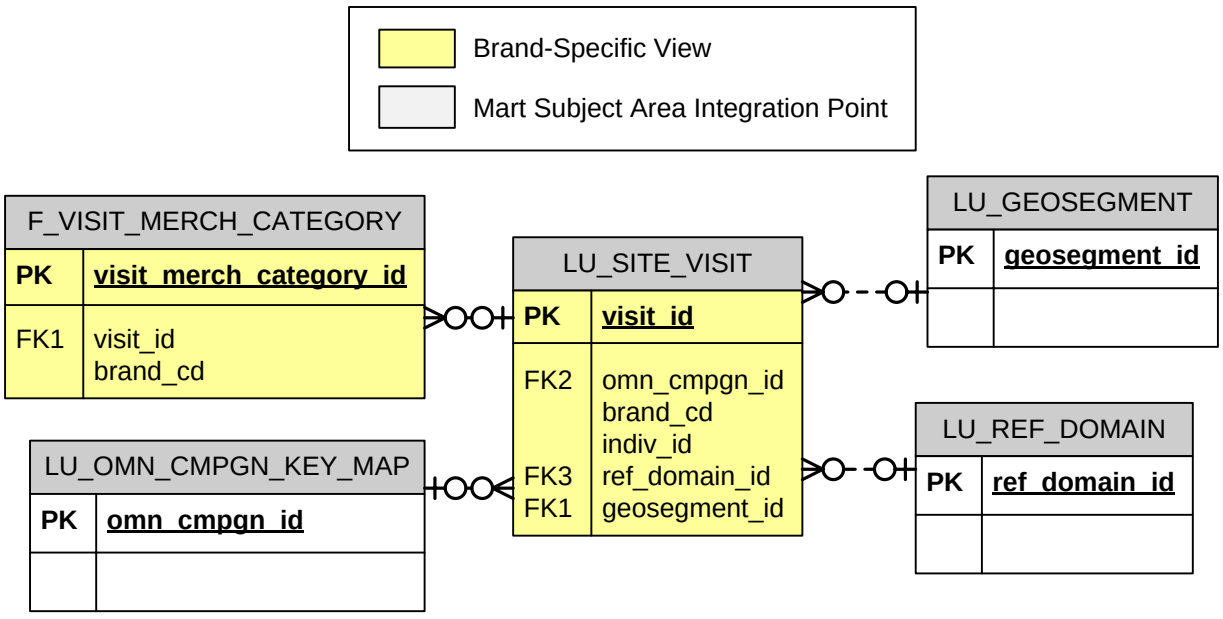
	model_run_id order by model_rank)) as rnk from dw_owner.model_score) a Where a.rnk = 1
BU31.6	The system will maintain the new customer lifetime value model score that is built in SAS by ANN INC. This will provide multiple model scores for each client across each brand/channel.
BU31.6.1	The ongoing lifetime scores are loaded by the process described in **BU31.1**
BU31.7	The system will maintain other predictive model scores (e.g. assimilation, cross-brand, multi-channel, attrition, etc). The models will be built in SAS by ANN INC.
BU31.7.1	The solution supports any customized scoring algorithm for which the predictive variables exist in the Data Mart.
BU31.7.2	The scores must conform to the interface agreement defined in **BU31.1**

Call Center and Online Activity Subject Area

Online Activity Entity Key Relationships – Data Warehouse



Online Activity Entity Key Relationships – Data Mart



32. Call Center

#	Design
BU32.1	The system will retain disposition data from the call center. This data will be supplied by RightNow. Only 2 dispositions per phone call are retained.
BU32.1.1	In the Data Warehouse, the call center data is stored in the CSC_INCIDENT table as a relationship to ACCOUNT
BU32.1.2	In the Data Mart, the call center data is stored in the LU_CSC_INCIDENT table as a dimension of SUM_INDIVIDUAL
BU32.1.3	The incident disposition is retained in the INCIDENT_DISPOSITION field. The initial disposition is initially loaded and potentially later updated with the final disposition based on the INCIDENT_NBR
BU32.2	The system will load 5 years of historical data from RightNow.
BU32.2.1	RightNow provides a one-time historical load dataset in the same format used to load incremental data.
BU32.2.2	The incremental feed updates the historical load based on incident number and date
BU32.3	The system will retain such data as: • Call Type • Category ID • Disposition ID • Emotive Rating
BU32.3.1	The Call Type is retained in the 40-byte CALL_TYPE field
BU32.3.2	The Category ID is retained in the 150-byte INCIDENT_CATEGORY field as a description of the ID
BU32.3.3	The Disposition ID is retained in the 150-byte INCIDENT_DISPOSITION field as a description of the ID
BU32.3.4	The Emotive Rating is retained in the numeric EMOTIVE_RATING field
BU32.3.5	All of the above attributes are updated with each load based on the INCIDENT_NBR and ACTIVITY_DT

33. Web Behavior

#	Design
BU33.1	The system will collect information regarding visits to Ann Taylor and LOFT websites.
BU33.1.1	Omniture provides an aggregate of the low level clickstream data tracked upon each site visit. The aggregate is performed at the Date, Visitor ID, and Visit number level.
BU33.1.2	This daily feed is loaded into SITE_VISIT_UNID and normalized into the structures described under **BU33.3**
BU33.2	The system will collect information regarding merchandise category captured during visits.
BU33.1.2	Omniture provides an aggregate of the low level ClickStream data tracked upon each site visit. The aggregate is performed at the Date, Visitor ID, Visit Number, and Product Category level
BU33.1.3	This daily feed is loaded into the VISIT_MERCH_CATEGORY_UNID table
BU33.3	The system will collect descriptive information regarding marketing efforts related to visits.
BU33.3.1	The REF_DOMAIN table stores the Referring Domain for each visit
BU33.3.2	The GEOSEGMENT table stores the Omniture-derived Geographic data related to the visit
BU33.3.3	The OMN_CAMPAIGN_KEY_MAP stores up to five (5) Omniture campaign identifiers that drove the site visit. This is parsed out from a single field of “>” delimited values. The ordering of the values in the following fields indicates the campaign sequence on the

	source feed: • STACKING1_CMPGN_ID • STACKING2_CMPGN_ID • STACKING3_CMPGN_ID • STACKING4_CMPGN_ID • STACKING5_CMPGN_ID
BU33.4	The system will tie visit data back to an individual if there was an associated order. These are referred to as identified visits.
BU33.4.1	Once the daily transaction load completes an attempt is made to associate unidentified transactions with the current set of transactions
BU33.4.2	The SITE_VISIT_UNID.TXN_NBR field is used to join to ATG TXN_HEADER TXN_NBR
BU33.5.3	Upon successful join, the SITE_VISIT_UNID data is loaded into SITE_VISIT and removed from the UNID counterpart. The ACCT_NBR found on that transaction is then permanently assigned to the visit.
BU33.5.4	In the event that the system later loads the same transaction assigned to a different VISITOR_ID, there is no attempt made to realign/reassign existing Visit ACCOUNTs
BU33.5.5	When a visit has been identified it's VISIT_MERCH_CATEGORY counterpart is similarly moved into the identified tables.
BU33.5.6	Only the identified visits are promoted to the Data Mart each refresh.
BU33.5	The system will continuously try to tie back unidentified visits to individuals for TBD amount of time before the data is archived off the system. Specific details to be defined during the Design Phase and captured in the Solution Design Document.
BU33.5.1	The retry rates are to be determines once the process run regularly in production for a period of time and we're best able to assess based on match performance and storage availability.
BU33.6	Archiving strategy will be determined later during the database implementation – due to unknown volumes at this time.
BU33.6.1	Once an unidentified visit has aged past the period defined in **BU33.5** based on the VISIT_DT, the table is backed up via the database backup and archiving mechanism, and the unidentified visits are deleted.
BU33.6.2	All orphaned VISIT_MERCH_CATEGORY records are then deleted.
BU33.6.2	The number of deleted rows is retained in the ELT_OWNER.PROESS_SAVESTATS table for historical auditing purposes.

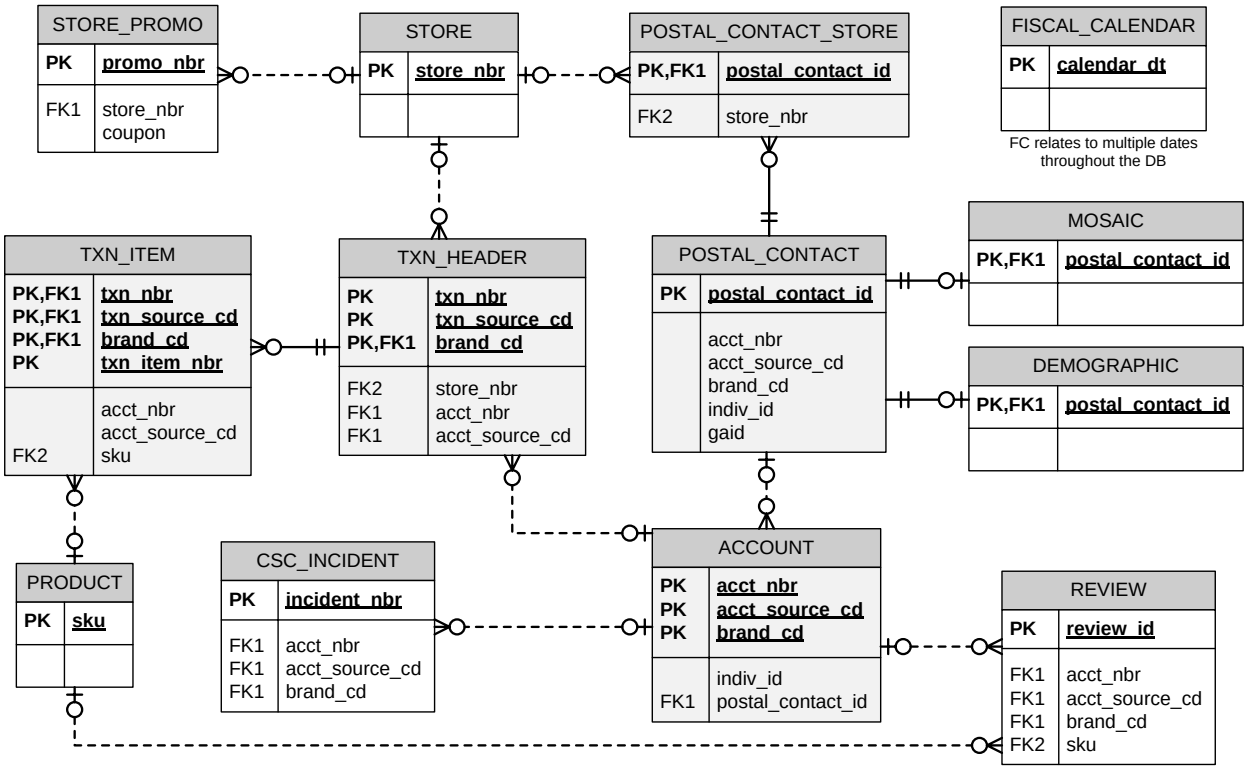
34. Product Reviews

#	Design
BU34.1	The system will retain product review data.
BU34.1.1	In the Data Warehouse, the review data is stored in the REVIEW table as a relationship to ACCOUNT
BU34.1.2	In the Data Mart, the review data is stored in the F_REVIEW table as a a fact
BU34.2	The system will have the ability to link the BV Client reviews back to the ATG Ecommerce order. This would be done using the profile id on BV and the order id on ATG.
BU34.2.1	The integration of the Ratings data in the Data Warehouse is through the ACCOUNT table. The REVIEWER_ID in the ratings feed joins to the ACCT_NBR (Profile_ID) from the ATG PROFILE feed
BU34.2.2	The ratings feed also has a foreign key relationship to the SAP PRODUCT feed on PRODUCTID to SKU
BU24.2.3	The RATINGS feed does not include an attribute used to link to a specific ATG transaction, though one could likely be inferred trough the SKU, PROFILE_ID, and Review Date

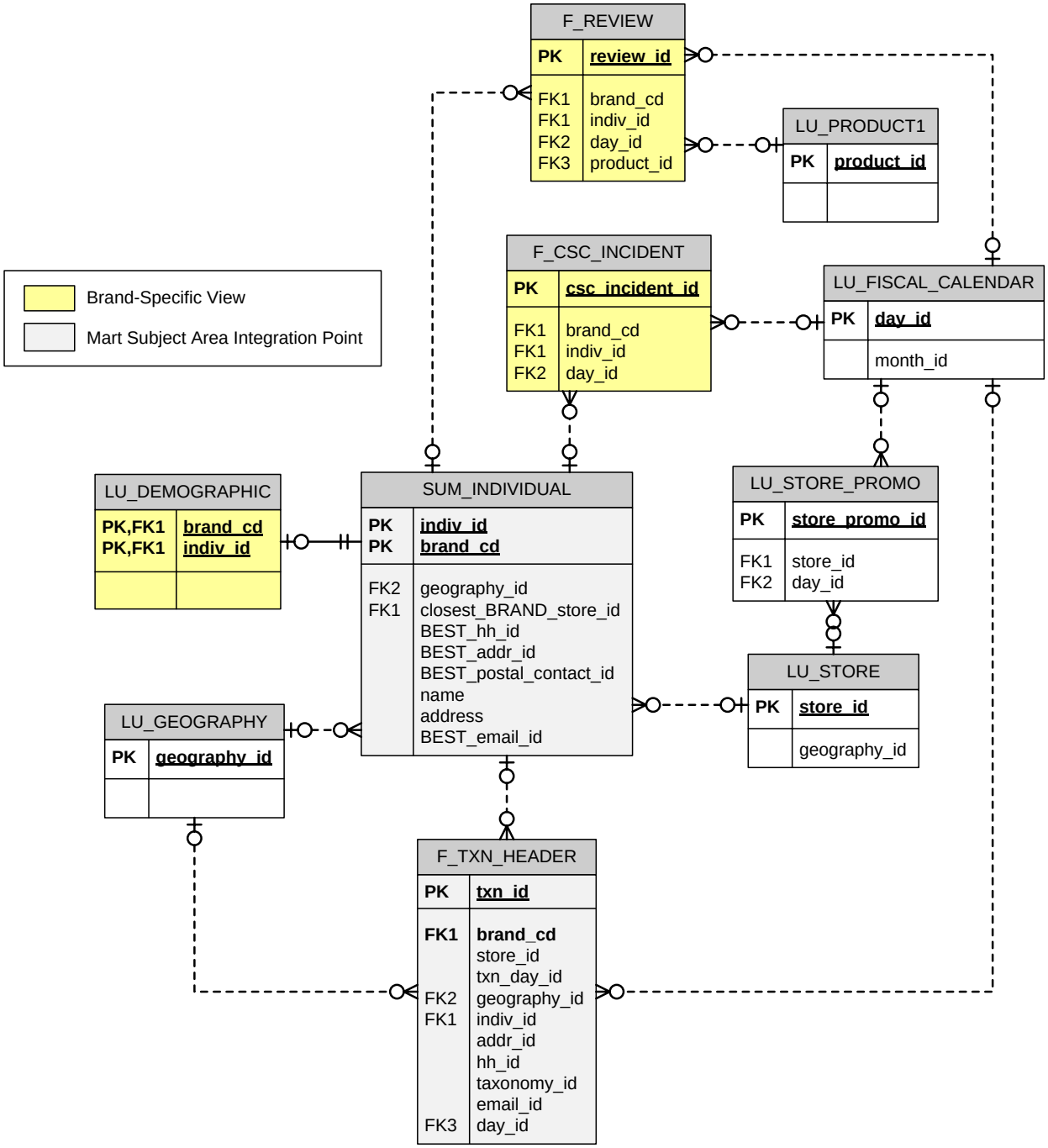
BU34.3	The system will retain a minimum of the reviewer summarized data which contains information such as; • Feedback Qty – Number of times the Client gave product feedback • Negative Feedback Qty – Number of times the Client gave negative feedback • Positive Feedback Qty – Number of times the Client gave positive feedback • Inappropriate Feedback Qty – Number of times the Client gave inappropriate feedback
BU34.3.1	RATING and RATINGRANGE are attributes on the LU_REVIEW table which is a dimension of SUM_INDIVIDUAL. These can be aggregated to infer Positive and Negative feedback at that level.
BU34.3.2	There is no method by which INAPPROPRIATE reviews submitted by a reviewer available in the underlying data.
BU34.3.2	The feedback counts provided by BazaarVoice are aggregates based on user community response to the REVIEW NUMFEEDBACKS NUMPOSITIVEFEEDBACKS NUMNEGATIVEFEEDBACKS NUMINAPPROPRIATEFEEDBACKS

Reference Subject Area

Reference Entity Key Relationships – Data Warehouse



Reference Entity Key Relationships – Data Mart



35. Reference

#	Design
BU35.1	The system will retain Product Reference data as follows: a. A master product reference table containing the following information: sku, style, department, class, color, size, vendor, season, merchandising category, and additional style attributes as outlined in the style feed.

	b. The primary source of product information on an ongoing basis will be from SAP. The historical MarketWorks data may contain reference information prior to what is provided on SAP. As part of the migration process, the system will augment the master product reference with MarketWorks data not contained in the SAP feeds. In all cases where there is overlap between MarketWorks and SAP, the SAP version will take precedence. c. The system will not attempt to align historical hierarchy information from MarketWorks with the hierarchy contained in SAP as part of the migration. d. For ongoing updates, the SAP product information will be provided as a full file. The system will add any new products encountered and update any matching products. Products in the data warehouse that are not present in the latest product update process will not be altered or deleted.
BU35.1.1	The PRODUCT table contains the SAP STYLE, PRODUCT, and VENDOR data denormalized down to the SKU
BU35.1.2	The PRODUCT table is augmented by any STYLE, COLOR, SIZE combinations for transactions in MarketWorks that were not found in the initial full SAP product data
BU25.1.3	The SAP Product feed is Upserted into the data warehouse on the SKU key
BU25.1.4	The PRODUCT table is augmented with SKUs from transaction that are not found in the SAP product table to prevent transaction orphans
BU35.2	The system will retain Store Reference data as follows: a. The system will maintain a store reference table containing the attributes in the SAP feed. b. Open stores, defined as those stores without a closing date and where the opening date is not set in the future will be sent to Agility for use in closest store assignment. c. Online store numbers 611, 612, 618, 619 will be excluded from being sent to Agility for use in the closest store assignment. d. The system shall distinguish between US and Non-US Retail Store locations
BU35.2.1	The STORE table contains the data from the SAP Store feed. It is upserted based on the STORE NBR
BU35.2.2	The STORE table is extracted on initial load and immediately prior to each quarterly AGILITY refresh for use in store proximity calculations. The online stores 611, 612, 618, and 619 are excluded from the extraction.
BU35.2.3	The STORE_STATUS_CD attribute is derived with values of “O” of “C” based on the rule above.
BU35.2.4	The LU_STORE table is promoted to the Data Mart as a dimension of Transaction and Individual
BU35.2.5	The STORE_PROMO table contain promotion information related to a STORE keyed on Promotion Number
BU35.2.6	The STORE_PROMO tables relates to transactions through the PROMO_ID and COUPON_CD derived from AuditWorks POS transactions and stored on TXN_ITEM_DISCOUNT. The PROMO_NBR join is required The COUPON_CD (DISC_SERIAL_NBR) may join in one of the following ways • If the AW coupon code length = 12, join on the first 3 bytes of it to 3 digit codes associated with the PROMO_NBR in SAP • If the AW coupon code length = 0/null, permit join on ONLY PROMO_NBR

	• Otherwise join on the full 8, 9, 10, 11 byte coupon code All of the above joins are qualified for the transaction date occurring within the SAP Promo offer start and end dates. The join in the mart is then simplified to join on a singe PROMO_ID field while the native PROMO_NBR and COUPON_CDs from both AW and SAP are retained on F_TXN_ITEM_DISCOUNT and LU_STORE_PROMO respectively.
BU35.2.7	The LU_STORE_PROMO table is promoted to the data mart as a relationship to STORE and TXN_ITEM_DISCOUNT
BU35.3	The system will retain Employee Reference data as follows: a. The employee reference will be used in determining which individuals and households contain an active employee. This reference will contain an account_key (brand_cd, account source, account number) for the account containing the contact information along with the location. b. The ongoing feed will only contain active employees. The system will maintain a status of 'A' for the active associates on the feed and a status of 'I' for the employees loaded previously to the system that are no longer on the feed. c. The system will retain the date that the status of an employee was made inactive. d. The brand code for the employees loaded into the account table will be 'ENT'.
BU35.3.1	The Associate feed is loaded into WRK_SRC_ACCOUNT as described in the **Customer section**. As such, these are found in the Data Warehouse ACCOUNT, POSTAL_CONTACT, and INDIVIDUAL tables with a ACCT_SOURCE_CD = “”EMP” and a BRAND_CD = ”ENT”
BU35.3.2	Associates are found in the Data Mart in the SUM_INDIVIDUAL table with an EMP_IND = 1
BU35.4	The system will retain Fiscal Calendar Reference data. ANN will send a complete full refresh file on an annual basis.
BU35.4.1	The FISCAL_CALENDAR table is loaded from the data provided on an ad-hoc basis by ANN INC
BU35.4.2	The table includes a row for 1900-01-01 to facilitate joins in the mart for dates that do not exist in the fiscal calendar
BU35.4.3	The LU_FISCAL_CALENDAR table is promoted to the Data Mart as a relationship to date fields in the database may include those sourced from the following Data Warehouse date fields o BRAND_EMAIL.EMAIL_ACQ_DT o CSC_INCIDENT.INCIDENT_DT o DEMOGRAPHIC.BNKRPT_DOCKET_DT o DEMOGRAPHIC.BUY_CATG_LAST_OFFLINE_ORD_DT o DEMOGRAPHIC.BUY_CATG_LAST_ONLINE_ORD_DT o DEMOGRAPHIC.BUY_CATG_LAST_ORD_DT o DEMOGRAPHIC.BUY_RET_LAST_ORD_DT o DEMOGRAPHIC.CRW_ACT_LAST_POST_DT o DEMOGRAPHIC.SURV_DT o EMAIL.UNDELIVERABLE_DT o EMAIL_BRAND_AGGREGATE.LAST_CLICK_DT o EMAIL_BRAND_AGGREGATE.LAST_OPEN_DT o EMAIL_BRAND_AGGREGATE.LAST_SEND_DT o EMAIL_BRAND_AGGREGATE.LAST_SUB_DT o EMAIL_BRAND_AGGREGATE.LAST_TRANS_DT

	○ EMAIL_BRAND_AGGREGATE.LAST_UNSUB_DT ○ EMAIL_PROFILE.ACTIVITY_DT ○ EMAIL_PROFILE.AT_BROWSE_REMARKET_DT ○ EMAIL_PROFILE.AT_REMARKET_DT ○ EMAIL_PROFILE.BIRTHDAY_3RD_PARTY_DT ○ EMAIL_PROFILE.BIRTHDAY_PREF_DT ○ EMAIL_PROFILE.CREATED_DT ○ EMAIL_PROFILE.LOFT_BROWSE_REMARKET_DT ○ EMAIL_PROFILE.LOFT_REMARKET_DT ○ EMAIL_PROFILE.MATERNITY_BROWSE_DT ○ EMAIL_PROFILE.MATERNITY_DT ○ EMAIL_PROFILE.MATERNITY_DUE_DT ○ EMAIL_PROFILE.MATERNITY_ORIG_PREF_DT ○ EMAIL_PROFILE.MATERNITY_PREF_DT ○ EMAIL_PROFILE.MODIFIED_DT ○ EMAIL_PROFILE.PETITE_AT_BROWSE_DT ○ EMAIL_PROFILE.PETITE_AT_PREF_DT ○ EMAIL_PROFILE.PETITE_AT_PUR_DT ○ EMAIL_PROFILE.PETITE_LOFT_BROWSE_DT ○ EMAIL_PROFILE.PETITE_LOFT_PREF_DT ○ EMAIL_PROFILE.PETITE_LOFT_PUR_DT ○ EMAIL_PROFILE.STUDENTS_BROWSE_DT ○ EMAIL_PROFILE.STUDENTS_GRAD_DT ○ EMAIL_PROFILE.STUDENTS_PREF_DT ○ EMAIL_PROFILE.TALL_AT_BROWSE_DT ○ EMAIL_PROFILE.TALL_AT_PREF_DT ○ EMAIL_PROFILE.TALL_AT_PUR_DT ○ EMAIL_PROFILE.TALL_LOFT_BROWSE_DT ○ EMAIL_PROFILE.TALL_LOFT_PREF_DT ○ EMAIL_PROFILE.TALL_LOFT_PUR_DT ○ EMAIL_PROFILE.TEACHERS_BROWSE_DT ○ EMAIL_PROFILE.TEACHERS_ORIG_PREF_DT ○ EMAIL_PROFILE.TEACHERS_PREF_DT ○ EMAIL_PROFILE.TRANS_T2P_AT_FIRST_DT ○ EMAIL_PROFILE.TRANS_T2P_AT_LAST_DT ○ EMAIL_PROFILE.TRANS_T2P_LOFT_FIRST_DT ○ EMAIL_PROFILE.TRANS_T2P_LOFT_LAST_DT ○ EMAIL_PROFILE.WEDDING_BROWSE_DT ○ EMAIL_PROFILE.WEDDING_DT ○ EMAIL_PROFILE.WEDDING_ORIG_PREF_DT ○ EMAIL_PROFILE.WEDDING_PREF_DT ○ EMAIL_PROFILE.WEDDING_PUR_DT ○ EMAIL_PROMOTION.PROMOTION_DT ○ EMAIL_RESPONSE.RESPONSE_DT ○ INDIVIDUAL.ADDR_END_DT ○ INDIVIDUAL.ADDR_START_DT ○ MODEL_RUN.MODEL_RUN_DT ○ PROMO_HISTORY.PROMOTION_DT

	○ REVIEW.FIRST_PUBLISH_DT ○ REVIEW.LAST_PUBLISH_DT ○ SITE_VISIT.VISIT_DT ○ STORE.CLOSED_DT ○ STORE.OPEN_DT ○ TXN_HEADER.RMB_DT ○ TXN_HEADER.TXN_COMPLETE_DT ○ TXN_HEADER.TXN_DT ○ TXN_ITEM.ITEM_LAST_SHIP_DT ○ TXN_ITEM.ITEM_RET_DT ○ TXN_TNDR.TNDR_DT

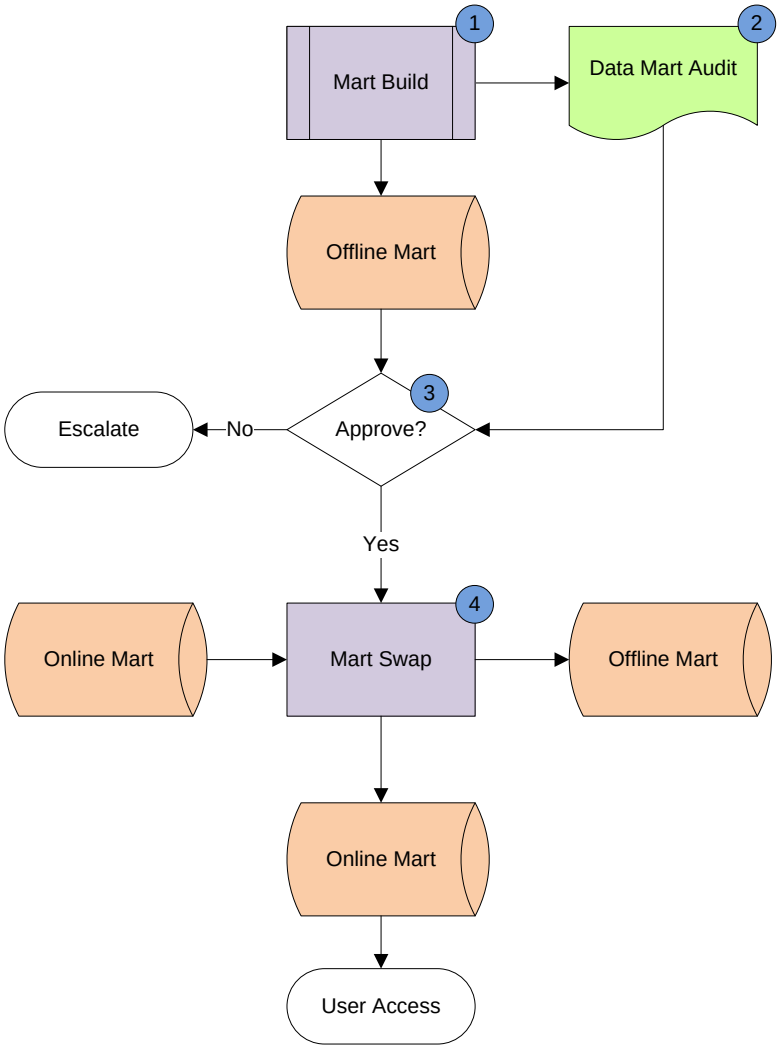
Data Mart Design

The Data Mart provides the end user with a consolidated view of all key data structures maintained in the Data Warehouse. The ANN INC CRM Solution includes the lowest level “Fact” data as well as summarized rollups. The Data Mart defines the customer and supports descriptive dimensions as foreign key relationships. These structures are used to support Campaign Management, Analytic, and Reporting needs. As such these structures are accessed directly through the Business Intelligence tools SAS VDD, SAS MA, and Business Objects.

The Data Mart build process is performed by executing fully refreshed (rebuilt) tables based on the underlying Data Warehouse structures. The updated mart data is made available daily.

The Data Mart is built as a background process in an offline instance such that user access to the online instance of the Mart is not affected. In this manner the complete mart can be built and audited internally before it is released or “swapped” in for external user access.

Data Mart Build Process Flow



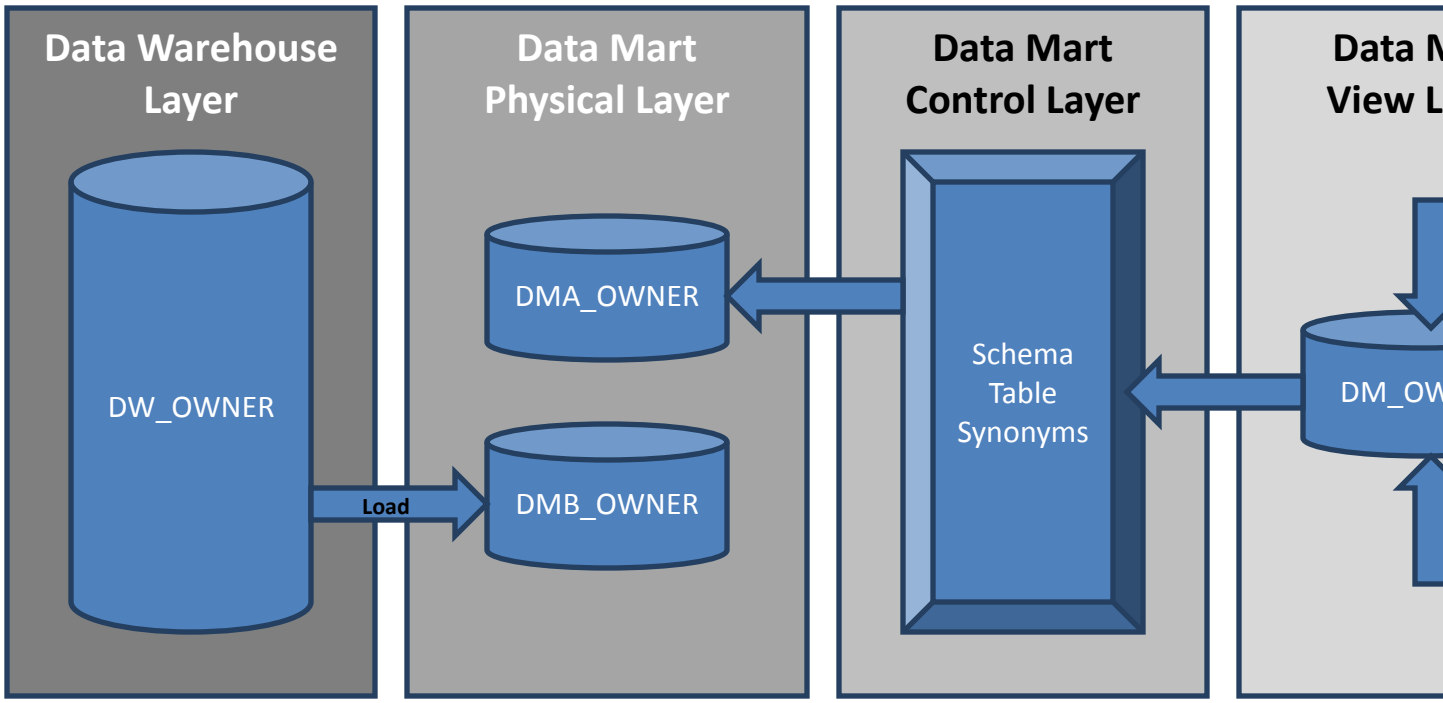
Data Mart Build Process Flow Annotations

1. **Mart Build** – The mart structures are built from the underlying Data Warehouse structures. This includes the business rule implementation to perform such tasks as
    a. Additional code standardization
    b. Denormalizing DW tables
    c. Referential Integrity enforcement
    d. Fact rollups
    e. Conversion of compound to single foreign key relationships
    f. Fact Rollups
    g. BEST Contact/Email/Phone identification
    h. Derived fields and table structures
    i. Brand context designation

    All elements are intended to provide a simplified/usable view of the ANN INC customer data while maintaining the flexibility to support customized analytic activities requiring visibility down to the granular level. The data is loaded into an offline instance of the mart as users continue to access the online instance as normal.
2. **Data Mart Audit** – the rebuilt data mart structures and profiled and compared to historical trends. Any anomalies are highlighted and escalated with ANN INC.
3. **Mart Approval** – The audits from Staging, Data Warehouse, and Data Mart are interpreted comprehensively. Those audit failures identified as critical result in the halt of the update process, otherwise the update proceeds.
4. **Mart Swap** – the configuration of each day's offline/online instance of the Data Mart is done through Oracle table synonyms while logically point the processes at the correct physical tables. The swap then becomes an instantaneous rebuild of synonyms are the pre-determined schedule. During the time when the swap occurs, external users cannot access the system. Once the swap completes users immediately access to the updated data set.

Data Mart Access Diagram

This diagram illustrates the Oracle schema configuration as it pertains to mart access from both the Data Warehouse and from the User perspective.



The Data Warehouse process loads all inbound source data feeds.
As mart tables are built, an ELT mechanism identifies which instance of the Data Mart is OFFLINE and loads to it.

The solution maintains an offline and online instance of the Mart.
The mart is build in the background permitting undisturbed user access to the ONLINE instance

Access to the ONLINE instance is controlled through Schema Table synonyms.
Once the new Mart has been audited and approved, a scheduled ELT process re-points the synonyms to the correct instance.

The users access
through Views cr
each table. These
direct or defined
specific/enterpris
views.

36. Data Mart Update

#	Design
BU36.1	The system will have the data mart daily refresh complete and available to users by Monday morning at 5:00AM CST, Tuesday – Sunday morning 6:00 AM CST.
BU36.1.1	Once the build of the mart tables completes and automated audit checks for critical failures have executed, the mart is "swapped' by switching the synonyms utilized by the DM_OWNER views to point at the newly updated 'offline' copy of the data mart tables. The catalogs of the various tools are pointed at these views. Thus, once the underlying synonyms are swapped the tools are effectively pointing to the new version of the data. Once the swap completes the updated Data Mart is immediately available to the end users of SAS VDD, SAS CI, and Business Objects.
BU36.1.2	The swap process is a discrete process in the job control process that will not occur if a preceding automated audit check fails or if the swap step is specifically disabled as part of the job stream.
BU36.2	The system will make sure that fields that are part of a foreign key in the data mart will have NULL values replaced by a NULL equivalent: • Date fields that are part of primary keys are defaulted to xx/xx/xxxx, will be specified in design and captured in the Solution Design Document. • String fields that are part of primary keys are defaulted to |~| • Numeric fields that are part of primary keys are defaulted to zero
BU36.2.1	All date fields in the database are foreign keys to LU_FISCAL_CALENDAR as such wherever they are NULL based on source data they are defaulted to 1900-01-01. The LU_FISCAL_CALENDAR table contains an entry for the invalid date
BU36.2.2	The LU_FISCAL_CALENDAR table contains a row for the invalid 1900-01-01 date to maintain RI
BU36.2.3	In general, for performance considerations whenever possible numeric surrogates for String keys are employed, however when necessary when building tables implementing a LEFT JOIN the string key fields include the nvl(${field_nm, '|~|') logic.
BU36.2.4	Numeric fields that fail a left join are defaulted to zero via nvl(${field_nm, 0)
BU36.3	The system will apply the address determined to be the "best postal contact address", in the data mart. If there are multiple Client records that link to the same individual id, only one record will be in the data mart. To choose the "best postal contact address" use this priority: • Recent NCOA movers then, • Rank deliverability scores 1-3 as the same rank, group those addresses as the best then, • Most recent activity date which is defined as any activity of the underlying account then, • Take the lowest mail score (1 is the best) then, • Residential over business address then, • Source of record (use the same account ranking that is used for the account table) *** Note we may include mail scores of 4 in the ranking of the addresses above, this will depend on results from Agility iterations.
BU36.3.1	The postal contact COA move date is used to determine mover recency. This includes both NCOA and PCOA moves.
BU36.3.2	The AGILITY-derived Postal Contact Mail score is used to rank the most likely deliverable addresses.
BU36.3.3	The ACCOUNT activity_dt is used as defined based on the following account source feeds

Source	Feed	Activity Date

MW	Address	sysdate	
MW	Customer	sysdate	
ADS	Customer Daily PLCC	sysdate	
ADS	Customer Daily CO Brand	sysdate	
ADS	Customer Monthly PLCC (type 0)	sysdate	
ADS	Customer Monthly COBRAND (type 0)	sysdate	
RIGHTNOW	Call Center Incident/Customer	CREATED	
BV	Product Review ANN Taylor	CRDATE	
BV	Product Review LOFT	CRDATE	
SDL	Subscriptions	DATETIME_SUB	
SDL	Change of Demographics	DATE_CHANGED	
SAP	Associate Reference	DATE_TIME	
ATG	ECOM Customer Profile	sysdate	
ATG	ECOM Transaction - Orders	LASTMODIFIEDDATE	
ATG	ECOM Catalog Requests	REQUESTED_DATE	
AW	POS Customer Detail (Type C)	TRANSACTION_DATE	
EXP	Reverse Email Append	sysdate	
EXP	Name/Zip Address Append	sysdate	
EXP	Trade Area Append	sysdate	

BU36.3.4

The USPS DSF2 residential/business indicator posted from AGILITY sorted to select Residential addresses over Businesses

BU36.3.5

The account sources are ranked according to the following designation:

ACCT_SOURCE_CD	ACCT_SOURCE_DESC	ACCT_SOURCE_RANK
EMP	ANN INC (SAP) ASSOCIATES	5
ADS	ALLIANCE DATA SYSTEMS	10
WEBB	ECOMM (ATG) BILL-TO	20
WEBC	ECOMM (ATG) CATALOGUE REQUESTOR	30
CM	CHEETAHMAIL (SDL) DEMO/SUBS	40
AWB	POS (AUDITWORKS) BILL-TO	50
RN	CALL CENTER (RIGHT NOW)	60
MW	MARKETWORKS	100
WEBS	ECOMM (ATG) SHIP-TO	110
AWS	POS (AUDITWORKS) SHIP-TO	120
NZA	NAME/ZIP APPEND (KBM)	130
TAA	TRADE AREA APPEND (KBM)	140
REA	REVERSE EMAIL APPEND (KBM)	150

BU36.3.6

The SUM_INDIVIDUAL table contains the rollups for all the ACCOUNT references through POSTAL_CONTACTS linked to an individual. Because an individual (person) can be associated with multiple addresses over time the following logic is employed to select the BEST:

```
select a.indiv_id, b.postal_contact_id, c.acct_source_cd, c.acct_nbr, c.brand_cd,
        b.coa_move_dt, b.mail_score, c.activity_dt, b.dsf2_res_bus_cd,
d.acct_source_rank,
        (ROW_NUMBER () OVER (PARTITION BY a.indiv_id
                ORDER BY
                        Nvl(b.coa_move_dt, '1900-01-01') desc,
```

	(case when b.mail_score >= '3' then 1 else 0 end), c.activity_dt desc, b.mail_score, (case when b.dsf2_res_bus_cd = 'R' then 0 else 1 end), d.acct_source_rank)) as RANKING from dw_owner.individual a inner join dw_owner.postal_contact b on a.indiv_id = b.indiv_id inner join dw_owner.account c on b.postal_contact_id = c.postal_contact_id inner join dw_owner.account_source d on c.acct_source_cd = d.acct_source_cd
BU36.3.7	The same ranking is used to assign the BEST postal contact based on brand interaction: select a.indiv_id, b.postal_contact_id, c.acct_source_cd, c.acct_nbr, c.brand_cd, b.coa_move_dt, b.mail_score, c.activity_dt, b.dsf2_res_bus_cd, d.acct_source_rank, (ROW_NUMBER () OVER (PARTITION BY a.indiv_id ORDER BY b.coa_move_dt desc, (case when b.mail_score >= '3' then 1 else 0 end), c.activity_dt desc, b.mail_score, (case when b.dsf2_res_bus_cd = 'R' then 0 else 1 end), d.acct_source_rank)) as RANKING from dw_owner.individual a inner join dw_owner.postal_contact b on a.indiv_id = b.indiv_id inner join dw_owner.account c on b.postal_contact_id = c.postal_contact_id inner join dw_owner.account_source d on c.acct_source_cd = d.acct_source_cd where c.brand_cd = ${BRAND_CD}
BU36.3.8	The Postal Contact ID of the selected BEST postal contact is posted to the SUM_INDIVIDUAL table along with its associated attributes including: • HouseHold ID • Address ID • Name • Name Attributes (Gender, Vulgar) • Address • Address Attributes (CASS/SERP/COA/EPAC) • Lat/Long • US Possession, APO/FPO, DMA Pander, Prison, Deceased Indicators • Mail Score
BU36.3.9	Because the mart provides both the brand specific view and enterprise view of an individual, the BEST postal contact selected based upon the above ranking may be different for each brand when the underlying account data is filtered.
BU36.3.10	Similarly, the SUM_HOUSEHOLD table provides the BEST postal contact at the household level. All of the same rules and ranking described at the individual level are applied to the group/partition at the HH_ID level on the INDIVIDUAL table.
BU36.3.11	Because there are individuals associated with multiple households and transactions for an individual follow the person to the household level, there may be instances where the household relationship for an individual that exists in the Data Warehouse does not exist in the Data Mart. This is because the individual selected as the BEST postal contact at the

<table>
<tr><td></td><td>household level was from a different household.</td></tr>
<tr><td>BU36.4</td><td>The system will retain the previous week’s data mart tables and keep them accessible to users while the current week’s data mart tables are being built offline. Once the process is finished and qc’d there will be a table swap performed to move the current tables online and retire the previous week’s tables.</td></tr>
<tr><td>BU36.4.1</td><td>All mart tables are refreshed daily as described in BU36.5</td></tr>
<tr><td>BU36.5</td><td>The system will retain the daily data mart tables and keep them accessible to users while the current day’s data mart tables are being built offline. Once the process is finished and qc’d there will be a table swap performed to move the current tables online and retire the previous day’s tables.</td></tr>
<tr><td></td><td>The following data mart tables are refreshed DAILY:
<table>
<tr><th>Subject</th><th>DM Table</th></tr>
<tr><td>Customer</td><td>XREF_MW_ACCT_INDIV</td></tr>
<tr><td>Model</td><td>F_MODEL_SCORE_CURRENT</td></tr>
<tr><td>Model</td><td>F_MODEL_SCORE_HISTORY</td></tr>
<tr><td>Model</td><td>LU_MODEL_TYPE</td></tr>
<tr><td>Promotion</td><td>F_DM_PROMO_HISTORY</td></tr>
<tr><td>Promotion</td><td>LU_CAMPAIGN_TAXONOMY</td></tr>
<tr><td>Reference</td><td>LU_DEMOGRAPHIC</td></tr>
<tr><td>Reference</td><td>LU_FISCAL_CALENDAR</td></tr>
<tr><td>Reference</td><td>LU_GEOGRAPHY</td></tr>
<tr><td>Reference</td><td>LU_PRODUCT</td></tr>
<tr><td>Reference</td><td>LU_STORE</td></tr>
<tr><th>Subject</th><th>DM Table</th></tr>
<tr><td>Customer</td><td>SUM_HOUSEHOLD</td></tr>
<tr><td>Customer</td><td>SUM_INDIVIDUAL</td></tr>
<tr><td>Email</td><td>F_BRAND_EMAIL</td></tr>
<tr><td>Email</td><td>LU_EMAIL_LIST_AGGR</td></tr>
<tr><td>Email</td><td>XREF_INDIV_EMAIL</td></tr>
<tr><td>Promotion</td><td>F_EMAIL_PROMO_HISTORY</td></tr>
<tr><td>Promotion</td><td>F_EMAIL_RESPONSE</td></tr>
<tr><td>Promotion</td><td>LU_EMAIL_RESPONSE_TYPE</td></tr>
<tr><td>Reference</td><td>LU_CSC_INCIDENT</td></tr>
<tr><td>Reference</td><td>LU_REVIEW</td></tr>
<tr><td>Reference</td><td>LU_STORE_PROMO</td></tr>
<tr><td>Transaction</td><td>F_TXN_HEADER</td></tr>
<tr><td>Transaction</td><td>F_TXN_ITEM</td></tr>
<tr><td>Transaction</td><td>F_TXN_ITEM_DISCOUNT</td></tr>
<tr><td>Transaction</td><td>F_TXN_TENDER</td></tr>
<tr><td>Web Tracking</td><td>LU_GEOSEGMENT</td></tr>
<tr><td>Web Tracking</td><td>LU_OMN_CMPGN_MAP</td></tr>
<tr><td>Web Tracking</td><td>LU_REF_DOMAIN</td></tr>
<tr><td>Web Tracking</td><td>F_SITE_VISIT</td></tr>
<tr><td>Web Tracking</td><td>F_VISIT_MERCH_CATEGORY</td></tr>
</table></td></tr>
<tr><td>BU36.5.2</td><td>The DAILY mart swap occurs at 5AM CST on Monday and 6AM CST Tuesday through Sunday</td></tr>
</table>

37. Data Mart Contents

#	Design
BU37.1	The system will support in the data mart: • Fact detail tables that correspond to events in the data warehouse; For example: Transactions, promotion history, promotion response activity, products etc…..This will be detailed out in the solution design document

BU37.1.1

The Fact tables are pre-pended with "F_" and are listed below:

Subject	DM Table	Description	Brand View
Email	F_BRAND_EMAIL	Contains all email addresses, list preferences, demographics	Y
Model	F_MODEL_SCORE_CURRENT	Scored individual segmentations	Y
Model	F_MODEL_SCORE_HISTORY	Scored individual segmentations	Y
Promotion	F_DM_PROMO_HISTORY	Direct Mail Campaign promotion history	Y
Promotion	F_EMAIL_PROMO_HISTORY	Email Campaign promotion history	Y
Promotion	F_EMAIL_RESPONSE	Email Campaign response activity (open, click, unsub, txn)	
Transaction	F_TXN_HEADER	Transaction Header	Y
Transaction	F_TXN_TENDER	Transaction Tender	
Transaction	F_TXN_ITEM	Transaction Detail	Y
Transaction	F_TXN_ITEM_DISCOUNT	Transaction Detail Coupon data	Y
Web Tracking	F_SITE_VISIT	Omniture Web Activity Visits	Y
Web Tracking	F_VISIT_MERCH_CATEGORY	Omniture Web Activity Visits by Merchandise Category	Y

BU37.1.2

The Data Warehouse data that is promoted into these structures is described in **Appendix**

BU37.2

The system will support in the data mart:

- Dimension tables that will categorize events in the fact tables;
  For example: Store, products, time etc…..This will be detailed out in the solution design document

BU37.2.1

The Dimension tables are pre-pended with "LU_" (lookup) and are listed below:

Subject	DM Table	Description	Brand View
Model	LU_MODEL_TYPE	Model Score Type lookup	
Promotion	LU_CAMPAIGN_TAXONOMY	Marketing Campaign Lookup (EM & DM); Note: this structure will likely be normalized further as the hierarchy is further clarified	Y
Promotion	LU_EMAIL_RESPONSE_TYPE	Email Response Activity Types Lookup	
Reference	LU_CSC_INCIDENT	Call Center incidents tied to an individual	Y
Reference	LU_DEMOGRAPHIC	KBM/Mosiac append attributes for BEST postal contact	Y
Reference	LU_FISCAL_CALENDAR	Fiscal Calendar Lookup	
Reference	LU_GEOGRAPHY	Postal Code Geography Lookup	
Reference	LU_PRODUCT	Product/Merchandise	
Reference	LU_REVIEW	BazaarVoice Ratings linked to Individual	Y
Reference	LU_STORE	Store Reference	
Reference	LU_STORE_PROMO	Store Promotion Reference	
Web Tracking	LU_GEOSEGMENT	Omniture GeoSegment Loookup	
Web Tracking	LU_OMN_CMPGN_MAP	Omniture Campaign Lookup	
Web Tracking	LU_REF_DOMAIN	Omniture Web Activity Referring Domain	

BU37.2.2

The Data Warehouse data that is promoted into these structures is described in **Appendix**

BU37.2.3

The Cross-Reference tables are pre-pended with "XREF_" and are listed below:

Subject	DM Table	Description	Brand View
Customer	XREF_ACCT_INDIV	Relates an Individual to historical (pre-deduped) MarketWorks Customer Numbers. This table also provides a	

<table>
<tr><td></td><td>
<table>
<tr><td></td><td></td><td>reference for each individuals source as well as source account ID (i.e. ADS). This is where users will be able to determine if a person was loaded from “X” source.</td><td></td></tr>
<tr><td>Email</td><td>XREF_INDIV_EMAIL</td><td>Relates an Individual/Customer to all of their underlying Email Addresses as obtained through the various sources</td><td>Y</td></tr>
</table>
</td></tr>
<tr><td>BU37.3</td><td>The system will provide in the data mart, summarized data elements defined during data mart discovery to support reporting and selection at the following levels:
• Individual
• Household
• Email (from CheetahMail Email aggregate file)
These are fully indentified and described in Appendix G – Data Mart Rollups
Note: All dollar aggregates are derived from US transaction dollars only</td></tr>
<tr><td>BU37.3.1</td><td>The Summary tables are pre-pended with ‘SUM_” and are listed below:
<table>
<tr><th>Subject</th><th>DM Table</th><th>Description</th><th>Brand View</th></tr>
<tr><td>Customer</td><td>SUM_HOUSEHOLD</td><td>Best Postal Contact [Head] of a Household (and all underlying accounts); includes derived rollups & account preferences/attributes</td><td>Y</td></tr>
<tr><td>Customer</td><td>SUM_INDIVIDUAL</td><td>Best Postal Contact of an Individual (and all underlying accounts); includes derived rollups & account preferences/attributes</td><td>Y</td></tr>
<tr><td>Email</td><td>LU_EMAIL_LIST_AGGR</td><td>Email rollups by PID & Month</td><td>Y</td></tr>
</table>
</td></tr>
<tr><td>BU37.3.2</td><td>The summary tables are primarily built from the relationship of individual to Postal Contact to Account. The elements summarized are derived from their relationship to an Account (i.e. transaction/email) or from the Postal Contact level and it’s dimensions (i.e. demographics)</td></tr>
<tr><td>BU37.3.3</td><td>Customer Preferences at the customer level are derived from underlying ACCOUNT preferences such that if any account is opted out to a given channel, the individual is opted out of the channel.</td></tr>
<tr><td>BU37.3.4</td><td>Email Preferences are defined at the Email level thorough CheetahMail/SDL on the BRAND_EMAIL table. This table is keyed at the Email ID and Brand CD level and provides separate fields for the lists supported by each brand as described in BU18.1.6

An email subscriber indicator can be posted to the SUM_INDIVIDUAL/HOUSEHOLD tables as an aggregate view of the CheetahMail subscriber status rolled up to the individual level. This is being evaluated as a Future Enhancement.</td></tr>
<tr><td>BU37.3.5</td><td>The Individual summary performs its rollups on INDIVIDUAL.INDIV_ID
The Household summary performs its rollups on INDIVIDUAL.HH_ID</td></tr>
<tr><td>BU37.3.6</td><td>The Email Aggregates are performed externally at SDL, as such they are loaded directly as a Dimension of the F_BRAND_EMAIL table</td></tr>
<tr><td>BU37.4</td><td>The system will provide in the data mart the standardized and cleansed name and address information for direct mail and salutation purposes.</td></tr>
<tr><td>BU37.4.1</td><td>The Postal Contact table contains the Standardized Name/Address data returned from AGILITY. This table is used to select the BEST postal contact for an individual on the SUM_INDIVIDUAL table. The best postal contact information for the individual selected as head of household is utilized on the SUM_HOUSEHOLD table.</td></tr>
<tr><td>BU37.4.2</td><td>The mart does not provide the ability to reference original/unstandardized address/email data. This level of detail only resides within the Data Warehouse.</td></tr>
<tr><td>BU37.5</td><td>The system will make the Fiscal Calendar Reference data accessible for any date field in any dimension or fact within the data mart.</td></tr>
<tr><td>BU37.5.1</td><td>The LU_FISCAL_CALENDAR has a DATE defined as the primary key. This key joins to every date field in the solution.</td></tr>
<tr><td>BU37.5.2</td><td>If a date field in the mart does not exist in the LU_FISCAL_CALENDAR table it is</td></tr>
</table>

	defaulted to 1900-01-01.
BU37.5.3	Date fields that are regularly used for analytical purposes are included in the audit process to identify any problems with source date data and/or the Fiscal Calendar itself.
BU37.6	The system will map the data warehouse transaction_no to a numeric field called transaction_id, to be used in queries and joins in the data mart.

BU37.6.1	The EPS_TXN_HEADER_ID is derived in the data warehouse TXN_HEADER update process as a sequenced unique numeric surrogate for the TXN_HEADER Primary Key
	• TXN_SOURCE_CD
	  ○ AW = POS (AuditWorks)
	  ○ ATG = E-Commerce
	  ○ MW = POST & E-Commerce Migration (Marketworks)
	• TXN_NBR
	  ○ AW = Store, Register, Date, Transaction ID
	  ○ ATG = OrderID
	  ○ MW = Transaction ID
	• BRAND_CD
	  ○ AT = Ann Taylor
	  ○ LOFT = Loft
	  ○ ATF = Ann Taylor Factory
	  ○ LOS = Loft Outlet
	  ○ ENT = Unknown

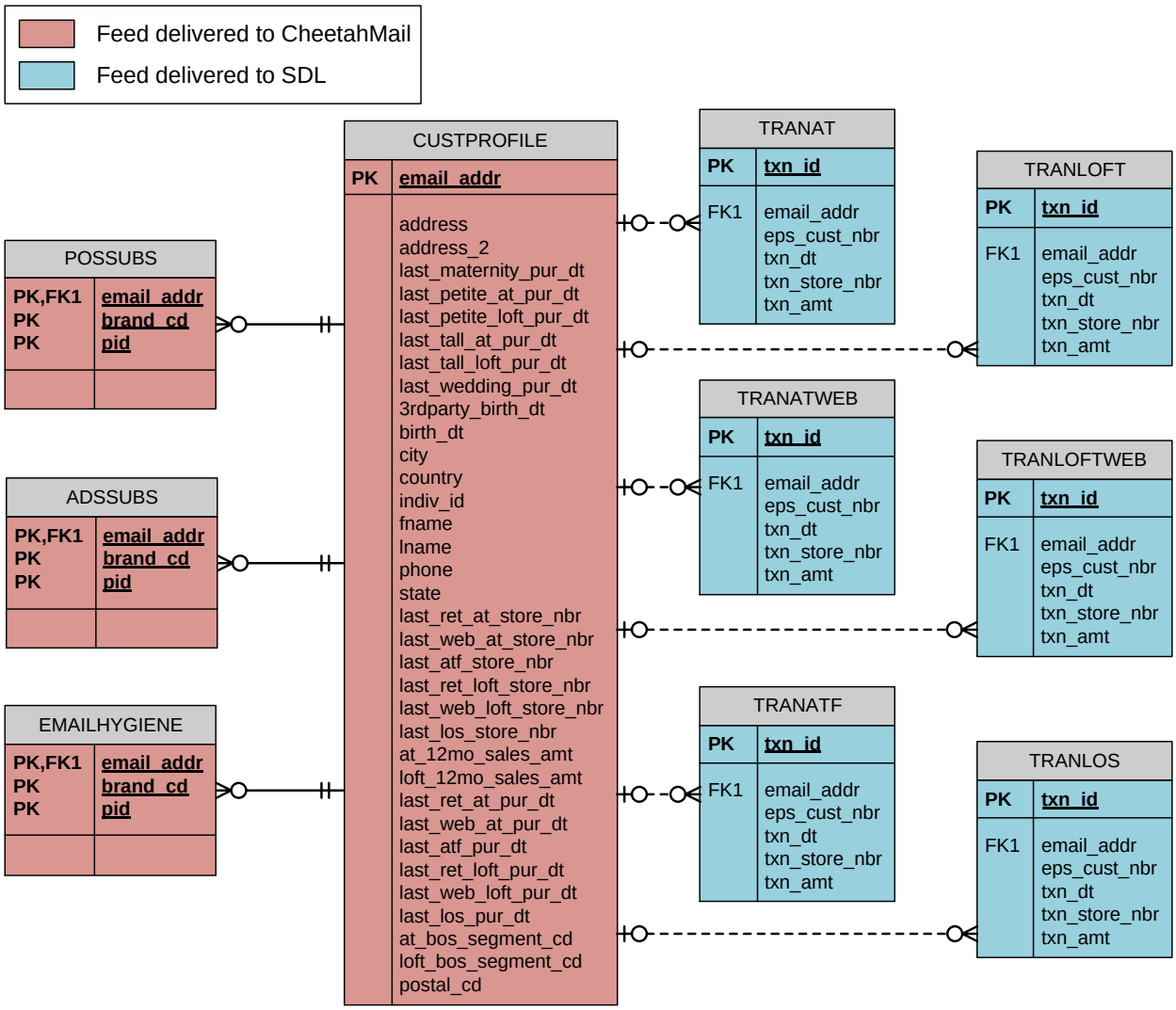
BU37.7	When consolidating multiple email addresses, the following rules apply to select the “best” email address: 1. First choose the valid email address over a non-valid email address (valid means non-chronic bouncer, not opted-out, and not invalid format). 2. Then choose the email address that has the most recent purchase activity. 3. Then choose the email address that has the most recent non-purchase activity (positive and Client initiated, such as opens or clicks). 4. Then choose sequentially based on email identifier stored in the table.f
BU37.7.1	The SUM_INDIVIDUAL and SUM_HOUSEHOLD each designate a BEST_EMAIL_ID for both the Enterprise (ENT) level as well as by Brand.
BU37.7.2	Email validation is determined by EMAIL.VALID_IND = 1 and EMAIL.UNDELIVERABLE_IND = 0 and EMAIL_NUCLEAR_OPTOUT_IND = 0 and NOT opted out of ANY CheetahMail PIDs
BU37.7.3	Transaction Recency is determined by: 1. First joining EMAIL directly to TXN_HEADER and selecting max(TXN_DT) 2. Then join to EMAIL_RESPONSE where EMAIL_RESPONSE_TYPE_CD = ‘TR’ and select max(EMAIL_RESPONSE_DT) – these are T2P (track to purchase) email addresses 3. Then select the max date between Transaction and Email Response – sorted descending
BU37.7.4	Response Activity Recency is determined by EMAIL_RESPONSE where EMAIL_RESPONSE_TYPE_CD in (‘OP’,’CL’) and selecting max(EMAIL_RESPONSE_DT) – sorted descending
BU37.7.5	The system uses email_id sorted desc as a final tie-breaker if the other sort criteria do not result in a best email address.

BU37.7.6	the following logic is employed to select the BEST email ID at the Enterprise level: select a.email_id as BEST_EMAIL_ID, a.valid_ind, a.undeliverable_ind, a.nuclear_optout_ind, c.last_txn_dt, d.last_txn_resp_dt, (row_number() over (partiton by a.indiv_id order by (case when a.valid_ind = 1 and a.undeliverable_ind = 0 and a.nuclear_optout_ind = 0 and nvl(e.nbr_optouts, 0) = 0 then 1 else 0 end) desc, (case when c.last_txn_dt > d.last_txn_resp_dt then c.last_txn_dt else d.last_txn_resp_dt end) desc, d.last_em_resp_dt desc, a.email_id desc)) from dw_owner.email a left join (select a.email_id, max(b.txn_dt) as last_txn_dt, from dw_owner.account_email a inner join dw_owner.txn_header b on a.acct_nbr = b.acct_nbr and a.acct_source_cd = b.acct_source_cd and a.brand_cd = b.brand_cd group by a.email_id) c on a.email_id = c.email_id left join (select a.email_id, max(case when a.email_response_type_cd = 'TR' then a.email_response_dt else null end) as last_txn_resp_dt, max(case when a.email_response_type_cd in ('OP','CL') then a.email_response_dt else null end) as last_em_resp_dt from dw_owner.email_response a group by a.email_id) d on a.email_id = d.email_id left join (select email_id, sum(email_optout_ind) as nbr_optouts from brand_email group by a.email_id) e on a.email_id = e.email_id
BU37.7.7	The same ranking is used to assign the BEST postal contact based on brand interaction: select a.email_id as BEST_EMAIL_ID, a.valid_ind, a.undeliverable_ind, a.nuclear_optout_ind, c.last_txn_dt, d.last_txn_resp_dt, (row_number() over (partiton by a.indiv_id order by (case when a.valid_ind = 1 and a.undeliverable_ind = 0 and a.nuclear_optout_ind = 0 and nvl(e.nbr_optouts, 0) = 0 then 1 else 0 end) desc, (case when c.last_txn_dt > d.last_txn_resp_dt then c.last_txn_dt else

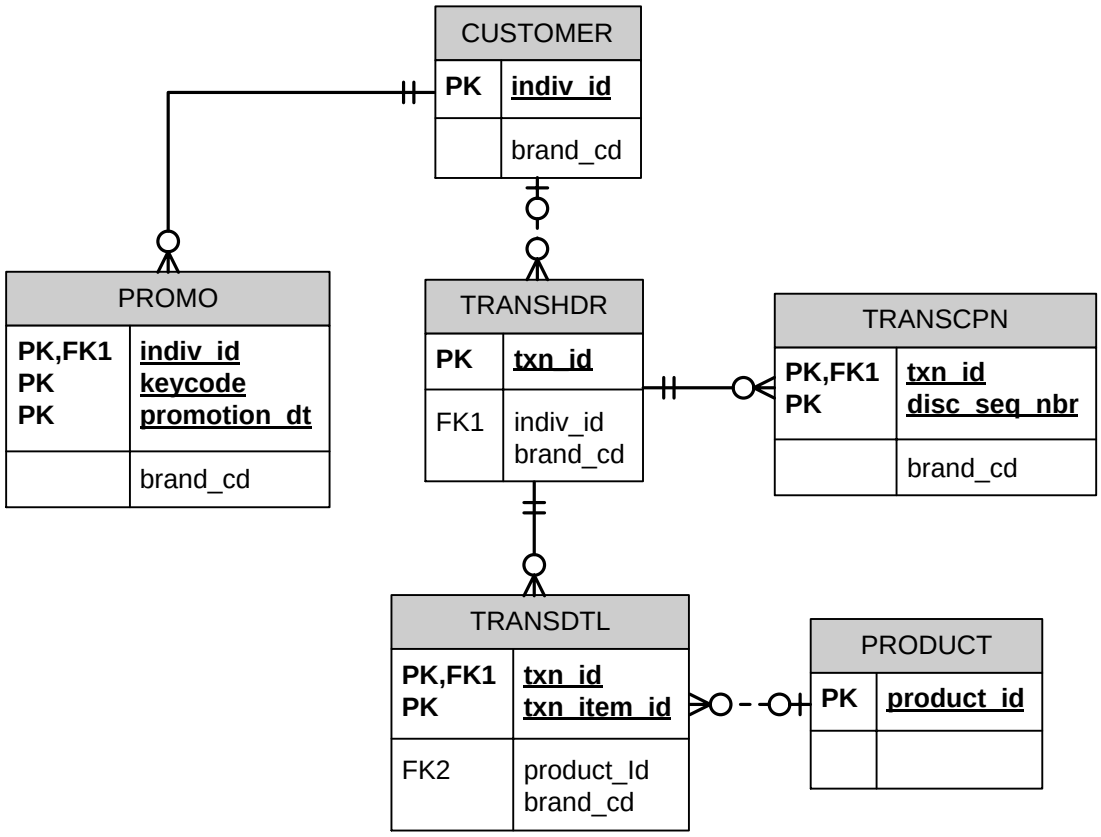
```
d.last_txn_resp_dt end) desc,
                    d.last_em_resp_dt desc,
                    a.email_id desc
          )
)
from dw_owner.email a
left join (
select a.email_id, max(b.txn_dt) as last_txn_dt,
from dw_owner.account_email a
inner join dw_owner.txn_header b on a.acct_nbr = b.acct_nbr and a.acct_source_cd =
b.acct_source_cd and a.brand_cd = b.brand_cd
group by a.email_id
) c on a.email_id = c.email_id
left join (
select a.email_id,
          max(case when a.email_response_type_cd = 'TR' then a.email_response_dt else
null end) as last_txn_resp_dt,
          max(case when a.email_response_type_cd in ('OP','CL') then
a.email_response_dt else null end) as last_em_resp_dt
from dw_owner.email_response a
group by a.email_id
) d on a.email_id = d.email_id
left join (
select email_id, brand_cd, sum(email_optout_ind) as nbr_optouts
from brand_email
group by a.email_id, brand_cd
) e on a.email_id = e.email_id and b.brand_cd = e.brand_cd
Where  b.brand_cd = '${BRAND_CD}
```

Extracts/ Audits/ Acceptance Design

SDL/CheetahMail Extract Entity Relationships



ADS Extract Entity Key Relationships



38. Extracts

#	Design
BU38.1	The system will allow users to create ad-hoc data extractions against any level of data accessible within the data mart.
BU38.1.1	SAS CI and VDD support the extraction of data from the tool. Files are written to the /outbound folder on the SAS server.
BU38.1.2	The /outbound directory is polled every 5 minutes for extracts. When found they are pushed out to /home/anntaylor/outbound on the DB Gateway server where they can be retrieved by the user through a SFTP client over the VPN tunnel.
BU38.2	The system will land all extractions in a secure location accessible only via SFTP.
BU38.2.1	The extracts are landed on the DB Gateway PC1ANNTELT03 that is accessible only on port 22 (SFTP). Additionally external access is only permitted through the ANN VPN tunnel for authorized users.
BU38.3	The system will produce standard data extractions to support the daily and periodic Agility processes and will need to be generated off of the data staging and warehouse tables.
BU38.3.1	The daily and quarterly Agility processes are fully described in **Section 8 – Agility/PCI**
BU38.4	The system will produce standard data extractions daily that will be sent via SFTP to CheetahMail. The extracts include; • Transaction header level data that can be directly tied to an email address, by brand for any new/updated orders occurring within the latest daily cycle. • Customer profile data linked to an email address when a transaction is made within the latest daily cycle.
BU38.4.1	The CheetahMail extracts described in **BU1.47** include the email address keyed profile aggregates
BU38.4.2	The SDL extracts described in **BU1.49** include the transaction data by brand with an email address foreign key
BU38.4.3	The CheetahMail profile data is passed through to SDL via an external process.
BU38.4.4	The data transfers for both SDL and CheetahMail are performed via SFTP as a PUSH from MERLIN through ALSAN to the external source. The Epsilon support team is notified of each outbound transfer.
BU38.4.5	The CheetahMail/SDL extracts are defined fully in **Appendix J**
BU38.5	The system will produce standard data extractions to support periodic 3rd party processes and will need to be generated off of the data staging and warehouse environment (KBM, Experian, CheetahMail and ADS) .
BU38.5.1	The KBM Demographic extract is created from the Data Warehouse Postal Contact table as described in **BU1.48**
BU38.5.2	The Experian MOSAIC extract is created from the Data Warehouse Postal Contact table as described in **BU1.48**
BU38.5.3	The CheetahMail (and SDL) Extracts are created from the Data Mart as described in **BU1.47** and **BU1.49**
BU38.5.4	The ADS Extracts are created from the Data Mart as described in **BU1.46**
BU38.6	The system will produce an extract of new email addresses identified from either POS (AuditWorks) or ADS sources and SFTP to CheetahMail as subscriber/ rescubscriber requests by brand.
BU38.6.1	The CheetahMail extracts described in **BU1.47** include the POS and ADS new email subscription requests

39. Quality Audits

#	Design
BU39.1	The system will create audit reports that will be generated out of the daily update. Format TBD during the design phase.

BU39.1.1	The Staging, AGILITY, Data Warehouse, and Data Mart audit reports are generated as described in **Appendix H – Audit Reports**
BU39.1.2	Retail Matchback audits are performed by providing transaction counts by • Method • Vendor • Date
BU39.1.3	For external Retail Matchback processes, counts are provided on sent and return to assess incremental match rates
BU39.1.4	File counts for daily extracts are performed to assess trending over time
BU39.2	The system will create audit counts to verify data migration counts against ANN's source counts.
BU39.2.1	Epsilon provides ANN INC with MarketWorks by brand loaded counts for the following structures: • ACCOUNT • ACCOUNT_EMAIL • ACCOUNT_HARD_KEY • ACCOUNT_PHONE • BRAND_EMAIL • EMAIL • BRAND_EMAIL • INDIVIDUAL • POSTAL_CONTACT • POSTAL_CONTACT_ARCHIVE • POSTAL_CONTACT_STORE • PHONE • PRODUCT • RMB_IDENTIFIER • STORE • TENDER • TXN_HEADER • TXN_ITEM • TXN_ITEM_DISCOUNT • XREF_MW_ACCOUNT
BU39.2.2	Epsilon works with ANN INC to determine what transaction and account aggregates are appropriate for acceptance
BU39.3	The system will create profile reports and/or audit counts on the warehouse tables. Format and frequency TBD during the design phase.
BU39.3.1	Profile reports will be run on each warehouse table to show distribution in field counts. Only the tables needed by ANN INC will be loaded to the Ann INC. SharePoint portal.
BU39.4	The system will create audit reports that will be generated out of the daily update. Format TBD during the design phase.
BU39.4.1	The Staging, AGILITY, Data Warehouse, and Data Mart audit reports are generated as described in **Appendix H – Audit Reports**
BU39.5	The system will create profile reports and/or audit counts on the data mart tables. Format and frequency TBD during the design phase.
BU39.4.1	See **BU39.3**. Comparison profile reports are leverage for the current vs. the previous aggregates.

40. Initial Acceptance

#	Design
BU40.1	It is required that a Test Plan document be created by Epsilon and ANN INC. (combined) which will detail specific activities required for the acceptance of the completed initial solution
BU40.1.1	ANN INC. to provide UAT test case scenarios. ANN Inc to perform UAT with Epsilon's support.

BU40.2	Pass Full System Integration Testing (SIT) phase to validate
	• Data transfer
	• Automated file handling and processing queuing
	  ○ PCI
	  ○ Staging
	  ○ Agility
	  ○ Data Warehouse Update
	  ○ Data Mart Update
	• Adherence to file naming conventions
	• Notifications
	• Reject and exception handling
	• Escalation processing

BU40.2.1	SIT testing will be performed by Epsilon which is an end to end run of the database update using JCS (Job Control Scheduler). All issues during SIT will be documented and resolved between Epsilon and ANN INC.
BU40.3	All non-conformities identified during UAT are documented. These include: • Data accuracy and integrity issues • CRM performance issues • Application performance issues • CRM system data security
BU40.3.1	UAT issues will be documented in the UAT test case scenario document and will be resolved where deemed necessary by Epsilon and ANN INC.
BU40.4	All non-conformities identified by ANN INC. as critical are resolved during UAT. Examples of these include but are not limited to: • Unexplained discrepancy in Client counts • Unexplained discrepancy in transaction counts • Unexplained discrepancy in aggregates that are critical for campaign execution Examples of non-critical non-conformities include: • Non-conformity in reports • Unexplained discrepancy in Client counts • Unexplained discrepancy in transaction counts **ANN Inc. and Epsilon will work through discrepancies during UAT and decide on acceptable variances per the above.
BU40.4.1	See as 40.3.1 Note: Unexplained discrepancy in Client and/or transaction counts are examples of CRITICAL non-conformities.

Marketing Automation Requirements

The Marketing Automation Requirements section documents ANN INC.'s end-user requirements for all Epsilon Marketing Application tools and functions, including:

- Suppression Handling
- Segmentation assignments and test/control designations
  - Storage and Re-usage of selection criteria
- Waterfall and "train-of-thought" analysis
- Campaign file extraction
  - Formatted according to channel and recipient
  - Transferred via SFTP
  - Stored locally for 30 days, and will be periodically reviewed by Epsilon to monitor volumes. If needed, the retention may need to be adjusted.
- Reporting
- SAS VDD

ANN INC. will have access to the following environments to perform campaigns, reporting and analysis with their client data.

Campaign Environment

41. SAS CI

#	Requirement
MA41.1	The system will allow users the ability to select records at the individual, household, address or email level. Email campaigns will be selected, output, and tracked at the email address level. All other outbound channels will be at the address level.
MA41.1.1	SAS CI has the concept of Subject IDs which are the levels that the campaign selection will be done at and exported at the individual, household or address level.
MA41.2	The system will allow users the ability to create, save, and re-use selection algorithms using naming conventions for use in current and future campaign efforts.
MA41.2.1	Selection Diagrams in SAS CI can be reused.
MA41.3	The system will allow users the ability to define standard Export layout templates to ensure file formats adhere to third party deployment platform requirements.
MA41.3.1	Standard export layouts will be set up by the SAS Administrator.
MA41.4	The system will allow for suppression criteria to be defined at individual, household, address or email level.
MA41.4.1	Suppression criteria is created in selection diagrams and can be at the subject level required.
MA41.5	The system will retain output files on the application server and available for download by end users for 60 days from time of extraction and will be periodically reviewed by Epsilon to monitor volumes. If needed, the retention may need to be adjusted.
MA41.5.1	Archiving will be placed on the server to remove files greater than 60 days. Depending on volumes and number of files this retention period may need to be adjusted.
MA41.6	The system will allow users the ability to define prioritized segmentation logic and waterfall execution of selections.
MA41.6.1	Waterfall selections can be done in selection diagrams by the user.
MA41.7	The system will allow users the ability to schedule list selection and output on a regular period basis as well as ad-hoc in advance of a specific date and time.
MA41.7.1	SAS CI has the ability to schedule selection diagrams.

MA41.8	The system will allow the users the ability to define periodic holdout groups, test panels, and control panels within selection cells.
MA41.8.1	Selection Diagram can be utilized for this purpose.
MA41.9	The will allow users the ability to set up suppressions based on previous/other/volume of promotion activity.
MA41.9.1	Promotion history tables will be available in the data mart to be used in selection or suppression criteria in selection diagrams.

Reporting Environment

42. Business Objects Reporting

#	Requirement
MA42.1	The system will allow Basic Users the ability to view reports created by other Power Users of the system.
MA42.1.1	ALL BO users will have the ability to review reports. Different security groups are set up to manage basic users and power user's privileges.
MA42.2	The system will allow users the ability to create, save, and re-use report templates with flexible naming conventions to allow for current and future use.
MA42.2.1	Power users set up in BO will be able to create and save reports. Basic users will be able to view and run reports. Report templates can be reused.
MA42.3	The system will allow users the ability to modify existing reports to create new reports without affecting the underlying report template.
MA42.3.1	Existing reports can be modified without affecting the report template.
MA42.4	The system will support summarization within the reporting environment at any level of granularity available within the report mart.
MA42.4.1	Summarized tables will be available within the report mart that will reflect the lowest level needed for the required email and zip capture reports being produced during the database implementation.
MA42.5	The system will allow users the ability to export report results in PDF or Excel format for viewing and distribution to non BO users.
MA42.5.1	Reports can be exported to in different formats and delivered via email or client share point portal
MA42.6	The system includes up to 10 defined reports (or 100 hours of development) as defined in **Appendix H – Marketing Reports**
MA42.6.1	The reports that will be developed for the implementation of the database will be Zip Capture, Email capture and Top 25 Store capture reports. The reports are produced in weekly and monthly intervals.
MA42.7	The system will be updated on a daily basis (7x per week) to ensure consistency with the Data Warehouse and Campaign Data Mart.
MA42.7.1	The report mart tables will be built during every update cycle, which is 7x per week.
MA42.8	The system will support delivering standard reports to ANN INC. via the Client portal or email.
MA42.8.1	During discovery ANN INC. decided they wanted the reports delivered each week and monthly via email. An email distribution will be set up for BO to access and deliver the reports.

SAS Analytic Environment

43. SAS VDD

#	Requirement

MA43.1	Install and configure SAS VDD on the Dedicated Infrastructure with the following modules assumed (versions are noted in Section 6.1.6 herein): a. Base b. Stat c. Graph d. Enterprise Guide e. Access for Oracle f. Access for PC Files g. Enterprise Miner h. JMP
MA43.1.2	The SDS VDD components listed above are installed on pc1uanntsas04 and accessed via the user interface published on the Citrix portal PC1WANNTCTX01
MA43.2	The system will integrate SAS VDD with the analytics data mart.
MA43.2.1	SAS VDD is installed on pc1uanntsas04 and hosted through CITIRX. The VDD toolkit accesses the online instance of the Data Mart as described in **BU36**
MA43.3	The analytic data mart will be defined during the design phase.
MA43.3.1	The analytic data mart provides access to the low-level fact data as well as all dimension and summary data described in **BU37**
MA43.4	The system will allow users the ability to pull extract files and transmit files via SFTP outside of SAS environment.
MA43.4.1	Extracts from SAS VDD are generated to the /outbound folder on pc1uanntsas04
MA43.4.2	Data posted into /outbound is polled at an interval of 5 minutes. Any new files are pushed to the /outbound folder on the DB Gateway host where they can be retrieved via an SFTP client over the VPN tunnel.
MA43.5	The system will be updated on a daily basis (7x per week) to ensure consistency with the Data Warehouse and Campaign Data Mart .
MA43.5.1	The analytic data mart is updated and refreshed daily as described in **BU36**

System Architecture Design

The ANN INC. Solution is a complex system that integrates many operational and architectural components. Physically, it contains personal computers, integration software, communication networks, servers and database management systems. Beyond physical components, it contains operational resources and procedures, a support organization, a backup and recovery plan and a capacity plan providing scalability. The primary goal is to develop a solution that delivers consistent, quality data and that the solution supports business goals of various units and associates within the organization.

To that end, there should be a common level of understanding and acceptance among technical, non-technical and management participants of the system architecture. The system architecture provides end-to-end integration, ensuring success of business activities and application of appropriate technologies. The following sections provide the system architecture requirements for the ANN INC. Solution.

Network/ Hardware Environments

44. Network

#	Design
SA44.1	A corp-to-corp VPN connection exists between ANN INC. and Epsilon which will enable ANN INC. users to access the CRM, the SAS environment and Business Objects
SA44.1.1	The VPN tunnel is setup for access between Epsilon and the ANN INC New York and Connecticut end-point.
SA44.1.2	The VPN has access to the following hosts on the Epsilon side: • DB Gateway (SFTP) - pc1uanntetl03 - 10.231.37.5/port 22 • Citrix Application Gateway - pc1wanntctx01 - 10.230.19.1/port 1494 and 443
SA44.1.3	The VPN tunnel is managed such that each data center has responsibility for maintaining their end of the connection.
SA44.1.4	The SSL certificate on the Citrix server is named: anncrm.epsilon.com
SA44.2	All access to ANN INC. data at Epsilon will require authentication, and explicit authorizations
SA44.2.1	Epsilon security mandates do not permit the use of shared accounts on either the DB Gateway, Database Hosts, or Application Servers. These exist strictly as service accounts to run the update process and the database team does not have access to these credentials.
SA44.2.2	Internal and External access is only provided through named users with accounts subject to Epsilon password expiration and complexity standards
SA44.2.3	External Access is provided to ANN INC associates via named accounts to the following applications as detailed in **Appendix G – External Access**: • Citrix • SFTP – DB Gateway • SAS VDD • SAS CI • Business Objects • SharePoint
SA44.2.4	Vendor inbound SFTP access is performed through SSH key authentication (without the need to manage passwords). These keys connect though a separate account setup for each vendor which have access only to their data landing zones: • ADS - anntads • BazaarVoice - anntbazaar • CheetahMail - anntcheetah • Experian - anntexp

	• KBM - anntkbm • Omniture - anntomni • RightNow - anntrtnow • SDL - anntalt

45. Hardware

#	Design
SA45.1	Provides an SFTP site for file transfer between Epsilon, ANN INC. and their business partners
SA45.1.1	The external SFTP site provided for ANN INC vendors is ASLAN.EPSILON.COM for the following sources: • ADS • BazaarVoice • CheetahMail • Experian • KBM • Omniture • RightNow • SDL
SA45.1.2	Once a transfer on Aslan completes it is immediately pushed onto MERLIN.EPSILON.COM via SFTP and removed from the externally facing ASLAN host
SA45.1.3	Once the transfer to Merlin completes a copy is pushed to the ANN DB Gateway host pc1uanntetl03 with a unique timestamp appended in the format yyyymmddhh24miss
SA45.1.4	ANN INC is provided direct access to the ANN DB Gateway host pc1uanntetl03 via SFTP over the VPN tunnel to provide the following data sources • ADS xref • AuditWorks • ATG • SAP
SA45.2	Supports input and output files transferred between Epsilon, ANN INC. and their business partners via SFTP
SA45.2.1	The outbound extract to provided via SFTP push as described in **Appendix C – Data Flow Diagram** and **Appendix G – External Access**
SA45.3	Includes an Application / Web server located at Epsilon, for access to Business Intelligence application
SA45.3.1	SAS CI and VDD are hosted via the CITRIX portal on PC1WANNTCTX01
SA45.3.2	Business Objects is hosted on the shared BO environment, BIGDOG

Backup and Disaster Recovery

46. Backup and Disaster Recovery

#	Design
SA46.1	Epsilon will back-up the System incrementally daily with full backups weekly
SA46.1.1	Daily Incremental backups are done on inbound feeds on both the external (aslan), internal (merlin). Inbound feeds are retained online for 2 days. Outbound feed are retained online for 7 days. Inbound and Outbound feeds are retained on tape for 45 days.

SA46.1.2	Full database backups are performed twice a week. Incremental backups are performed daily (except when full backups are performed). Once a week, the database is exported out to the /db99 file system on the oracle host.
SA46.2	Historical Data is retained on tape for 2 years
SA46.2.2	The entire database is backed up daily as described in **SA46.1**. Data Warehouse and Data Mart retention is 2 years. Inbound feed retention is 45 days.

Database Support and End User Access

47. Database Support

#	Design
SA47.1	Requires an Epsilon account team that is first line of support for the CRM team at ANN INC. available Monday through Friday 8:00 AM to 7:00 PM EST Time excluding Epsilon holidays and special events that have been communicated in advance to ANN INC.
SA47.1.1	The Epsilon Manage Support team is identified in **Appendix F – Escalation Procedures**
SA47.2	Requires that the 24/7 data center support be provided to ANN INC. solution components that reside in Epsilon's data center
SA47.2.1	Escalation procedures are documented in **Appendix F**
SA47.3	Requires a Relationship Manager on the Epsilon account team responsible for activities covered in all statements of work between ANN INC. and Epsilon
SA7.3.1	The dedicated ANN INC account relationship manager is identified in **Appendix F – Escalation Procedures**
SA47.4	Requires an Epsilon account team member available for auditing files sent to Epsilon and processed through the update system
SA47.4.1	The production analyst has the responsibility for auditing the daily update process and is identified in **Appendix F – Escalation Procedures**
SA47.5	Provides ANN INC. a weekly conference call to discuss issues and projects within the ANN INC. Customer Relationship Management
SA47.5.1	The weekly ANN INC CRM project management call is to be scheduled as defined in **Appendix F – Escalation Procedures**
SA47.6	Requires Epsilon support hardware and software required for the solution assuming hardware and/or software is located at Epsilon's facilities
SA47.6.1	Epsilon provides support for all hosted applications, data, and infrastructure.
SA47.7	Epsilon will directly contact source data providers to resolve issues with the data, notifying ANN INC.'s team of the problem and its resolution, or the need to escalate
SA47.4.1	Epsilon escalates with ANN INC vendors as described in **Appendix F – Escalation Procedures**
SA47.8	All SAS CI list selections for the email and direct mail to be done by Ann Inc.
SA47.8.1	As a self-service client, ANN INC performs all campaign management functions. Epsilon support is available to assist with tool and data questions.
SA47.9	Requires Epsilon account team to access ANN web portal to store project documentation, contact information and other collaborative information related to the project
SA47.9.1	Epsilon project management has been provided access to the ANN INC hosted SharePoint portal http://annywhere.com/crm
SA47.10	Requires Epsilon account team to access ANN web portal to store project documentation, contact information and other collaborative information related to the project

SA47.10.1	See **SA47.9**

48. End User Access

#	Requirement
SA48.1	The Business Intelligence tool must support up to a maximum of 10 named users
SA48.1.1	The Business Objects users are identified in **Appendix G – External Access**
SA48.2	SAS CI tool must support up to a maximum of 10 named users
SA48.2.1	The SAS CI users are identified in **Appendix G – External Access**
SA48.3	The SAS VDD tool must support up to a maximum of 10 named users
SA48.3.1	The SAS VDD users are identified in **Appendix G – External Access**

Security

49. Security

#	Design
SA49.1	Requires a working security system that monitors and manages physical access to ANN INC.'s environment at Epsilon's site. Security breach requirements and escalation policy stipulated in Manage SOW must be adhered to
SA49.1.1	Epsilon has a formal Information Security Policy which is reviewed at least annually and is based on the ISO 27001 standards; as, Epsilon is ISO 27001/2 Certified.
SA49.1.2	Epsilon's approach to Information and Physical Security is based on restricted levels of access and associated privileges following the "least required" principle. Each point of physical access requires that an employee have the required privilege to access that specific level or area and utilize their assigned key card. For example, there is building access, sensitive areas access and data center access. Each level requires that the employee have appropriate authorization to access that particular area based on job functionality. Additionally, all card accesses are logged and maintained.
SA49.1.3	Visitor access to the facility must be properly verified and approved; and, all visitor badges are numbered. If an access card is assigned, the access privilege level must be the lowest required and properly maintained/returned.
SA49.1.4	Epsilon's Data Center locations contain several layers of security; including, but not limited to: approved key card access with logging functionality, secure server racks, video camera systems, temperature and humidity monitoring devices, raised flooring, pre-action fire suppression system and 24x7x365 staffing or monitoring. Access rights and privileges are reviewed at least quarterly.
SA49.1.5	Epsilon has a formal Incident Response Policy and Plan which are reviewed at least annually. Incident Response is based on a 24 x 7 x 365 availability and on the principles of: Preparation, Identification, Assessment, Containment, Eradication, Recovery, Follow-up and Post Mortem. Escalation occurs as is required by the IR Policy and all contractual commitments based on the severity of the incident; including, the possible contact of Senior Management, Clients (in compliance with all contractual and regulatory requirements) and External Resources such as Law Enforcement and Forensics Personnel. The IR Plan includes participation from members of Senior Management, Information Security, Human Resources, Legal, Privacy, Public Relations, Information Technology, Compliance and appropriate external resources. The Chief Information Security Officer will lead the investigation with Legal's guidance to ensure appropriate evidence is properly identified and maintained, and the proper chain of custody occurs.
SA49.2	Requires a secure internal network with appropriate protocols and passwords to ensure that only approved personnel have access to ANN INC.'s data
SA49.2.1	Epsilon performs a quarterly compliance review subject to project and technical delivery sign-off for access to all systems. User access and privileges is one component of this mandate.
SA49.3	Requires a secure firewall for transfer of data between ANN INC., Epsilon and ANN INC.'s vendors. Systems behind the firewall can only be accessed from within the Epsilon network
SA49.3.1	Vendor transfer through Aslan/Merlin employ the firewall rule.
SA49.3.2	ANN INC has restricted to access to the DB gateway host over VPN and SFTP for sending and retrieving data. Access is limited to only the directories setup for this activity.
SA49.3.3	Data Transfers through Citrix are double encrypted over VPN and SSL

SA49.3.4	Internal access is only permitted to authorized users within Epsilon. Access to production systems is restricted.
SA49.4	Requires double logins and provides secure directories for maintenance of privacy for SFTP transfers
SA49.4.1	Vendor transfers into ASLAN are authenticated via SSH only from the key's originating IP address. Once the transfer an internal sign-on to a SFTP host inside the firewall is employed and the data is transferred and removed from the externally facing directories.
SA49.4.2	ANN INC transfer in and out of the solution are double-encrypted over SSH and through the VPN tunnel
SA49.4.3	All vendor and ANN INC landing zones are maintained independently as described in **Appendix C – Data Flow Diagram**
SA49.5	Requires that access to ANN INC.'s Customer Relationship Management require sign-off by an ANN INC. representative
SA49.5.1	Access requests to any ANN CRM host are approved as described in **Appendix G – External Access**
SA49.6	Requires passwords to be reset at initial login and at least every 90 days. Reset passwords must be different from any of the 4 previous passwords. System must warn the user 14 days prior to expiration
SA49.6.1	These adhere to Epsilon password complexity standards on the Windows/Citrix hosts
SA49.7	Requires a monthly system access report be provided to ANN INC. to report which End Users have access to the CRM
SA49.7.1	Epsilon maintains an Internal/External access and role matrix for all systems that is subject to quarterly review and sign-off within Epsilon to remain in compliance.
SA49.7.2	The external user access portion of this document is shared with ANN INC management during the first week of each month and lists out each users access points and privileges
SA49.8	Requires prior approval from the appropriate ANN INC. point of contact, if downtime related to security patches and / or upgrades need to occur outside of the regularly scheduled maintenance window
SA49.8.1	Epsilon notifies ANN INC as soon as the need to request downtime has been identified. There is a minimum of 24 hour notice unless the need is urgent (security patch, etc)
SA49.9	Minimum password requirements are; • Must be seven characters or more • Must contain at least one number or special character
SA49.9.1	These adhere to Epsilon password complexity standards on the Windows/Citrix hosts
SA49.10	Accounts are locked after 5 unsuccessful logon attempts
SA49.10.1	Account locking is done at the Citrix presentation layer

Training Design

50. Training

#	Design
SA50.1	5 days total to include; system (data model), SAS CI and BO user training
SA50.1.1	The first day of UAT includes training supplied by the Epsilon support team on the design of the solution
SA50.1.2	The second and third day of UAT includes training on SAS CD and Business Objects.
SA50.1.3	The training sessions are intended to equip the ANN INC testers and approvers with the necessary tools to effectively perform tool and data tests. Epsilon resources are made available throughout the Acceptance Testing phase to respond to questions on system design and tool functionality.

Approvals

This design document requires a signature approval containing the date, signature, and name of the individual approving its contents. Signature approval can be received in either written format (facsimile) or via e-mail under the individual's e-mail stationery. Signature approval shall represent that the Parties have read this document, understand its contents and agree to same.

Epsilon Data Management, Inc.

By: _____

Name: _____

Title: _____

Date: _____

ANN INC

By: _____

Name: _____

Title: _____

Date: _____

By: _____

Name: _____

Title: _____

Date: _____

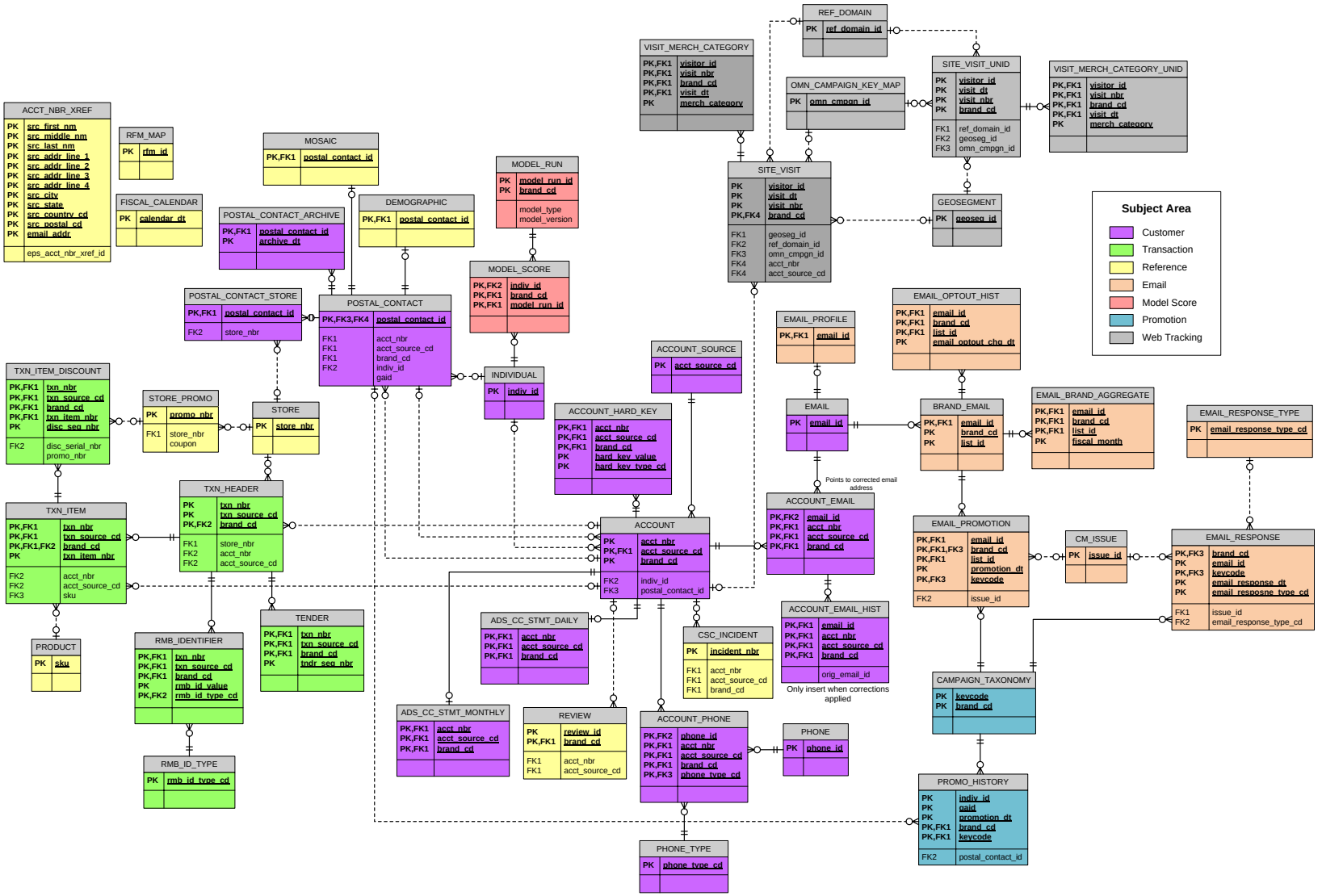
Appendix A – Data Models

This section illustrates the table relationships in both the Data Warehouse and Data Mart. These are simplified views intended to provide only key relationships and some of the more critical fields to the end user. The full detail data model, including all fields, is referenced in **Supporting Documents.**

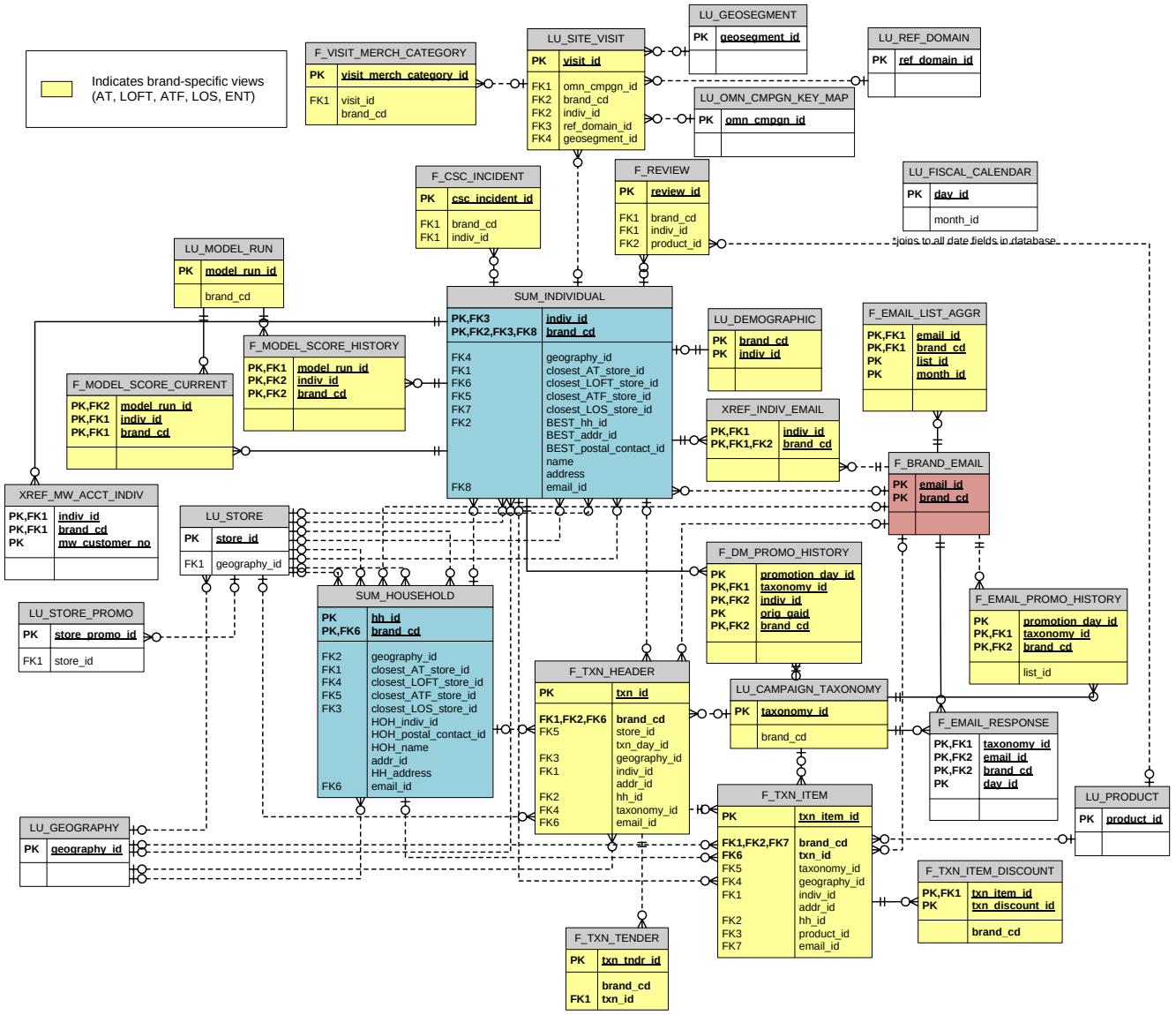
The Physical diagrams show all relationships as they exist in the underlying structures.

The Logical diagrams are focused on providing the user with the central BASE structure and an illustration of the pathways that are traversed in order to reference DIMENSION and FACT data structures appropriately. These diagrams are leveraged particularly for Campaign Management and Analytic exercises.

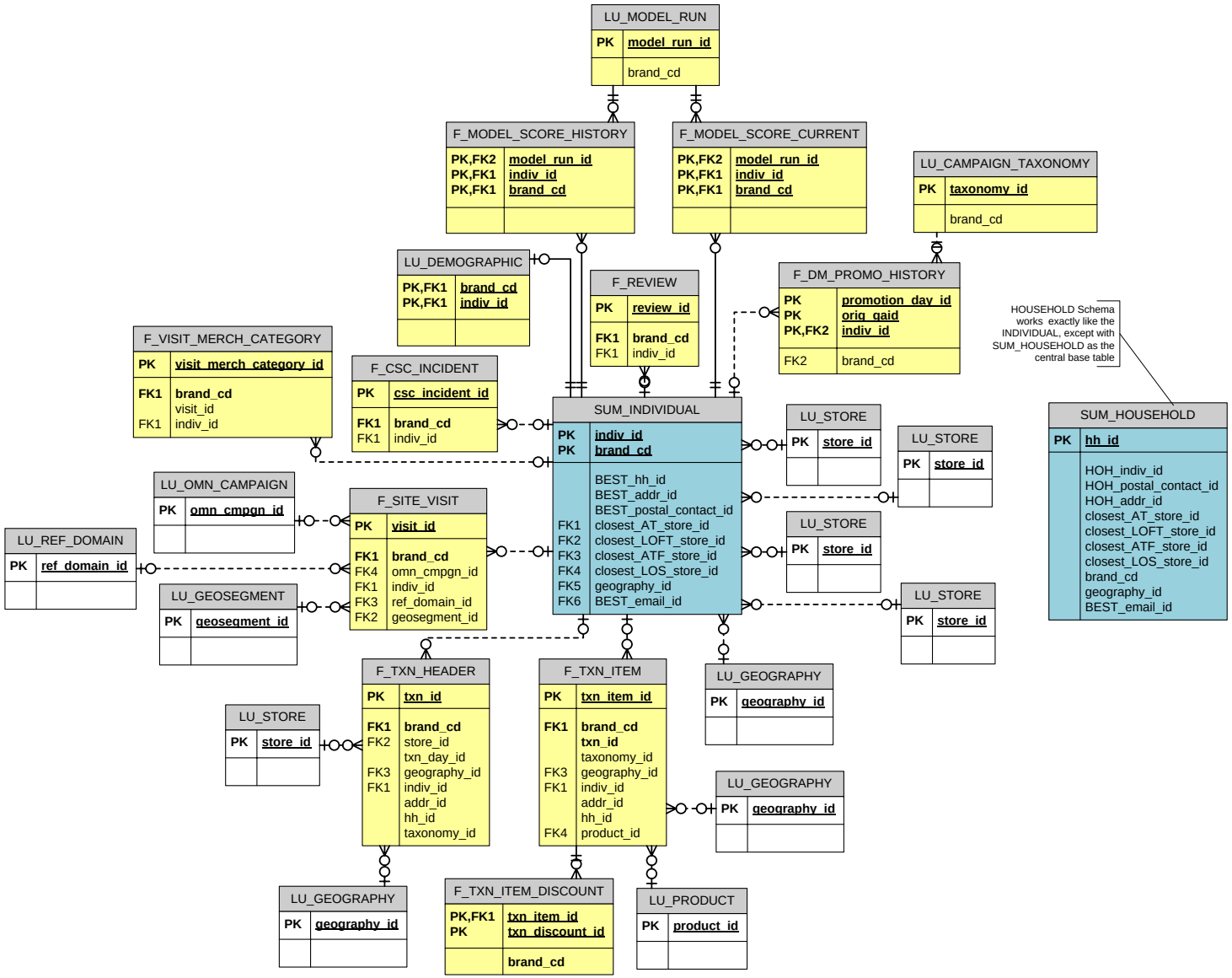
Data Warehouse Physical Diagram



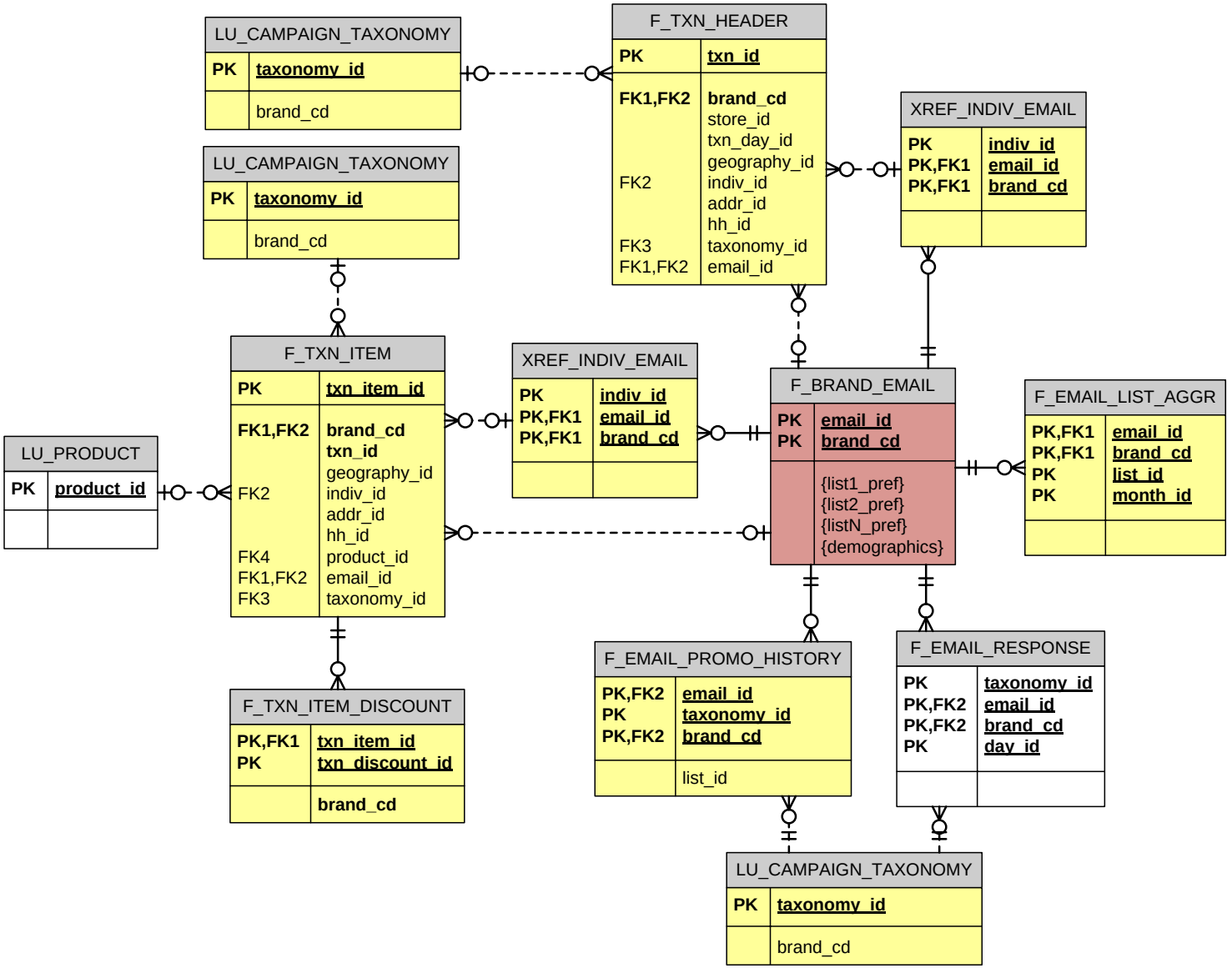
Data Mart Physical Diagram



Data Mart Logical Diagram – Individual/Household



Data Mart Logical Diagram – Email



Appendix B – Source to Target Overview

This section outlines the one or more structures that are mapped through each phase of the process as detailed in the **Source To Target Mapping Documents.**

Stage to Work

Source	STG_OWNER	WRK (primary)	WRK (secondary)	Migration Specific?
ADS	STG_ADS_CUSTDAILY	WRK_SRC_ACCOUNT		
ADS	STG_ADS_CUSTMTHLY0	WRK_SRC_ACCOUNT		
ATG	STG_ATG_BILLING_ADDRESS	WRK TXN_HEADER		
ATG	STG_ATG_BILLING_ADDRESS	WRK_SRC_ACCOUNT	XREF_ACCT_NBR	
ATG	STG_ATG_CATALOG_REQUEST	WRK_SRC_ACCOUNT	XREF_ACCT_NBR	
ATG	STG_ATG_COMMERCE_ITEM	WRK TXN_ITEM		
ATG	STG_ATG_CREDIT_CARD	WRK TXN_HEADER		
ATG	STG_ATG_DTL_ITM_PRICE_INFO	WRK TXN_ITEM		
ATG	STG_ATG_HARDGOOD_SHIP_GRP	WRK TXN_ITEM		
ATG	STG_ATG_ITEM_PRICE_INFO	WRK TXN_ITEM		
ATG	STG_ATG_ORDER	WRK TXN_HEADER		
ATG	STG_ATG_ORDER	WRK TXN_ITEM		
ATG	STG_ATG_ORDER_PRICE_INFO	WRK TXN_HEADER		
ATG	STG_ATG_PAYMENT_GROUP	WRK TXN_HEADER		
ATG	STG_ATG_PRICE_INFO	WRK TXN_HEADER		
ATG	STG_ATG_PROFILE	WRK_SRC_ACCOUNT		
ATG	STG_ATG_RETURN_ITEM	WRK TXN_ITEM		
ATG	STG_ATG_RETURN_REQUEST	WRK TXN_ITEM		
ATG	STG_ATG_SHIPPING_ADDRESS	WRK TXN_ITEM		
ATG	STG_ATG_SHIPPING_ADDRESS	WRK_SRC_ACCOUNT	XREF_ACCT_NBR	
ATG	STG_ATG_SHIPPING_GROUP	WRK TXN_ITEM		
AW	STG_AW_CUST_DTL	WRK_SRC_ACCOUNT	XREF_ACCT_NBR	
AW	STG_AW_CUST_DTL	WRK_TENDER		
AW	STG_AW_CUST_DTL	WRK_TXN_HEADER		
AW	STG_AW_CUST_DTL	WRK_TXN_ITEM		
AW	STG_AW_DISC_DTL	WRK_TXN_ITEM_DISC		
AW	STG_AW_LINE_NOTES	WRK_TENDER		
AW	STG_AW_MERCH_DTL	WRK_TXN_ITEM		
AW	STG_AW_RETURN_DTL	WRK_TXN_ITEM		
AW	STG_AW_TXN_HEADER	WRK_TENDER		
AW	STG_AW_TXN_HEADER	WRK_TXN_HEADER		
AW	STG_AW_TXN_HEADER	WRK_TXN_ITEM		
AW	STG_AW_TXN_LINE	WRK_TENDER		
AW	STG_AW_TXN_LINE	WRK_TXN_HEADER		
AW	STG_AW_TXN_LINE	WRK_TXN_ITEM		
KBM	STG_KBM_NZA	WRK_SRC_ACCOUNT	XREF_ACCT_NBR	
KBM	STG_KBM_REA	WRK_SRC_ACCOUNT	XREF_ACCT_NBR	
KBM	STG_KBM_TRADEAREA	WRK_SRC_ACCOUNT	XREF_ACCT_NBR	
MW	ADDRESS	WRK_SRC_ACCOUNT		Y
MW	CUSTOMER	WRK_SRC_ACCOUNT		Y
MW	ATTRIBUTE_GROUPING	WRK_SRC_ACCOUNT		Y
MW	ATTRIBUTE_LIST	WRK_SRC_ACCOUNT		Y
MW	CUSTOMER_ATTRIBUTE	WRK_SRC_ACCOUNT		Y
SAP	STG_SAP_ASSOCIATE	WRK_SRC_ACCOUNT	XREF_ACCT_NBR	
SDL	STG_SDL_DEMO	WRK_SRC_ACCOUNT	XREF_ACCT_NBR	

SDL	STG_SDL_SUBS	WRK_SRC_ACCOUNT	XREF_ACCT_NBR	

Work to Data Warehouse

Source	WRK	DW_OWNER
<various>	WRK_SRC_ACCOUNT	ACCOUNT
<various>	WRK_SRC_ACCOUNT	ACCOUNT_EMAIL
<various>	WRK_SRC_ACCOUNT	ACCOUNT_HARD_KEY
<various>	WRK_SRC_ACCOUNT	ACCOUNT_PHONE
<various>	WRK_SRC_ACCOUNT	BRAND_EMAIL
<various>	WRK_SRC_ACCOUNT	BRAND_EMAIL
<various>	WRK_SRC_ACCOUNT	EMAIL
<various>	WRK_SRC_ACCOUNT	INDIVIDUAL
<various>	WRK_SRC_ACCOUNT	PHONE
<various>	WRK_SRC_ACCOUNT	POSTAL_CONTACT
<various>	WRK_SRC_ACCOUNT	POSTAL_CONTACT
<various>	WRK_SRC_ACCOUNT	POSTAL_CONTACT_ARCHIVE
<various>	WRK_SRC_ACCOUNT	POSTAL_CONTACT_STORE
ATG	WRK_TXN_HEADER	TXN_HEADER
ATG	WRK_TXN_ITEM	TXN_ITEM
AW	WRK_TENDER	TENDER
AW	WRK_TXN_HEADER	TXN_HEADER
AW	WRK_TXN_ITEM	TXN_ITEM
AW	WRK_TXN_ITEM_DISC	TXN_ITEM_DISCOUNT

Stage to Data Warehouse

Source	STG_OWNER	DW_OWNER (primary)	DW_OWNER (secondary)	Migration Specific?
ADS	STG_ADS_CUSTDAILY	ADS_CC_STMT_DAILY		
ADS	STG_ADS_CUSTMTHLY0	ADS_CC_STMT_DAILY		
ADS	STG_ADS_CUSTMTHLY0	ADS_CC_STMT_MONTHLY		
ADS	STG_ADS_CUSTMTHLY7	ADS_CC_STMT_MONTHLY		
ADS	STG_ADS_XREFDAILY	ACCOUNT_HARD_KEY		
ATG	STG_ATG_ADJUSTMENT	TXN_ITEM_DISCOUNT		
ATG	STG_ATG_BILLING_ADDRESS	RMB_IDENTIFIER		
ATG	STG_ATG_COMMERCE_ITEM	TXN_ITEM_DISCOUNT		
ATG	STG_ATG_CREDIT_CARD	RMB_IDENTIFIER		
ATG	STG_ATG_CREDIT_CARD	TENDER		
ATG	STG_ATG_ORDER	RMB_IDENTIFIER		
ATG	STG_ATG_ORDER	TENDER		
ATG	STG_ATG_ORDER	TXN_ITEM_DISCOUNT		
ATG	STG_ATG_PAYMENT_GROUP	RMB_IDENTIFIER		
ATG	STG_ATG_PAYMENT_GROUP	TENDER		
ATG	STG_ATG_PRICE_INFO	TXN_ITEM_DISCOUNT		
BV	STG_BV_RATINGS	REVIEW		
CM	STG_CM_LOADER	EMAIL		
CRM	STG_CRM_CMPGN	CAMPAIGN_TAXONOMY		
CRM	STG_CRM_MODEL	MODEL_RUN		
CRM	STG_CRM_MODEL	MODEL_SCORE_CURRENT		
CRM	STG_CRM_MODEL	MODEL_SCORE_HISTORY		
CRM	STG_CRM_PROMO	PROMO_HISTORY		
CRM	STG_FISCAL_CALENDAR	FISCAL_CALENDAR		

EXP	STG_EXP_MOSAIC	MOSIAC		
EXP	STG_EXP_MOSAICMIGR	MOSIAC		Y
KBM	STG_KBM_DEMOGRAPHIC	DEMOGRAPHIC		
KBM	STG_KBM_DEMOGRAPHICMIGR	DEMOGRAPHIC		Y
MW	CLASS	PRODUCT		Y
MW	COLOR	PRODUCT		Y
MW	CUSTOMER_XREF	XREF_MW_ACCOUNT		Y
MW	DEPARTMENT	PRODUCT		Y
MW	PRODUCT_XREF	PRODUCT		Y
MW	STORE	STORE	Used to define BRAND on TXN	Y
MW	STYLE	PRODUCT		Y
MW	STYLE_XREF	PRODUCT		Y
MW	TENDER	TENDER		Y
MW	TI_TRANSACTION_COUPON_ERROR	TXN_ITEM_DISCOUNT		Y
MW	TI_TRANSACTION_DETAIL_ERROR	TXN_ITEM		Y
MW	TI_TRANSACTION_HEADER_ERROR	TXN_HEADER		Y
MW	TRANSACTION_COUPON	TXN_ITEM_DISCOUNT		Y
MW	TRANSACTION_DETAIL	TXN_ITEM		Y
MW	TRANSACTION_HEADER	TXN_HEADER		Y
MW	STG_MW_TRAN_TEND_ERR	RMB_IDENTIFIER		Y
MW	ATTRIBUTE_GROUPING	MODEL_SCORE_HISTORY		Y
MW	ATTRIBUTE_LIST	MODEL_SCORE_HISTORY		Y
MW	CUSTOMER_ATTRIBUTE	MODEL_SCORE_HISTORY		Y
OMNI	STG_OMNI_SAINT	OMN_CAMPAIGN_KEY_MAP		
OMNI	STG_OMNI_VISIT	SITE_VISIT	SITE_VISIT_UNID; GEOSEGMENT	
OMNI	STG_OMNI_VISITCATG	VISIT_MERCH_CATEGORY	VISIT_MERCH_CATEGORY_UNID	
RN	STG_RIGHTNOW_INCIDENT	CSC_INCIDENT		
SAP	STG_SAP_PRODUCT	PRODUCT		
SAP	STG_SAP_PROMO	STORE_PROMO		
SAP	STG_SAP_STORE	STORE		
SAP	STG_SAP_STYLE	PRODUCT		
SAP	STG_SAP_VENDOR	PRODUCT		
SDL	STG_SDL_AGGREGATE	EMAIL_BRAND_AGGREGATE		
SDL	STG_SDL_DATA_CLICK	EMAIL_RESPONSE		
SDL	STG_SDL_DATA_OPEN	EMAIL_RESPONSE		
SDL	STG_SDL_DATA_SENT	EMAIL_PROMOTION		
SDL	STG_SDL_DATA_TXN	EMAIL_RESPONSE		
SDL	STG_SDL_DEMO	EMAIL		Y
SDL	STG_SDL_DATA_UNSUB	EMAIL_RESPONSE		Y
SDL	STG_SDL_DEMO	EMAIL		
SDL	STG_SDL_DEMO	EMAIL_PROFILE		
SDL	STG_SDL_IIDKEYS	CM_ISSUE		
SDL	STG_SDL_SUBS	EMAIL		
SDL	STG_SDL_SUBS	EMAIL_PROFILE		
SDL	STG_SDL_SUBS	BRAND_EMAIL		
SDL	STG_SDL_UNSUBS	EMAIL		
SDL	STG_SDL_UNSUBS	BRAND_EMAIL		

Stage Unmapped

Source	STG_OWNER	Notes	Migration Specific?
ADS	STG_ADS_CUSTMTHLY9	n/a - used for audit	

ATG	STG_ATG_CATALOG_PREF		
ATG	STG_ATG_LOCAL_STORE		
ATG	STG_ATG_PROFBILL_ADDRESS		
ATG	STG_ATG_REFUND_METHOD		
ATG	STG_ATG_RELATED_ORDERS		
AW	STG_AW_AUTH_DTL		
AW	STG_AW_EXPCUST_DTL		
MW	DIVISION		Y
MW	TENDER_MAPPING		Y
MW	STG_MW_CUST_ALT_KEY		Y
MW	STG_MW_EMAIL_POS_ARCH	excluded from solution (cannot be used in RMB_IDENTIFIER)	Y

Data Warehouse to Data Mart

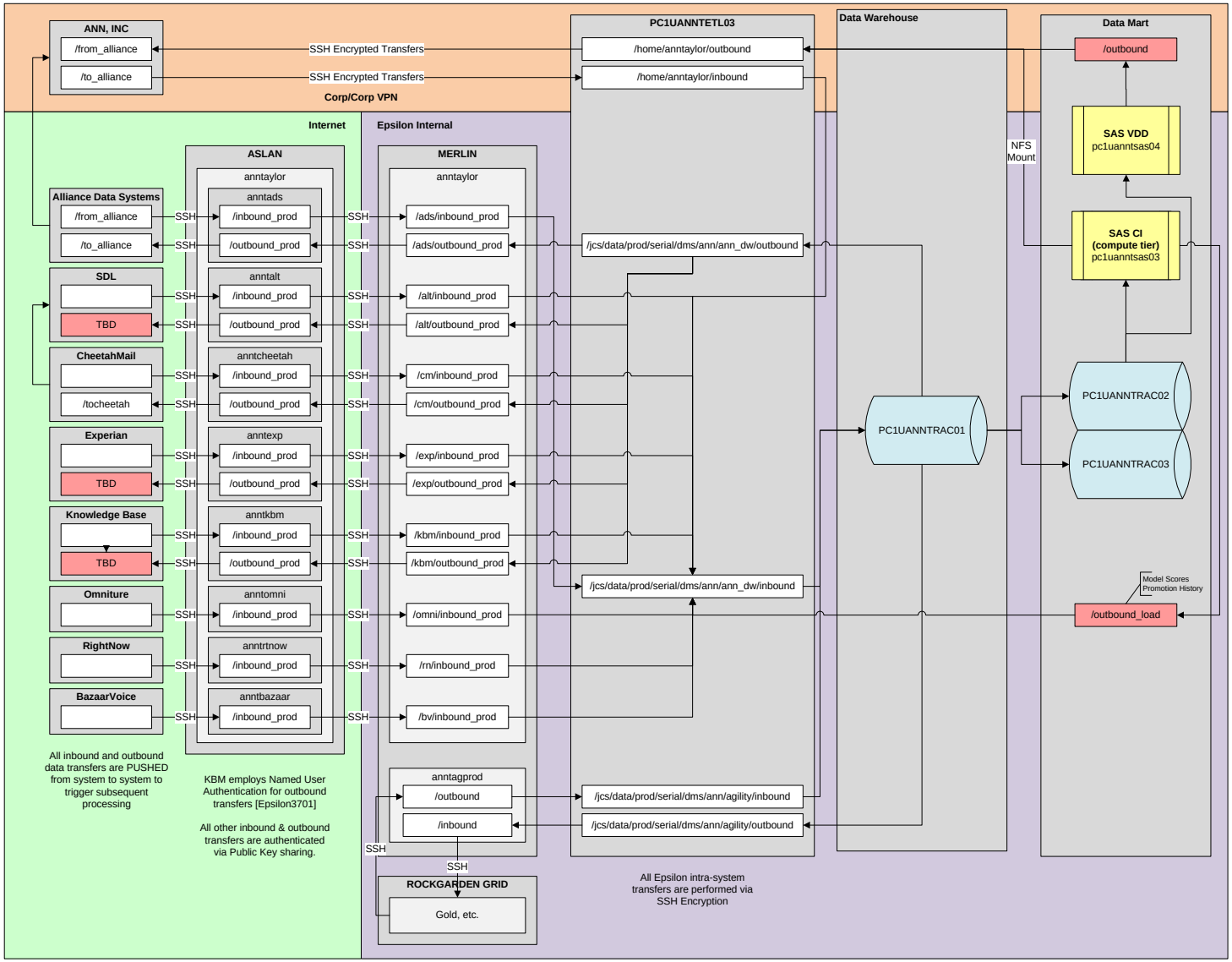
Subject	DW_OWNER	MART_OWNER	Notes
Customer	ACCOUNT_EMAIL	SUM_INDIVIDUAL	select BEST email
Customer	ACCOUNT_EXTENSION	SUM_INDIVIDUAL	selected based upon BEST postal contact
Customer	ACCOUNT_PHONE	SUM_INDIVIDUAL	select BEST phone
Customer	INDIVIDUAL	SUM_INDIVIDUAL	Includes rollups
Customer	INDIVIDUAL	XREF_INDIV_EMAIL	Used for linking all email addresses to an individual
Customer	INDIVIDUAL	XREF_INDIV_EMAIL	Used for linking all email addresses to an individual
Customer	PHONE	SUM_INDIVIDUAL	select BEST phone
Customer	POSTAL_CONTACT	SUM_HOUSEHOLD	Includes rollups
Customer	POSTAL_CONTACT	SUM_INDIVIDUAL	select BEST postal contact
Customer	POSTAL_CONTACT_STORE	SUM_INDIVIDUAL	Used to assign closest store(s) to an Individual/Household
Customer	SUM_INDIVIDUAL	SUM_HOUSEHOLD	Includes rollups
Customer	XREF_MW_ACCOUNT	XREF_MW_ACCT_INDIV	
Email	BRAND_EMAIL	F_BRAND_EMAIL	
Promotion	CM_ISSUE	LU_CAMPAIGN_TAXONOMY	use issue_id as keycode for email campaigns
Email	EMAIL	F_BRAND_EMAIL	
Email	EMAIL_BRAND_AGGREGATE	LU_EMAIL_BRAND_AGGREGATE	
Email	EMAIL_PROFILE	F_BRAND_EMAIL	
Model	MODEL_SCORE	F_MODEL_SCORE	
Model	MODEL_SCORE	LU_MODEL_TYPE	
Promotion	CAMPAIGN_TAXONOMY	LU_CAMPAIGN_TAXONOMY	
Promotion	EMAIL_PROMOTION	F_ONLINE_PROMO_HISTORY	
Promotion	EMAIL_RESPONSE	F_EMAIL_RESPONSE	Roll up responses by promotion/day/email
Promotion	PROMO_HISTORY	F_DM_PROMO_HISTORY	
Reference	<ref>	LU_GEOGRAPHY	Reference Table (postal geo)
Reference	CSC_INCIDENT	LU_CSC_INCIDENT	
Reference	DEMOGRAPHIC	LU_DEMOGRAPHIC	Demographic based on BEST postal contact
Reference	FISCAL_CALENDAR	LU_FISCAL_CALENDAR	
Reference	MOSIAC	LU_DEMOGRAPHIC	selected based upon BEST postal contact
Reference	PRODUCT	LU_PRODUCT	
Reference	REVIEW	LU_REVIEW	
Reference	STORE	LU_STORE	
Reference	STORE_PROMO	LU_STORE_PROMO	
Transaction	TENDER	F_TXN_TENDER	
Transaction	TXN_HEADER	F_TXN_HEADER	
Transaction	TXN_ITEM	F_TXN_ITEM	
Transaction	TXN_ITEM_DISC	F_TXN_ITEM_DISCOUNT	Might not be able to do this if we need coupon detail in mart
Web Tracking	GEOSEGMENT	LU_GEOSEGMENT	

Web Tracking	OMN_CAMPAIGN_KEY_MAP	LU_OMN_CMPGN_MAP	
Web Tracking	REF_DOMAIN	LU_REF_DOMAIN	
Web Tracking	SITE_VISIT	F_SITE_VISIT	
Web Tracking	VISIT_MERCH_CATEGORY	F_VISIT_MERCH_CATEGORY	

Data Warehouse Unmapped

DW_OWNER	Notes
ACCOUNT_HARD_KEY	Used for AGILITY
ACCOUNT_SOURCE	Used for selecting
ADS_CC_STMT_DAILY	
ADS_CC_STMT_MONTHLY	
DOMAIN_CORRECTION	METADATA - email hygiene engine
EMAIL_OPTOUT_HIST	Historical Tracking
MODEL_RUN	METADATA - model scoring
PHONE_TYPE	Used to select BEST phone (home)
POSTAL_CONTACT_ARCHIVE	Used to archive merged postal contacts
RFM_MAP	METADATA - model scoring
RMB_CURR_POOL	METADATA - retail matchback
RMB_ID_TYPE	METADATA - retail matchback
RMB_IDENTIFIER	METADATA - retail matchback
RMB_METHOD_VALUE	METADATA - retail matchback
RMB_POOL	METADATA - retail matchback
RMB_POOL_MEMBER	Retail matchback
SITE_VISIT_UNID	Promoted to mart once visits become identified
VISIT_MERCH_CATEGORY_UNID	Promoted to mart once visits become identified
XREF_ACCT_NBR	Used for Account assignment (AW txn, ATG txn, SAP assoc)

Appendix C – Data Flow Diagram



Appendix D – Infrastructure

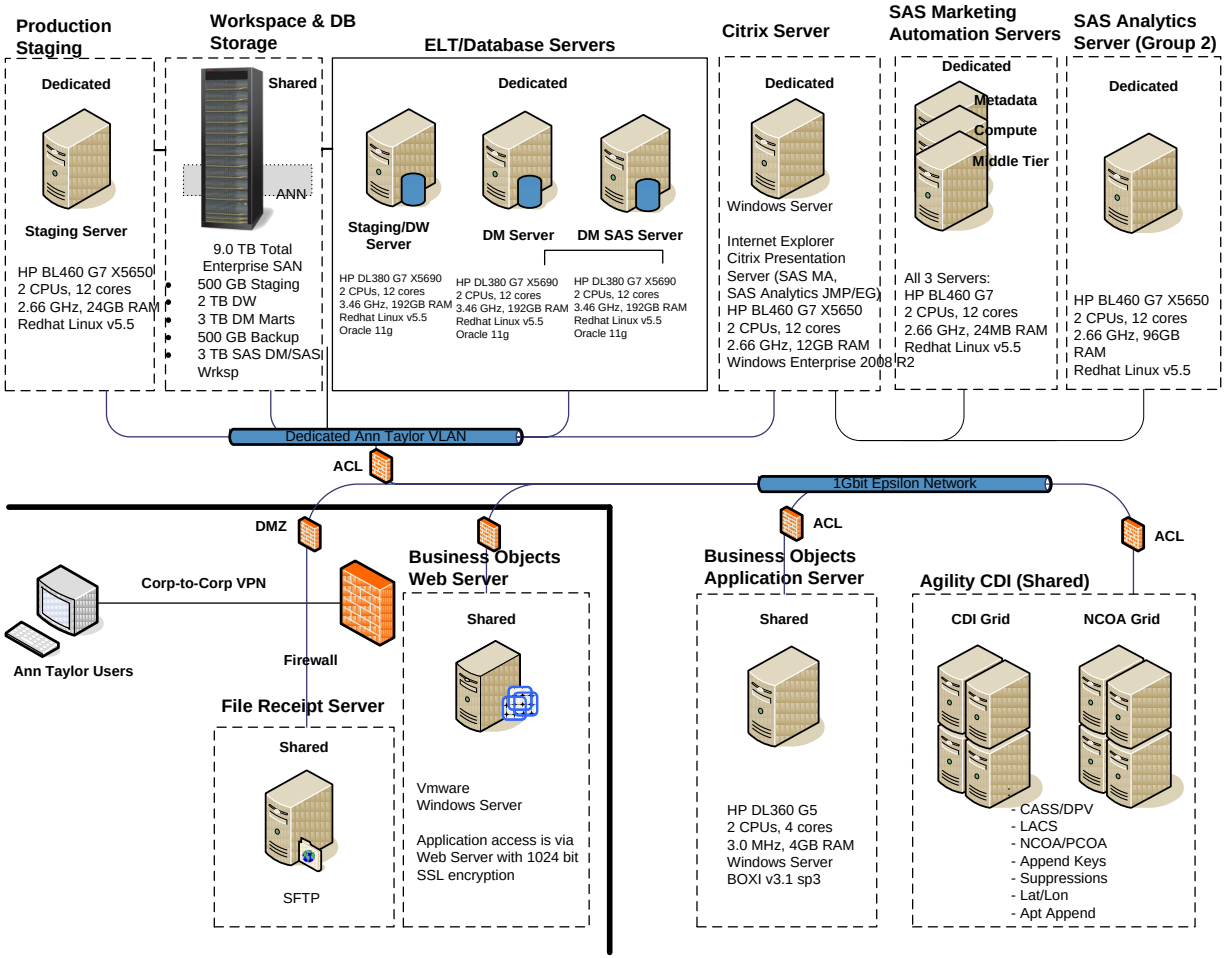
Hosts

Hostname	IP Address	Function	Environment	Description	Server	CPU	RAM	OS	Local Disk	Allocation	SAN
pc1uanntetl03	10.231.37.5	Staging Server	Production	HP Blade	BL460 G7 X5650 Blade Server	2 CPU, 12 Cores	24GB	Redhat Linux v5.5		ETL	500GB
pc1uanntrac01	10.231.38.1	Data Warehouse/ELT Server	Production	HP Blade	HP DL380 G7 X5690	2 CPU, 12 Cores	144GB	Redhat Linux v5.5		ASM	5TB
										ASM Vote	60GB
										Backup	1TB
										Oracle	60GB
										Install	30GB
pc1uanntrac01-vip	10.231.38.2								n/a	n/a	n/a
pc1uanntrac1-scan	10.231.38.7								n/a	n/a	n/a
pc1uanntrac02	10.231.38.3	Data Mart/ELT Server (for SAS)	Production	HP Blade	HP DL380 G7 X5690	2 CPU, 12 Cores	144GB	Redhat Linux v5.5		see above	see above
pc1uanntrac02-vip	10.231.38.4								n/a	see above	see above
pc1uanntrac1-scan	10.231.38.8								n/a	see above	see above
pc1uanntrac03	10.231.38.5	Data Mart/ELT Server	Production	HP Blade	HP DL380 G7 X5690	2 CPU, 12 Cores	144GB	Redhat Linux v5.5		see above	see above
pc1uanntrac03-vip	10.231.38.6								n/a	see above	see above
pc1uanntrac1-scan	10.231.38.9								n/a	see above	see above
pc1wanntctx01	10.230.19.1	Citrix Server	Production	HP Blade	BL460 G7 X5650 Blade Server	2 CPU, 12 Cores	12GB	Windows	50GB	NTFS	100GB
pc1uanntsas01	10.231.37.1	SAS Campaign Metedata	Production	HP Blade	BL460 G7 X5650 Blade Server	2 CPU, 12 Cores	24GB	Redhat Linux v5.5		SAS servers	1TB
										Reserve	2.5TB
pc1uanntsas02	10.231.37.2	SAS Campaign Compute	Production	HP Blade	BL460 G7 X5650 Blade Server	2 CPU, 12 Cores	24GB	Redhat Linux v5.5		see above	see above
pc1uanntsas03	10.231.37.3	SAS Campaign Middle Tier	Production	HP Blade	BL460 G7 X5650 Blade Server	2 CPU, 12 Cores	24GB	Redhat Linux v5.5		see above	see above
pc1uanntsas04	10.231.37.4	SAS Analytics Application Server	Production	HP Blade	BL460 G7 X5650 Blade Server	1 CPU, 4 Cores	48GB	Redhat Linux v5.5		/sasdata	-300GB
utxanntetld01	159.127.171.39	Staging Server	Dev/QA/DR	HP Blade	BL460 G7 X5650 Blade Server	2 CPU, 12 Cores	24GB	Redhat Linux v5.5	10GB	ETL	200GB
utxanntracd01	159.127.171.40	Data Warehouse/Mart/ELT	Dev/QA/DR	HP Blade	BL460 G7 X5650 Blade Server	2 CPU, 12 Cores	96GB	Redhat Linux v5.5		ASM	2.1TB
										Backup	700GB

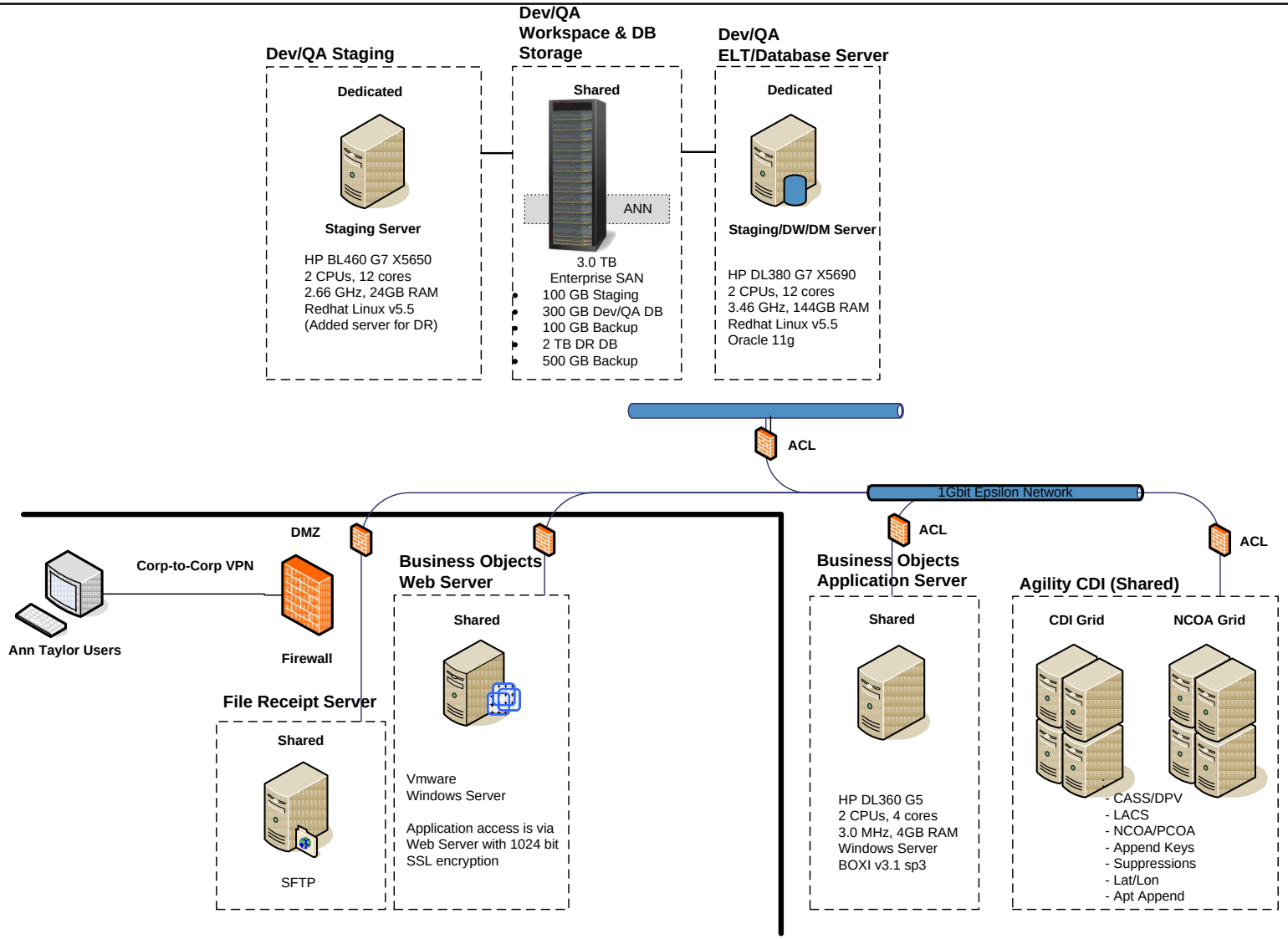
		Server								Oracle Install	60GB 30GB

Infrastructure Diagram

Ann Taylor Marketing DB Physical Architecture - Production



Ann Taylor Marketing DB Physical Architecture – Dev/QA



ELT Standards

RAC - two instances on same node - separate TNS entry	
Node 1	STG/DW
Node 2	SAS Mart
Node 3	BO Mart

DB Schemas	Description
RAW_OWNER	Raw tables
STG_OWNER	Stage tables
DW_OWNER	Data warehouse tables
OUT_OWNER	Outbound feed tables
DMA_OWNER	Mart A-side tables
DMB_OWNER	Mart B-side tables
BI_OWNER	Owner of BI views
CM_OWNER	Owner of Campaign Management views
SAS_OWNER	Owner of SAS views
ELT_OWNER	Framework meta-data
PKG_OWNER	DBA package owner: analyze, index rebuild, truncate, synonym, swap, etc.
ELT_USER	Run ETL jobs
BI_USER	Business Objects Application Account
CM_USER	SAS MA Application Account

DB Roles	Description
BI_READ	Read on appropriate views
CM_READ	Read on appropriate views
SAS_READ	Read on appropriate views
ELT	Read, Insert, Update, Delete on all tables
ELT_SUPPORT	Read, Insert, Update, Delete on framework meta-tables; non-ELT reference tables
GLOBAL_READ	Read all views/tables
ELT_READ	Required for Framework
ELT_WRITE	Required for Framework
ELT_READWRITE	Required for Framework

Object Prefix	
WRK_	
MAP_	code/desc pairs
LU_	Dimension
F_	Fact
SUM_	Summary
RPT_	Reporting Tables
AUD_	Audit
LKP_	Framework temp
FLY_	Framework temp
TMP_	Framework temp

Object Suffix	
_PK	Primary Key
_IX	Index
_FK	Build Contstraints
_UK	Unique Key
_SEQ	Field Name, Sequence

_VW	View

Field Suffix	
_ID	Epsilon derived ID
_NBR	Source ID
_DT	Date
_NM	Name
_CD	Code
_AMT	Amount (15,2)
_QTY	Quantity
_DESC	Description
_PCT	Percent (5,2)
_FLG	Flag (Y,N,U)
_IND	Indicator (0/1)
_TS	Timestamp

Framework RAW	Description
FILE_ID	
RECORD_NBR	
ALTERNATE_KEY	
Framework STG	
ACTIVITY_DT	
FILE_ID	
RECORD_NBR	
ALTERNATE_KEY	
FILE_TS	
SOURCE_SYSTEM_CODE	
Framework DW	
ACTIVITY_DT	TIMESTAMP?
CREATED_JOB_ID	NUMERIC
CREATED_DT	DATE
MODIFIED_JOB_ID	NUMERIC
MODIFIED_DT	DATE
ALTERNATE_KEY	Numeric (20,10)
ORIG_ALTERNATE_KEY	Numeric (20,10)

Agility Keys	
INDIV_ID	NUMERIC
HH_ID	NUMERIC
GAID	NUMERIC

Version Control	Subversion

Process Standard	Consideration
LOAD	append YYYYMMDDHH24MISS
RAW	original format
STG	data types cast; upper (except where noted); trim
DW	business rules

Tablespace Standards	
PCTFREE=0	
NOCOMPRESS	
LOGGING	

Script Naming Convention	

Create Table	cr_tb_${tablename}.sql
	Note: no DROP, no OWNER, PK, INX, Contraints
Create Sequence	cr_seq_${seqname}.sql
Load Table	ld_tb_${tablename}.sql
Create Synonym	cr_syn_${owner}
Create View	cr_vw_${viewname}.ddl
Extract Table	xt_tb_${tablename}.sql

Directory Structure	
Scripts	/jcs/sand/${platform}/dms/ann
Data Warehouse	ann_dw
Korn Shell	run
ELT SQL	sql
Control Files	db
DB DDL	ddl
Raw Schema	RAW_OWNER
Stage Schema	STG_OWNER
Data Warehouse Schema	DW_OWNER
Data Warehouse	ann_dm
Korn Shell	run
ELT SQL	sql
Control Files	db
DB DDL	ddl
DMA/DMB Schemas	DM_OWNER
Data	/jcs/data/${platform}/prod/serial/dms/ann
DW/DM data	ann_dw
Inbound	inbound
Outbound	outbound
Logs	logs
Rejects	reject
Work/Temp	work
Archive	archive
Agility Data	agility
Inbound	inbound
Outbound	outbound
Logs	logs
Rejects	reject
Work/Temp	work
Archive	archive
Config	/jcs/conf
JCS Environment	config.${hostname}.${platform}
Path/Authorization	config.ann.ods.${platform}

Appendix E – Process Dependencies

This section describes which update processes must be complete before another process may execute because the downstream process requires the most current dataset in order to update properly.

Process dependencies are fully defined upon final code review prior to System Integration Testing

Source	Target	Dependency
<various>	WRK_SRC_ACCOUNT	<staging process>
<various>	TXN_HEADER	WRK_SRC_ACCOUNT
<various>	TXN_ITEM	WRK_SRC_ACCOUNT
<various>	AGILITY	STORE
<various>	AGILITY	WRK_SRC_ACCOUNT
STG_BV_RATINGS	REVIEW (bv)	EMAIL_ACCOUNT

Appendix F – Escalation Procedures

ANN INC and Vendor Escalation

In the event that an audit failure occurs Epsilon escalates with the appropriate vendor as described below.

This section is to be fully defined as part of the transition period into the implementation MANAGE phase

Vendor	Primary Contact	Email Address	Phone	Alternate Contact	Email Address	P
ADS	Jennifer Brown	Jennifer.Brown@alliancedata.com	TBD	Leigh Helsel	Leigh.Helsel@alliancedata.com	T
ANN	Suresh K Kancherla	Suresh_Kancherla@anninc.com	2125413660	Geoffrey Oliver	Geoffrey_Oliver@anninc.com	T
BazaarVoice	Jon Hauser	jonathan.hauser@bazaarvoice.com	TBD	Christensen Sarah	Sarah_Christensen@anninc.com	T
CheetahMail	Andrew Perell	Andrew.Perell@experian.com	TBD	Rob Andrezzi	Rob.Andrezzi@experian.com	T
Experian	Matt Schwarz	Matt.Schwarz@experian.com	TBD	TBD	TBD	T
KBM	Gary Zupke	Gary.Zupke@kbmg.com	TBD	TBD	TBD	T
Omniture	Mark Heidbrink	mheidbri@adobe.com	TBD	Emily Gasner	emily@gasneranalytics.com	T
RightNow	Crystal Ashton	Crystal.Ashton@teleperformance.com	801.257.6275	TBD	TBD	T
Alterian (SDL)	Bobbi Scelfo	bscelfo@sdl.com	312 919 1685			T

All vendor communication includes the ANN INC personnel listed below:

Name	Contact
Chuck Chapek	
TBD	

Epsilon Escalation

The Epsilon support team is available to address requests fielded by the ANN INC CRM team. All escalation requests including data/process questions, tool support, and infrastructure issues are managed as follows:

Escalation Level	Role	Contact
1	Support Team	SDS-SCH-ANN_Support@epsilon.com
2	Relationship Manager	See Below
3	General Manager	See Below

The Epsilon Manage Support Team and contact information are identified below

Role	Name	Email	Location	Office	Mobile
Relationship Manager	Barnum Bennett	bbennett@epsilon.com	Lafayette, CO		
Business Lead	Denise Bernat	dbernat@epsilon.com	Schaumburg, IL		
Technical Lead	Mark Beening	mbeening@epsilon.com	Schaumburg, IL	847.592.1151	847.845.0043
Business Analyst	TBD – as we approach manage phase				
Production Analyst	TBD – as we				

	approach manage phase				

Epsilon Manage Responsibilities

This section describes the manage responsibilities beyond solution support that have been specifically requested.

Responsibility	Schedule
Weekly project management meeting	TBD as we transition in the Manage phase
Monthly user access report	First week of month
Downtime Request	24 hour notification

Appendix G – External Access

Data Transfer

Vendor	user	Direction	SSH Key File	platform	aslan path	merlin path	dedicated host	dedicated path	external host
ADS	anntads	Inbound	ssh_host_key.pub	dev	/inbound_dev	/ads/inbound_dev	utxanntetld01	/jcs/data/dev/serial/dms/ann/ann_dw/inbound	est.alliancedata.com
				qa	/inbound_qa	/ads/inbound_qa	utxanntetld01	/jcs/data/qa/serial/dms/ann/ann_dw/inbound	
				prod	/inbound_prod	/ads/inbound_prod	pc1uanntetl03	/jcs/data/prod/serial/dms/ann/ann_dw/inbound	
		Outbound	ssh_host_*_key.pub	prod	/outbound	/ads/outbound	pc1uanntetl03	/jcs/data/prod/serial/dms/ann/ann_dw/outbound	est.alliancedata.com
ANN	anntaylor	Inbound		dev	/inbound_dev	/inbound_dev	utxanntetld01	/jcs/data/dev/serial/dms/ann/ann_dw/inbound	
				qa	/inbound_qa	/inbound_qa	utxanntetld01	/jcs/data/qa/serial/dms/ann/ann_dw/inbound	
				prod	/inbound_prod	/inbound_prod	pc1uanntetl03	/jcs/data/prod/serial/dms/ann/ann_dw/inbound	
BazaarVoice	anntbazaar	Inbound		dev	/inbound_dev	/bv/inbound_dev	utxanntetld01	/jcs/data/dev/serial/dms/ann/ann_dw/inbound	
				qa	/inbound_qa	/bv/inbound_qa	utxanntetld01	/jcs/data/qa/serial/dms/ann/ann_dw/inbound	
				prod	/inbound_prod	/bv/inbound_prod	pc1uanntetl03	/jcs/data/prod/serial/dms/ann/ann_dw/inbound	
CheetahMail	anntcheetah	Inbound	rsa.txt	dev	/inbound_dev	/cm/inbound_dev	utxanntetld01	/jcs/data/dev/serial/dms/ann/ann_dw/inbound	
				qa	/inbound_qa	/cm/inbound_qa	utxanntetld01	/jcs/data/qa/serial/dms/ann/ann_dw/inbound	
				prod	/inbound_prod	/cm/inbound_prod	pc1uanntetl03	/jcs/data/prod/serial/dms/ann/ann_dw/inbound	
		Outbound	ssh_host_*_key.pub	prod	/outbound_prod	/cm/outbound_prod	pc1uanntetl03	/jcs/data/prod/serial/dms/ann/ann_dw/outbound	tt.cheetahmail.com
Experian	anntexp	Inbound		dev	/inbound_dev	/exp/inbound_dev	utxanntetld01	/jcs/data/dev/serial/dms/ann/ann_dw/inbound	
				qa	/inbound_qa	/exp/inbound_qa	utxanntetld01	/jcs/data/qa/serial/dms/ann/ann_dw/inbound	
				prod	/inbound_prod	/exp/inbound_prod	pc1uanntetl03	/jcs/data/prod/serial/dms/ann/ann_dw/inbound	
		Outbound		prod	/outbound_prod	/exp/outbound_prod	pc1uanntetl03	/jcs/data/prod/serial/dms/ann/ann_dw/outbound	stm.experian.com
KBM	anntkbm	Inbound		dev	/inbound_dev	/kbm/inbound_dev	utxanntetld01	/jcs/data/dev/serial/dms/ann/ann_dw/inbound	
				qa	/inbound_qa	/kbm/inbound_qa	utxanntetld01	/jcs/data/qa/serial/dms/ann/ann_dw/inbound	
				prod	/inbound_prod	/kbm/inbound_prod	pc1uanntetl03	/jcs/data/prod/serial/dms/ann/ann_dw/inbound	
		Outbound		prod	/outbound_prod	/kbm/outbound_prod	pc1uanntetl03	/jcs/data/prod/serial/dms/ann/ann_dw/outbound	securexfer.kbm1.com
Omniture	anntomni	Inbound	authorized_keys	dev	/inbound_dev	/omni/inbound_dev	utxanntetld01	/jcs/data/dev/serial/dms/ann/ann_dw/inbound	
				qa	/inbound_qa	/omni/inbound_qa	utxanntetld01	/jcs/data/qa/serial/dms/ann/ann_dw/inbound	
				prod	/inbound_prod	/omni/inbound_prod	pc1uanntetl03	/jcs/data/prod/serial/dms/ann/ann_dw/inbound	
RightNow	anntrtnow	Inbound		dev	/inbound_dev	/rn/inbound_dev	utxanntetld01	/jcs/data/dev/serial/dms/ann/ann_dw/inbound	
				qa	/inbound_qa	/rn/inbound_qa	utxanntetld01	/jcs/data/qa/serial/dms/ann/ann_dw/inbound	
				prod	/inbound_prod	/rn/inbound_prod	pc1uanntetl03	/jcs/data/prod/serial/dms/ann/ann_dw/inbound	
Alterian (SDL)	anntalt	Inbound		dev	/inbound_dev	/alt/inbound_dev	utxanntetld01	/jcs/data/dev/serial/dms/ann/ann_dw/inbound	
				qa	/inbound_qa	/alt/inbound_qa	utxanntetld01	/jcs/data/qa/serial/dms/ann/ann_dw/inbound	
				prod	/inbound_prod	/alt/inbound_prod	pc1uanntetl03	/jcs/data/prod/serial/dms/ann/ann_dw/inbound	
		Outbound	ssh_host_*_key.pub	prod	/outbound_prod	/kbm/outbound_prod	pc1uanntetl03	/jcs/data/prod/serial/dms/ann/ann_dw/outbound	Xfer.alterianconnect.com

Agility	anntagqa	Inbound	unknown	dev	n/a	/inbound	utxanntetld01	/jcs/data/prod/serial/dms/ann/agility/inbound	n/a	
				qa	n/a		utxanntetld01	/jcs/data/prod/serial/dms/ann/agility/inbound	n/a	
	anntagprod			prod	n/a	/outbound	utxanntetld01	/jcs/data/prod/serial/dms/ann/agility/inbound	n/a	
	anntagqa	Outbound	ssh_host_*_key.pub	dev	n/a	/outbound	utxanntetld01	/jcs/data/prod/serial/dms/ann/agility/outbound	n/a	
				qa	n/a		utxanntetld01	/jcs/data/prod/serial/dms/ann/agility/outbound	n/a	
	anntagprod			prod	n/a	/outbound	utxanntetld01	/jcs/data/prod/serial/dms/ann/agility/outbound	n/a	

Tool Access Matrix

SAS CI Users Remaining	2
SAS VDD Users Remaining	2
BusObj Users Remaining	2

Access to any hosts included in the ANN INC CRM solution must be approved by **Chuck Chapek**

ID	User	Originating IP Address	Citrix User	SAS VDD User	SAS CI User	BO User
1	Chuck Chapek	VPN	1	1	1	1
2	Dan Williams		1	1	1	1
3	Sue Lee		1	1	1	1
4	Patricia Laing		1	1	1	1
5	Jaclyn Miller		1	1	1	1
6	Bob Arciero		1	1	1	1
7	Dianne Binford		1	1	1	1
9	Jessica Ng		1	1	1	1
9						
10						
Total			8	8	8	8

Appendix H – Audit Reports

Samples of the standard audit reports are provided below.

Staging

The Staging audit assess each feed as it is received, compared to control, loaded and prepared for load into the Data Warehouse. Any rejects are compared to the expected tolerance.

ANN CRM Staging Audit

Update Week 2013-05-08

	ADS	ATG	AW	BV	CM	KBM	OMNI	RN
CCXREF	2013-05-08	N/A	N/A	N/A	N/A	N/A	N/A	N/A
CUST	2013-05-08	2013-05-08	N/A	N/A	N/A	N/A	N/A	N/A
TXNDTL	N/A	2013-05-08	2013-05-08	N/A	N/A	N/A	N/A	N/A
TXNHDR	N/A	2013-05-08	2013-05-08	N/A	N/A	N/A	N/A	N/A
TXNTNDR	N/A	2013-05-08	2013-05-08	N/A	N/A	N/A	N/A	N/A
TXNCOUP	N/A	N/A	2013-05-08	N/A	N/A	N/A	N/A	N/A
PRODRATING	N/A	N/A	N/A	2013-05-08	N/A	N/A	N/A	N/A
DEMO	N/A	N/A	N/A	N/A	2013-05-08	N/A	N/A	N/A
EVENTCLICK	N/A	N/A	N/A	N/A	2013-05-08	N/A	N/A	N/A
EVENTOPEN	N/A	N/A	N/A	N/A	2013-05-08	N/A	N/A	N/A
EVENTUNSUB	N/A	N/A	N/A	N/A	2013-05-08	N/A	N/A	N/A
IIDKEYS	N/A	N/A	N/A	N/A	2013-05-08	N/A	N/A	N/A
LOADER	N/A	N/A	N/A	N/A	2013-05-08	N/A	N/A	N/A
SUBS	N/A	N/A	N/A	N/A	2013-05-08	N/A	N/A	N/A
UNSUBS	N/A	N/A	N/A	N/A	2013-05-08	N/A	N/A	N/A
NZAPPEND	N/A	N/A	N/A	N/A	N/A	2013-05-08	N/A	N/A
SAINT	N/A	N/A	N/A	N/A	N/A	N/A	2013-05-08	N/A

VISIT	N/A	N/A	N/A	N/A	N/A	N/A	2013-05-08	N/A
VISITPROD	N/A	N/A	N/A	N/A	N/A	N/A	2013-05-08	N/A
INCIDENT	N/A	N/A	N/A	N/A	N/A	N/A	N/A	2013-05-08

Source	Feed	Source Extract	File Name	Control Dates	Expected Qty	File Qty	Load Qty	Load Reject	Stage Qty	Stage Reject	File ID	Audit Feed Missing	Audit Control Missing	Audit Control Quantity	Audit Control Dates	Audit Multiple Files	Audit Reject Threshold	Audit PK Threshold	Audit RI Threshold
ADS	CCXREF	INCREMENTAL	ANN_ADS_CCXREF_20130508.dat	2013-05-08 - 2013-05-08	171965	171965	171965	0	171965	0	122315	N	N	N	N	N	N	N	N
ADS	CUST	INCREMENTAL	ANN_ADS_CUST_20130508.dat	2013-05-08 - 2013-05-08	4303358	4303358	4303358	0	4303358	0	122320	N	N	N	N	N	N	N	N
Source	**Feed**	**Source Extract**	**File Name**	**Control Dates**	**Expected Qty**	**File Qty**	**Load Qty**	**Load Reject**	**Stage Qty**	**Stage Reject**	**File ID**	**Audit Feed Missing**	**Audit Control Missing**	**Audit Control Quantity**	**Audit Control Dates**	**Audit Multiple Files**	**Audit Reject Threshold**	**Audit PK Threshold**	**Audit RI Threshold**
ATG	CUST	INCREMENTAL	ANN_ATG_CUST_20130508.dat	2013-05-08 - 2013-05-08	2991359	2991359	2991359	0	2991359	0	122368	N	N	N	N	N	N	N	N
ATG	TXNDTL	INCREMENTAL	ANN_ATG_TXNDTL_20130508.dat	2013-05-08 - 2013-05-08	2085738	2085738	2085732	6	2085732	0	122371	N	N	N	N	N	N	N	N
ATG	TXNHDR	INCREMENTAL	ANN_ATG_TXNHDR_20130508.dat	2013-05-08 - 2013-05-08	220070	220070	220070	0	220070	0	122366	N	N	N	N	N	N	N	N
ATG	TXNTNDR	INCREMENTAL	ANN_ATG_TXNTNDR_20130508.dat	2013-05-08 - 2013-05-08	837806	837806	837806	0	837806	0	122367	N	N	N	N	N	N	N	N
Source	**Feed**	**Source Extract**	**File Name**	**Control Dates**	**Expected Qty**	**File Qty**	**Load Qty**	**Load Rej**	**Stage Qty**	**Stage Rej**	**File ID**	**Audit Fee**	**Audit Con**	**Audit Cont**	**Audit Con**	**Audit Mul**	**Audit Reject**	**Audit PK Thres**	**Audit RI Thres**

								ect		ect		d Missing	trol Missing	rol Quantity	trol Dates	tiple Files	Threshold	hold	hold
AW	TXNDTL	INCREMENTAL	ANN_AW_TXNDTL_20130508.dat	2013-05-08 - 2013-05-08	2085738	2085738	2085732	6	2085732	0	122371	N	N	N	N	N	N	N	N
AW	TXNHDR	INCREMENTAL	ANN_AW_TXNHDR_20130508.dat	2013-05-08 - 2013-05-08	220070	220070	220070	0	220070	0	122366	N	N	N	N	N	N	N	N
AW	TXNTNDR	INCREMENTAL	ANN_AW_TXNTNDR_20130508.dat	2013-05-08 - 2013-05-08	837806	837806	837806	0	837806	0	122367	N	N	N	N	N	N	N	N
AW	TXNCOUP	INCREMENTAL	ANN_AW_TXNCOUP_20130508.dat	2013-05-08 - 2013-05-08	837806	837806	837806	0	837806	0	122367	N	N	N	N	N	N	N	N
Source	**Feed**	**Source Extract**	**File Name**	**Control Dates**	**Expected Qty**	**File Qty**	**Load Qty**	**Load Reject**	**Stage Qty**	**Stage Reject**	**File ID**	**Audit Feed Missing**	**Audit Control Missing**	**Audit Control Quantity**	**Audit Control Dates**	**Audit Multiple Files**	**Audit Reject Threshold**	**Audit PK Threshold**	**Audit RI Threshold**
BV	PRODRATING	REPLACEMENT	ANN_BV_PRODRATING_20130508.dat	2013-05-08 - 2013-05-08	1007832	1007832	1007234	598	1007233	1	122274	N	N	N	N	N	N	N	N
Source	**Feed**	**Source Extract**	**File Name**	**Control Dates**	**Expected Qty**	**File Qty**	**Load Qty**	**Load Reject**	**Stage Qty**	**Stage Reject**	**File ID**	**Audit Feed Missing**	**Audit Control Missing**	**Audit Control Quantity**	**Audit Control Dates**	**Audit Multiple Files**	**Audit Reject Threshold**	**Audit PK Threshold**	**Audit RI Threshold**
CM	SUBS	INCREMENTAL	ANN_CM_SUBS_20130508.dat	-	0	41241	41241	0			122379	N	N	N	N	N	N	N	N
CM	UNSUBS	INCREMENTAL	ANN_CM_UNSUBS_20130508.dat	-	0	814717	814717	0			122378	N	N	N	N	N	N	N	N
CM	IID_KEYS	INCREMENTAL	ANN_CM_IID_KEYS_20130508.dat	-	0	1293	1293	0			122415	N	N	N	N	N	N	N	N
CM	DATA_BULK	INCREMENTAL	ANN_CM_DATA_20130508_bulk.dat	-	0	100980	100980	0			122420	N	N	N	N	N	N	N	N

						2	2												
CM	DATA_CLICK	INCREMENTAL	ANN_CM_DATA_20130508_click.dat	-	0	29751	29751	0			122419	N	N	N	N	N	N	N	N
CM	DATA_OPEN	INCREMENTAL	ANN_CM_DATA_20130508_open.dat	-	0	204122	204122	0			122418	N	N	N	N	N	N	N	N
CM	DATA_UNSUB	INCREMENTAL	ANN_CM_DATA_20130508_unsub.dat	-	0	2062	2062	0			122417	N	N	N	N	N	N	N	N
CM	DEMO	INCREMENTAL	ANN_CM_DEMO_20130508.dat	-	0	381744	381385	359			122416	N	N	N	N	N	N	N	N
CM	LOADER	INCREMENTAL	ANN_CM_LOADER_20130508_7.dat	-	0	8	8	0			122402	N	N	N	N	N	N	N	N
CM	LOADER	INCREMENTAL	ANN_CM_LOADER_20130508_5.dat	-	0	6	6	0			122404	N	N	N	N	N	N	N	N
CM	LOADER	INCREMENTAL	ANN_CM_LOADER_20130508_2.dat	-	0	34	34	0			122407	N	N	N	N	N	N	N	N
CM	LOADER	INCREMENTAL	ANN_CM_LOADER_20130508_3.dat	-	0	22	22	0			122406	N	N	N	N	N	N	N	N
CM	LOADER	INCREMENTAL	ANN_CM_LOADER_20130508_10.dat	-	0	6	6	0			122409	N	N	N	N	N	N	N	N
Source	**Feed**	**Source Extract**	**File Name**	**Control Dates**	**Expected Qty**	**File Qty**	**Load Qty**	**Load Reject**	**Stage Qty**	**Stage Reject**	**File ID**	**Audit Feed Missing**	**Audit Control Missing**	**Audit Control Quantity**	**Audit Control Dates**	**Audit Multiple Files**	**Audit Reject Threshold**	**Audit PK Threshold**	**Audit RI Threshold**
KBM	NZAPPEND	INCREMENTAL	ANN_KBM_NZAPPEND_20130508.dat	-	0	11567	11567	0			122421	N	N	N	N	N	N	N	N
Source	**Feed**	**Source Extract**	**File Name**	**Control Dates**	**Expected Qty**	**File Qty**	**Load Qty**	**Load Reject**	**Stage Qty**	**Stage Reject**	**File ID**	**Audit Feed Missing**	**Audit Control Missing**	**Audit Control Quantity**	**Audit Control Dates**	**Audit Multiple Files**	**Audit Reject Threshold**	**Audit PK Threshold**	**Audit RI Threshold**
OM	SAINT	INCREM	ANN_OMNI_SAINT	2013-05-08 -	160	160	160	0	160	0	122	N	N	N	N	N	N	N	N

NI		ENTAL	20130508.dat	2013-05-08							326								
OMNI	VISIT	INCREMENTAL	ANN_OMNI_VISIT_20130508.dat	2013-05-08 - 2013-05-08	235	235	235	0	235	0	122323	N	N	N	N	N	N	N	N
OMNI	VISITPROD	INCREMENTAL	ANN_OMNI_VISITPROD_20130508.dat	2013-05-08 - 2013-05-08	3877	3877	3877	0	3877	0	122324	N	N	N	N	N	N	N	N
Source	**Feed**	**Source Extract**	**File Name**	**Control Dates**	**Expected Qty**	**File Qty**	**Load Qty**	**Load Reject**	**Stage Qty**	**Stage Reject**	**File ID**	**Audit Feed Missing**	**Audit Control Missing**	**Audit Control Quantity**	**Audit Control Dates**	**Audit Multiple Files**	**Audit Reject Threshold**	**Audit PK Threshold**	**Audit RI Threshold**
RN	INCIDENT	INCREMENTAL	ANN_RN_INCIDENT_20130508.dat	2013-05-08 - 2013-05-08	0	0	0	0			122327	N	N	N	N	N	N	N	N

AGILITY

The AGILITY audits are created to assess key assignment and name/address quality metrics.

annt_dqa_20120504172226.xls

Data Warehouse

The Data Warehouse Audit show the impact of the update on each DW table and a breakout of how each feed impacted each table, Primary and Surrogate keys are validated as well as Referential Integrity assessments.

ANN CRM Data Warehouse Audit
Update Week 2013-05-08
This section shows each DW table change this update cycle

Table Name	Input	Updated	%	Inserted	%	Rolled Off	%	Output	%	Unique PKs	Unique EPS Keys
ACCOUNT	167	160	95.8	0	0.0	0	0%	167	100.0	167	N/A
ACCOUNT_HARD_KEY	1729198	564306	32.6	3327	0.2	0	0%	1732525	100.2	1732525	1732525
POSTAL_CONTACT	332047	0	0.0	757	0.2	0	0%	332804	100.2	332804	332804
ACCOUNT_EMAIL	8436265	6688033	79.3	25851	0.3	0	0%	8462116	100.3	8462116	8462116
CSC_INCIDENT	311	311	100.0	3	1.0	0	0%	314	101.0	314	N/A
EMAIL	13375339	5308951	39.7	394260	2.9	0	0%	13769599	102.9	13769599	13769599
EMAIL_PROMOTION	54294729	907769	1.7	1084744	2.0	0	0%	55379473	102.0	55379473	55379473
EMAIL_RESPONSE	12568232	0	0.0	233598	1.9	0	0%	12801830	101.9	12801830	12801830
IID_KEYS	1413	291	20.6	26	1.8	0	0%	1439	101.8	1439	1439
PRODUCT	16152638	281734	1.7	28660	0.2	0	0%	16181298	100.2	16181298	16181298
REFERRAL	1081594	418751	38.7	1066	0.1	0	0%	1082660	100.1	1082660	1082660
STORE	291298	270811	93.0	845	0.3	0	0%	292143	100.3	292143	N/A
TXN_HEADER	121149839					0	0%	121149839	100.0	95586222	95586222
TXN_PAYMENT	120393795	235852	0.2	756044	0.6	0	0%	121149839	100.6	121149839	121149839
TXN_ITEM	452132622	353137	0.1	2985802	0.7	0	0%	455118424	100.7	455118424	455118424
VISIT	4132637	0	0.0	4080	0.1	0	0%	4136717	100.1	4136717	4136717
VISIT_MERCH_CATEGORY	7917203	6373438	80.5	58267	0.7	0	0%	7975470	100.7	7975470	7975470

This section shows how each feed updated the DW this cycle

Source	Feed	Source Extract	File Name	File ID	Stage Qty	Target Table	DW Input	Reject Qty	Ignore Qty	Dedupe Qty	Insert Qty	Update Qty
ADS	CCXREF	INCREMENTAL	ANN_ADS_CCXREF_20130508.dat	122315	171965	TBD	3610	0	0	0	1558	2052
ADS	CUST	INCREMENTAL	ANN_ADS_CUST_20130508.dat	122315	171965	TBD	1363	0	0	0	1049	314
Source	Feed	Source Extract	File Name	File ID	Stage Qty	Target Table	DW Input	Reject	Ignore Qty	Dedupe	Insert Qty	Update Qty

								Qty		Qty		
ATG	CUST	INCREMENTAL	ANN_ATG_CUST_20130508.dat	122368	291359	TBD	542729	0	0	0	1388	541341
ATG	TXNDTL	INCREMENTAL	ANN_ATG_TXNDTL_20130508.dat	122368	291359	TBD	1085	0	0	5	9	1071
ATG	TXNHDR	INCREMENTAL	ANN_ATG_TXNHDR_20130508.dat	122368	291359	TBD	46158	0	0	0	66	46092
ATG	TXNTNDR	INCREMENTAL	ANN_ATG_TXNTNDR_20130508.dat	122368	291359	TBD	288595 2	0	0	0	7951	287801
Source	Feed	Source Extract	File Name	File ID	Stage Qty	Target Table	DW Input	Reject Qty	Ignore Qty	Dedupe Qty	Insert Qty	Update Qty
AW	TXNCOUP	INCREMENTAL	ANN_CA_WM_CUST_20130508.dat	122353	427869	TBD	20714	0	0	13	40	20661
AW	TXNDTL	INCREMENTAL	ANN_AW_TXNDTL_20130508.dat	122353	427869	TBD	596	0	0	0	2	594
AW	TXNHDR	INCREMENTAL	ANN_AW_TXNHDR_20130508.dat	122353	427869	TBD	70131	0	0	1	79	70051
AW	TNDR	INCREMENTAL	ANN_AW_TXNTNDR_20130508.dat	122353	427869	TBD	402120	0	0	506	955	400659
Source	Feed	Source Extract	File Name	File ID	Stage Qty	Target Table	DW Input	Reject Qty	Ignore Qty	Dedupe Qty	Insert Qty	Update Qty
CM	SUBS	INCREMENTAL	ANN_CM_SUBS_20130508.dat	122379	41241	EMAIL_CONTACT	41240	0	0	200	1103	39937
CM	SUBS	INCREMENTAL	ANN_CM_SUBS_20130508.dat	122379	41241	PID_EMAIL	41240	0	0	0	41230	4
CM	UNSUBS	INCREMENTAL	ANN_CM_UNSUBS_20130508.dat	122378	814717	EMAIL_CONTACT	814716	0	14	407380	0	407322
CM	UNSUBS	INCREMENTAL	ANN_CM_UNSUBS_20130508.dat	122378	814717	PID_EMAIL	814716	0	779449	14	0	35250
CM	IID_KEYS	INCREMENTAL	ANN_CM_IID_KEYS_20130508.dat	122415	1293	IID_KEYS	1292	0	0	975	26	291

CM	**IID_KEYS**	INCREMENTAL	ANN_CM_IID_KEYS_20130508.dat	122415	1293	SOURCE_TAXONOMY	1292	0	0	975	26	291
CM	**DATA_BULK**	INCREMENTAL	ANN_CM_DATA_20130508_bulk.dat	122420	1009802	EMAIL_PROMOTION	1009802	329	0	0	101704	907769
CM	**DATA_CLICK**	INCREMENTAL	ANN_CM_DATA_20130508_click.dat	122419	29751	EMAIL_RESPONSE	29751	40	0	1044	28667	0
CM	**DATA_OPEN**	INCREMENTAL	ANN_CM_DATA_20130508_open.dat	122418	204122	EMAIL_RESPONSE	204122	225	0	0	203897	0
CM	**DATA_UNSUB**	INCREMENTAL	ANN_CM_DATA_20130508_unsub.dat	122417	2062	EMAIL_RESPONSE	2062	8	0	1020	1034	0
CM	**DEMO**	INCREMENTAL	ANN_CM_DEMO_20130508.dat	122416	381385	EMAIL_CONTACT	381385	1	0	3	378833	2548

This section trends out each foreign key relationship in the database (example below)
Key Referential tolerances exception reporting can be customized

Source	Parent	Child	Parent Rows	Orphan Rows	%
ANN	POSTAL_CONTACT	INDIVIDUAL	666069	0	0.00%
ANN	ACCOUNT	INDIVIDUAL	666069	1463	0.22%
ANN	ACCOUNT_HARD_KEY	ACCOUNT	4315730	0	0.00%
ANN	ACCOUNT_EMAIL	ACCOUNT	4315730	92416	2.14%

Data Mart

The Data Mart audit compares each table size to the previous update's mart to flag any table that dropped quantity or grew outside the expected threshold based on historical trends.

Mart tables are profiled on critical fields and compared to the previous mart to identify any anomalies.

ANN CRM Data Mart Audit

Update Week 2013-05-08

This section compares each mart table to that of the previous update any trends outside of the expected threshold need to be explained. it is also typical to perform comparison trending of data field profiles or aggregates vs the prior mart to help insure update accuracy

Table	Current Rows	Previous Rows	Delta	Delta %
BASE_EMAIL_CONTACT	13769599	13375339	394260	2.95
BASE_INDIVIDUAL	14886500	14841344	45156	0.30
LU_CAMPAIGN_TAXONOMY	33	33	0	0.00
SUM_EMAIL	6226593	6197734	28859	0.47
LU_POSTAL_CONTACT	912790	909291	3499	0.38
F_EMAIL_RESPONSE	12799266	12566533	232733	1.85
F_OFFLINE_PROMO_HIST	15026668	15000893	25775	0.17
F_ONLINE_PROMO_HIST	55379473	54294729	1084744	2.00
F_TXN_HDR	92446416	91860385	586031	0.64
F_TXN_ITEM	452642557	449658446	2984111	0.66
F_TXN_PMT	120384690	119629389	755301	0.63
SUM_INDIVIDUAL	1082660	1081594	1066	0.10
LU_PID_EMAIL	8087825	8046595	41230	0.51
LU_PRODUCT	3357272	3343315	13957	0.42
LU_STORE	51101	50938	163	0.32

Period	Current Total	Current Orders	Current %	Previous Total	Previous Orders	Previous %	Delta Total	Delta Orders	Delta %
200804	96925632	2067023	2.1%	96280980	2067023	2.1%	644652	0	0.00%
200805	96925632	2793838	2.9%	96280980	2793838	2.9%	644652	0	0.00%
200806	96925632	2656469	2.7%	96280980	2656469	2.8%	644652	0	0.00%
200807	96925632	2732972	2.8%	96280980	2732972	2.8%	644652	0	0.00%
200808	96925632	3448139	3.6%	96280980	3448139	3.6%	644652	0	0.00%
200809	96925632	2835169	2.9%	96280980	2835169	2.9%	644652	0	0.00%
200810	96925632	2544847	2.6%	96280980	2544847	2.6%	644652	0	0.00%
200811	96925632	2887385	3.0%	96280980	2887385	3.0%	644652	0	0.00%

200812	96925632	2883359	3.0%	96280980	2883359	3.0%	644652	0	0.00%
200901	96925632	2608056	2.7%	96280980	2608056	2.7%	644652	0	0.00%
200902	96925632	3139810	3.2%	96280980	3139810	3.3%	644652	0	0.00%
200903	96925632	2396718	2.5%	96280980	2396718	2.5%	644652	0	0.00%
200904	96925632	2314226	2.4%	96280980	2314226	2.4%	644652	0	0.00%
200905	96925632	2862154	3.0%	96280980	2862154	3.0%	644652	0	0.00%
200906	96925632	2373979	2.4%	96280980	2373978	2.5%	644652	1	0.00%
200907	96925632	2414391	2.5%	96280980	2414391	2.5%	644652	0	0.00%
200908	96925632	3078614	3.2%	96280980	3078614	3.2%	644652	0	0.00%
200909	96925632	2533026	2.6%	96280980	2533026	2.6%	644652	0	0.00%
200910	96925632	2367155	2.4%	96280980	2367155	2.5%	644652	0	0.00%
200911	96925632	2753239	2.8%	96280980	2753239	2.9%	644652	0	0.00%
200912	96925632	2824268	2.9%	96280980	2824268	2.9%	644652	0	0.00%
201001	96925632	2531694	2.6%	96280980	2531694	2.6%	644652	0	0.00%
201002	96925632	3134607	3.2%	96280980	3134607	3.3%	644652	0	0.00%
201003	96925632	2450862	2.5%	96280980	2450862	2.5%	644652	0	0.00%
201004	96925632	2412877	2.5%	96280980	2412877	2.5%	644652	0	0.00%
201005	96925632	2824308	2.9%	96280980	2824308	2.9%	644652	0	0.00%
201006	96925632	2253577	2.3%	96280980	2253577	2.3%	644652	0	0.00%
201007	96925632	2446053	2.5%	96280980	2446053	2.5%	644652	0	0.00%
201008	96925632	3073707	3.2%	96280980	3073707	3.2%	644652	0	0.00%
201009	96925632	2568237	2.6%	96280980	2568236	2.7%	644652	1	0.00%
201010	96925632	2356644	2.4%	96280980	2356641	2.4%	644652	3	0.00%
201011	96925632	2805270	2.9%	96280980	2805269	2.9%	644652	1	0.00%
201012	96925632	2815006	2.9%	96280980	2815006	2.9%	644652	0	0.00%
201101	96925632	2551647	2.6%	96280980	2551645	2.7%	644652	2	0.00%
201102	96925632	3155069	3.3%	96280980	3155051	3.3%	644652	18	0.00%
201103	96925632	2431147	2.5%	96280980	2386611	2.5%	644652	44536	1.87%

201104	96925632	600090	0.6%	0	0	0.0%	96925632	600090	0.00%
Status Code	**Current Total**	**Current accounts**	**Current %**	**Previous Total**	**Previous accounts**	**Previous %**	**Delta Total**	**Delta accounts**	**Delta %**
ACTIVE	1082660	908472	83.9%	1081594	907547	83.9%	1066	925	0.10%
AT RISK	1082660	25556	2.4%	1081594	25562	2.4%	1066	-6	-0.02%
INACTIVE	1082660	148449	13.7%	1081594	148302	13.7%	1066	147	0.10%
PROSPECT	1082660	62	0.0%	1081594	62	0.0%	1066	0	0.00%
TPLT	1082660	121	0.0%	1081594	121	0.0%	1066	0	0.00%

Other Audits

- Verify that outbound Subscribers feeds (POSSUBS, ADSSUBS) which are supposed to be auto-subbed are returned by CM in the SUBS feed on the subsequent update. The Email Hygiene subs feed requires authorization by the email recipient before they may be subscribed.

Appendix I – Future Enhancements

ID	Enhancement
FE1	ACCOUNT_EVENT table for storing additional information about ad-hoc loaded accounts
FE2	International address processing
FE3	Third-part phone type identification
FE4	Email Campaign deployment directly from CRM solution o Promotion History Freeze Field
FE5	Email Model Scoring
FE6	Campaign Attribution Engine implementation
FE7	Process will be developed to feed the actual mailed records back into direct mail promotion history table, post data processing and prospect data.
FE8	Linking of CheetahMail email promotion data with Omniture web behavior data. This will happen at a later point and will be the responsibility of CheetahMail.
FE9	Promotion of EMAIL_OPTOUT_HIST data to the mart
FE10	Include aggregate EMAIL_SUBSCRIBER_IND on SUM_INDIVIDUAL and SUM_HOUSEHOLD for use in conjuction with best email_address on multi-channel campaigns
FE11	Automated process for the reactivation of Email Nuclear Opt-Out (stoplisted) email addresses
FE12	Automated calculation of store proximity based on postal code only
FE13	Email Campaign promotion history including control group data
FE14	Direct attribution from Email Response data to transactions
FE15	BazaarVoice in-store product review data

Appendix J – CheetahMail/SDL Outbound Extracts

Email Profile

This feed provides current physical address and transaction rollup data as it relates to an email address. The trigger for extracting this information is a change to any element on the profile feed. This is used as a supplemental email demographic data source within CheetahMail

Email Profile Driver (Extract Triggers)

The email address is the primary key of the CUSTPROFILE extract provided to CheetahMail to update the demographic profile data they maintain. The first step of the process is to identify which email addresses to include. This is triggered based on Data Warehouse row last modified dates on the following tables:

Trigger 1: Transaction Occurs

- **Direct Email Relationship** – the email address is provided directly on the transaction through either an e-commerce order or POS email capture
- **Indirect Email Relationship** – one or more email addresses linked to an individual through their underlying account relationships in the database.

For example, if Mark Beening places an order online we'll establish a DIRECT email relationship on the ATG transaction, We will also perform a lookup on Mark Beening in the database and identify all of his email addresses (whether they were provided through other ATG orders, CheetahMail Subscription, BazaarVoice review, Marketworks history, etc.)

Trigger 2: Update to Account / Postal Contact

Identify any individuals linked to email address who's underlying Account or Postal Contact data has been updated or newly created within the latest cycle

The union of all of these email address becomes the driver of the extraction process. These email address are used to build/refresh the customer profile for CheetahMail. This includes a restatement or update to name/address elements as well as a refresh of transaction aggregates, regardless of the event that identified the email address for inclusion in the extract on any particular cycle.

Customer Attributes

For the purposes of the Profile extract data can be sourced from one of two place in the CRM database

- Epsilon – POSTAL_CONTACT, ACCOUNT (as provided by multiple sources through the various account source feeds)
- CheetahMail - EMAIL_PROFILE (as provided from SDL through the DEMO and SUBS feeds)

As illustrated in the table below, the rule is to check one source first and where the criteria is not available, to "fail over" to the secondary source

Extract Field	Primary	Secondary	Notes
ADDRESS	EPSILON	CM	Address elements use POSTAL_CD to identify source viability
ADDRESS_2	EPSILON	CM	POSTAL_CD
CITY	EPSILON	CM	POSTAL_CD
STATE	EPSILON	CM	POSTAL_CD
ZIP	EPSILON	CM	

COUNTRY	EPSILON	CM	POSTAL_CD
FNAME	CM	EPSILON	Name elements use FNAME || LNAME to identify source viability
LNAME	CM	EPSILON	See FNAME note
PHONE	EPSILON	n/a	Always provide Epsilon's most recently loaded HOME phone number
BIRTHDAY_3RD_PARTY	CM	n/a	
BIRTHDAY_PREF	CM	EPSILON	
CUSTOMER NUMBER	EPSILON	n/a	

Fundamentally this means that:

1. Whenever available, use Epsilon as the source for physical address. The CheetahMail "fail over" process really just provides back to CM what was already provided to Epsilon without any changes. This "fail over" is put in place to insure that Epsilon does not overwrite CheetahMail addresses data with NULL in the event that CheetahMail had an address and Epsilon did not

2. Whenever available, use CheetahMail as the source for self-reported Name and Birthday data. This is likely to be the most trustworthy source. Epsilon data is only applied (if available) when CheetahMail does not already have these attributes.

3. The BIRTHDAY_3RD_PARTY data is not available within the CRM solution (may eventually come from KBM/Experian/or some other demographic append process). It will not be populated on the extract from Epsilon. This may result in the overwrite of NULL in the update from Epsilon to CheetahMail if CM did have Birthday 3rd party data attributed to an email address.

Transaction/Promotion Attributes

These attributes are rollups of transaction data. Effectively these are email-level rollups that get applied to all email addresses associated to the individual making transactions.

- An email address shared by multiple individuals accounts for transactions made by all individuals such that if Person A and Person B shared an email address and Person A's last transaction date was 2010 and Person B's last transaction date was 2011, the email addresses last purchase date would be 2011.

- An individual with multiple email addresses refers to the same underlying transactions for all email addresses (example below)

Consider the following examples:

Example 1 – mbeening@epsilon.com is provided on an ATG transaction occurring on 6/15/2012. The last transaction date for mbeening@epsilon.com is 6/15/2012

Example 2 – Three months later, Mark Beening walks into an Ann Taylor retail store and makes a purchase on 9/15/2012. He DOES NOT provide his email address to the cashier.

We associate Mark Beening to the transaction through the credit card swipe.

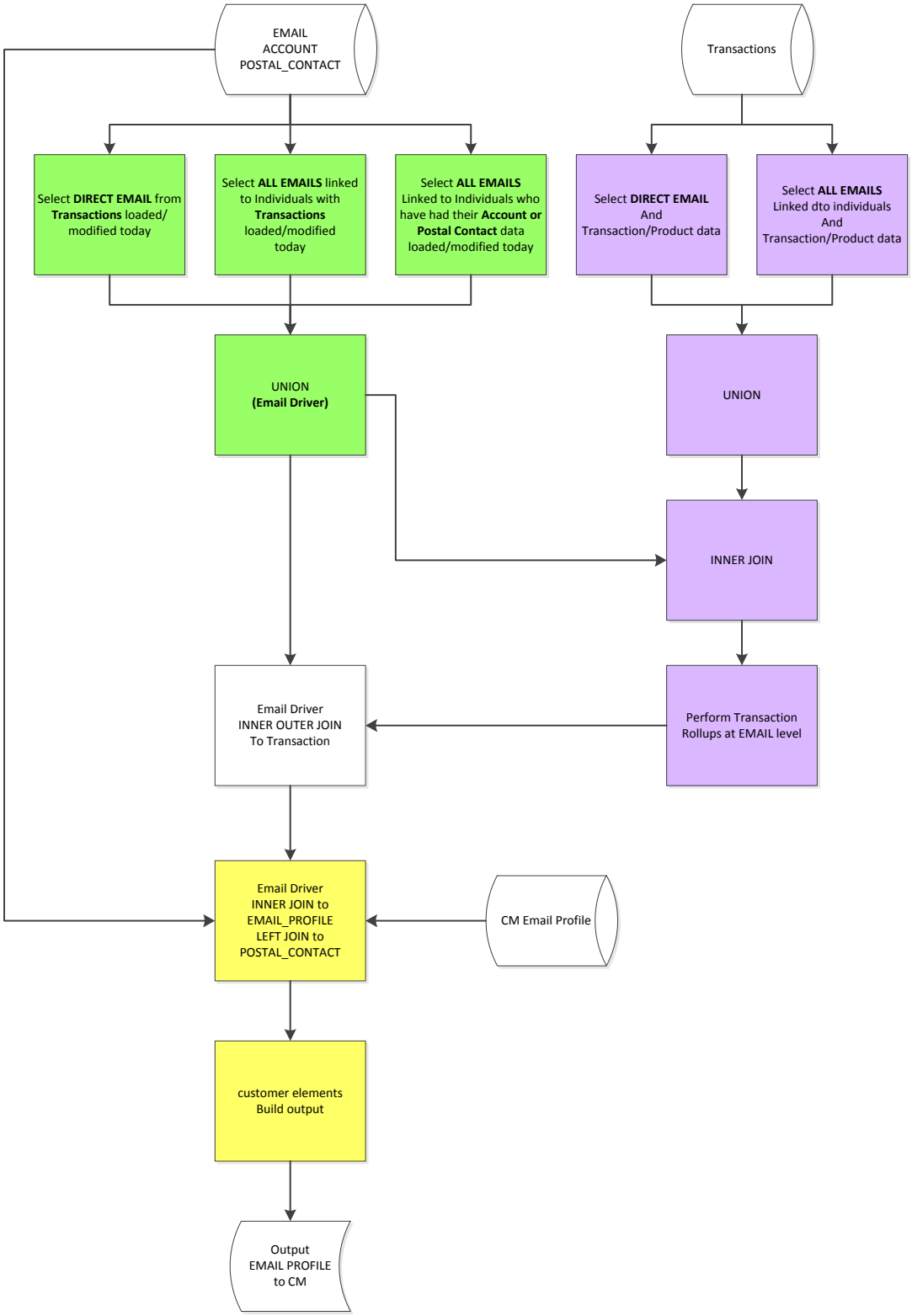
We then discover that Mark Beening is linked to three email addresses.
1) mbeening@epsilon.com was provided on a 6/15 ATG order (see Example 1)
2) mbeening@hotmail.com was provided on an old BazaarVoice product review
3) mbeening@gmail.com was provided on an SDL Demographic feed

The **last transaction date rollups** for the Epsilon.com, Hotmail.com, and gmail.com email addresses are provided to CheetahMail as 9/15 on all 3 email profile rows

Extract Field	
ATTR_MATERNITY_PUR_DATE	Last TXN_DT
ATTR_PETITE_AT_PUR_DATE	Last TXN_DT
ATTR_PETITE_LOFT_PUR_DATE	Last TXN_DT
ATTR_TALL_AT_PUR_DATE	Last TXN_DT
ATTR_TALL_LOFT_PUR_DATE	Last TXN_DT
ATTR_WEDDING_PUR_DATE	Last TXN_DT
STORE_NO_ATS	Last STORE_NBR
STORE_NO_ATS_O	Last STORE_NBR
STORE_NO_FAC	Last STORE_NBR
STORE_NO_LOFT	Last STORE_NBR
STORE_NO_LOFT_O	Last STORE_NBR
STORE_NO_OUTLET	Last STORE_NBR
TRANS_AMOUNT_ATS	Sum(TXN_AMT)
TRANS_AMOUNT_LOFT	Sum(TXN_AMT)
TRANS_DATE_ATS	Last TXN_DT
TRANS_DATE_ATS_O	Last TXN_DT
TRANS_DATE_FAC	Last TXN_DT
TRANS_DATE_LOFT	Last TXN_DT
TRANS_DATE_LOFT_O	Last TXN_DT
TRANS_DATE_LOS	Last TXN_DT
TRANS_ID_ATS	B-O-S Segment
TRANS_ID_LOFT	B-O-S Segment

Email Profile Extract Process Flow

The flowchart below describes the process for extracting EMAIL PROFILE data for load into the CheetahMail system



Transaction Feeds

The transaction feeds provide SDL with each transaction (header) loaded into the CRM solution with a relationship to an email address. Only the initial instance of a transaction is provided though the CRM solution does provide the ability to update/modify transactions after initial load.

Seq	Field Name	PK	NULL	Data Type	Length	Description
1	EMAIL_ADDR			String	256	Email Address
2	EPS_CUST_NO				20	Epsilon Customer Number
3	TXN_ID			Numeric	10	Transaction ID
4	TXN_DT			Date		Transaction Date (YYYY/MM/DD)
5	TXN_STORE_NO				5	Transaction Store Number
6	TXN_AMT			Decimal2	22	Transaction Amount

The email address is assigned to a transaction through either 1) direct relationship through POS email capture or providing an email address on an e-commerce order or 2) the BEST email address associated to an individual at the time the transaction was loaded into the CRM database.

The logic for selecting the BEST email address is found in the BRD as follows:

BU37.7	When consolidating multiple email addresses, the following rules apply to select the "best" email address: 5. First choose the valid email address over a non-valid email address (valid means non-chronic bouncer, not opted-out, and not invalid format). 6. Then choose the email address that has the most recent purchase activity. 7. Then choose the email address that has the most recent non-purchase activity (positive and Client initiated, such as opens or clicks). 8. Then choose sequentially based on email identifier stored in the table.

While relatively unlikely to occur, this means that the email selected for the same individual may shift over time.

Email Transaction Extract Process Flow

The flowchart below describes the process for extracting transaction data for load into the SDL system.