Virgin Trains West Coast – Single Customer View (SCV)

Rail Refunds and Rail Supplements ToC+ VTWC

Commercial in confidence

Author:

Russell Jackson – Solution Architect

Confidential

## Table of Contents

[Table of Contents 2](#_Toc522025898)

[Document Management 3](#_Toc522025899)

[1. Document Purpose 4](#_Toc522025900)

[2. Overview 4](#_Toc522025901)

[3. Process 4](#_Toc522025902)

[3.1. High Level Steps – Rail Supplements 4](#_Toc522025903)

[3.2. Functional Steps – Rail Supplements 5](#_Toc522025904)

[3.3. Table Mappings and Business Rules – Rail Supplements 6](#_Toc522025905)

[3.4. High Level Steps – Rail Refunds 8](#_Toc522025906)

[3.5. Functional Steps – Rail Refunds 8](#_Toc522025907)

[3.6. Table Mappings and Business Rules – Rail Refunds 10](#_Toc522025908)

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

|  |  |  |  |
| --- | --- | --- | --- |
| **Version Control** | | |  |
| **Version** | **Date** | **Author(s)** | **Section Changed** |
| 0.1 | 14/08/2018 | 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 cleansing 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) |
| Journey, Journey Leg and Leg |  | [Click here](https://merkleinc.sharepoint.com/:w:/r/sites/VirginTrainsWestCoast-CRMProject/Shared%20Documents/General/Development%20Documentation/Design%20Documentation/Staging/TOC+/Merkle%20VTWC%20ToC+%20Journey,%20Journey%20Leg%20and%20Rail%20Leg.docx?d=w7b4d90c8514d4464af533e0735354fc2&csf=1&e=dSXCJJ) |
| Transactions |  | [Click here](https://merkleinc.sharepoint.com/:w:/r/sites/VirginTrainsWestCoast-CRMProject/Shared%20Documents/General/Development%20Documentation/Design%20Documentation/Staging/TOC+/Merkle%20VTWC%20TOC+%20Load%20Design%20-%20Transaction%20-%20v0.1.docx?d=wa6226e8a8f3a47bf90351003f8c19f7a&csf=1&e=nfYUFv) |

|  |  |  |
| --- | --- | --- |
| **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\_Rail Refunds and ToC\_Rail Supplements 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\_RailRefunds
* ToC\_RailSupplements

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 – Rail Supplements

Supplements information is loaded and stored in the SalesDetail table.

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 | Identify unique records within the CBE\_SupplementsSale table |
| 004 | Insert into STG\_SalesDetails table |
| 005 | Log processing end time |
| 006 | Log operation status and data related statistics such as updates |

## Functional Steps – Rail Supplements

|  |  |  |
| --- | --- | --- |
| **No.** | **Database Object** | **Steps** |
| 001 | [PreProcessing].[CBE\_SalesDetail\_Supplement\_Insert] | Capture logging and auditing information |
| 002 | Identify unique records by ordering on SupplementSaleID descending and DataImportDetailID is the same |
| 003 | Insert into SalesDetail  FROM [Staging].[STG\_SalesTransaction] a WITH (NOLOCK)  INNER JOIN CTE\_CBE\_SupplementSales b WITH (NOLOCK) ON a.ExtReference = CAST(b.ST\_ID AS NVARCHAR(256))  AND a.InformationSourceID = @informationsourceid  INNER JOIN [Reference].[Product] c WITH (NOLOCK) ON b.Supplement\_Code = c.SupplementCode  AND c.InformationSourceID = @informationsourceid  AND a.SalesTransactionDate BETWEEN c.StartDate AND c.EndDate  AND c.ArchivedInd = 0  LEFT JOIN [Staging].[STG\_SalesDetail] d WITH (NOLOCK) ON 'FF\_ID=' + ISNULL(CAST(b.FF\_ID AS NVARCHAR(256)),'NULL') +  ',ST\_ID=' + ISNULL(CAST(b.ST\_ID AS NVARCHAR(256)),'NULL') +  ',PF\_ID=' + ISNULL(CAST(b.PF\_ID AS NVARCHAR(256)),'NULL') +  ',TKT\_ID=' + ISNULL(CAST(b.TKT\_ID AS NVARCHAR(256)),'NULL') +  ',JRND\_ID='+ ISNULL(CAST(b.JRND\_ID AS NVARCHAR(256)),'NULL') +  ',SU\_ID=' + ISNULL(CAST(b.SU\_ID AS NVARCHAR(256)),'NULL') = d.ExtReference  AND d.InformationSourceID = @informationsourceid  WHERE d.SalesDetailID IS NULL  AND b.Net\_Value IS NOT NULL  AND b.RANKING = 1 |
| 004 | Flag those records that have been processed by populating the ProcessedInD flag and the LastModifiedDateETL |
| 005 | Capture logging and auditing information |

## 

## Table Mappings and Business Rules – Rail Supplements

|  |  |  |  |
| --- | --- | --- | --- |
| **Feed Details** | | | |
| **Feed ID/#:** | 57 | **Feed Format** | Delimited |
| **Feed Name** | Supplement | **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\_Supplement.csv | **Count file received** | No |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Inbound Field\_Name** | **[PreProcessing].[TOC\_Supplement]** | **Field\_Data\_type** | **Length** | **Mandatory** | **Staging.** SalesTransactionDetail **filenames** | **Business Rules** |
| SupplementID | SupplementID | bigint |  |  | [Staging].[SalesTransactionDetail].[Extreference] | insert or update into the SalesTransactionDetail table |
| TCSBookingID | TCSBookingID | bigint |  |  | [Staging].[SalesTransactionDetail].[TCSBookingID] |  |
| TCSTransactionId | TCSTransactionId | bigint |  |  | [Staging].[SalesTransactionDetail].[TCSTransactionID] | match to the ExtReference in the SalesTransactions table, which will return the  salestransactionID |
| SupplementTypeCode | SupplementTypeCode | nvarchar | 5 |  | [Staging].[salesdetailtable].[productid] | Build reference table with supplementtypecode and supplementtypedesc |
| SupplementTypeDesc | SupplementTypeDesc | nvarchar | 50 |  | [reference].[Product].[supplementtypedesc] |  |
| AppliesTo | AppliesTo | nvarchar | 5 |  |  | The database has TCSBooking, which can be joined to the Journey table and on the  SalesDetail which will give the user the same result |
| Cost | Cost | numeric | 6, 2 |  | [staging].[salesdetail].Unitprice |  |
| Quantity | Quantity | int |  |  | [staging].[salesdetail].Quanity |  |
| TotalCost | TotalCost | numeric | 6, 2 |  | [staging].[salesdetail].SalesAmount |  |
| Zone | Zone | nvarchar | 50 |  | [Staging].Product].[Zone] | (the product table will be at a supplement type level.  each product could have a London  zone travel card, you can't have a London travel card without a product) |
| DateUpdated | DateUpdated | datetime |  |  | [Staging].[Salesdetail].DateUpdated |  |
| DateCreated | DateCreated | datetime |  |  | [Staging].[Salesdetail].DateCreated |  |

## High Level Steps – Rail Refunds

Supplements information is loaded and stored in the SalesDetail table.

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 | Identify unique records in the CBE\_Refund table but using the ID and order on Date\_Modified and CBE\_RefundID descending |
| 004 | Insert data into STG\_Refund |
| 005 | Flag those records that have been processed in the STG\_Refund table |
| 006 | Log processing end time |
| 007 | Log operation status and data related statistics such as updates |

## Functional Steps – Rail Refunds

|  |  |  |
| --- | --- | --- |
| **No.** | **Database Object** | **Steps** |
| 001 | [PreProcessing].[CBE\_Refund\_Insert] | Capture logging and auditing information |
| 002 | Identify unique records in the CBE\_Refund table but using the ID and order on Date\_Modified and CBE\_RefundID descending |
| 003 | Insert records using the following logic;  FROM CTE\_CBE\_Refunds a WITH (NOLOCK)  INNER JOIN [Staging].[STG\_SalesTransaction] b WITH (NOLOCK) ON CAST(a.ST\_ID AS NVARCHAR(256)) = b.ExtReference  LEFT JOIN [Reference].[RetailChannel] c WITH (NOLOCK) ON a.[Retail\_Channel\_Code] = c.Code  AND b.SalesTransactionDate BETWEEN c.ValidityStartDate AND c.ValidityEndDate  AND c.InformationSourceID = @informationsourceid  LEFT JOIN [Reference].[RefundReasonCode] d WITH (NOLOCK) ON a.[Reason\_For\_Refund] = d.ExtReference  AND d.InformationSourceID = @informationsourceid  AND a.Refund\_Date BETWEEN d.ValidityStartDate AND d.ValidityEndDate  LEFT JOIN [Reference].[RefundDecisionCode] e WITH (NOLOCK) ON a.[Refund\_Decision\_Code] = e.ExtReference  AND a.Refund\_Date BETWEEN e.ValidityStartDate AND e.ValidityEndDate  AND e.InformationSourceID = @informationsourceid  LEFT JOIN [Staging].[STG\_Refund] f WITH (NOLOCK) ON 'ID=' + ISNULL(CAST(a.ID AS NVARCHAR(256)),'NULL') +  ',ST\_ID=' + ISNULL(CAST(a.ST\_ID AS NVARCHAR(256)),'NULL')  = f.ExtReference  AND f.InformationSourceID = @informationsourceid  WHERE f.RefundID IS NULL  AND a.RANKING = 1 |
| 004 | Flag those records that have been processed by populating the ProcessedInD flag and the LastModifiedDateETL |
| 005 | Capture logging and auditing information |

## Table Mappings and Business Rules – Rail Refunds

|  |  |  |  |
| --- | --- | --- | --- |
| **Feed Details** | | | |
| **Feed ID/#:** | 58 | **Feed Format** | Delimited |
| **Feed Name** | Refunds | **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\_Refunds.csv | **Count file received** | No |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Inbound Field\_Name** | **[PreProcessing].[TOC\_Refunds]** | **Field\_Data\_type** | **Length** | **Mandatory** | **Staging.STG\_Refunds filenames** | **Business Rules** |
| TcsBookingID | TcsBookingID | bigint |  |  | [Staging].[STG\_Refund].ExtReference | Please note; on the booking table there is a BookingID.  This is Merkle’s own ID and should not be matched on.    TCSBookingID can't be used to match back to the booking table.    Match this field to Journey using TcsBookingID, which can then be used to match back to Booking table |
| ArRefArrivalId | ArRefArrivalId | bigint |  |  | [Staging].[STG\_Refund].RefundNumber |  |
| RefundType | RefundType | nvarchar | 20 |  | [Reference].[RefundType].[RefundTypeID] |  |
| Percentage | Percentage | int |  |  | [Staging].[STG\_Refund].[Percentage] |  |
| GrossRefund | GrossRefund | numeric | 6, 2 |  | [Staging].[STG\_Refunds].[GrossRefund] | refund amount minus the tax |
| AdminFee | AdminFee | int |  |  | [Staging].[STG\_Refund].[AdminChargeAmount] |  |
| RefundAmount | RefundAmount | numeric | 6, 2 |  | [Staging].[STG\_Refund].[RefundAmount] |  |
| DatamartCreateDate | DatamartCreateDate | datetime |  |  | [Staging].[STG\_Refund].[SourceCreatedDate] |  |
| DatamartUpdateDate | DatamartUpdateDate | datetime |  |  | [Staging].[STG\_Refund].[SourceModifiedDate] |  |
| RequestedDate | RequestedDate | datetime |  |  | [Staging].[STG\_Refund].[RequestedDate] |  |
| RefundedIssuedDate | RefundedIssuedDate | datetime |  |  | [Staging].[STG\_Refund].[RefundDate] |  |
| RefundReason | RefundReason | nvarchar | 20 |  | [Staging].[STG\_Refund].[RefundReasonCodeID] | match to reference.refundreasoncode and infromationsourceID |
| RefundReasonDesc | RefundReasonDesc | nvarchar | 50 |  |  | use to build reference.refundreasoncode table |