Virgin Trains West Coast – Single Customer View (SCV)

Matching, Merging and Cleansing

Commercial in confidence

Author:

Russell Jackson – Solution Architect

Confidential

## Table of Contents

[Table of Contents 2](#_Toc519758470)

[Document Management 3](#_Toc519758471)

[1. Document Purpose 4](#_Toc519758472)

[2. Document Overview 4](#_Toc519758473)

[2.1. Solution-Level Objectives 4](#_Toc519758474)

[3. Matching 5](#_Toc519758475)

[3.1. Overview 5](#_Toc519758476)

[3.2. Match Keys 5](#_Toc519758477)

[3.2.1. Overview 5](#_Toc519758478)

[3.2.2. Match Keys 5](#_Toc519758479)

[3.3. Functional Specifications – Processing Steps 6](#_Toc519758480)

[3.3.1. Customer 6](#_Toc519758481)

[3.3.2. Prospect 10](#_Toc519758482)

[3.4. Functional Specifications – Database Object 11](#_Toc519758483)

[3.5. Constraints, considerations and dependencies 14](#_Toc519758484)

[4. Merging 15](#_Toc519758485)

[4.1. Overview 15](#_Toc519758486)

[4.2. Prioritisation 16](#_Toc519758487)

[5. Cleasning/Standardisation 18](#_Toc519758488)

[5.1. Overview 18](#_Toc519758489)

[5.2. Functional Overview 19](#_Toc519758490)

## **Document** **Management**

|  |  |  |  |
| --- | --- | --- | --- |
| **Version Control** | | |  |
| **Version** | **Date** | **Author(s)** | **Section Changed** |
| 0.1 | 13/07/2018 | Russell Jackson | Initial Draft |
|  |  |  |  |

|  |  |  |
| --- | --- | --- |
| **Review** | | |
| **Version** | **Date** | **Review By** |
|  |  |  |
|  |  |  |

|  |  |  |
| --- | --- | --- |
| **Supporting Documents** | | |
| **Document Name** | **Document Purpose** | **Document Link** |
| Merkle VTWC Technical Design | Technical Specification | [Click here](https://teams.microsoft.com/_#/vsd/viewer/teams/https%3A~2F~2Fmerkleinc.sharepoint.com~2Fsites~2FVirginTrainsWestCoast-CRMProject~2FShared%20Documents~2FGeneral~2FDevelopment%20Documentation~2FDesign%20Documentation~2FTechnical%20Overview~2FVTWC%20-Process%20Overview%20v0.1.vsd?thread) |
|  |  |  |

|  |  |  |
| --- | --- | --- |
| **Glossary of Terms** | | |
| **Term Name** | **Description** | **Acronym, Synonyms** |
| SCV | Single Customer View | SCV |
| Customer | A person who has registered with VTWC resulting in a record being created in Tracs. This record does not require the person to have made a purchase |  |
| Prospect | A person who has interacted with a VTWC system or is a prospect who VTWC wishes to communicate with. Prospects are linked to customers but held separately in the Single Customer View (SCV) |  |
| Guest Customer | A guest customer is a customer and is treated the same way, other than a flag indicating that the guest customer has been captured as a guest | Customer |

## Document Purpose

The objective of the Matching, Merging and Cleansing document is to provide detailed technical information on the following technical deliverables;

* Matching – How inbound information matches existing data within the SCV database
* Merging – Once a match has been identified, how should the inbound information enhance the existing information
* Cleansing – to improve the quality of information within the database as well as improve the matching success rate, a series of data cleansing rules will be applied to the inbound data

## Document Overview

The Matching, Merging and Cleansing document serves as a supporting document for the master [Merkle VTWC Technical Design](http://ct-sharepoint/PMO/Projects%20Folder/Forms/AllItems.aspx?RootFolder=%2FPMO%2FProjects%20Folder%2FElsevier%2FMDP%20Project%2FCommercials%2FMDP%20Phase%202%20%2D%20Implementation&FolderCTID=0x012000171F192B898D724B9534917D702E9A8E&View=%7B434C34D1%2D3231%2D4BDD%2D838D%2D8EA2ACF05835%7D) documentation.

Each section of the document contains:

* a high-level description to inform the audience of a specific objective and to align on terminology;
* a functional description on how the component part will operate and interact with other dependencies;
* Consideration, constraints and dependencies sub-section. This section should be interpreted as follows:
  + a consideration is something to note and should be considered as part of the configuration of the solution;
  + a constraint is an issue which is constraining the MDP in its own design;
  + a dependency is an action which must be completed or resolved for the successful delivery of the MDP.

### Solution-Level Objectives

The objective of the SCV is to become the basis of the VTWC CRM solution, replacing the current Trainline CRM solution. The solution will deliver a SCV of all VTWC customers and prospects. To achieve this objective information will be loaded into a database on regular intervals, the data will be cleansed to ensure any erroneous characters are removed, matched to existing records within the database and merge inbound information with existing information to provide a complete view of a VWTC individual (customer or prospect).

## Matching

### Overview

Inbound information is matched to existing information leveraging a specific VTWC SCV solution match key.

Information is received and once uploaded into the staging location, the match key is generated using specific pre-defined rules. The pre-defined rules were agreed through the development and implementation of the Virgin Trains East Coast solution, which the VTWC solution is based off.

Once the match key has been created, the application of the match key is via **exact matching** (like-for-like) and therefore no fuzzy logic is applied.

### Match Keys

### Overview

To achieve a high match rate, there will be several attempts at matching information using a number of match keys.

The purpose for leveraging multiple match keys is:

1. In some cases, information may have been matched in an existing solution within VTWC data landscape. In this instance a system ID will be provided which can be used to match inbound information with existing information
2. Not in all instance will information be provided fully populated manner. It is common for records to have missing attributes such as missing email address or missing address lines, therefore the matching logic most accommodate this scenario

### Match Keys

Matching will be performed in the following order, therefore if a match has not occurred using email address then the system will attempt to match on email address (and so on).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Priority No.** | **Type** | **Key/Field** | **Example** | **Source** |
| 001 | Source System Identifier | Source\_ID | VTWC0011123 | Trainline Only |
| 002 | Email | Email\_Address | [r.jackson@merkleinc.com](mailto:r.jackson@merkleinc.com) | All sources |
| 003 | Mobile Phone number | Mobile\_Phone | 079162747474 | All sources |
| 004 | Name and (partial) Address | Initial+Surname+Postcode+AddressLine1  ‘R’, ‘Jackson’,’ SM1 2BX’, ’10 Smith Street’ | RJacksonSM12BX10SmithStreet  \*All spaces are removed for consistency | All sources |

### Functional Specifications – Processing Steps

The following information outlines how the ingestion process manages the different scenarios when handling different individual types (Customer/Prospects).

### Customer

Functionals steps for customer records only

Once matched in prioirty order, record processing is completed.

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Individual Type** | **Senario** | **Steps** |
| 001 | Customer | Inbound Customer Information | The data will match to the STG\_ElectronicAddress table and apply the appropriate match rule (prioirty outlined in the match ket [section](#_Match_Keys)). If the process unsuccessfully identifies a match with the first key, the process will attempt to match on the second key, an so on.  Each feed will check the processing tables to validate what rule to apply, based on the pre-processing tables having the following columns populated:   * TCSCusomerID - If the TCSCustomerID field is populated, then the database validates on STG\_ElectronicAddress if we have a match against an existing TCSCustomerID, to see if we have seen this customer from the booking engine before; * Email - If we have an email address the process checks on STG\_ElectronicAddress to see if we have a match against AddressType 3 records (Email); * Mobile - If the record has a mobile number then the process checks on STG\_ElectronicAddress table to see if we have a match against AddressType 4 records (Mobile); * Name and Address - If the inbound record has a valid name and address matchkey, then the process validates STG\_ElectronicAddress to identify if we have a match against AddressType. |
| 002 | Inbound Known Customer Record | For each incoming customer record, where   * an incoming database customer record has matched an existing customer record on SCV CustomerID, we will then load all relating transactional data using the matched CustomerID.Email, Mobile number or Name and address key are different, the process will add this information to the STG\_ElectronicAddress table (The ElectronicAddress table provides an audit history of all identifiers for that person).   For email and mobile the process will set the current PrimaryInd = 0 and add a new row for the incoming email or mobile and make this the primary address for that type.  The process then updates a Subscription preferences against that SCV CustomerID.  If the existing customer is set to IsPersonInd = 1 (Guest customer) then this will be set to IsPersonInd = 0 in STG\_Customer allowing the previous guest record to be passed through to production.customer table  (The primaryInd indicates the primary identifier at person and channel level) |
| 003 | Customer Matches a Prospect Record | Where an inbound customer has matched to an existing prospect, the process will allocate a new CustomerId and will then load all relating transactional data.  The process will then add a row to the STG\_Keymapping table for this customer and update the prospect record with the customerid and the individualID, creating a link between customers and prospects.  Where there is a different Email, Mobile number or Name and address key, we will add this to the STG\_ElectronicAddress table.  For email and mobile we will set the current PrimaryInd = 0 and add a new row for the incoming email or mobile and make this the primary address for that type. |
| 004 | New Customer | Where an incoming customer is a new record, the process will allocate a new customerid and load all relating transactional data using the allocated CustomerID.  The process will then add a row to the STG\_Keymapping.  Where we have an Email, Mobile number or Name and address key, we will add this to the STG\_ElectronicAddress table. For email and mobile we will set this to have a PrimaryInd = 1 for that address type. |
| 005 | Guest Customer | Known Customer | (treated the exact same way as a known customer to customer)  The database will match to the STG\_ElectronicAddress table and apply the appropriate match rule.  These for each feed will check the processing tables to validate what rule to apply, based on the preprocessing table having columns:   * TCSCusomerID - If the TCSCustomerID field is populated, then the database validates on STG\_ElectronicAddress if we have a match against an existing TCSCustomerID, to see if we have seen this customer from the booking engine before; * Email - If we have an email address the process checks on STG\_ElectronicAddress to see if we have a match against AddressType 3 records (Email); * Mobile - If the record has a mobile number then the process checks on STG\_ElectronicAddress table to see if we have a match against AddressType 4 records (Mobile); * Name and Address - If the inbound record has a valid name and address matchkey, then the process validates STG\_ElectronicAddress to identify if we have a match against AddressType. |
| 006 | New Guest | Where the inbound records contain a new customer, the process will allocate a new customerID and load the associated data.  A record will be added to STG\_Keymapping and add the new customerID, creating a placeholder for mapping between prospect and customer.  We will then add a row to the STG\_ElectronicAddress table for this source and make this the primaryind = 1.  As this is a new customer record but we don’t have personal details provided then the process will flag this on the STG\_Customer table as IsPersonInd = 0, this will stop the customer being processed into the production.customer table and being made available for campaigning or profiling.  If the customer then registers or matches via another account then we will update the IsPersonInd = 1 |
| 007 | Guest matches a Prospect | Where there is an email address, this will go through the existing email matching process.  Where it matches to an existing prospect then the process will allocate a new CustomerID and load the associated data.  The process will then create an entry in the STG\_Keymapping table and add the new customerID and the existing IndividualId from the prospect record, creating a mapping between prospect and customer  The process will then add a row to the STG\_ElectronicAddress table for this source and make this the primaryind = 1 , then make the previous record for this address type to primaryInd = 0.  As this is a new customer record but we don’t have personal details provided yet then we will flag this on the STG\_Customer table as IsPersonInd = 0, this will stop the customer being processed into the production.customer table and being made available for campaigning or profiling. If the customer then registers or matches via another account then we will update the IsPersonInd = 1 |

### Prospect

Functionals steps for prospect records only

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Individual Type** | **Task** | **Steps** |
| 001 | Prospect | Inbound Prospect Records | The database will match to the STG\_ElectronicAddress table and apply the appropriate match rule.  For each feed will check the processing tables to validate what rule to apply, based on the pre-processing table having columns:   * TCSCusomerID - If the TCSCustomerID field is populated, then the database validates on STG\_ElectronicAddress if we have a match against an existing TCSCustomerID, to see if we have seen this customer from the booking engine before; * Email - If we have an email address the process checks on STG\_ElectronicAddress to see if we have a match against AddressType 3 records (Email); * Mobile - If the record has a mobile number then the process checks on STG\_ElectronicAddress table to see if we have a match against AddressType 4 records (Mobile); * Name and Address - If the inbound record has a valid name and address matchkey, then the process validates STG\_ElectronicAddress to identify if we have a match against AddressType. |
| 002 | Prospect Matches a Customer | Where an incoming prospect record has matched an existing customer record we will load this data with the matched customerID.  If this is also a new prospect record the process will allocate a new individualid and update the keymapping table with the new individualid, creating a link between customer and individual.  If this matched a prospect only record then the data will be loaded using the returned individualID  A customer record cannot become a prospect. |
| 003 | Unknown Prospect (unmatched) | Where a prospect record has not matched a record in the STG\_ElectronicAddress table then a new IndividualID will be allocated and the data loaded.  An entry will be added to STG\_KeyMapping. |

### Functional Specifications – Database Object

The following section outlines the objects within the database that are utilised to fulfil this objective.

|  |  |  |
| --- | --- | --- |
| **No.** | **Object** | **Description** |
| 001 | STG\_Customer | Holds Customer Data |
| 002 | STG\_Individual | Holds Prospect Data |
| 003 | STG\_KeyMapping | Holds relationships of CRM Prospect, CRM Customer and Customer Source Keys  (TCSCustomerID)  STG\_Keymapping will be the main relationship table between a customer and a prospect.  This table tracks ID’s where the record is a customer or a prospect, it also allows us to store multiple source customer id’s i.e TCSCustomerID, this is used as a match key to identify whether we have seen this customer before. If VTWC migrated the booking engine we would add a new source key to this table to allow us to link across old and new booking engines and allow us to track migrated customers or prospects. |
| 004 | Gen\_key | [insert additional rules |
| 004 | STG\_ElectronicAddress | all variations of AddressTypes for a CRM Customer or CRM prospect  This table holds all variations of a customer/Prospect for each source file and address type. Currently it tracks all email addresses, mobile phone numbers and Beam VisitorID’s . This table also holds a PrimaryInd that points to the latest address type that is valid for use. This is based on which email or mobile phone number we have seen most recently for this customer or prospect.  Depending on the source of data, the process can choose to allow this source to become the primary address for each type or just store the new email or mobile but not override the primary but still allow is to bring different sources together. For each source that we receive that has email, mobile or address we will document in the data feed section in this document on the level of interaction this source can have on the overall SCV results and the setting of the address type for each marketing channel.  This table holds one row per address type and a column called AddressType indicates what channel the Address column relates to; this table will also hold an encrypted version of the address for each type. Current address types that are supported are:   * Email * Mobile * Beam Visitor ID   We will address a further address type to the STG\_ElectronicAddress table support name and address matching   * Namad   The Electronic address table will hold valid and invalid email addresses and will be identifiable by the ParsedInd column. The table holds an Address column and a ParsedAddress column, only he ParsedAddress column will be used for matching. |
| 005 | CRM.Reference.AddressType | Reference table holding the different types of addresses |
| 006 | ElectronicAddress | A core element of the VTWC solution is the ElectronicAddress table, this holds the emails, mobile numbers and various source keys for matching, tracking and bringing together both customers and prospect data.  This table, along with the KeyMapping table, allow us to bring together various data sources of both customer and prospect data without the need to merge the data into the same staging tables. This helps to meet the requirements of GDPR and separate 3rd party data while still giving the ability to look across differing sources.  The process is developed in a way that allows us to flexibly add in new keys, i.e account numbers, Namad (Name and household address key), NI numbers for VTWC as appropriate.  Due to the way data is captured on VTWC the data is deemed to be ‘clean’ and requires limited cleansing (documented above), and hence matching, the single Namad key detailed above is being used to bring together people at the same postal address. The ElectronicAddress table would hold all variations of a person’s household address where they have moved. |

### Constraints, considerations and dependencies

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Type** | **Description** | **DB Object** |
| 001 | Consideration | If a name and address matchkey is deemed too weak to use for matching then the key will be deleted and no attempt to match this record against existing data or add this key the STG\_ElectronicAddress table  The Reasons for a weak matchkey are;   * Address line 1 is empty * First name is empty * Last name is empty * Final key is less than or equal to 15 characters in length | Staging.rejectReason |
| 002 | Consideration | If the first name is not populated then the process will attempt to populate it from the middle name or surname.  (This function is configurable to only use the first name and the benefits of extending this to middle name and surname will be evaluated as part of the SCV testing) |  |
| 003 | Consideration | The Electronicaddress table holds all channel varidations for all individauls, and can be easily queried to identify how many records were matched, and how many new records were created on a given day for a given source. This level of auditing will aid | Staging.stg\_electronicaddress |

## Merging

### Overview

Whilet merging records for each overlapping attribute, a decision is required on whether to retain the original value or choose to overwrite the values with incoming source information.

This is to ensure a complete and consistent view of all information within the SCV, without impacting the quality of the data by leveraging untrusted source information to overwrite existing data.

The preference for VTWC is to keep all values where possible. In cases where a survivor needs to be selected a combination of source precedence and recency is used. The following provide the priority and recency rules which will be applied;

1. If the conflicting values appear in different groups, the group with the highest precedence survives (see below for survivorship rule)
2. If there are conflicts within a group, the most recent value is chosen

For attributes where all values are kept, but a primary value needs to be identified, the same rules as above apply.

### Prioritisation

In all cases where recency is required, the following field will be referenced; SourceChangeDate.

Reference;

|  |  |
| --- | --- |
| **Priority No.** | **Description** |
| -1 | Those sources that do not contain information to update customer or prospect information |
| 0 | Those sources that contain information to update customer or prospect information, but it not utilised |
| 1 or above | 1 being the lowest priority |

The following table outlines how the process managed inbound feed for customer and prospect contact information only according to priority.

|  |  |  |
| --- | --- | --- |
| **Priority No.** | **Source** | **Notes** |
| -1 | ADR | Source contains only transaction information |
| -1 | Evolvi | All data is loaded into separate table and therefore does not overwrite existing contact information.  (email is used to match back) |
| 0 | Nexus Alpha Journey Check Alerts | This source will create a prospect record but not a customer, and applies updates at prospect level but need to confirm whether we should use this information to overwrite existing information (trusted?) |
| 1 | ITG Creator (Push and Pull) | * Push file contains little information other than email. We do not do anything with that email other than match to existing record. * Pull files contains name, email and optin. The process creates individual customer records and updates optin preference information (as this is a trusted source) |
| -1 | Adobe | Inbound campaign information, therefore does not contain contact information for merging |
| 0 | Beam | Firstname, surname and optin.  The process creates only a prospect record.  Where we match to a customer the process updates the optin information. |
| -1 | Wifi (Onboard and On station) | Matches only update attribute information and not individuals contact details, other than marketing preference information. |
| 2 | ToC (Train Operating Customer) | Master feed takes priority and is the only source that contains customer information |

## Cleasning/Standardisation

### Overview

To achieve a higher match rate and improve data quality, a series of cleansing steps are performed on all inbound data.

This section will detail the steps that are taken to improve data quality as well as the relevenat database objects.

### Functional Overview

|  |  |  |
| --- | --- | --- |
| **No.** | **Type** | **Description** |
| 001 | Email Address | Email address will be parsed into the following component:   * User name; * Domain; * Top level domain.   The format of the email will be validated and where it is deemed a correctly formatted email it will be given a parsed score of 100 and a parsedInd of 1, else it will be given a parsed score of 0 and a parsedInd of 0.  This will apply to all sources of incoming data which could contain email addresses. |
| 002 | Mobile Phone Numbers | Mobile phone numbers will be validated for length and format, where they meet the format then they will be given a parsed score of 100 and a parsedInd of 1  Where the mobile number is deemed to be a US number then the parsed score will be set to 80 |
| 003 | Profanity Checks | All emails, forename, middle names and Last names will be check for profanity words and flagged appropriately  Once this processing is complete, the pre-processing tables will be updated with the following details:   * + ParsedEmail, ParsedInd and ParsedScore   + ParsedMobile, ParsedInd and ParsedScore   + Profanity Flag   The development of the algorithms to process the incoming data will be an iterative process using actual data. As the data is processed, additional information may be added to support further processing – the integrity of the original source information will be maintained within CEM. |
| 004 | Remove Spurious Characters | The following stored procedure will remove non alphanumeric (a-z/0-9)characters from all fields;  Staging.RemoveNonAlphaCharacter |
| 005 | Remove Duplicate Characters | The following stored procedure will remove duplicated characters within the name or address elements. Such as misspelt address lines or surnames i.e ‘Garry’ v ‘Gary’ , ‘Skinner’ v ‘Skiner’. This will not change a field that has duplicated characters that are together, so ‘Garary’ will still return ‘Garary’;  Staging.removeDuupCharsv2 |
| 006 | Populate Initial Field | The following stored procedure will return the first initial of the first name, if the first name is not populated then it will attempt to populate this from the middle name or surname.  Staging.getFirstInitial  (This function is configurable to only use the first name and the benefits of extending this to middle name and surname will be evaluated as part of the SCV testing) |