Virgin Trains West Coast – Single Customer View (SCV)

Journey, Journey Leg and Leg ToC+ VTWC

Commercial in confidence

Author:

Russell Jackson – Solution Architect

Confidential

## Table of Contents

[Table of Contents 2](#_Toc521933587)

[Document Management 3](#_Toc521933588)

[1. Document Purpose 4](#_Toc521933589)

[2. Overview 4](#_Toc521933590)

[3. Process 4](#_Toc521933591)

[3.1. High Level Steps – Journey 4](#_Toc521933592)

[3.2. Functional Steps - Journey 5](#_Toc521933593)

[3.3. Table Mappings and Business Rules - Journey 6](#_Toc521933594)

[3.4. High Level Steps – Journey Legs 12](#_Toc521933595)

[3.5. Functional Steps – Journey Legs 12](#_Toc521933596)

[3.6. Table Mappings and Business Rules – Journey Legs 14](#_Toc521933597)

[3.7. High Level Steps – (Rail) Legs 17](#_Toc521933598)

[3.8. Functional Steps – Journey Legs 18](#_Toc521933599)

[3.9. Table Mappings and Business Rules - Legs 21](#_Toc521933600)

## **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) |
| Address Matching, Merging and Cleansing - VTWCv2 | Address matching, merging and cleasning specification | [Click here](https://teams.microsoft.com/_#/docx/viewer/teams/https%3A~2F~2Fmerkleinc.sharepoint.com~2Fsites~2FVirginTrainsWestCoast-CRMProject~2FShared%20Documents~2FGeneral~2FDevelopment%20Documentation~2FDesign%20Documentation~2FTechnical%20Overview~2FAddress%20Matching%2C%20Merging%20and%20Cle) |
| Customer ToC+ VTWC | Functional information on handling customer and prospect information | [Click here](https://teams.microsoft.com/_#/docx/viewer/teams/https%3A~2F~2Fmerkleinc.sharepoint.com~2Fsites~2FVirginTrainsWestCoast-CRMProject~2FShared%20Documents~2FGeneral~2FDevelopment%20Documentation~2FDesign%20Documentation~2FTechnical%20Overview~2FCustomer%20ToC%2B%20VTWC.docx?threadId=19%3) |
|  |  |  |

|  |  |  |
| --- | --- | --- |
| **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 |
| TVM | Records or information that has been captured using a Train Vending Machine (TVM). | TVM |

## Document Purpose

This document provides detailed instructions on how the ToC\_Journey, ToC\_JourneyLeg and ToC\_Rail feed files are processed by the system, the data transformation rules that are applied and functional database object that are utilised.

## Overview

The main source of information ingested into the VTWC solution is provided by the Trainlines booking engine, ToC. ToC (Train Operating Company) provides several feed files on a daily basis which all inter-depend on each other by a relationship defined on a database (primary/foreign) key.

This document intends to detail the following files;

* ToC\_Journey - A Journey represents the origin and final destination Locations for a Ticket. Depending on the type of Ticket purchased a Journey could include departure and arrival times. Also, depending on the related information available, it will possible to identify whether the Journey includes a VTWC component.
* ToC\_JourneyLeg - A Journey Leg represents a portion of a Journey attributable to one TOC. Depending on the type of Ticket purchased, the information recorded will include the origin and destination Locations, the departure and arrival date and times, train ID and the associated TOC. The TOC information will be used to determine if a Journey includes a VTWC component.
* ToC\_Rail - the Rail Leg file contains the rail trip for all passengers that the Booker purchased tickets for.

This document should be used in conjunction with other ToC feed files documents, all of which are referenced in the Master ToC feed documentation.

## Process

## High Level Steps – Journey

The following provides a high level process flow of the main processes that are applied to the Customer feed file;

|  |  |
| --- | --- |
| **No.** | **Steps** |
| 001 | Log processing start time. |
| 002 | Log operation status and data related statistics. |
| 003 | Populate temporary table with location information from [Reference].[Location\_NLCCode\_VW] |
| 004 | Create unique file set of the location data as potentially there could be duplicates |
| 005 | Insert data into STG\_Journey |
| 006 | Flag those records that have been processed in the STG\_Journey table |
| 007 | Log processing end time |
| 008 | Log operation status and data related statistics such as updates |

## Functional Steps - Journey

|  |  |  |
| --- | --- | --- |
| **No.** | **Database Object** | **Steps** |
| 001 | [PreProcessing].[CBE\_Journey\_Insert] | Capture logging and auditing information |
| 002 | Populate temporary table with location information from [Reference].[Location\_NLCCode\_VW] |
| 003 | Identify unique records using the ID and TkT\_ID fields, ordering on on DateModified |
| 004 | Insert data into STG\_Journey from SalesDetail, joining on the Unique dataset (step 003)  and referencing the [Reference].[Location\_NLCCode\_VW] on Origin\_NLC, Destination\_NLC, referencing [Reference].[TOC] on ShortCode  and referencing STG\_Journey on ExtReference  where f.SalesDetailID IS NULL  AND a.InformationSourceID = @informationsourceid  AND b.RANKING = 1  AND a.IsTrainTicketInd = 1  [ FROM [Staging].[STG\_SalesDetail] a WITH (NOLOCK)  INNER JOIN CTE\_CBE\_JourneyDirections b WITH (NOLOCK) ON SUBSTRING(a.ExtReference,CHARINDEX('TKT',a.ExtReference,1)+7,LEN(a.ExtReference)) = CAST(b.TKT\_ID AS NVARCHAR(256))  LEFT JOIN #tmp\_NLCCode\_LU c WITH (NOLOCK) ON c.NLCCode = b.Origin\_NLC  LEFT JOIN #tmp\_NLCCode\_LU d WITH (NOLOCK) ON d.NLCCode = b.Destination\_NLC  LEFT JOIN [Reference].[TOC] e WITH (NOLOCK) ON e.ShortCode = b.Primary\_TOC\_Code  LEFT JOIN [Staging].[STG\_Journey] f WITH (NOLOCK) ON f.ExtReference = CAST(b.ID AS NVARCHAR(256))  AND f.InformationSourceID = @informationsourceid  WHERE f.SalesDetailID IS NULL  AND a.InformationSourceID = @informationsourceid  AND b.RANKING = 1  AND a.IsTrainTicketInd = 1 ] |
| 005 | Update the lastModifiedDate |
| 006 | Cpature logging and auditing infromation |

## Table Mappings and Business Rules - Journey

|  |  |  |  |
| --- | --- | --- | --- |
| **Feed Details** | | | |
| **Feed ID/#:** | 46 | **Feed Format** | Delimited |
| **Feed Name** | Journey | **Field Delimiter** | | |
| **Feed Source** | TOC+ | **Record Delimiter** | LF |
| **Frequency** | Daily | **Header/Footer/Both** | Header |
| **Input/Output** | Input | **Text Qualifiers** | Yes |
| **Output Recipient** | Merkle | **Encryption** | Other |
| **CDI Processing** | No | **Delivery Mechanism** | Client SFTP |
| **Update Critical** | No | **File Location** | \data |
| **Incremental / Full Refresh** | Incremental | **Contact Information** | N |
| **Physical File Naming Convention** | VT\_MKT\_YYYYMMDDHHMMSS\_Journey.csv | **Count file received** | No |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Inbound Field\_Name** | **[PreProcessing].[TOC\_Journey] Field Name** | **Field\_Data\_type** | **Length** | **Mandatory** | **Staging.STG\_Journey Field Name** | **Business Rules** |
| journeyid | [journeyid] | bigint |  |  | [Staging].STG\_Journey.[ExtReference] |  |
| purchaseid | [purchaseid] | bigint |  |  | [Staging].STG\_Journey.[PurchaseID] | New column |
| tcscustomerid | [tcscustomerid] | bigint |  |  |  |  |
| tcstransactionid | [tcstransactionid] | bigint |  |  | [Staging].STG\_Journey.[tcstransactionid] |  |
| tcsbookingid | [tcsbookingid] | bigint |  |  | [Staging].STG\_Journey.[BookingID] | New column |
| transactiondate | [transactiondate] | datetime |  |  |  | (not required as being brought through on the Transactions feed file) |
| channelcode | [channelcode] | nvarchar | 15 |  |  | (populated in the transaction table) |
| businessorleisure | [businessorleisure] | nchar | 1 |  |  | (to be taken from the booking table) |
| faresettingtoc | [faresettingtoc] | nvarchar | 3 |  | [Staging].STG\_Journey.ToCIDPrimary | reference.toc |
| origstation | [origstation] | nvarchar | 50 |  |  |  |
| deststation | [deststation] | nvarchar | 50 |  |  |  |
| reasonfortravel | [reasonfortravel] [nvarchar](50) NULL, | nvarchar | 50 |  |  | information will be taken from Booking table |
| outdatedep | [outdatedep] [datetime] NULL, | datetime |  |  | [Staging].STG\_Journey.[departuredatetime] | A single journey record will be split into two which will include rtn and arrival dates.  IsReturnIndicator - 1 inferrs that the journey was on the return leg |
| outdatearr | [outdatearr] [datetime] NULL, | datetime |  |  | [Staging].STG\_Journey.[Arrivaldatetime] |
| retdatedep | [retdatedep] [datetime] NULL, | datetime |  |  | [Staging].STG\_Journey.[departuredatetime] |
| retdatearr | [retdatearr] [datetime] NULL, | datetime |  |  | [Staging].STG\_Journey.[Arrivaldatetime] |
| journeytype | [journeytype] [nchar](1) NULL, | nchar | 1 |  | [Staging].STG\_Journey.[IsReturnIndicator] |  |
| travellername | [travellername] [nvarchar](100) NULL, | nvarchar | 100 |  |  | Will retrive this infromation from customer feed file |
| railcarddesc | [railcarddesc] [nvarchar](50) NULL, | nvarchar | 50 |  | [Staging].Reference.toc.name | Match on Railcard1 = [RailcardTypeID] |
| railcard1 | [railcard1] [nvarchar](5) NULL, | nvarchar | 5 |  | [Staging].STG\_SalesDetail.[RailcardTypeID] | Match on Railcard1 = [RailcardTypeID] |
| noofrailcards | [noofrailcards] [int] NULL, | int |  |  | [Staging].STG\_Journey.[NoOfRailCards] |  |
| totaladults | [totaladults] [int] NULL, | int |  |  | [Staging].STG\_Journey.[TotalAdults] |  |
| totalchildren | [totalchildren] [int] NULL, | int |  |  | [Staging].STG\_Journey.[TotalChildern] |  |
| totalreturningadults | [totalreturningadults] [int] NULL, | int |  |  | [Staging].STG\_Journey.[TotalReturningAdults] |  |
| totalreturningchildren | [totalreturningchildren] [int] NULL, | int |  |  | [Staging].STG\_Journey.[TotalReturningChildern] |  |
| costoftickets | [costoftickets] [numeric](6, 2) NULL, | numeric | 6, 2 |  | [Staging].STG\_Journey.CostofTickets |  |
| totalcost | [totalcost] [numeric](6, 2) NULL, | numeric | 6, 2 |  | [Staging].STG\_Journey.TotalCost |  |
| savingsmade | [savingsmade] [numeric](6, 2) NULL, | numeric | 6, 2 |  | [Staging].STG\_Journey.SavingsMade |  |
| procode | [procode] [nvarchar](15) NULL, | nvarchar | 15 |  | [Staging].STG\_Journey.ProCode |  |
| tickettypecode | [tickettypecode] [nvarchar](5) NULL, | nvarchar | 5 |  | [Staging].STG\_Sales\_Detail.productID | match to reference.product table |
| tickettypedesc | [tickettypedesc] [nvarchar](50) NULL, | nvarchar | 50 |  | [reference].[product] | Match to refernece.product table using [Staging].STG\_Sales\_Detail.productID |
| deliverymethodcode | [deliverymethodcode] [nvarchar](20) NULL, | nvarchar | 20 |  | [Staging].STG\_Sales\_Detail.deliverymethodcode | match to reference.FulfilmentMethod |
| deliverymethoddescription | [deliverymethoddescription] [nvarchar](50) NULL, | nvarchar | 50 |  | [Staging].STG\_Sales\_Detail.deliverymethodescription |  |
| journeydirection | [journeydirection] [nvarchar](5) NULL, | nvarchar | 5 |  | [Staging].STG\_Journey.IsOutboundIndicator  and [Staging].STG\_Journey.IsReturnIndicator |  |
| journeyreference | [journeyreference] [nvarchar](20) NULL, | nvarchar | 20 |  | [Staging].STG\_Sales\_transactions.bookingreference |  |
| cmddateupdated | [cmddateupdated] [datetime] NULL, | datetime |  |  |  | CMD is the Train Lines campaign tool |
| class | [class] [nchar](1) NULL, | nchar | 1 |  |  | Captured in the Journey Leg file |
| origstationcode | [origstationcode] [nvarchar](5) NULL, | nvarchar | 5 |  | [Staging].STG\_Journey.locationidorigin | join to reference.location on CRS code where IsStation = 1 |
| deststationcode | [deststationcode] [nvarchar](5) NULL, | nvarchar | 5 |  | [Staging].STG\_Journey.locationiddestination | join to reference.location on CRS code where IsStation = 1 |
| outboundmileage | [outboundmileage] [numeric](6, 2) NULL, | numeric | 6, 2 |  | [Staging].STG\_Journey.outboundmileage |  |
| availabilitycode | [availabilitycode] [nvarchar](5) NULL, | nvarchar | 5 |  |  | Create new reference table to support availabilitycode  and match on availabilitycode to populate AvailablilityCode |
| DisabledInd | [DisabledInd] [nchar](1) NULL, | nchar | 1 |  | [Staging].STG\_Journey.[DisableInd] |  |
| NoFullFareAdults | [NoFullFareAdults] [int] NULL, | int |  |  | [Staging].STG\_Journey.NoFullFareAdults |  |
| NoDiscFareAdults | [NoDiscFareAdults] [int] NULL, | int |  |  | [Staging].STG\_Journey.NoDiscFareAdults |  |
| NoFullFareChildren | [NoFullFareChildren] [int] NULL, | int |  |  | [Staging].STG\_Journey.NoFullFareChildren |  |
| NoDiscFareChildren | [NoDiscFareChildren] [int] NULL, | int |  |  | [Staging].STG\_Journey.NoDiscFareChildren |  |
| NoRailcard1 | [DateCreated] [datetime] NULL, | int |  |  | [Staging].STG\_Journey.NoRailcard1 |  |
| DateCreated | [DateUpdated] [datetime] NULL, | datetime |  |  | [Staging].STG\_Journey.DateCreated |  |
| DateUpdated | [FareId] [int] NULL, | datetime |  |  | [Staging].STG\_Journey.DateUpdated |  |
| FareId | [PromoCode] [nvarchar](30) NULL, | int |  |  | [Staging].STG\_Journey.FareId |  |
| PromoCode | [FullAdultFare] [numeric](6, 2) NULL, | nvarchar | 30 |  | [Staging].STG\_Journey.PromoCode |  |
| FullAdultFare | [DiscAdultFare1] [numeric](6, 2) NULL, | numeric | 6, 2 |  | [Staging].STG\_Journey.FullAdultFare |  |
| DiscAdultFare1 | [DiscAdultFare2] [numeric](6, 2) NULL, | numeric | 6, 2 |  | [Staging].STG\_Journey.DiscAdultFare1 |  |
| DiscAdultFare2 | [DiscAdultFare3] [numeric](6, 2) NULL, | numeric | 6, 2 |  | [Staging].STG\_Journey.DiscAdultFare2 |  |
| DiscAdultFare3 | [FullChildFare] [numeric](6, 2) NULL, | numeric | 6, 2 |  | [Staging].STG\_Journey.DiscAdultFare3 |  |
| FullChildFare | [DiscChildFare1] [numeric](6, 2) NULL, | numeric | 6, 2 |  | [Staging].STG\_Journey.FullChildFare |  |
| DiscChildFare1 | [DiscChildFare2] [numeric](6, 2) NULL, | numeric | 6, 2 |  | [Staging].STG\_Journey.DiscChildFare1 |  |
| DiscChildFare2 | [DiscChildFare3] [numeric](6, 2) NULL, | numeric | 6, 2 |  | [Staging].STG\_Journey.DiscChildFare2 |  |
| DiscChildFare3 | [NoAdultsFullFare] [int] NULL, | numeric | 6, 2 |  | [Staging].STG\_Journey.DiscChildFare3 |  |
| NoAdultsFullFare | [NoAdultsDiscFare1] [int] NULL, | int |  |  | [Staging].STG\_Journey.NoAdultsFullFare |  |
| NoAdultsDiscFare1 | [NoAdultsDiscFare2] [int] NULL, | int |  |  | [Staging].STG\_Journey.NoAdultsDiscFare1 |  |
| NoAdultsDiscFare2 | [NoAdultsDiscFare3] [int] NULL, | int |  |  | [Staging].STG\_Journey.NoAdultsDiscFare2 |  |
| NoAdultsDiscFare3 | [NoChildFullFare] [int] NULL, | int |  |  | [Staging].STG\_Journey.NoAdultsDiscFare3 |  |
| NoChildFullFare | [NoChildDiscFare1] [int] NULL, | int |  |  | [Staging].STG\_Journey.NoChildFullFare |  |
| NoChildDiscFare1 | [NoChildDiscFare2] [int] NULL, | int |  |  | [Staging].STG\_Journey.NoChildDiscFare1 |  |
| NoChildDiscFare2 | [NoChildDiscFare3] [int] NULL, | int |  |  | [Staging].STG\_Journey.NoChildDiscFare2 |  |
| NoChildDiscFare3 | [NoRailcard1] [int] NULL, | int |  |  | [Staging].STG\_Journey.NoChildDiscFare3 |  |
| Railcard2 | [Railcard2] [nvarchar](5) NULL, | nvarchar | 5 |  | [Staging].STG\_Journey.Railcard2 |  |
| NoRailcard2 | [NoRailcard2] [int] NULL, | int |  |  | [Staging].STG\_Journey.NoRailcard2 |  |
| Railcard3 | [Railcard3] [nvarchar](5) NULL, | nvarchar | 5 |  | [Staging].STG\_Journey.Railcard3 |  |
| NoRailcard3 | [NoRailcard3] [int] NULL, | int |  |  | [Staging].STG\_Journey.NoRailcard3 |  |
| NoGroupTicketsFullFare | [NoGroupTicketsFullFare] [int] NULL, | int |  |  | [Staging].STG\_Journey.NoGroupTicketsFullFare |  |
| NoGroupTicketsDiscFare | [NoGroupTicketsDiscFare] [int] NULL, | int |  |  | [Staging].STG\_Journey.NoGroupTicketsDiscFare |  |
| FullGroupFare | [FullGroupFare] [numeric](6, 2) NULL, | numeric | 6, 2 |  | [Staging].STG\_Journey.FullGroupFare |  |
| DiscGroupFare | [DiscGroupFare] [numeric](6, 2) NULL, | numeric | 6, 2 |  | [Staging].STG\_Journey.DiscGroupFare |  |
| Railcard2Desc | [Railcard2Desc] [nvarchar](50) NULL, | nvarchar | 50 |  | [Staging].STG\_Journey.Railcard2Desc |  |
| Railcard3Desc | [Railcard3Desc] [nvarchar](50) NULL, | nvarchar | 50 |  | [Staging].STG\_Journey.Railcard3Desc |  |
| FareSettingTOCDesc | [FareSettingTOCDesc] [nvarchar](50) NULL, | nvarchar | 50 |  | [Staging].STG\_Journey.FareSettingTOCDesc |  |
| AvailabilityCodeDesc | [AvailabilityCodeDesc] [nvarchar](50) NULL, | nvarchar | 50 |  | reference.AvailabilityCode | Create new reference table to support availabilitycode and match on  availabilitycode to populate AvailablilityCode |
|  | [CreatedDateETL] [datetime] NULL, |  |  |  |  |  |
|  | [LastModifiedDateETL] [datetime] NULL, |  |  |  |  |  |
|  | [ProcessedInd] [nchar](1) NULL, |  |  |  |  |  |
|  | [DataImportDetailID] [int] NULL, |  |  |  |  |  |

## High Level Steps – Journey Legs

The following provides a high level process flow of the main processes that are applied to the Customer feed file;

There is a section on the original code that refers to JourneyTrain. There is no longer a requirement to populate/reference this object. (this was confirmed by Gary)

|  |  |
| --- | --- |
| **No.** | **Steps** |
| 001 | Log processing start time. |
| 002 | Log operation status and data related statistics. |
| 003 | Populate temporary table with location information from [Reference].[Location\_NLCCode\_VW] |
| 004 | Identify the latest journey |
| 006 | Insert into STG\_JourneyLeg from SalesDetail and STG\_Journey table |
| 007 | Update those records that have been processed in the STG\_JourneyLeg |
| 008 | Log processing end time |
| 009 | Log operation status and data related statistics such as updates |

## Functional Steps – Journey Legs

|  |  |  |
| --- | --- | --- |
| **No.** | **Database Object** | **Steps** |
| 001 | [PreProcessing].[CBE\_JourneyLeg\_Insert] | Capture logging and auditing information |
| 002 | Populate temporary table with location information from [Reference].[Location\_NLCCode\_VW] |
| 003 | Identify unique records using the ID, JRND\_ID and LEG\_ID fields, ordering on on DateModified and CBE\_JourneyID |
| 004 | Insert data into [Staging].[STG\_JourneyLeg] joining to [Staging].[STG\_Journey] on SalesDtailID and #tmp\_JourneyLegs on b.JRND\_ID and informationsourceID  Referencing #tmp\_NLCCode\_LU matching NLCCode to Origin\_NLC, Destination\_NLC, Train\_Origin\_NLC, Train\_Destination\_NLC  Referencing [Reference].[TOC] matching TOC = SHORTHand  Referencing [Staging].[STG\_JourneyLeg] matching ID = extReference and InformationSourceID  Referencing [Reference].[Product] matching sd.ProductID = j.ProductID |
| 005 | Update the lastModifiedDate and ProcessedInD fields in STG\_JourneyLeg |
| 006 | Cpature logging and auditing infromation |

## Table Mappings and Business Rules – Journey Legs

|  |  |  |  |
| --- | --- | --- | --- |
| **Feed Details** | | | |
| **Feed ID/#:** | 47 | **Feed Format** | Delimited |
| **Feed Name** | Journey Legs | **Field Delimiter** | | |
| **Feed Source** | TOC+ | **Record Delimiter** | LF |
| **Frequency** | Daily | **Header/Footer/Both** | Header |
| **Input/Output** | Input | **Text Qualifiers** | Yes |
| **Output Recipient** | Merkle | **Encryption** | Other |
| **CDI Processing** | No | **Delivery Mechanism** | SFTP |
| **Update Critical** | No | **File Location** | \data |
| **Incremental / Full Refresh** | Incremental | **Contact Information** | 0 |
| **Physical File Naming Convention** | VT\_MKT\_YYYYMMDDHHMMSS\_Journey\_leg.csv | **Count file received** | No |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Inbound Field\_Name** | **[PreProcessing].[TOC\_JourneyLeg] Field Name** | **Field\_Data\_type** | **Length** | **Mandatory** | **Staging.STG\_JourneyLeg Fieldname** | **Business Rules** |
| LegId | LegId | bigint |  |  | Staging.STG\_JourneyLeg.ExtReference |  |
| JourneyId | JourneyId | bigint |  |  | Staging.STG\_JourneyLeg.JourneyID |  |
| TCSCustomerId | TCSCustomerId | bigint |  |  |  |  |
| TransactionDate | TransactionDate | datetime |  |  |  | Infomration being pulled through from the Transaction table |
| LegNo | LegNo | int |  |  | Staging.STG\_JourneyLeg.LegNo |  |
| LegOrigStation | LegOrigStation | nvarchar | 50 |  |  | join on CRS Code to reference.location.NLCCode\_VW |
| LegDestStation | LegDestStation | nvarchar | 50 |  |  | join on CRS Code to reference.location.NLCCode\_VW |
| DepDateTime | DepDateTime | datetime |  |  | [Staging].STG\_JourneryLeg.DepartureDateTime |  |
| ArrDateTime | ArrDateTime | datetime |  |  | [Staging].STG\_JourneryLeg.ArrivalDateTime |  |
| ModeOfTransport | ModeOfTransport | char | 3 |  | [Staging].STG\_JourneryLeg.ModeOfTransport |  |
| SeatReserved | SeatReserved | char | 1 |  |  |  |
| SeatingClass | SeatingClass | char | 1 |  | [Staging].STG\_JourneryLeg.TicketClassID | join on seatingclassdesc to ticketclass table. Standard and both (both = first class) |
| ReservationRequired | ReservationRequired | char | 3 |  |  |  |
| OperatorCode | OperatorCode | int | 3 |  | [Staging].STG\_JourneryLeg.ToCID | Join reference.toc on operatorCode = ShortCode |
| OperatorType | OperatorType | nvarchar | 10 |  |  | (TocID will reference back to the TrainDescription) |
| OperatorDescription | OperatorDescription | nvarchar | 75 |  |  | (TocID will reference back to the TrainDescription) |
| RetailTrainId | RetailTrainId | nvarchar | 10 |  | [Staging].STG\_JourneryLeg.RSID | Concatenate '00+RSID' (also known as RSID) |
| TCSTransactionId | TCSTransactionId | bigint |  |  |  |  |
| CMDDateUpdated | CMDDateUpdated | datetime |  |  |  |  |
| LegOrigStationCode | LegOrigStationCode | nvarchar | 50 |  |  | aka CRS Code (please reference G21) |
| LegDestStationCode | LegDestStationCode | nvarchar | 50 |  |  | aka CRS Code (please reference G22) |
| Coach | Coach | char | 1 |  | [Staging].[STG\_JourneyLeg].SeatResveration | concatenate Coach and Seat |
| Seat | Seat | int |  |  |
| QuietZone\_YN | QuietZone\_YN | char | 1 |  | [Staging].[STG\_JourneryLeg].QuietZone |  |
| TrainUId | TrainUId | nvarchar | 20 |  | [Staging].[STG\_JourneryLeg].TrainUId |  |
| JLType | JLType | char | 1 |  | [Staging].[STG\_JourneryLeg].JLType | aka JourneyType = outbound or return |
| CMDDateCreated | CMDDateCreated | datetime |  |  |  |  |
| FareSettingTOCDesc | FareSettingTOCDesc | nvarchar | 50 |  |  | inferring this information from transaction or booking |
| ModeOfTransportDesc | ModeOfTransportDesc | nvarchar | 50 |  | [Staging].STG\_JourneryLeg.ModeOfTransportDesc |  |
| SeatingClassDesc | SeatingClassDesc | nvarchar | 50 |  |  |  |
| SeatReservedDesc | SeatReservedDesc | nvarchar | 50 |  |  |  |
| OperatorCodeDesc | OperatorCodeDesc | nvarchar | 50 |  |  |  |

## High Level Steps – (Rail) Legs

The following provides a high level process flow of the main processes that are applied to the Customer feed file;

|  |  |
| --- | --- |
| **No.** | **Steps** |
| 001 | Log processing start time. |
| 002 | Log operation status and data related statistics. |
| 003 | Update STG\_SalesDetail table with the relevant seat and reservation information |
| 004 | Update the STG\_JourneyLeg table with the processed flag |
| 006 | Set variables for onwards processing with successful and unsuccessful updates |
| 007 | Log processing end time |
| 008 | Log operation status and data related statistics such as updates |

## Functional Steps – Rail Legs

|  |  |  |
| --- | --- | --- |
| **No.** | **Database Object** | **Steps** |
| 001 | [PreProcessing].[CBE\_JourneyLeg\_Insert] | Capture logging and auditing infromation |
| 002 | Update STG\_SalesDetail joining to Staging.STG\_Journey  On SalesDeatilID  Join to Staging.STG\_JourneyLeg on JourneyID  Join to CBE\_SeatReservation on extreference to JL\_ID, informationsourceID and PF\_ID  WHERE b.[DataImportDetailID] = @dataimportdetailid  AND sd.InformationSourceID = @informationsourceid  AND b.[ProcessedInd] = 0  AND b.[Unit\_Type\_Code] = 'SEAT'  AND sd.IsTrainTicketInd = 1  Example code  [ UPDATE a  SET SeatReservation = CASE WHEN ISNULL(b.Coach\_Identifier,'') + ' ' + ISNULL(Seat\_Berth\_Number,'') = ''  THEN NULL  ELSE 'COACH ' + ISNULL(b.Coach\_Identifier,'') + ' ' + 'SEAT ' + ISNULL(Seat\_Berth\_Number,'')  END  ,LastModifiedDate = GETDATE()  ,LastModifiedBy = 0  FROM Staging.STG\_SalesDetail sd  INNER JOIN Staging.STG\_Journey j ON sd.SalesDetailID = j.SalesDetailID  INNER JOIN Staging.STG\_JourneyLeg a ON j.JourneyID = a.JourneyID  INNER JOIN PreProcessing b ON a.ExtReference = CAST(b.JL\_ID AS NVARCHAR(256))  AND a.InformationSourceID = @informationsourceid  AND CAST(b.PF\_ID AS nvarchar(256)) = CAST(SUBSTRING(sd.ExtReference,  CHARINDEX('PF\_ID=',sd.ExtReference,1)+LEN('PF\_ID='),  CHARINDEX(',',SUBSTRING(sd.ExtReference,CHARINDEX('PF\_ID=',sd.ExtReference,1)+LEN('PF\_ID='),LEN(sd.ExtReference)),1)-1) AS NVARCHAR(256))  WHERE b.[DataImportDetailID] = @dataimportdetailid  AND sd.InformationSourceID = @informationsourceid  AND b.[ProcessedInd] = 0  AND b.[Unit\_Type\_Code] = 'SEAT'  AND sd.IsTrainTicketInd = 1  ] |
| 003 | Update ProcessInD and lastmodifieddateETL fields in [STG\_JourneyLeg] |
| 006 | Cpature logging and auditing infromation |

## Table Mappings and Business Rules - Legs

|  |  |  |  |
| --- | --- | --- | --- |
| **Feed Details** | | | |
| **Feed ID/#:** | 48 | **Feed Format** | Delimited |
| **Feed Name** | Legs | **Field Delimiter** | | |
| **Feed Source** | TOC+ | **Record Delimiter** | LF |
| **Frequency** | Daily | **Header/Footer/Both** | Header |
| **Input/Output** | Input | **Text Qualifiers** | Yes |
| **Output Recipient** | Merkle | **Encryption** | Other |
| **CDI Processing** | No | **Delivery Mechanism** | Client SFTP |
| **Update Critical** | No | **File Location** | \data |
| **Incremental / Full Refresh** | Incremental | **Contact Information** | N |
| **Physical File Naming Convention** | VT\_MKT\_YYYYMMDDHHMMSS\_VT\_LEGS.csv | **Count file received** | No |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Inbound Field\_Name** | **[PreProcessing].[TOC\_Leg]** | **Field\_Data\_type** | **Length** | **Mandatory** | **Staging.STG\_Leg Fieldname** | **Business Rules** |
| Operator | Operator | char | 3 |  |  |  |
| JL\_Id | JL\_Id | bigint |  |  | LegID | Create journey record for each record in the Rail Leg file, as the Rail Leg file contains the rail  trip for all passaengers that the Booker purchased tickets for.  This field links back to the LegID in the JourneyLeg table |
| coach | coach | char | 1 |  | [Staging].[STG\_JourneyLeg].SeatResveration | concatenate Coach and Seat |
| seat\_number | seat\_number | int |  |  |
| quiet\_coach | quiet\_coach | char | 1 |  | [Staging].[STG\_JourneryLeg].QuietZone |  |
| DateCreated | DateCreated | datetime |  |  |  | (not required as being brought through on the Transactions feed file) |
| DateUpdated | DateUpdated | datetime |  |  |  | (not required as being brought through on the Transactions feed file) |