

Transportation Manager API Reference Guide

Version 6.0



One i2 Place
11701 Luna Rd.
Dallas, TX 75234 USA



i2® Transportation Manager API Reference Guide
Version 6.0, August 2002

Copyright © 1995-2002 i2 Technologies US, Inc. All Rights Reserved.

This notice is intended as a precaution against inadvertent publication and does not imply any waiver of confidentiality. Information in this document is subject to change without notice. No part of this document may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording, or information storage or retrieval systems, for any purpose without the express written permission of i2 Technologies US, Inc.

The software and/or database described in this document are furnished under a license agreement or nondisclosure agreement. It is against the law to copy the software onto any medium except as specifically allowed in the license or nondisclosure agreement. **If software or documentation is to be used by the federal government, the following statement is applicable: In accordance with FAR 52.227-19 Commercial Computer Software -- Restricted Rights, the following applies: This software is Unpublished--rights reserved under the copyright laws of the United States.**

The text and drawings set forth in this document are the exclusive property of i2 Technologies US, Inc. Unless otherwise noted, all names of companies, products, street addresses, and persons contained in the scenarios are designed solely to document the use of i2 Technologies US, Inc. products.

The brand names and product names used in this document are the trademarks, registered trademarks, service marks, or trade names of their respective owners. i2 Technologies US, Inc. is not associated with any product or vendor mentioned in this publication unless otherwise noted.

The following trademarks and service marks are the property of i2 Technologies US, Inc. and its affiliates: TradeMatrix; RHYTHM Transportation Enabled Planning; Rhythmtone; eBPO; eBPO Electronic Business Process Optimization; AeroMatrix; AeroMatrix.com; AeroSpaceMatrix; AeroSpaceMatrix.com; AutoMatrix; AutoMatrix.com; eServiceMatrix; FreightMatrix; FreightMatrix.com; SoftgoodsMatrix; B2A; ConstructionMatrix; ConstructionMatrix.com; TelecoMatrix; TelecoMatrix.com; HightechMatrix; HightechMatrix.com; C-StoreMatrix; C-StoreMatrix.com; Heterocasting.

The following registered trademarks are the property of i2 Technologies US, Inc. and its affiliates: i2; i2 Technologies and design; Global Supply Chain Management; PLANET; RHYTHM; “juggler” design; RhythmLink; TradeMatrix.com.

This product may be protected by one or more of the following patents:

Australian Patent No. 692929
Australian Patent No. 708311
Australian Patent No. 709987
Australian Patent No. 710420
Australian Patent No. 711699
Australian Patent No. 712238
Australian Patent No. 728154
Australian Patent No. 731197
PATENTED NO. EP0752134
PATENTED NO. EP0755537
PATENTED NO. EP0776506
PATENTED NO. EP0776507
PATENTED NO. EP0776508
PATENTED NO. EP0776509
Singapore PATENT NO. 37930
Singapore PATENT NO. 37931
Singapore PATENT NO. 51047
Singapore PATENT NO. 51051
Singapore PATENT NO. 64044

Singapore PATENT NO. 67163
Taiwan PATENT NO. 80326
Taiwan PATENT NO. 93090
Taiwan PATENT NO. 100569
Taiwan PATENT NO. 108409
Taiwan PATENT NO. 110827
Taiwan PATENT NO. 113331
Taiwan PATENT NO. 113820
Taiwan PATENT NO. 114592
Taiwan PATENT NO. 127358
United Kingdom Patent No. 2316506
U. S. Patent No. 5,630,123
U. S. Patent No. 5,764,543
U. S. Patent No. 5,832,532
U. S. Patent No. 5,845,258
U. S. Patent No. 5,930,156
U. S. Patent No. 5,931,900
U. S. Patent No. 5,937,155
U. S. Patent No. 5,943,244

U. S. Patent No. 5,974,395
U. S. Patent No. 5,983,194
U. S. Patent No. 5,995,945
U. S. Patent No. 6,031,984
U. S. Patent No. 6,047,290
U. S. Patent No. 6,055,519
U. S. Patent No. 6,055,533
U. S. Patent No. 6,076,108
U. S. Patent No. 6,085,220
U. S. Patent No. 6,119,149
U. S. Patent No. 6,167,380
U. S. Patent No. 6,188,989
U. S. Patent No. 6,222,533
U. S. Patent No. 6,233,493
U. S. Patent No. 6,233,572
U. S. Patent No. 6,266,655
U. S. Patent No. 6,289,384
U. S. Patent No. 6,289,385

Contents

1	Introduction	13
	Overview of API	13
	API Changes	14
	Structure Changes	14
	Service Changes	15
	Parameter Changes	15
	Additional Information	15
2	API Elements	17
	System Overview	17
	Architecture Overview	18
	Using Transportation Manager API Services	18
	Standard Data Types	18
	Creating Objects	19
	Retrieving Objects	20
	Updating Objects	20
	Doing Block Operations	21
	Error Handling	21
	Domain Table Type Map	23
	API Parameter Set	24
	VentureFactory IOR	26
3	API Services	29
	Overview	29
	Multi-Thread API Processing	30
	Address Service	30
	void AddrSrvc::Validate	30
	Delivery Schedule Services	30
	void DeliveryScheduleSrvc::CreateDeliverySchedule	30
	void DeliveryScheduleSrvc::CreateDeliveryScheduleEntity	31
	void DeliveryScheduleSrvc::DeleteDeliverySchedule	31
	void DeliveryScheduleSrvc::DeleteDeliveryScheduleEntity	31

void DeliveryScheduleSrcv::DetachAllItineraries	32
void DeliveryScheduleSrcv::RetrieveDeliverySchedule.....	32
void DeliveryScheduleSrcv::RetrieveDeliveryScheduleEntity	32
void DeliveryScheduleSrcv::UpdateDeliverySchedule.....	33
void DeliveryScheduleSrcv::UpdateDeliveryScheduleEntity	33
Entity Services	33
void EntitySrcv::Create	33
void EntitySrcv::Delete	34
void EntitySrcv::DeleteINCOTerms	34
void EntitySrcv::Retrieve	34
void EntitySrcv::RetrieveINCOTerms	35
void EntitySrcv::RemoveItems.....	35
void EntitySrcv::SetStatusV1	35
void EntitySrcv::Update	36
void CEntityProcessor::RetrieveTMEntityIds.....	36
Financial Services	37
void FinancialsSrcv::CancelFrhtBill	37
void FinancialsSrcv::CancelFrhtBillDetail.....	38
void FinancialsSrcv::CommitAPTransaction	38
void FinancialsSrcv::CommitARTransaction.....	38
void FinancialsSrcv::CommitGLTransaction.....	38
void FinancialsSrcv::CreateFrhtBill.....	39
void FinancialsSrcv::CreateFrhtBillDetail	39
void FinancialsSrcv::CreateNOF.....	39
void FinancialsSrcv::DeleteNOF.....	40
void FinancialsSrcv::ResponsibleCustomerOverride.....	40
void FinancialsSrcv::RetrieveAPTransaction.....	40
void FinancialsSrcv::RetrieveAPTransactionQty.....	40
void FinancialsSrcv::RetrieveARTransaction	41
void FinancialsSrcv::RetrieveARTransactionQty	41
void FinancialsSrcv::RetrieveFrhtBill	41
void FinancialsSrcv::RetrieveFrhtBillDetail	42
void FinancialsSrcv::RetrieveGLTransaction	43
void FinancialsSrcv::RetrieveGLTransactionQty	43
void FinancialsSrcv::RetrieveNOF.....	44
void FinancialsSrcv::RetrieveResponsibleCustomerOverride	44
void FinancialsSrcv::UpdateFrhtBill.....	45
void FinancialsSrcv::UpdateFrhtBillDetail	45
void FinancialsSrcv::UpdateNOF.....	45
Load Services.....	46
void LoadSrcv::AssignToLoad.....	46
void LoadSrcv::AssignToNewLoad	47
void LoadSrcv::CreatePlan.....	47
void LoadSrcv::DisableAssignToTrip.....	47
void LoadSrcv::DisableTripContinuation	48
void LoadSrcv::LoadConfirm.....	48

void LoadSrvc::LoadCreate	48
void LoadSrvc::LoadManifest.....	49
void LoadSrvc::LoadSpotRate	49
void LoadSrvc::LoadUpdateProgress	49
void LoadSrvc::PayableCarrierOverride.....	50
void LoadSrvc::PreBuiltLoadCreate	50
void LoadSrvc::PreBuiltLoadDelete	52
void LoadSrvc::RemoveShipment	53
void LoadSrvc::RemoveShipmentLeg.....	53
void LoadSrvc::RetrieveCondensedLoad	54
void LoadSrvc::RetrieveLoad	54
void LoadSrvc::RetrieveLoadsByDate	55
void LoadSrvc::RetrieveLoadsByDesc	55
void LoadSrvc::RetrievePayableCarrierOverride	56
void LoadSrvc::RetrievePlanLoads.....	56
void LoadSrvc::RetrievePlanShipments	56
void LoadSrvc::SelectLoads	57
void LoadSrvc::SetAllToPlanned	57
void LoadSrvc::SetLoadInstruction	58
void LoadSrvc::SetShipmentPOD.....	58
void LoadSrvc::SetStopETA.....	59
void LoadSrvc::SetStopPOD.....	59
void LoadSrvc::SetStopToDelivered	60
void LoadSrvc::SetToPlanned.....	60
void LoadSrvc::ShipmentUpdateProgress	61
void LoadSrvc::StopConfirm	63
void LoadSrvc::TriggerLoadEvent	63
Load Tender Services	64
Reference Number.....	64
void LoadSrvc::RetrieveTender	65
void LoadSrvc::ReturnToOpen	65
void LoadSrvc::Tender.....	65
void LoadSrvc::TenderAccept	66
void LoadSrvc::TenderCancel.....	67
void LoadSrvc::TenderReject	67
Rate Quotation Services	67
void RateQuotationSrvc::GetPossibleCharges.....	68
void RateQuotationSrvc::RateQuotation.....	68
Shipment Services	69
void ShipmentSrvc::AttachToPlan.....	69
void ShipmentSrvc::CancelShipment	69
void ShipmentSrvc::PayableCarrierOverride.....	69
void ShipmentSrvc::ResponsibleCustomerOverride.....	70
void ShipmentSrvc::RetrievePayableCarrierOverride	70
void ShipmentSrvc::RetrieveResponsibleCustomerOverride	71
void ShipmentSrvc::RetrieveShipmentLegs	71

void ShipmentSvc::RetrieveShipmentLegsById.....	71
void ShipmentSvc::ReturnShipmentLegFromPlanned	72
void ShipmentSvc::SelectForOptimization.....	72
void ShipmentSvc::SendToOptimizer.....	73
void ShipmentSvc::SetToPlanned.....	73
void ShipmentSvc::ShipmentCompositionChange	73
void ShipmentSvc::ShipmentConfirm	74
void ShipmentSvc::TriggerShipmentLegEvent	74
void ShipmentSvc::UpdateShipmentLegs.....	75
Shipment Order Entry Services	75
void ShipmentOrderEntrySvc::CancelShipment.....	75
void ShipmentOrderEntrySvc::CreateShipment	76
void ShipmentOrderEntrySvc::DeleteShipment	77
void ShipmentOrderEntrySvc::PlanShipments	78
void ShipmentOrderEntrySvc::RetrieveShipment	78
void ShipmentOrderEntrySvc::UpdateShipment	80
Tariff Services.....	81
void TariffSvc::CreateTariff.....	82
void TariffSvc::CreateTariffEntity.....	82
void TariffSvc::DeleteTariff.....	82
void TariffSvc::DeleteTariffEntity.....	82
void TariffSvc::RetrieveTariff	83
void TariffSvc::RetrieveTariffEntity.....	83
void TariffSvc::UpdateTariff	84
void TariffSvc::UpdateTariffEntity.....	84
Transport Order Services	84
void TransportOrderSvc::CreateTO	84
void TransportOrderSvc::DeleteTO	85
void TransportOrderSvc::RetrieveTO.....	85
void TransportOrderSvc::RetrieveTOShipments	86
void TransportOrderSvc::SetTOUpdate.....	86
void TransportOrderSvc::TriggerTOEvent.....	87
void TransportOrderSvc::UpdateTO	87
Updating Containers on a Transport Order	88
4 API Structures.....	89
Structure Overview	89
Common Structures	90
Address_V1	90
AutoAppliedOption_V1	91
BusinessHours_V1	93
CEAConstraints_V1	93
Charge_V1	94
ChargeOverride_V1.....	94
ComponentType_V1.....	95
ComponentTypeGroup_V1	96

Contact_V1 (Person).....	96
ContactInfo_V1.....	97
CustCarrFrhtAudit_V1.....	97
CustShpgLocXRef_V1.....	98
GeoArea_V1.....	98
DomicileEquipment_V1.....	99
ExternalAlias_V1.....	99
Holiday_V1.....	100
LaneAvailability_V1.....	100
Memo_V1.....	101
NamedResult.....	101
PayableCarrOvr_V1.....	101
POD_V1.....	102
RefNumber_V1.....	102
RspbCustOvr_V1.....	102
SEC_V1.....	103
ShippingInfo_V1.....	104
StopEvent_V1.....	104
UMsr_V1.....	104
Delivery Schedule Structures.....	105
BusinessDays_V1.....	105
FixedPntTimeTable_V1.....	105
DayWeekPntTimeTable_V1.....	106
ElapsedPntTimeTable_V1.....	106
ItrnTimeTableEntry_V1.....	107
ItineraryPoint_V1.....	107
ItrnTimeTable_V1.....	108
Itinerary_V1.....	108
DeliverySchedule_V1.....	109
ItrnLanePerformance_V1.....	110
Perf_V1.....	110
Entity Structures.....	111
Carrier_V1.....	111
Consignee_V1.....	115
Customer_V1.....	118
DistributionCenter_V1.....	124
Domicile_V1.....	127
EquipmentType_V1.....	128
HarmTariff_V1.....	129
Hub_V1.....	130
INCOTerms_V1.....	133
ItemGroup_V1.....	133
ItemMaster_V1.....	134
LoadAt_V1.....	135
Zone_V1.....	138
Entity Structures - CIS.....	139

CXX_CarrierXRef_V1	139
CXX_ComponentTypeXRef_V1	139
CXX_CustomerXRef_V1	139
CXX_CustomerLocationXRef_V1	139
CXX_EquipmentTypeXRef_V1	140
CXX_RefNumberTypeXRef_V1	140
CXX_RetrieveTMEntityId_V1	140
CXX_ServiceXRef_V1	140
Financial Structures	141
APTransaction_V1	141
ARTransaction_V1	142
FreightBill_V1	143
FreightBillDetail_V1	145
FreightBillDetailCharge_V1	146
GLTransaction_V1	147
Load Structures	149
Load_V1	149
LoadBuildPlan_V1	153
LoadConfirmData_V1	154
LoadCreateData_V1	154
LoadInstruction_V1 Structure	155
LoadManifestData_V1	155
LoadPlan_V1	156
LoadSelectionCriteria_V1	156
LoadSpotRateData_V1	156
LoadTenderContact_V1	157
LoadTenderData_V1	157
LoadUpdateProgress_V1	159
NotificationData_V1	159
ShipmentKeyInfo_V1	160
ShipmentLeg_V1	160
ShipmentLegHeader_V1	163
ShipmentUpdateProgress_V1	164
Stop_V1	164
StopConfirmData_V1	165
StopUpdateProgress_V1	166
Non-Operational Freight Structures	166
NonOperationalFreight_V1	166
RatingInfo_V1	170
NOFWeightByFreightClass_V1	171
NOF_Stop_V1	171
Rate Quotation Structures	172
ChargeCriteria_V1	172
ChargeDetail_V1	173
FreightDetail_V1	174
RateCriteria_V1	174

Schedule_V1	177
StopSchedule_V1	177
ValidLane_V1	177
Shipment Structures.....	178
CompTrackingNum_V1.....	178
SelectionCriteria_V1	178
ShipmentCompChangeData_V1	179
ShipmentConfirmData_V1.....	180
Shipment Order Entry Structures	181
Container_V1	181
ContainerPlan_V1	183
NMFC_V1.....	184
PlanningRequest_V1	184
Shipment_V2.....	185
ShipmentHeader_V2	189
ShipmentPlan_V1.....	191
ShipmentThruPoint_V1	191
Tariff Structures.....	191
FC_Crs_Rfrc_V1	191
LaneAssc_V1	191
Rate_V1.....	192
RateCd_V1	193
RngRate_V1	194
SchgChrg_V1	194
SchgRate_V1.....	194
SchgSrvc_V1.....	195
Rstc_V1	195
Tariff_V1.....	196
TffChrg_V1.....	199
TffSrvc_V1.....	203
TffSrvcEqmt_V1	205
Transport Order Structures.....	206
Specifying Origin, Destination and Throughpoints	206
Consolidating Multiple Components	207
Consolidating Multiple Shipments.....	207
CntrOrtn_V1.....	208
Component_V1	209
CompRefNumber_V1	211
ItineraryStep_V1	211
Shipment_V1.....	212
ShipmentItem_V1	214
TransportOrder_V1	216
WeightByFreightClass_V1	219
5 Flat File Driver.....	221
Overview	221

Terms and Definitions	221
System Interfaces	222
Flat File Driver Command Line Options	222
Using the Flat File driver	225
Functions and Facilities	226
Common Services	226
Flat File Structures	226
Flat File Driver Daemon	227
Flat File Driver Daemon Command Line Options	227
6 Flat File Driver API Services	229
Address Service	229
Delivery Schedule Services	229
Entity Services	235
Data Structure Contents	235
Data Transfer	236
Destination of Retrieved Data and Errors	236
Create	237
Delete	246
Remove	247
Retrieve	247
Set Status	249
Update	249
Financial Services	252
Examples	252
Load Services	258
Data Structure	258
Data Transfer	259
Examples	259
Rate Quotation Services	267
Examples	267
Shipment Services	268
Examples	268
Shipment Order Entry Services	271
Examples	271
Tariff Services	276
Examples	276
Tariff Entity Services – Service Equipment	277
Examples	277
Transport Order Services	278
Data Structures	279
Data Transfer	279
Examples	280
7 CORBA API	283
Overview	283

CORBA Program Development.....	284
Running A CORBA Program.....	284
Venture Factory.....	285
Venture Session.....	285
8 XML API.....	287
XML Overview.....	287
XML API Server.....	287
Process Overview.....	287
Naming Procedure for XML Elements.....	288
XML Server Parameters.....	289
XML Parser.....	289
Document Type Definitions (DTDs).....	289
DTD Directories.....	290
XML APIs.....	292
XML Client.....	294
Overview of SSL.....	294
XML API Connection Settings.....	295
XMLClient.jar.....	295
9 API Error Messages.....	297
A Sample C++ API Client.....	333
B XML API File Samples.....	337
Create Customer XML File.....	337
Retrieve Customer XML File.....	339
Create Shipment XML File.....	339
Sample Java Program.....	341
C API Changes for VisiBroker Support.....	345
API Server Port from Orbix to VisiBroker.....	345
API Server Parameter Set Entries.....	345
Starting the VisiBroker Name Server.....	346
Flat File Driver Command Line Parameters.....	346
Flat File Driver Daemon Parameters.....	347
Changes to Custom CORBA Clients.....	348
Replacing the _bind() call in an Orbix 3.0.1 client.....	348
Porting an Orbix 3.0.1 client to VisiBroker 4.1 for C++.....	349
Connecting from a VisiBroker client using the IOR String File.....	351
Connecting from a VisiBroker client using the Naming Service.....	352
Porting an OrbixWeb 3.2 client to VisiBroker 4.1 for Java.....	353
Connecting from a VisiBroker for Java client using the IOR String File.....	353
Connecting from a VisiBroker for Java client using the Naming Service.....	354
Changes to the Java API Client Library.....	355

Index357

Chapter 1

Introduction

Welcome to the Application Program Interface (API) Reference for Transportation Manager.

This manual is for advanced programmers and developers who need to integrate Transportation Manager data with their other systems. A general understanding of Transportation Manager is recommended.

Overview of API

An API is a set of routines, protocols, and tools for building software applications. It exchanges data between different programs.

The following API integration methods are available for Transportation Manager:

Flat File Driver Interface

This is an extension to the CORBA object interface. The flat file is a standard ASCII text file containing the data. You do not need to know CORBA programming to use this interface.

The flat file driver interface is described in:

- [“Flat File Driver” on page 221](#)
- [“Flat File Driver API Services” on page 229](#)

CORBA Object Interface

This uses Common Object Request Broker Architecture (CORBA) client programs which interface with API server objects.

CORBA was developed by the Object Management Group industry consortium. It is an architecture that enables various program objects to communicate with one another, independent from the programming language and operating system. CORBA is designed to work in object-oriented environments. It automates many common network programming tasks such as object registration, location, and enabling.

A program requests objects through an object request broker, or ORB, Information about the structure of the program from where the object comes is not required.

The CORBA interface is described in:

- [“CORBA API” on page 283](#)
- [“Sample C++ API Client” on page 333](#)

Items common to both CORBA API and the flat file driver are described in:

- [“API Elements” on page 17](#)
- [“API Services” on page 29](#)
- [“API Structures” on page 89](#)

XML

The Extensible Markup Language (XML) API server uses XML to exchange information between the server and its clients.

The XML API is described in:

- [“XML API” on page 287](#)
- [“XML API File Samples” on page 337](#)

API Changes

This section describes the changes to the API Reference Manual since the last version.

Structure Changes

CIS structures have been added in this version: refer to [“Entity Structures - CIS” on page 139](#).

The following fields have been added to the current API structures since the last version.

- **Carrier_V1:** Extl_cd1, Extl_cd2, IgnoreLaneAvailabilities, LaneAvailabilities
- **ComponentType_V1:** Actv_yn, Extl_cd
- **Consignee_V1:** Extl_cd1, Extl_cd2
- **Customer_V1:** Extl_cd1, Extl_cd2
- **DistributionCenter_V1:** Extl_cd1, Extl_cd2
- **EquipmentType_V1:** Extl_cd
- **Hub_V1:** Extl_cd1, Extl_cd2
- **Load_V1:** Tff_Id, Rate_Cd
- **LoadAt_V1:** Extl_cd1, Extl_cd2
- **POD_V1:** Extl_cd
- **StopConfirmData_V1:** Shpg_Loc_cd, Shpg_Loc_Typ_enu, Stop_Seq_Num

The following fields have been removed from the API structures.

- **Carrier_V1:** EDI_Capb
- **ComponentType_V1:** EDI_Pck_cd_typ
- **EquipmentType_V1:** EDI_Eqmt_typ
- **POD_V1:** Rsn_Cd_typ

All hour fields have been expanded one decimal place to `VIS::num6_2`. They were previously `VIS::num5_2`. These fields are all those ending in “_hrs” and the `Lnst_Stop_Wait_Tm` field in `NonOperationalFreight_V1`.

Service Changes

A new service has been added that uses the new CIS structures: refer to [“void CEntityProcessor::RetrieveTMEntityIds”](#) on page 36.

Parameter Changes

A new parameter, `IsService`, has been added to the API Parameter set. For details, refer to [“IsService”](#) on page 25.

Additional Information

For more information on Transportation Manager refer to the *Release Notes*. The *Release Notes* include:

- Related Documentation
- If You Need Assistance

Chapter 2

API Elements

This chapter describes the elements that apply to both the CORBA API and the flat file driver. It includes the following topics:

- [System Overview](#)
- [Using Transportation Manager API Services](#)
- [Domain Table Type Map](#)

System Overview

The flat file driver interface is an extension of the CORBA API and provides the same services as the CORBA API. The flat file interface resides on top of the CORBA API, making calls directly to it.

You can create, retrieve, update, and delete data for most services, for both the flat file driver and CORBA API. To do these operations, you define the appropriate structures and commands within the input files. The file format creates complex data structures that are passed to the Transportation Manager database during an operation.

With the flat file driver, each input file for the Transportation Manager database has a separate operational session. However, each session can have many transactions. For example, one input file can have all of the load-at data and can populate Transportation Manager using one or more transactions.

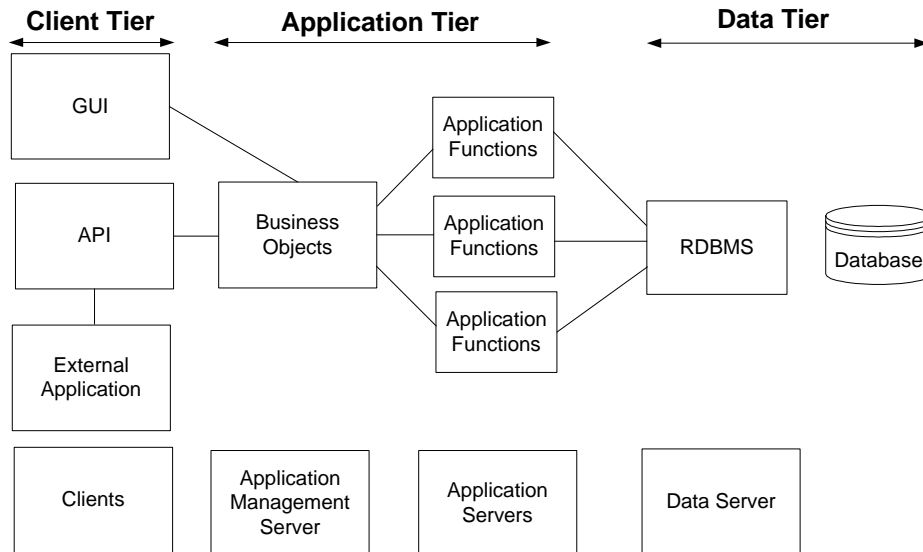
Flat file operations are equivalent to CORBA API operations. These include sending retrieved data to output files and sending transaction exceptions to log files. The output files have data which maps to data requests from an input file. Both files have the same data format.

Enumerated Types

To see the enumerated types that are common for all entities, view the file `VISTypes.idl`.

Architecture Overview

The following chart gives an overview of the Transportation Manager integration architecture.



Using Transportation Manager API Services

The following general concepts apply to all Transportation Manager API services. For a detailed description of each service, refer to “API Services” on page 29.

Standard Data Types

The integration service is called VIS. The VIS modules described here define the standard data types. All data passed into and returned from the Transportation Manager API services use these data types.

An empty string (“”) represents a null value, except for `VIS::vbool`.

VIS::vbool

This represents boolean values. Its values are:

- `VIS::bTRUE`
- `VIS::bFALSE`
- `VIS::bNULL`

VIS::date

This represents a date value as a string in the form `MM/DD/YYYY`. For example, `10/02/1997`. The year must be within the range 1800 to 2055.

VIS::time

This represents a 24 hour time value as a string in the form HH:MM:SS. For example, 10:15:00. The minimum and maximum time values are 00:00:00 and 23:59:59.

VIS::timestamp

This represents a timestamp value as a string in the form MM/DD/YYYY@HH:MM:SS. For example, 10/02/1997@10:15:00.

VIS::num<s>_<p>

This represents a fixed point numeric value with a precision of <p> and a scale of <s>. A string value represents the data type. For example, the following list of legal values is for the type VIS::num5_2: 100, 100.23, 0.23, -100.2, or -0.03.

Do not store values greater than the scale and precision you have set, because this will generate a data overflow error.

VIS_V1::t<domain table>

Within Transportation Manager, a domain table defines string values that represent domain values for a particular type. For example, the Carrier Type domain table has a list of different carrier types such as Less Than Truckload (LTL), Truckload (TL), and Rail. (LTL is a freight service used to move a shipment or group of shipments that do not require a full truck's capacity.)

Types of the form t<domain table> represent a domain table. The possible values depend on the table setup in Transportation Manager. For a map of the internal domain table names used by the Transportation Manager API services and those in the Graphical User Interface (GUI) setup applet, refer to the "Domain Table Type Map" on page 23.

Creating Objects

When you create objects in Transportation Manager, the state (data values) of the object you are creating is passed into the appropriate Create method by using a structure. It reduces the amount of network traffic by sending one request, rather than sending many individual set attribute requests. This helps ensure reasonable performance over the network.

Defaulting Values

Assign the default object attribute values within the Transportation Manager API services. Set the corresponding structure values to the appropriate null value before calling the create method. The following table lists the appropriate null values for each data type.

Type	Null value
VIS::vbool	VIS::bNULL
VIS::date	""

Type	Null value
VIS::time	""
VIS::timestamp	""
VIS::num<p>_<s>	""
VIS_V1::t<domain table>	""
VIS_V1::e<enumeration>	VIS_V1::<tag>_NULL
string	""
long	VIS::NULLlong
short	VIS::NULLshort
unsigned short	VIS::NULLushort

Read-Only Values

Some structure values are read-only. If you set these values before calling a create or update, the value set will be ignored. Ensure you initialize all string values to null values.

Handling Relationships

Most of the objects in Transportation Manager are related to others. Using structures shows these relationships in different ways.

If the relationship is one to one, the owner object has a structure that represents the owned object. For example, a carrier has an associated address object that is stored as a structure member of the `Address_V1` type.

If the relationship is one to many, the parent object has a sequence of structures representing the related objects. For example, a consignee can have one or more contacts. The `Consignee_V1` structure represents this by having a sequence of `Contact_V1` structures.

If an object references a shared object, the ID of the referenced object is stored as a structure element. For example, a carrier can reference a preferred service object. The carrier structure has a string attribute which contains the ID of the service object.

Retrieving Objects

Retrieving objects uses the same structure format as creating objects. The `retrieve` method usually takes a sequence of object identifiers as an argument. It then returns a sequence of structures that represent the state of the returned objects.

Updating Objects

Updating objects uses the same structure format as creating objects. The `update` method usually takes a sequence of object structures representing the new state of the objects to update. The ID attribute of each structure identifies which object in the system to update.

Controlling Updated Attributes

To update specific attributes, provide values only for the attributes you want to update. Set the other attributes to the appropriate null value before calling the update method.

Updating Objects with Other Structures

If an object structure has a sequence of other structures, then you pass all the related structures to update the sequence. For example, a carrier object can refer to two contact objects. To update the carrier by adding a new contact, pass the two current contacts and the new one during the update. Note that if you did not pass any contacts when updating the carrier, this would remove all the contacts from the carrier.

Most sequences have an associated boolean variable, usually `Ignore<Sequence name>`. Use this flag to update an object without resending all the related objects. If this flag is set to `False` during a create or update operation, then the contents of the related collection are ignored.

Doing Block Operations

For most methods, you can do an operation on several objects with just one request. Packaging a set of separate requests into one can increase performance over a network. Also, it treats an operation on a set of objects as a single logical unit of work. This unit either fully completes or it fails. If it fails, then the data will not be changed.

In the event that the transaction fails due to a database deadlock or an optimistic lock failure, the API will retry the transaction up to five times. Each failure and retry is recorded in the API transaction log (if the logging level is set high enough). However, if any of the retries is successful, no error will be reported to the caller (see [Error Handling](#) next). Only if all of the retries fail is the transaction abandoned, and an error message returned to the caller.

To support block operations, most methods use arrays of input or output parameters (usually state structures) as arguments.

Error Handling

There are two different modes of error handling: immediate and deferred.

Immediate Mode

In immediate mode, when you get an error running a method, a `VISError::Immediate` exception is relayed to you. All changes made to Transportation Manager are reversed.

`VISError::Immediate` is defined as:

```
Module VISError {
    exception Immediate
    {
        long item;
        long code;
        string structTag;
        string elementTag;
        string msg;
    }
}
```

```

        string sysMsg;
    };
};

```

Where:

item	The item that caused the error when doing a block operation. It is the zero-based index of the item in the input array.
code	The error code. For error explanations, refer to “API Error Messages” on page 297 .
structTag	The type of object structure that the error relates to, for example, <code>Carrier_v1</code> . This element can be blank if the error is not specific to a certain structure type.
elementTag	The structure element that the error refers to, for example, <code>Pstl_Code</code> . This element can be blank if the error is not specific to a given structure element.
msg	A description of the error.
sysMsg	A description of the system error, used mostly in debugging.

When using immediate mode, only situations in which the requested operation does not finish will cause an error. It will not give you minor warnings.

Deferred mode

This mode gives much greater control over error handling. You can better determine the errors that occur during an operation.

This mode reports the following severity levels of errors:

Warning	A low severity error or an event of interest that does not affect the outcome of a requested operation.
Recoverable	A more serious error than <i>Warning</i> . The operation succeeds, however the status of the object is set to indicate that you need to do a manual correction in Transportation Manager.
Serious	A more serious error than <i>Recoverable</i> . The operation stops completely, but other errors can still be reported.
Fatal	The current operation completely stops and no more errors can be reported.

Deferred mode errors and warnings are returned as a sequence of `VISError::Detail` structures. If the size of the returned sequence is zero, it means no errors or warnings have occurred. If one or more errors have occurred, the following information will be returned for each error:

```

Module VISError {
    struct Details{
        long item;
        eSeverity severity;
        long code;
        string structTag;
    };
};

```

```

        string attrTag;
        string msg;
        string sysMsg;
    };
};

```

For an explanation of the structure elements, refer to “Immediate Mode” on page 21.

In deferred error mode, the system can report multiple errors for the same item. More than one `VISError::Detail` structure appearing with the same item number indicates multiple errors.

Domain Table Type Map

The following table lists the corresponding domain tables in the setup applet to which each structure type maps. The structure types are defined in `VISTypes.idl`.

Type	Domain Table in Setup
tAPInvoiceFormat	A/P invoice print formats
tAPterms	A/P terms
tARInvoiceFormat	A/R invoice print formats
tBarcodeType	bar code type
tBOLFormat	bill of lading print formats
tCancelReasonCode	cancel reason code
tCarrierType	carrier types
tCarrMnftFmt	carrier manifest print formats
tConsigneeGroup	consignee groups
tCostCenter	cost center
tCreditTerm	credit terms
tCurrency	currency codes
tCustSrvRep	customer service representatives
tGLCategory	G/L category
tGLType	G/L type
tHoldReasonCode	hold reason code
tInvoiceType	invoice type
tLanguage	language codes
tMinorityGroup	minority groups
tProfitCenter	profit centers
tRateShopReasonCode	rate shop reason code

Type	Domain Table in Setup
tReferenceNumType	reference number qualifier type
tRole	contact types
tSCAC	sCAC
tShipLabelFormat	shipment label print formats

API Parameter Set

The database connection information is stored in the `dsc.ini` file.

`-ps` is the parameter set to load, relative to the database. All the other API server parameters are loaded from this parameter set.

The parameter set type of an API server parameter set must be “`APISrv`”. The API server usually uses the parameter set “`API`”.

As with other Transportation Manager servers, the Secant DSC must be running when the API server is started, to allow access to the Routing and Rating Engine and the Distance Engine. That is, the API server is a VisiBroker server, and a client of the Secant servers.

Parameter	Description
APOnly	indicates whether to support A/P transactions only do not set this parameter if you will be supporting A/R transactions.
CacheReloadFrequency	the time in seconds between each refresh of the Transportation Manager global cache
ConsoleLog	indicates whether certain Transportation Manager operations will be logged to the console
DefaultDetailLevel	the detail level to use if <code>fcnArea</code> is not listed in the <code>DetailLevels</code> parameter -1 indicates no logging.
DetailLevels	a comma-separated list of <code>fcnArea:detailLevel</code> pairs example: "Common:2,Comm:3"
DistCache	refer to the <code>Dist</code> parameter within the <code>DistSrv</code> parameter set
DistEnable	indicates whether to enable distance calculation "Y" is the recommended setting
DistEngine	refer to the <code>Dist</code> parameter in the <code>DistSrv</code> parameter set
DistOvrPoll	refer to the <code>OvrPoll</code> parameter in the <code>DistSrv</code> parameter set
DistRoute	refer to the <code>Route</code> parameter in the <code>DistSrv</code> parameter set
ErrorMapFile	the path of the file containing Transportation Manager mapping error codes and error messages
ExternalSrvHost	the host running the Secant Distributed Service Coordinator (DSC)

Parameter	Description
FlowControlFeatureOn	indicates whether to accept logs using the flow control feature
IDInNamingService	<p>the identifier (or NameComponent.id field) with which the API Server will identify itself in the naming service</p> <p>if multiple API servers are launched, each can be given a descriptive name so that clients can choose to connect to a specific one</p> <p>this parameter has a default value of "TransportationManager".</p> <p>note that all API Server IORs published to the naming service will have a type (NameComponent.type field) of "VentureFactory", so that a client can browse the naming service for all registered API Server instances</p>
IORFilePath	<p>a full path to a file</p> <p>if this path exists and if the API Server has permission to write this file, then the API Server's IOR will be written to this file in string format</p> <p>an IOR string file should ideally be written to a location on the network where the server and all clients can access it, such as a Windows file system share</p> <p>if there are clients that do not share access to the same file system that the server does, then the IOR string file may be copied to the client machine once, and used repeatedly, since the server's IOR is "long-lived"</p>
IsService	<p>indicates whether the API Server is running as an operating-system service.</p> <p>set this parameter to "Y" only if you are using the Transportation Manager Process Monitor to start the API server.</p> <p>this value causes VisiBroker to keep the API server running when a user logs out of the machine on which it is running</p>
ListenerPort	<p>the TCP/IP port number on which the API Server will listen for client connections</p> <p>instead of allowing the API server to be assigned the next available port number each time it is run, the API server is forced to listen at this user-specified port</p> <p>this creates the "long lived" IOR described above.</p>
LogApplicationId	indicates whether an application ID will be logged
LogCreateTime	indicates whether a creation time error will be logged
LogFcnAreald	indicates whether a function area ID will be logged
LogFileInfo	indicates whether source file information will be logged
LogSessionid	indicates whether a session ID will be logged.
LogThreadId	<p>includes whether a thread ID will be logged</p> <p>use this parameter to trace entries for an individual operation</p>
LogTransactionId	indicates whether an transaction ID will be logged
MaxThreadsInPool	<p>the maximum number of threads in the servant thread pool</p> <p>each thread in this pool can process an API request concurrently</p> <p>this parameter can be set to 0 to use the VisiBroker default</p>

Parameter	Description
MinThreadsInPool	the minimum number of worker threads in the servant thread pool each thread in this pool can process an API request concurrently this parameter can be set to 0 to use the VisiBroker default
NamingServiceHost	holds the TCP/IP hostname of the machine running the VisiBroker naming service
NamingServicePort	the TCP/IP port number of the naming service
OutDBGMsg	indicates whether to display a trace of input structures and output error messages to the API server's console window.
Secant	indicates whether to log to a Secant file
SecantLogLevel	the Secant log level "0" indicates no logging and "5" is the maximum log level level 1 logs SQL queries and errors to enable Transportation Manager logging, refer to DefaultDetailLevel.
SessionList	a comma-separated list of session IDs to be logged an empty list indicates all sessions are logged
TimingFeatureOn	indicates whether to include a timing feature in logs
UseExternalSrv	indicates whether to use external servers for the routing and rating engine if "N", a collocated engine is used

Note: `IDInNamingService`, `NamingServiceHost`, and `NamingServicePort` must be populated for the API Server to register its IOR with this naming service.

VentureFactory IOR

To connect to the API Server, a program must obtain an Interoperable Object Reference (IOR) to the server's `VentureFactory` object. The API Server can write the `VentureFactory` IOR to a file in string format, and it can publish it through the CORBA naming service.

The `IORFilePath` key in the `APISrv` parameter set type controls whether the IOR is written to a file. This key contains the path and filename of this file. If the path exists and the server has write access to this file, then it will write the IOR string to this file.

The `IDInNamingService`, `NamingServiceHost`, and `NamingServicePort` keys in the `APISrv` parameter set type control whether the IOR is published to the naming service.

`NamingServiceHost` identifies the machine running a `NameServer`, and `NamingServicePort` identifies the TCP/IP port that it is listening for connections on.

`IDInNamingService` identifies the `NameComponent.id` field that the API Server's IOR will be bound to. The `NameComponent.type` field will always be

"ventureFactory". If all three of these values are populated, then the API Server will write its IOR to the specified NameServer.

The ventureFactory object has a persistent object reference. This means it has the same object identifier each time the API server is run. This is a useful attribute for a client program that uses the IOR string file to obtain the IOR, but does not have network access to the location where the API server writes the IOR string file. In this case, the file can be copied by any means to the client machine, and reused as long as the API Server setup does not change. Another option for a client that does not have access to the location where the API server writes its IOR file is to use the naming service.

Chapter 3

API Services

This chapter describes the Transportation Manager API services. It includes the following topics:

- [Overview](#)
- [Address Service](#)
- [Delivery Schedule Services](#)
- [Entity Services](#)
- [Financial Services](#)
- [Load Services](#)
- [Load Tender Services](#)
- [Rate Quotation Services](#)
- [Shipment Services](#)
- [Shipment Order Entry Services](#)
- [Tariff Services](#)
- [Transport Order Services](#)

Overview

All Transportation Manager API services are listed alphabetically, with synopses, descriptions, and parameters. For all services:

- the return value is “void”
- the Errors parameter is a sequence of `VISError::Details` structures which describes any errors or warnings

For examples and information about using these services with the flat file driver, refer to “Flat File Driver” on page 221.

Multi-Thread API Processing

The API server can process requests from more than one client at a time. It runs multiple threads on a “thread-per-request” basis. That is, a thread is launched for each API operation. This can improve performance when multiple clients are accessing the API server concurrently.

Address Service

There is only one address service: `AddrSrvc`. Use this service to validate addresses.

`void AddrSrvc::Validate`

Synopsis

```
void Validate(in any AddrList,
             in eAddrChkMode ValidateMode,
             out VISError::Details Errors)
raises(VISError::Immediate);
```

Description Validates the specified addresses.

Parameters

AddrList A sequence of address structures of the type of `Address_V1` to validate. The supported sequence type is `AddressList_V1`.

ValidateMode The validation mode, either `AddrSrvc::POSTAL` or `AddrSrvc::STREET`.

In `POSTAL` mode, the zip/postal code information is assumed to be correct and is validated against the rest of the address.

In `STREET` mode, the street information is assumed to be correct and is validated against the rest of the address.

The errors list is filled if the error mode for the session is set to `VentureSession::Deferred`.

Delivery Schedule Services

Use the delivery schedule services to create, retrieve, update, and delete delivery schedules, and to detach itineraries from timetables.

The error list is filled if the error mode for the session is `VentureSession::Immediate`.

`void DeliveryScheduleSrvc::CreateDeliverySchedule`

Synopsis

```
void CreateDeliverySchedule(in any DlvSchdList,
                          out VISError::Details Errors)
raises(VISError::Immediate);
```

Description Creates delivery schedules.

Parameters

DlvySchdList A sequence of `DeliverySchedule_V1` type entity structures of the type to be created. The supported sequence type is `DeliveryScheduleList_V1`.

void DeliveryScheduleSrcv::CreateDeliveryScheduleEntity**Synopsis**

```
void CreateDeliveryScheduleEntity
  in any DlvySchdEntities,
    out VISError::Details Errors)
raises(VISError::Immediate);
```

Description Creates delivery schedule entities of the specified type.

Parameters

DlvySchdEntityList

A sequence of entity structures of the type of entity to be created. The supported sequence types are: `Itinerary_V1`, `ItnrTimeTable_V1`, `ItnrLanePerformance_V1`.

void DeliveryScheduleSrcv::DeleteDeliverySchedule**Synopsis**

```
void DeleteDeliverySchedule
  (in VIS::StrIdList DlvySchdCdList,
    out VISError::Details Errors)
raises(VISError::Immediate);
```

Description Deletes delivery schedules.

Parameters

DlvySchdCdList A sequence of strings identifying the codes of delivery schedules to delete.

void DeliveryScheduleSrcv::DeleteDeliveryScheduleEntity**Synopsis**

```
void DeleteDeliveryScheduleEntity
  (in any DlvySchdEntities,
    out VISError::Details Errors)
raises(VISError::Immediate);
```

Description Deletes delivery schedule entities.

Parameters

DlvySchdEntities

A sequence of entity structures of the type of entity to be deleted. The supported sequence types are: `Itinerary_V1`, `ItnrTimeTable_V1`, `ItnrLanePerformance_V1`.

void DeliveryScheduleSrcv::DetachAllItineraries

Synopsis

```
void DetachAllItineraries
  (in any DlvvSchdEntities,
   out VISError::Details Errors)
raises(VISError::Immediate);
```

Description Detaches all itineraries of defined timetables. Indicates timetables from which itineraries should be detached.

Parameters

DlvvSchdEntities

A sequence of itinerary timetables codes for delivery schedules. They indicate which itineraries should be detached from these timetables.

void DeliveryScheduleSrcv::RetrieveDeliverySchedule

Synopsis

```
void RetrieveDeliverySchedule
  (in VIS::StrIdList CdList,
   inout any DlvvSchds,
   out VISError::Details Errors)
raises(VISError::Immediate);
```

Description Retrieves delivery schedules.

Parameters

CdList A sequence of strings identifying the codes of delivery schedules to retrieve.

DeliveryScheduleList

A sequence of DeliverySchedule_V1 type delivery schedule structures to be retrieved. The supported sequence type is DeliveryScheduleList_V1.

void DeliveryScheduleSrcv::RetrieveDeliveryScheduleEntity

Synopsis

```
void RetrieveDeliveryScheduleEntity
  (inout any DlvvSchdEntities,
   out VISError::Details Errors)
raises(VISError::Immediate);
```

Description Retrieves delivery schedule entities of the specified type.

Parameters

DlvvSchdEntities

A sequence of entity structures of the type of entity to be retrieved. The supported sequence types are: Itinerary_V1, ItnrTimeTable_V1, ItnrLanePerformance_V1.

void DeliveryScheduleSrcv::UpdateDeliverySchedule

Synopsis

```
void UpdateDeliverySchedule
  (in any DlvSchdList,
   out VISError::Details Errors)
raises(VISError::Immediate);
```

Description Updates delivery schedules.

Parameters

DlvSchdList A sequence of `DeliverySchedule_V1` type delivery schedule structures to be updated. The supported sequence is `DeliveryScheduleList_V1`.

void DeliveryScheduleSrcv::UpdateDeliveryScheduleEntity

Synopsis

```
void UpdateDeliveryScheduleEntity
  (in any DlvSchdEntities,
   out VISError::Details Errors)
raises(VISError::Immediate);
```

Description Updates delivery schedule entities of the specified type.

Parameters

DlvSchdEntities

A sequence of entity structures of the type of entity to be updated. The supported sequence types are: `Itinerary_V1`, `ItnrTimeTable_V1`, `ItnrLanePerformance_V1`.

Entity Services

Use the entity services to create, retrieve, update, and delete entities such as load-ats, carriers, and customers.

The error list is filled if the error mode for the session is `VentureSession::Immediate`.

void EntitySrcv::Create

Synopsis

```
void Create
  (in any EntityList,
   out VISError::Details Errors)
raises(VISError::Immediate);
```

Description Creates the specified entities.

Parameters

EntityList A sequence of entity structures of the entity type to create. The supported sequence types are: `CarrierList_V1`, `ConsigneeList_V1`, `CustomerList_V1`, `DCList_V1`,

EquipmentTypeList_V1, HubList_V1, LoadAtList_V1,
and ZoneList_V1.

void EntitySrvc::Delete

Synopsis

```
void Delete
  (in eEntityType EntityType,
   in VIS::StrIdList EntityList,
   out VISError::Details Errors)
raises(VISError::Immediate);
```

Description Deletes the specified entities.

Parameters

EntityType The type of entity to delete.
EntityList A sequence of strings that identify the objects to delete.

void EntitySrvc::DeleteINCOTerms

Synopsis

```
void DeleteINCOTerms
  ( in anyEntityList,
   inout any outEntityList,
   out VISError::Details Errors)
raises(VISError::Immediate);
```

Description Deletes INCO terms.

Parameters

EntityList A list of INCOTerms_V1 structures. To delete specific INCO terms, ensure the following fields are completed: INCO Terms code, INCO Terms Version, and Division code information.

void EntitySrvc::Retrieve

Synopsis

```
void Retrieve
  (in VIS::StrIdList IdList,
   inout any EntityList,
   out VISError::Details Errors)
raises(VISError::Immediate);
```

Description Retrieves the specified entities.

Parameters

IdList A sequence of strings that identify the objects to retrieve.
EntityList A sequence of entity structures of the entity type to retrieve. The supported sequence types are the same as in EntitySrvc::Create.

void EntitySrvc::RetrieveINCOTerms

Synopsis

```
void RetrieveINCOTerms
  (in any EntityList,
   inout any outEntityList,
   out VISError::Details Errors)
raises(VISError::Immediate);
```

Description Retrieves INCO terms.

Parameters

EntityList A list of INCOTerms_v1 structures. To retrieve specific INCO terms, ensure the following fields are completed: INCO Terms code, INCO Terms Version, and Division code information.

outEntityList A list of INCOTerms_v1 structures: INCOTermsList_v1

void EntitySrvc::RemoveItems

Synopsis

```
void AddItems
  (in eEntityType EntityType,
   in any EntityList,
   out VISError::Details Errors)
raises(VISError::Immediate);
```

Description Removes a component type from a component group or an item master from an item group.

Parameters

EntityType The entity type to be removed from the proper list. This value must be COMPONENT_TYPE or ITEM_MASTER.

EntityList A list of the items to be removed from the appropriate groups.

void EntitySrvc::SetStatusV1

Synopsis

```
void SetStatusV1
  (in eEntityType EntityType,
   in VIS::StrIdList EntityList,
   in VIS_V1::eStatus NewStatus,
   out VISError::Details Errors)
raises(VISError::Immediate);
```

Description Sets the status of the specified entities.

Parameters

EntityType The type of entity for which you set the status. (You cannot set the status of a zone or equipment type.)

EntityList A sequence of strings that identify the objects for which to set the status.

NewStatus The new status for the entities.

void EntitySrvc::Update

Synopsis

```
void Update
  (in any EntityList,
   out VISError::Details Errors)
raises(VISError::Immediate);
```

Description Updates the specified entities.

Parameters

EntityList A sequence of entity structures of the entity type to update. The supported sequence types are the same as in EntitySrvc::Create.

void CEntityProcessor::RetrieveTMEntityIds

This function uses the CIS entity structures listed in [“Entity Structures - CIS” on page 139](#).

Synopsis

```
void RetrieveTMEntityIds
  (inout any entityIds,
   out VISError::Details Errors)
```

Description Returns internal (Transportation Manager) entity codes from supplied external codes.

Parameters

entityIds A sequence of CXX_RetrieveTMEntityId_V1 entity structures.

CIS Structures and Code Extraction

The input for this service is the list of CXX_RetrieveTMEntityId_V1 structures. Each of these structures can contain the following structures

- serviceList of CXX_CarrierXRef_V1
- serviceList of CXX_CustomerXRef_V1
- serviceList of CXX_EquipmentTypeXRef_V1
- serviceList of CXX_ServiceXRef_V1
- serviceList of CXX_RefNumberTypeXRef_V1

The codes are extracted in the following way for these structures.

- CXX_CarrierXRef_V1 – the carrier code is extracted using the supplied SCAC type
- CXX_ServiceXRef_V1 – the service code is extracted using the supplied external code
- CXX_EquipmentTypeXRef_V1 – the equipment type code is extracted using the supplied external code

The codes are extracted in the following way for the `CXX_CustomerXRef_V1` structures.

- The customer code is extracted using the supplied external code.
- If the list of customer shipping location cross-references is provided, then the shipping point location code and location type will be extracted from the “customer external aliases by customer” code and the shipping location external code.
- If the list of component type cross-references is provided, then the component type code will be extracted from the “component types by customer” code and the component types external code.

For each of the `CXX_RefNumberTypeXRef_V1` structures, the reference number type code is extracted using the supplied external code.

Financial Services

Use the financial services to retrieve and commit Transportation Manager G/L, A/P, and A/R transactions. You commit transactions so that they will not be used in future retrievals.

Transactions that have not been processed have a status of `INTERFACE` or `RE_SEND`.

The error list is filled if the error mode for the session is `VentureSession::Immediate`.

void FinancialsSrcv::CancelFrhtBill

Synopsis

```
void CancelFrhtBill (in any FrhtBillList,
    in VIS_V1::eFrhtBillId FrhtBillIdType,
    out VISError::Details Errors)
```

Description Cancels a list of freight bills.

Parameters

FrhtBillList A sequence of input `FreightBill_V1` structures to be cancelled.

FrhtBillIdType An enumerated value which determines how the freight bill should be identified.

Comments

Freight Bill can be identified in several ways:

You can identify the freight bill by specifying:

- the freight bill ID (`FB_ID`)
- the freight bill number (`FB_NUM`)
- the freight bill number and carrier ID (`FB_NUM_CARR`)
- the freight bill reference number (`FB_RFRC`)

If a freight bill has been created manually, it will be deleted as a result of the Cancel operation.

void FinancialsSrcv::CancelFrhtBillDetail

Synopsis

```
void CancelFrhtBillDetail(in any FrhtBillDetlList,  
    out VISError::Details Errors)  
    raises(VISError::Immediate);
```

Description Cancels a list of freight bill details.

Parameters

FrhtBillDetlList A sequence of input `FreightBillDetail_V1` structures to be cancelled.

Comments

Freight Bill Detail can be identified in the same way as in “[void FinancialsSrcv::DeleteNOF](#)” on page 40. If the detail to be cancelled has been created manually, then it will be deleted as a result of Cancel operation.

void FinancialsSrcv::CommitAPTransaction

Synopsis

```
void CommitAPTransaction  
    (in VIS::StrIdList APTranIdList,  
    out VISError::Details Errors)  
    raises(VISError::Immediate);
```

Description Commits A/P transactions by changing their status from `INTERFACE` or `RE_SEND` to `ACCOUNT_PAYABLE`.

Parameters

APTranIdList A sequence of strings that identify the A/P transactions to commit.

void FinancialsSrcv::CommitARTransaction

Synopsis

```
void CommitARTransaction  
    (in VIS::StrIdList ARTranIdList,  
    out VISError::Details Errors)  
    raises(VISError::Immediate);
```

Description Commits A/R transactions by changing their status from `INTERFACE` or `RE_SEND` to `ACCOUNT_RECEIVABLE`.

Parameters

ARTranIdList A sequence of strings that identify the A/R transactions to commit.

void FinancialsSrcv::CommitGLTransaction

Synopsis

```
void CommitGLTransaction  
    (in VIS::StrIdList GLTranIdList,  
    out VISError::Details Errors)  
    raises(VISError::Immediate);
```

Description Commits G/L transactions by changing their status from INTERFACE or RE_SEND to GENERAL_LEDGER.

Parameters

GLTranIdList A sequence of strings that identify the G/L transactions to commit.

void FinancialsSrvc::CreateFrhtBill

Synopsis

```
void CreateFrhtBill(in any FrhtBillList,
    out VIS::StrIdList FrhtBillIdList,
    out VISError::Details Errors)
raises(VISError::Immediate);
```

Description Creates a list of freight bills.

Parameters

FrhtBillList A sequence of FreightBill_V1 structures.

FrhtBillIdList A list of created freight bills IDs.

void FinancialsSrvc::CreateFrhtBillDetail

Synopsis

```
void CreateFrhtBillDetail (in any FrhtBillDetlList,
    out VIS::StrIdList FrhtBillDetlIdList,
    out VISError::Details Errors)
raises(VISError::Immediate);
```

Description Creates a list of freight bill details.

Parameters

FrhtBillDetlList A sequence of FreightBillDetail_V1 structures.

FrhtBillDetlIdList A list of created freight bill details IDs.

void FinancialsSrvc::CreateNOF

Synopsis

```
void CreateNOF
    (in any NOFList,
    out VIS::StrIdListNOFNumList,
    out VISError::Details Errors)
raises(VISError::Immediate);
```

Description Creates one or more non-operational freight entities.

Parameters

NOFList A sequence of NonOperationalFreight_V1 entity structures to create.

NOFNumList A sequence of strings identifying the non-operational freight entities that have been created.

void FinancialsSrcv::DeleteNOF

Synopsis

```
void DeleteNOF
  (in VIS::StrIdList IdList,
   out VISError::Details Errors)
raises(VISError::Immediate);
```

Description Deletes one or more non-operational freight entities.

Parameters

IdList A sequence of strings identifying the codes of non-operational freight entities to delete.

void FinancialsSrcv::ResponsibleCustomerOverride

Synopsis

```
void ResponsibleCustomerOverride
  (in string<30> Non_Op_Frht_Num,
   inout any CustomerOverrideList,
   out VISError::Details Errors)
raises(VISError::Immediate);
```

Description A responsible customer override.

Parameters

Non_Op_Frht_Num

A string that identifies the NOF.

CustomerOverrideList

A list of RspbCustOvr_V1 structures:
RspbCustOvrList_V1.

void FinancialsSrcv::RetrieveAPTransaction

Synopsis

```
void RetrieveAPTransaction
  (inout any APTransactions,
   out VISError::Details Errors)
raises(VISError::Immediate);
```

Description Retrieves a set of A/P transactions that have not been processed.

Parameters

APTransactions

A sequence of APTransaction_V1 (APTransactionList_V1) transactions.

void FinancialsSrcv::RetrieveAPTransactionQty

Synopsis

```
void RetrieveAPTransaction
  (inout any APTransactions,
```



```

    out VISError::Details Errors)
    raises(VISError::Immediate);

```

Description Retrieves a specified quantity of A/P transactions that have not been processed.

Parameters

APTransactions

A sequence of APTransaction_V1 (APTransactionList_V1) transactions.

Qty

The quantity of transactions to retrieve.

void FinancialsSrcv::RetrieveARTransaction

Synopsis

```

void RetrieveARTransaction
    (inout any ARTransactions,
     out VISError::Details Errors)
    raises(VISError::Immediate);

```

Description Retrieves a set of A/R transactions that have not been processed.

Parameters

ARTransactions

A sequence of ARTransaction_V1 (ARTransactionList_V1).

void FinancialsSrcv::RetrieveARTransactionQty

Synopsis

```

void RetrieveARTransaction
    (inout any ARTransactions,
     out VISError::Details Errors)
    raises(VISError::Immediate);

```

Description Retrieves a specified quantity of A/R transactions that have not been processed.

Parameters

ARTransactions

A sequence of ARTransaction_V1 (ARTransactionList_V1).

Qty

The quantity of transactions to retrieve.

void FinancialsSrcv::RetrieveFrhtBill

Synopsis

```

void RetrieveFrhtBill(in any FrhtBillNumList,
    in VIS_V1::eFrhtBillId FrhtBillIdType,
    inout any FrhtBillList,
    out VISError::Details Errors)
    raises(VISError::Immediate);

```

Description Retrieves a list of freight bills

Parameters**FrhtBillNumList**

A sequence of input `FreightBill_V1` structures identifying the freight bills to be retrieved.

FrhtBillIdType An enumerated value which determines how the freight bill should be identified.

FrhtBillList A list of retrieved freight bills.

Comments

You can identify the freight bill by specifying:

- the freight bill ID (`FB_ID`)
- the freight bill number (`FB_NUM`)
- the freight bill number and carrier ID (`FB_NUM_CARR`)
- the freight bill reference number (`FB_RFRC`)

void FinancialsSrcv::RetrieveFrhtBillDetail**Synopsis**

```
void RetrieveFrhtBillDetail(in any inFrhtBillDetlList,  
    inout any outFrhtBillDetlList,  
        out VISerror::Details Errors)  
    raises(VISerror::Immediate);
```

Description Retrieves a list of freight bill details.

Parameters**inFrhtBillDetlList**

A sequence of input `FreightBillDetail_V1` structures identifying the freight bill details to be retrieved.

outFrhtBillDetlList

A list of retrieved Freight Bill Details.

Comments

The freight bill detail to be retrieved can be identified in a following ways:

- if freight bill detail ID is defined, then the specified detail is retrieved
- if a freight bill detail sequential number is defined, then the API identifies the freight bill, if possible, by either the freight bill ID or freight bill number, and then retrieves the freight bill detail by its sequential number.

If the data is ambiguous, for example, the freight bill detail object can be identified more than one way, but each way points to a different object, then the retrieve operation fails and an error message is displayed.

void FinancialsSrvc::RetrieveGLTransaction

Synopsis

```
void RetrieveGLTransaction
(inout any GLTransactions,
 in VIS_V1::eGLTrnsType  GL_Trns_Typ_enu,
 in VIS_V1::eGLClsc  GL_Clsc_enu,
 in string<12> Frht_Bill_Num,
 in string<8> Carr_Id,
 in string<12> Invc_Num,
 in string<12> Cust_Id,
 out VISError::Details Errors)
raises(VISError::Immediate);
```

Description Retrieves a set of G/L transactions that have not been processed.

Parameters

GLTransactions

A sequence of GLTransaction_V1 (GLTransactionList_V1) structures.

The following parameters are optional. The API service will select the transactions based on the parameters you include.

GL_Trns_Typ_enu

The G/L transaction type.

GL_Clsc_enu The G/L classification.

Frht_Bill_Num The freight bill number. (A carrier issues a freight bill to request payment for freight services.)

Carr_Id The carrier ID.

Invc_Num The invoice number.

Cust_Id The customer ID.

void FinancialsSrvc::RetrieveGLTransactionQty

Synopsis

```
void RetrieveGLTransaction
(inout any GLTransactions,
 in VIS_V1::eGLTrnsType  GL_Trns_Typ_enu,
 in VIS_V1::eGLClsc  GL_Clsc_enu,
 in string<12> Frht_Bill_Num,
 in string<8> Carr_Id,
 in string<12> Invc_Num,
 in string<12> Cust_Id,
 out VISError::Details Errors)
raises(VISError::Immediate);
```

Description Retrieves a specified quantity of G/L transactions that have not been processed.

Parameters**GLTransactions**

A sequence of `GLTransaction_V1` (`GLTransactionList_V1`) structures.

The following parameters are optional. The API service will select the transactions based on the parameters you include.

GL_Trns_Typ_enu

The G/L transaction type.

GL_Clsc_enu The G/L classification.

Frht_Bill_Num The freight bill number. (A carrier issues a freight bill to request payment for freight services.)

Carr_Id The carrier ID.

Inv Num The invoice number.

Cust_Id The customer ID.

Qty The quantity of transactions to retrieve.

void FinancialsSrvc::RetrieveNOF**Synopsis**

```
void RetrieveNOF
    (in VIS::StrIdList IdList,
     inout any NOFList,
     out VISError::Details Errors)
raises(VISError::Immediate);
```

Description Retrieves one or more non-operational freight entities.

Parameters

IdList A sequence of strings identifying the numbers of non-operational freight entities to retrieve.

NOFList A sequence of `NonOperationalFreight_V1` entity structures to retrieve.

void FinancialsSrvc::RetrieveResponsibleCustomerOverride**Synopsis**

```
void RetrieveResponsibleCustomerOverride
    (in string<30> Non_Op_Frht_Num,
     inout any CustomerOverrideList,
     out VISError::Details Errors)
raises(VISError::Immediate);
```

Description Retrieves responsible customer overrides.

Parameters

Non_Op_Frht_Num

A string that identifies the NOF.

CustomerOverrideList

A list of `RspbCustOvr_V1` structures: `RspbCustOvrList_V1`.

void FinancialsSrvc::UpdateFrhtBill**Synopsis**

```
void UpdateFrhtBill (in any FrhtBillList,
    in VIS_V1::eFrhtBillId FrhtBillIdType,
    out VISError::Details Errors)
```

Description Updates a list of freight bills.

Parameters

FrhtBillList A sequence of input `FreightBill_V1` structures to be updated.

FrhtBillIdType An enumerated value which determines how the freight bill should be identified.

Comments

You can identify the freight bill by specifying:

- the freight bill ID (`FB_ID`)
- the freight bill number (`FB_NUM`)
- the freight bill number and carrier ID (`FB_NUM_CARR`)
- the freight bill reference number (`FB_RFRC`)

void FinancialsSrvc::UpdateFrhtBillDetail**Synopsis**

```
void UpdateFrhtBillDetail(in any FrhtBillDetlList,
    out VISError::Details Errors)
raises(VISError::Immediate);
```

Description Updates a list of freight bill details.

Parameters

FrhtBillDetlList A sequence of input `FreightBill_V1` structures to be updated.

Comments

Freight Bill Details can be identified in the same way as in [“void FinancialsSrvc::DeleteNOF”](#) on page 40.

void FinancialsSrvc::UpdateNOF**Synopsis**

```
void UpdateNOF
    (in any NOFList,
    out VISError::Details Errors)
raises(VISError::Immediate);
```

Description Updates one or more non-operational freight entities.

Parameters

NOFList A sequence of `NonOperationalFreight_V1` entity structures to update.

Load Services

A load is a freight service provided by a carrier to move goods from certain points to other points. It is made up of shipment legs. A shipment leg is the route that shipments in a load take when moving from one stop or point to another. The stops are the shipping locations that make up the load.

Use load services to do any of the following:

- retrieve loads, shipment legs, and stops
- confirm and modify the status of loads
- create and retrieve load build plans
- accept, reject, and cancel load tenders
- manipulate components within transport orders
- manipulate shipments
- create a plan ID for creating transport orders

For more information about creating transport orders, refer to “Transport Order Services” on page 84.

As part of the load building process, you can use the plan ID as a marker for either manual or automatic (optimized) loads. By providing the means to create a plan ID, you can retrieve planned loads or planned shipments after load building. You can also retrieve individual loads and trigger status changes of loads based only on load IDs.

void LoadSvc::AssignToLoad

Synopsis

```
void AssignToLoad  
(in VIS::StrIdList ShipmentLegIdList,  
    in VIS::num28_0LoadId,  
    in booleanDiscardConflictingAppt_yn,  
    out VISError::DetailsErrors);
```

Description Attaches the set of shipment legs to the specified load.

Parameters

ShipmentLegIdList

A sequence of shipment leg IDs to be assigned to the load.

LoadId The ID of the load.

DiscardConflictingAppt_yn

If True, any conflicting appointments are discarded.

void LoadSrvc::AssignToNewLoad

Synopsis

```
void AssignToNewLoad
(in VIS::StrIdList ShipmentLegIdList,
 in string PlanId,
 out VISError::Details Errors)
raises(VISError::Immediate);
```

Description Attaches each of the specified shipment legs to a new load. If the list is empty, then each of the shipment legs that are being attached to the specified plan will be assigned to a new load.

Parameters

ShipmentLegIdList

A sequence of strings of shipment leg IDs to be assigned to a new load.

PlanId

The plan ID from which to select the shipment legs.

void LoadSrvc::CreatePlan

Synopsis

```
void CreatePlan
(in any Plan,
 out VIS::StrIdList PlanIdList,
 out VISError::Details Errors)
raises(VISError::Immediate);
```

Description Creates a load build plan.

Parameters

Plan

A LoadBuildPlan_V1 structure describing the load build plan to create.

PlanId

The new plan ID assigned by the system.

void LoadSrvc::DisableAssignToTrip

Synopsis

```
void DisableAssignToTrip
(in VIS::StrIdList LoadIdList,
 out VISError::Details Errors)
raises(VISError::Immediate);
```

Description Renders each specified load ineligible to be assigned to a trip.

Parameters

LoadIdList

A sequence of Load IDs. This operation will try to stop each load from being able to be assigned to a trip. If any of the loads cannot be stopped, none are. That is, the operation succeeds for all of the loads in the list, or for none of them.

void LoadSrvc::DisableTripContinuation

Synopsis

```
void DisableTripContinuation
  (in VIS::StrIdList TripIdList,
   out VISError::Details Errors)
raises(VISError::Immediate);
```

Description Renders each specified trip ineligible to continue.

Parameters

TripIdList A sequence of trip IDs. This operation will try to stop each trip from continuing. If any of the trips cannot be stopped, none are. That is, the operation succeeds for all of the trips in the list, or for none of them.

void LoadSrvc::LoadConfirm

Synopsis

```
void LoadConfirm
  (in any LoadConfirmData,
   out VISError::Details Errors)
raises(VISError::Immediate);
```

Description Confirms the specified load.

Parameters

LoadConfirmData

A structure of type `LoadConfirmData_V1` that identifies the load to confirm.

Note: You must specify only the SEC code, trailer number, seal number and driver SEC fields. If you specify any other SEC field, a warning message will appear and the value will be ignored.

void LoadSrvc::LoadCreate

Synopsis

```
void LoadCreate
  (in any LoadCreateDataList,
   out VIS::StrIdList LoadIdList,
   out VISError::Details Errors)
raises(VISError::Immediate);
```

Description Creates loads. `Plan_Id` is the only `LoadCreateData_V1` field that is mandatory. If you do not specify a load ID, the system will create a new load and assign it a new ID. You can create a load with a specific ID, but must be in the range of 80000000 to 99999999.

Parameters

LoadCreateDataList

A list of `LoadCreateData_V1` structures which define new loads to be created.

LoadIdList A sequence of created load IDs.

void LoadSrvc::LoadManifest

Synopsis

```
void LoadManifest
  (in any LoadManifestDataList,
   out VISError::Details Errors)
raises(VISError::Immediate);
```

Description Manifests loads. A manifest is a grouping of shipment legs being sent from a single shipping point to other locations using a single carrier.

Parameters

LoadManifestdataList

A sequence of LoadManifestData_V1 structures to be manifested.
The supported sequence type is LoadManifestDataList_V1.

void LoadSrvc::LoadSpotRate

Synopsis

```
void LoadSpotRate
  (in any LoadSpotRateDataList,
   out VISError::Details Errors)
raises(VISError::Immediate);
```

Description Applies a specified spot rate to a load.

Parameters

LoadSpotRateDataList

A sequence of LoadSpotRateData_V1 structures that contain data required to apply a spot rate to a load.

The supported sequence type is LoadSpotRateDataList_V1.

void LoadSrvc::LoadUpdateProgress

Synopsis

```
void LoadUpdateProgress(in any LoadUpdateProgressList,
  in boolean IdsAreRefNums,
  in tRefNumType RefNumType,
  out VISError::Details Errors)
raises (VISError::Immediate);
```

Description Sends a list of events that describe the progress of a load.

Parameters

LoadUpdateProgressList

A sequence of LoadUpdateProgress_V1 structures.

IdsAreRefNumsIndicates whether loads should be identified by their reference numbers instead of their ID numbers.

RefNumType The type of reference numbers used to identify the loads.

Notes

You can identify the load by specifying:

- the load ID (Load_Id)
- the Load Tracking Number (Load_Desc)
- the Master BOL (MBOL_Num)

You can also identify the load by the load reference number. If so, IdsAreRefNums should be set to True and RefNumType must be defined. The reference number itself should be defined in LoadUpdateProgress_V1 > Load_Id.

WARNING! Only specify **one** these four fields. If you specify more than one field, the service call will fail.

void LoadSrcv::PayableCarrierOverride

Synopsis

```
void PayableCarrierOverride
  (in VIS::num28_0 Load_Id,
   in any CarrierOverrideList,
   out VISError::Details Errors)
  raises(VISError::Immediate);
```

Description A payable carrier override.

Parameters

Load_ID A string that identifies the load.

CarrierOverrideList

A list of PayableCarrOvr_V1 structures:
PayableCarrOvrList_V1.

void LoadSrcv::PreBuiltLoadCreate

Synopsis

```
void PreBuiltLoadCreate(in any ShipmentList,
  in any LoadCreateData,
  in VIS_V1::ePreBuiltLoadStatus LoadStatus,
  in Boolean EnableLoadLevelNotification,
  in VIS_V1::eResultContents ResultContents,
  out VIS::num28_0 Load_Id,
  inout any OutShipmentList,
  out VISError::Details Errors)
```

Description

Creates loads and shipments, assigns shipments to the load, changes the load to a specified status.

Parameters

ShipmentList A sequence of `Shipment_V2` entity structures to be created. The supported sequence types is `ShipmentList_V2`.

LoadCreateData

The data for the load to be created. The supported structure is `LoadCreateData_V1`.

LoadStatus The status in which the load should be advanced. Valid values are: `LDS_OPEN`, `LDS_PLANNED`, and `LDS_TENDER_ACCEPTED`. The default value is `LDS_PLANNED`.

EnableLoadLevelNotification

If `False`, event notification messages will not be sent. The default value is `True`.

ResultContents This parameter allows clients to specify the output type. The possible values are

`RES_NULL` and `RES_NONE` – only IDs of new shipments will be sent back to the client

`RES_SHIPMENT_HEADER` – partial information about created shipments (shipment headers) will be sent to the client

`RES_SHIPMENT_FULL` – complete information about created shipments is produced

`RES_SHIPMENTLEG_HEADER` – partial information (shipment leg header) about shipment legs for the created shipments is produced

Load_Id A system identifier of the new load.

OutShipmentList

A sequence of output structures for the created shipments. The output depends on the `Result Contents` parameter:

`RES_NULL`, `RES_NONE` – the output will be a list of shipment IDs

`RES_SHIPMENT_HEADER` – the output will be a list of `ShipmentHeader_V2` structures

`RES_SHIPMENT_FULL` – the output will be a list of `Shipment_V2` structures

`RES_SHIPMENTLEG_HEADER` – the output will be a list of `ShipmentLegHeader_V1` structures

Restrictions

This API has the following restrictions:

- the plan for a new load should be created in advance
- each shipment sent as part of this API must have exactly one shipment leg, otherwise this API will fail.

- each shipment sent as part of this API is treated as a new shipment; it is not related back to existing shipments in Transportation Manager: only the Create Shipment API will be triggered for these shipments.
- the stops are built in Transportation Manager according to its rules
- the load will be routed and rated in Transportation Manager
- the plans under which all loads using this API are created must already be in Transportation Manager and reference the same division as all the Shipments sent as part of the same request
- all shipments from one source may have the same division; therefore only one plan referring that division must be created in advance in Transportation Manager, either manually or through an API

Restricted Functions and Options

Do not use the following functions and options in Transportation Manager (either manually or with API) with the LoadBuild API:

- trips
- integrated versions of Transportation Optimizer
- splitting shipment legs belonging to shipments that are part of this API
- assigning new shipment legs to a load created with this API
- updating load or their stops
- updating, cancelling, and deleting shipments, including itineraries and reference numbers
- creating, updating and deleting appointments

WARNING! If you use any of these functions, there will be discrepancies of the entity profiles in Transportation Manager, Transportation Optimizer and in the order management system.

void LoadSrcv::PreBuiltLoadDelete

```
void PreBuiltLoadDelete
    (in VIS::StrIdList LoadIdList,
     out VISError::Details Errors)
raises (VISError::Immediate);
```

Description

This API does the following sequence of operations:

- cancels the tender if the operational status is Tendered or Tender Accepted
- stops the auto-tender process if it is enabled
- retrieves the list of associated shipment legs
- cancels the load
- deletes the corresponding shipment for each associated shipment leg

This API does not look for a plan ID for loads that are in Open status. It allows these loads to be deleted regardless of whether they are in a plan.

Parameters

LoadIdList A list of load IDs to delete.

Restrictions

This API has the following restrictions:

- the load must be a built load and cannot be part of a trip
- each shipment on the load can have only one shipment leg
- the load must not have been included as part of an optimization request
- the load must be in one of the following operational statuses: Open/Planned/Tender/Tender Accepted/Tender Rejected.
- if the load is in Tender or Tender Accepted status, then the tender will be canceled; if the cancellation is successful, then the Delete operation can proceed

void LoadSrv::RemoveShipment

Synopsis

```
void RemoveShipment
  (in any ShipmentKeyInfoList,
   out VISError::Details Errors)
  raises(VISError::Immediate);
```

Description Removes shipments from specified loads.

Parameters

RemoveShipmentList

A sequence of `ShipmentKeyInfo` structures identifying the shipments to remove.

void LoadSrv::RemoveShipmentLeg

Synopsis

```
void RemoveShipmentLeg
  (in VIS::StrIdList
   ShipmentLegIdList,
   out VISError::Details Errors)
  raises(VISError::Immediate);
```

Description Removes shipment legs from specified loads.

Parameters

RemoveShipmentLegList

A sequence of shipment leg IDs of the shipments to remove.

void LoadSrcv::RetrieveCondensedLoad

Synopsis

```
void RetrieveCondensedLoad
  (in VIS::StrIdList LoadIdList,
   in boolean IdsAreRefNums,
   in tRefNumType RefNumType,
   inout any LoadList,
   out VISError::Details Errors)
raises(VISError::Immediate);
```

Description Retrieves the shipper's view of a load. After a customer places a shipping order with a carrier, the logistics provider may insert additional stops along the journey not relevant to the shipper's pickup and drop-off points. With the condensed load option enabled, the shipper can see only the portion of the load relevant to them.

Parameters

LoadIdList A sequence of strings that identify the loads to retrieve.

IdsAreRefNums A boolean flag that indicates whether the supplied IDs are load IDs or load reference numbers.

RefNumType If the previous flag is True, then `RefNumType` is the reference number qualifier type on which to search.

Loads A sequence of `Load_V1` structures (`LoadList_V1`).

void LoadSrcv::RetrieveLoad

Synopsis

```
void RetrieveLoad
  (in VIS::StrIdList LoadIdList,
   in boolean IdsAreRefNums,
   in tRefNumType RefNumType,
   in boolean ComputeCharges,
   inout any LoadList,
   out VISError::Details Errors)
raises(VISError::Immediate);
```

Description Retrieves the specified loads.

Parameters

LoadIdList A sequence of strings that identify the loads to retrieve.

IdsAreRefNums A boolean flag that indicates whether the supplied IDs are load IDs or load reference numbers. If you use reference numbers for identification, they must be unique across all loads. If a supplied reference number is attached to two or more loads, this will generate an error.

RefNumType If the previous flag is True, then `RefNumType` is the reference number qualifier type on which to search. Refer to the Reference

Number Qualifier domain table in Transportation Manager for the possible values.

ComputeCharges

A boolean flag that indicates whether to calculate a detailed charge breakdown for each load and load leg associated with each load.

Loads

A sequence of Load_V1 structures (LoadList_V1).

void LoadSrcv::RetrieveLoadsByDate

Synopsis

```
void RetrieveLoadsByDate
  (in eRetrieveLoadDate LoadDateType,
   in VIS::timestamp StartDate,
   in VIS::timestamp EndDate,
   out VIS::StrIdListLoadIdList,
   out VISError::Details Errors)
raises(VISError::Immediate);
```

Description Retrieves loads that meet the specified range of dates.

Parameters

LoadDateType The type of date for the loads to be retrieved. You can retrieve loads by the specified dates range for:

Load Start Date: date type is RLD_START.

Load End Date: date type is RLD_END.

Load Scheduled Date: date type is RLD_SCHED.

StartDate The start date of the specified range. The format is MM/DD/YYYY. If this date is omitted, then the system uses the earliest date in the database.

EndDate The end date of the specified range. The format is MM/DD/YYYY. The end date defaults to current date. If this date is omitted, then the system uses the current date in the database.

LoadIdList List of Load ID that meet the specified criteria.

void LoadSrcv::RetrieveLoadsByDesc

Synopsis

```
void RetrieveLoadsByDesc
  (in VIS::StrIdList LoadDescList,
   inout any Loads,
   out VISError::Details Errors)
raises(VISError::Immediate);
```

Description Retrieves loads by a specified load description.

Parameters

LoadDescList A sequence of strings representing load descriptions.

Loads A sequence of output Load_V1 structures for retrieved loads.

void LoadSrcv::RetrievePayableCarrierOverride

Synopsis

```
void RetrievePayableCarrierOverride
  (in VIS::num28_0 Load_Id,
   inout any CarrierOverrideList,
   out VISError::Details Errors)
  raises(VISError::Immediate);
```

Description Retrieves payable carrier overrides.

Parameters

Load_ID A string that identifies the load.

CarrierOverrideList

A list of PayableCarrOvr_V1 structures:
PayableCarrOvrList_V1.

void LoadSrcv::RetrievePlanLoads

Synopsis

```
void RetrievePlanLoads
  (in VIS::num28_0 PlanId,
   out any Loads,
   in boolean ComputeCharges,
   out VISError::Details Errors)
  raises(VISError::Immediate);
```

Description Retrieves a set of loads from the specified LoadBuildPlan.

Parameters

PlanId The plan ID from which to retrieve the loads.

ComputeCharges

A boolean flag that indicates whether to calculate a detailed charge breakdown for each load and for each load leg associated with each load.

Loads A sequence of Load_V1 structures (LoadList_V1).

void LoadSrcv::RetrievePlanShipments

Synopsis

```
void RetrievePlanShipments
  (in VIS::num28_0 PlanId,
   out any Shipments,
   out VISError::Details Errors)
  raises(VISError::Immediate);
```

Description Retrieves all the shipments for the specified plan.

Parameters

PlanId	The plan ID from which to retrieve the shipments.
Loads	A sequence of Shipment_V1 structures (ShipmentList_V1).

void LoadSrv::SelectLoads**Synopsis**

```
void SelectLoads
    (in any Criteria,
     out VIS::StrIdList LoadIdlist,
     out VISError::Details Errors)
raises(VISError::Immediate);
```

Description Retrieves the load ID of each load. These load IDs can be used to retrieve the Load_V1 structures using the RetrieveLoad operation.

Parameters

Criteria	Contains a LoadSelectionCriteriaList_V1. For each such structure in the list, SelectLoads identifies the loads that meet the criteria in the structure as a load ID list.
LoadIdList	Returns the resulting Load IDs.

void LoadSrv::SetAllToPlanned**Synopsis**

```
void SetAllToPlanned
    (in VIS::StrIdList PlanIdList,
     inout any Results,
     out VISError::Details Errors)
raises(VISError::Immediate);
```

Description Advances the status of all loads in a plan from Open to Planned and removes them from the plan.

If the load does not currently have valid routing and rating information, the API server will rate the load.

If the auto-tendering flag has been enabled, the load will be submitted for auto-tendering after successfully being set to planned.

You can successfully set a plan to Planned status, but still fail to submit it for auto-tendering. This is because the two operations are done in separate transactions.

Parameters

PlanIdList	The list of plan IDs. If some loads are not successfully advanced to Planned, you do not need to change this PlanIdList before trying again. This operation will advance as many loads as it can, regardless of whether some of them fail.
Results	A sequence of NamedResult structures. One entry is added to the sequence for each load processed. The ID field contains the Plan ID and the Desc field contains the Load ID. The Success_yn field

contains `VIS::bTRUE` if the load was successfully advanced. Otherwise the value is `VIS::bFALSE`.

Errors Errors contains a description of the problem for each entry in Results with a value of `VIS::bFALSE` in `Success_yn`. The index field of the entry contains the plan's index in `PlanIdList`, and the Attribute field contains the Load ID.

void LoadSrvc::SetLoadInstruction

Synopsis

```
void SetLoadInstruction
    in VIS::num28_0 LoadId,
    in VIS::num28_0 StopId,
    in any LoadInstruction,
    out VISError::Details Errors)
raises(VISError::Immediate);
```

Description Creates load instructions for a specified stop.

Parameters

LoadId A load ID.

StopId A stop ID.

Loads A sequence of load instructions for a specified stop. Supports `LoadInstruction_V1` structures.

void LoadSrvc::SetShipmentPOD

Synopsis

```
void SetShipmentPOD (in any ShipmentUpdateProgressList,
    in VIS_V1::eShipmentId ShipmentIdType,
    in boolean IdsAreRefNums,
    in tRefNumType RefNumType,
    out VISError::Details Errors)
raises (VISError::Immediate);
```

Description Sets the proof of delivery information for a shipment leg.

Parameters

ShipmentUpdateProgressList

A sequence of `ShipmentUpdateProgress_V1` structures.

ShipmentIdType An enumerated value defining how the shipment should be identified.

IdsAreRefNums Indicates whether shipments should be identified by their reference numbers instead of their ID numbers.

RefNumType The type of reference numbers used to identify the shipments.

Notes

The shipment leg is identified the same way as in `ShipmentUpdateProgress` service.

void LoadSrvc::SetStopETA

Synopsis

```
void SetStopETA(in any StopUpdateProgressList,
               in boolean IdsAreRefNums,
               in tRefNumType RefNumType,
               out VISError::Details Errors)
raises (VISError::Immediate);
```

Description Sets the estimated time of arrival for a stop.

Parameters

StopUpdateProgressList

A sequence of StopUpdateProgress_V1 structures.

IdsAreRefNums Indicates whether loads should be identified by their reference numbers instead of their ID numbers.

RefNumType The type of reference numbers used to identify the loads.

Notes

The stop for which the estimated time of arrival is being set is identified in two steps:

1. The load is identified in the same way as in the LoadUpdateProgress service.
2. The load's stop is identified in one the following ways:
 - o by the shipping location code and type - if this is not unique within the load, then an error occurs and the service call will fail
 - o by the stop sequential number

void LoadSrvc::SetStopPOD

Synopsis

```
void SetStopPOD(in any StopUpdateProgressList,
                in Boolean IdsAreRefNums,
                in tRefNumType RefNumType,
                out VISError::Details Errors)
raises (VISError::Immediate);
```

Description Sets Proof of Delivery information for a stop.

Parameters

StopUpdateProgressList

A sequence of StopUpdateProgress_V1 structures.

IdsAreRefNums Indicates whether stops should be identified by their reference numbers instead of their ID numbers.

RefNumType The type of reference numbers used to identify the stops.

Notes

Load and a shipment leg completed at the specified stop should be IN_TRANSIT or DELIVERED state.

The stop for which the ETA is being set is identified in the same way as in `SetStopETA`.

The load and shipment leg that are completed at the specified stop should be in a state of `IN_TRANSIT` or `DELIVERED`.

void LoadSrcv::SetStopToDelivered

Synopsis

```
void SetStopToDelivered(in any StopUpdateProgressList,  
    in Boolean IdsAreRefNums,  
    in tRefNumType RefNumType,  
    out VISError::Details Errors)  
raises (VISError::Immediate);
```

Description Sets a load stop to Delivered status.

Parameters

`StopUpdateProgressList`

A sequence of `StopUpdateProgress_V1` structures.

`IdsAreRefNums` Indicates whether loads should be identified by their reference numbers instead of their ID numbers.

`RefNumType` The type of reference numbers used to identify the loads.

Notes

The stop being set to delivered is identified in the same way as in `SetStopETA`.

The load and shipment legs that are completed at the specified stop should be in a state of `IN_TRANSIT`.

void LoadSrcv::SetToPlanned

Synopsis

```
void SetToPlanned  
    (in VIS::StrIdList LoadIdList,  
    out VISError::Details Errors)  
raises(VISError::Immediate);
```

Description Advances the status of a load from Open to Planned and removes it from the associated plan.

If the load does not currently have valid routing and rating information, the API server will rate the load.

If the auto-tendering flag has been enabled, the load will be submitted for auto-tendering after successfully being set to planned.

You can successfully set a load to Planned status, but still fail to submit it for auto-tendering. This is because the two operations are done in separate transactions.

Parameters

LoadIdList A sequence of Load IDs. If the status of any of the loads in this list cannot be advanced, then none will be.

void LoadSrcv::ShipmentUpdateProgress**Synopsis**

```
void ShipmentUpdateProgress(in any ShipmentUpdateProgressList,
    in VIS_V1::eShipmentId ShipmentIdType,
    in boolean IdsAreRefNums,
    in tRefNumType RefNumType,
    out VISError::Details Errors)
    raises (VISError::Immediate);
```

Description Updates the progress information of a shipment leg.

Parameters

ShipmentUpdateProgressList

A sequence of ShipmentUpdateProgress_V1 structures.

ShipmentIdType An enumerated value defining how the shipment should be identified.

IdsAreRefNums Indicates whether shipments should be identified by their reference numbers instead of their ID numbers.

RefNumType The type of reference numbers used to identify the shipments.

Identifying a Shipment Leg

The progress is updated for a shipment leg, not for a shipment. The shipment leg is identified in the following way:

1. The shipment is identified using the ShipmentUpdateProgress_V1 data structure and the parameter ShipmentIdType, as described in the following table.

ShipmentIdType	Shipment identified by a	Fields in ShipmentUpdateProgress_V1 that must be defined	Notes
SI_BOL_NUM_CARR	carrier code and BOL number	Carr_cd, MBOL_Num	parameter IdsAreRefNums should be set to FALSE
SI_ID (the default ShipmentIdType)	shipment ID	Shpm_Id	Shpm_Id identifies the shipment parameter IdsAreRefNums should be set to FALSE
SI_LEG_ID	shipment leg ID	Shpm_Leg_Id	parameter IdsAreRefNums should be set to FALSE
SI_NUM	shipment number	Shpm_Num	Shpm_Id identifies the shipment parameter IdsAreRefNums should be set to FALSE

ShipmentIdType	Shipment identified by a	Fields in ShipmentUpdateProgress_V1 that must be defined	Notes
SI_RFRC	reference number	Shpm_Id	contain reference number, parameter <code>IdsAreRefNums</code> should be set to TRUE parameter <code>RefNumType</code> should specify reference number type
SI_RFRC_DIV	reference number and division code	Shpm_Id Div_cd	Shpm_Id should contain a reference number parameter <code>IdsAreRefNums</code> should be set to TRUE parameter <code>RefNumType</code> should specify reference number type Div_cd should contain the division code if the user is restricted to a specific division and different division is specified in the input, then the operation will fail
SI_TRKG_NUM	shipment tracking number	Shpm_Desc	Shpm_Desc should contain a shipment tracking number that identifies uniquely the shipment Parameter <code>IdsAreRefNums</code> should be set to FALSE
SI_TRKG_NUM_DIV	shipment tracking number and division code	Shpm_Desc Div_cd	Shpm_Desc should contain a shipment tracking number that identifies uniquely the shipment Div_cd should contain the division code if the user is restricted to a specific division and different division is specified in the input, then the operation will fail Parameter <code>IdsAreRefNums</code> should be set to FALSE

- After the shipment is determined, the shipment leg should be identified within it. The shipment leg can be identified by the BOL number, the shipment leg ID, or the shipment leg sequential number. Only specify one of these fields in a `ShipmentUpdateProgress_V1` structure.

Note: The shipment and a shipment leg should be in a state of `IN_TRANSIT`.

void LoadSrcv::StopConfirm

Synopsis

```
void StopConfirm
  (in any StopConfirmData,
   out VISError::Details Errors)
  raises(VISError::Immediate);
```

Description Confirms the specified stop.

Parameters

StopConfirmData

A StopConfirmData_v1 structure that identifies the stop to be confirmed.

You can identify a stop using any of the following parameters:

- stop ID
- shipping location and shipping location type
- sequence number

Note: You must specify only the SEC code, trailer number, seal number and driver SEC fields. If you specify any other SEC field, a warning message will appear and the value will be ignored.

void LoadSrcv::TriggerLoadEvent

Synopsis

```
void TriggerLoadEvent
  (in VIS::StrIdList LoadIdList,
   in VIS_V1::eLoadEvent Event,
   in any SECInfo,
   in boolean IdsAreRefNums,
   in tRefNumType RefNumType,
   out VISError::Details Errors)
  raises(VISError::Immediate);
```

Description Triggers a status change in the specified loads.

Parameters

LoadIdList A sequence of strings that identify the loads on which to trigger an event.

Event The event to trigger.

SECInfo The service event code (SEC) information.

IdsAreRefNums

A boolean flag that indicates whether the supplied IDs are load IDs or load reference numbers.

RefNumType If the previous flag is True, then RefNumType is the reference number qualifier type on which to search.

Load Tender Services

A load tender is a request to a carrier to transport a load.

The following load services are specifically for load tenders. The status codes described are as follows:

Planned	STAT_LLH_O_PLANNED
Tendered	STAT_LLH_O_TENDERED
Accepted	STAT_LLH_O_TENDER_ACCEPTED
Rejected	STAT_LLH_O_TENDER_REJECTED

For these services, the errors list is filled only if the error mode for the session is set to `VentureSession::Immediate`.

Reference Number

You can generate the reference number before load confirmation. That is, the system can generate the reference number during tender acceptance. In load confirmation, the system will not generate a reference number if it finds one already generated (during tender acceptance).

Whether the reference number should be generated at tender accept time or confirmation time depends on the status in the Details tab of the Reference Number Qualifier.

If the status is `LL_TNDR_ACPT`, then reference number will be generated at tender accept time. If the status is `LL_LOADED_CN`, then reference number will be generated at load confirmation time. Note that if it is `LL_TNDR_ACPT`, it will also generate a reference number during load tender if `AutoAccept_yn` is `True`.

The reference number generated at tender accept time will be deleted in the following cases:

- if the tender is cancelled
- if the tender is rejected
- during carrier override in load confirmation: after deletion, a new reference number is generated

This reference number generation affects the following services:

- `LoadSrcv::Tender`
- `LoadSrcv::TenderAccept`
- `LoadSrcv::Confirm`
- `LoadSrcv::TenderCancel`
- `LoadSrcv::TenderReject`

void LoadSrcv::RetrieveTender

Synopsis

```
void RetrieveTender
  (in VIS::StrIdList LoadIdList,
   inout any TenderList,
   out VISError::Details Errors)
raises(VISError::Immediate);
```

Description Gives load tender information.

Parameters

LoadIDList A sequence of load IDs identifying load tenders to retrieve.

TenderList A sequence of LoadTenderData_V1 structures with the tender data of the specified loads.

void LoadSrcv::ReturnToOpen

Synopsis

```
void ReturnToOpen
  (in any LoadPlanList,
   out VISError::Details Errors)
raises(VISError::Immediate);
```

Description Returns the status of each load in `LoadPlanList` from `Planned` to `Open`. By default, this operation leaves the load unattached to any Plan. You can attach the load to a plan using the appropriate plan ID with the load ID in `LoadPlanList`.

The only API operations that can attach a load to a plan are the `LoadCreate` and `ReturnToOpen`. There is no direct API operation to attach a load to a plan.

Parameters

LoadPlanList A sequence of `LoadPlan_V1` structures. Each of these structure contains a Load ID, and optionally a plan ID to which you can attach the open load.

If any of the loads in the list cannot be returned to `Open` status, none of them can.

void LoadSrcv::Tender

Synopsis

```
void Tender
  (in any RequestList,
   in boolean AutoAccept_yn,
   inout any ResultList,
   out VISError::Details Errors)
raises(VISError::Immediate);
```

Description Advances the load status from `Planned` to `Tendered`. `AutoAccept_yn` applies to each entry in the list. The `ResultList` contains the resulting tender details for each entry in `RequestList`.

To tender a Load, it must be in planned status. The system checks the rating of the load to ensure it is still valid. (Rating is the calculation of freight charges for shipments, shipment legs, or loads.)

The system takes into consideration that the load may have been set to Planned since it was last rated. Also, you may have specified new carrier and service information for the load tender. If the rating is not valid, then the tender operation does a rating.

If `AutoAccept_yn` is true, or if the tariff and service associated with the load is flagged for auto-accept, then the status of the load is set to Tender Accepted. Otherwise, it is left at Tendered.

Parameters

RequestList A sequence of `LoadTenderData_V1` structures containing a load ID, and fields related to the tender operation. You override values for the tender operation by supplying values in the fields.

If you cannot tender any of the Loads then none of them can be tendered.

AutoAccept_yn

If True, the Tenders are automatically accepted. This value applies to all loads in `RequestList`.

This flag does not disable an auto-accept specified by the associated tariff. Both this flag and the tariff's auto-accept flag must be False to stop a tender from being automatically accepted.

ResultList A sequence of `LoadTenderData_V1` structures with the resulting tender data.

void LoadSrcv::TenderAccept

Synopsis

```
void TenderAccept
  (in any TenderList,
   out VISError::Details Errors)
raises(VISError::Immediate);
```

Description Advances a load status to Tender Accepted. To accept a tender, the load has to be in Tendered status.

If you request this operation for a list of loads and one of the loads fails, then the other loads will still be processed.

Parameters

TenderList A sequence of `LoadTenderData_V1` structures identifying load tenders to accept by the load ID. If the memo field is not empty, it is applied to the load.

If you cannot advance one of the loads in the list none of them will be advanced.

void LoadSrvc::TenderCancel

Synopsis

```
void TenderCancel
  (in any TenderList,
   out VISError::Details Errors)
raises(VISError::Immediate);
```

Description Reverses the status of loads to planned, clears tendering information from the load.

You can only cancel tenders for loads in Tendered or Tender Accepted status. You can cancel tenders for loads in Tender Rejected status using the API, but not through the GUI.

If you request this operation for a list of loads and one of the loads fails, then the other loads will still be processed.

Parameters

TenderList A sequence of LoadTenderData_v1 structures, identifying load tenders to cancel by load ID. If the memo field is not empty, it is applied to the load.

If one of the loads cannot in the list cannot be cancelled, none of them will be.

void LoadSrvc::TenderReject

Synopsis

```
void TenderReject
  (in any TenderList,
   out VISError::Details Errors)
raises(VISError::Immediate);
```

Description Rejects tenders for loads by advancing their load status to Tender Rejected, and clears tendering information from the load.

To reject a Tender, the load has to be in Tendered or Tender Accepted status.

Parameters

TenderList A sequence of LoadTenderData_v1 structures, identifying the load tenders to reject by load ID. If the memo field is not empty, it is applied to the load.

If one of the loads in the list cannot be rejected, none of them will be.

Rate Quotation Services

A zone is a geographic area that represents the origin or destination point of a shipment. A lane links two zones together for a specific carrier service.

The rate quotation services search for valid shipping lanes. If they find more than one valid lane, then all valid lanes found will be sorted from best to worst. The sorting is

based on `SortCriteria` which you provide in the `RateCriteria` parameter. The default value is `cost`. All lanes found will have a detailed list of costs.

void RateQuotationSvc::GetPossibleCharges

Synopsis

```
void GetPossibleCharges
(in any ChargeCriteria,
 inout any ChargeDetails)
raises(VISError::Immediate);
```

Description Returns a list of possible charge details.

Parameters

ChargeCriteria Optional criteria that limits which charges to get (refer to “ChargeCriteria_V1” in “API Structures” on page 172). This service requires an origin point (`LoadAtCode` or `HubFromCode`), and a destination point (`ConsigneeCode` or `HubToCode`).

ChargeDetails A list of possible charge details.

void RateQuotationSvc::RateQuotation

Synopsis

```
void RateQuotation
(in any RateCriteria,
 in VIS::vBool ReturnDiagnostics,
 inout any ValidLanes,
 out VISError::Details Errors)
raises(VISError::Immediate);
```

Description Tries to find a set of valid lanes and, if found, sorts them in increasing order of shipping costs.

For the Oracle Application integration interface, if you do not enter the freight class ID, the system uses the freight class of the customer as the default.

Parameters

RateCriteria The input criteria describing what is being shipped and how it should be shipped.

ReturnDiagnostics

Returns diagnostic messages with reasons why no valid lane was found for shipping. You can use this parameter only if the version of your server supports it.

ValidLanes Lists all valid lanes associated with the shipment including the best lane, and the cost of each. These lanes are in a sequence of `ValidLane_V1` structures (`ValidLanes_V1`). They are sorted from best to worst according to the `SortCriteria` parameter specified in the `RateCriteria` structure for the `RateQuotation` call.

Shipment Services

Use the shipment services to retrieve, confirm, and trigger shipments.

The error list is filled if the error mode for the session is `VentureSession::Immediate`.

void ShipmentSrvc::AttachToPlan

Synopsis

```
void AttachToPlan
  (in string PlanId,
   in VIS::StrIdList
   ShipmentLegIdList,
   out VISError::Details Errors)
raises(VISError::Immediate);
```

Description Attaches shipment legs to the specified plan. If the list is empty, then all shipment legs in process status will be attached to the plan.

Parameters

PlanId The plan ID to which the shipment legs can attach.

ShipmentLegList

A sequence of shipment leg ID strings to attach to the plan.

void ShipmentSrvc::CancelShipment

Synopsis

```
void CancelShipment
  (in VIS::StrIdList ShipmentIdList,
   out VISError::Details Errors)
raises(VISError::Immediate);
```

Description Cancels shipments.

Parameters

ShipmentIdList A sequence of shipment IDs of the shipments to cancel.

void ShipmentSrvc::PayableCarrierOverride

Synopsis

```
void PayableCarrierOverride
  (in VIS::num28_0 ShipmentLegId,
```

```
    in any CarrierOverrideList,  
    out VISError::Details Errors)  
raises(VISError::Immediate);
```

Description A payable carrier override.

Parameters

ShipmentLegId A string that identifies the shipment leg.

CarrierOverrideList

A list of PayableCarrOvr_V1 structures:
PayableCarrOvrList_V1.

void ShipmentSrvc::ResponsibleCustomerOverride

Synopsis

```
void ResponsibleCustomerOverride  
  (in string<30>ShipmentId,  
   inout any CustomerOverrideList,  
   out VISError::Details Errors)  
raises(VISError::Immediate);
```

Description A responsible customer override.

Parameters

ShipmentId A string that identifies the shipment.

CustomerOverrideList

A list of RspbCustOvr_V1 structures:
RspbCustOvrList_V1.

void ShipmentSrvc::RetrievePayableCarrierOverride

Synopsis

```
void RetrievePayableCarrierOverride  
  (in VIS::num28_0 ShipmentLegId,  
   inout any CarrierOverrideList,  
   out VISError::Details Errors)  
raises(VISError::Immediate);
```

Description Retrieves payable carrier overrides.

Parameters

ShipmentLegId A string that identifies the shipment leg.

CarrierOverrideList

A list of PayableCarrOvr_V1 structures:
PayableCarrOvrList_V1.

void ShipmentSrvc::RetrieveResponsibleCustomerOverride

Synopsis

```
void RetrieveResponsibleCustomerOverride
( in string<30> ShipmentId,
  inout any CustomerOverrideList,
  out VISError::Details Errors)
raises(VISError::Immediate);
```

Description Retrieves responsible customer overrides.

Parameters

ShipmentId A string that identifies the shipment.

CustomerOverrideList

A list of RspbcustOvr_V1 structures: RspbcustOvrList_V1.

void ShipmentSrvc::RetrieveShipmentLegs

Synopsis

```
void RetrieveShipmentLegs
( in VIS::StrIdList ShipmentIdList,
  inout any ShipmentLegList,
  out VISError::Details Errors)
raises(VISError::Immediate);
```

Description Retrieves all the shipment leg information for a specified shipment.

Parameters

ShipmentIdList A sequence of strings that identifies the shipment numbers to retrieve.

ShipmentLegList

A sequence of shipment leg structures of the type ShipmentLeg_V1.
The supported sequence type is ShipmentLegList_V1.

void ShipmentSrvc::RetrieveShipmentLegsById

Synopsis

```
void RetrieveShipmentLegsById
( in VIS::StrIdList ShipmentLegIdList,
  inout any ShipmentLegList,
  out VISError::Details Errors)
raises(VISError::Immediate);
```

Description Retrieves shipment leg information by ID.

Parameters

ShipmentLegIdList

A structure of the type VIS::StrIdList describing the list of shipment leg IDs to retrieve.

ShipmentLegList

A structure of the type `ShipmentLegList_V1`. A list of shipment leg information of the type `ShipmentLeg_V1`.

void ShipmentSrvc::ReturnShipmentLegFromPlanned

Synopsis

```
void ReturnShipmentLegFromPlanned
  (in VIS::StrIdList ShipmentLegIdList,
   out VISError::Details Errors)
raises(VISError::Immediate);
```

Description Return shipment legs from Planned status to Processed.

Parameters

ShipmentLegIdList

A sequence of shipment leg IDs of the shipments to return.

void ShipmentSrvc::SelectForOptimization

Synopsis

```
void SelectForOptimization
  (in any SelectionCriteria,
   in VIS::timestamp RequestdDateTime,
   in boolean BuildTrip_yn,
   in boolean AutoOptimize_yn,
   out VISError::Details Errors)
raises(VISError::Immediate);
```

Description Selects shipments for optimization and starts the optimization process for the selected shipments.

Parameters

SelectionCriteria

A structure of the type `SelectionCriteria_V1` describing the criteria for selecting shipments.

RequestdDateTime

The date and time that the optimization will be performed.

BuildTrip_yn

You can use this flag if you have not selected a shipment leg for optimization. If `True`, then the optimization request is sent to Transportation Optimizer. If `False`, a warning message appears and the service stops.

AutoOptimize_yn

Indicates whether to send the optimization request to Transportation Optimizer. The default value is `True`.

void ShipmentSrvc::SendToOptimizer

Synopsis

```
void SendToOptimizer
    (in string PlanId,
     in VIS::timestampRequestdDateTime,
     in boolean BuildTrip_yn,
     out VISError::DetatilsErrors)
raises(VISError::Immediate);
```

Description Sends a optimization request to Transportation Optimizer.

Parameters

PlanId The current plan ID.

RequestdDateTime

The date and time that the optimization will be performed

BuildTrip_yn

You can use this flag if you have not selected a shipment leg for optimization. If True, then the optimization request is sent to Transportation Optimizer. If False, a warning message appears and the service stops.

void ShipmentSrvc::SetToPlanned

Synopsis

```
void SetToPlanned
    (in VIS::StrIdList ShipmentLegList,
     out VISError::Details Errors)
raises(VISError::Immediate);
```

Description Advances the status of specified shipment legs from Processed to Planned and removes them from any associated plan.

Through this API, you can set a load to Planned status both before and after rating. However, using the GUI, the load must first be successfully rated.

Parameters

ShipmentLegList

A sequence of shipment leg IDs. If you cannot advance the status of the shipment legs, none of them will be.

void ShipmentSrvc::ShipmentCompositionChange

Synopsis

```
void ShipmentCompositionChange
    (in any ShipmentCompChangeDataList,
     in eAction Action,
     out VISError::Details Errors)
raises(VISError::Immediate);
```

Description Adds or deletes components from existing shipments. Use `UpdateTO` to update existing components (refer to “void `TransportOrderSvc::SetTOUpdate`” on page 86).

Parameters

`ShipmentCompChangeDataList`

A sequence of shipment composition change data structures of the type `ShipmentCompChangeData_V1`. The supported sequence type is `ShipmentCompChangeDataList_V1`.

`Action` The action to perform: Add or Delete.

void ShipmentSvc::ShipmentConfirm

Synopsis

```
void ShipmentConfirm
    (in any ShipmentConfirmDataList,
     out VISError::Details Errors)
raises(VISError::Immediate);
```

Description Confirms the specified shipments.

Parameters

`ShipmentConfirmDataList`

A sequence of shipment confirmation data structures of the type `ShipmentConfirmData_V1` to confirm. The supported sequence type is `ShipmentConfirmDataList_V1`.

void ShipmentSvc::TriggerShipmentLegEvent

Synopsis

```
void TriggerShipmentLegEvent
    (in VIS::StrIdList ShipmentLegIdList,
     in VIS_V1::eShipmentLegEvent Event,
     in any SECInfo,
     out VISError::Details Errors)
raises(VISError::Immediate);
```

Description Triggers shipment leg events. The supporting events are release and return of a shipment, and stopping the suspension of shipment legs.

Parameters

`ShipmentLegIdList`

A list of the shipment leg numbers to apply.

`Event` The event to trigger.

`SECInfo` The SEC information for the event.

void ShipmentSrvc::UpdateShipmentLegs

Synopsis

```
void UpdateShipmentLegs
    (in any ShipmentLegList,
     out VISError::Details Errors)
raises(VISError::Immediate);
```

Description Updates the commodity pick sequence number for specified shipment leg. Shipment legs are identified by their IDs. Only the following two fields must be specified in the ShipmentLeg_V1 structure: Shpg_Leg_Id and CdtY_Pick_Seq_num.

Parameters

ShipmentLegList

A sequence of shipment leg structures of the type ShipmentLeg_V1. The supported sequence type is ShipmentLegList_V1.

Shipment Order Entry Services

Use the shipment order entry services to create, retrieve, update, and delete shipments. You can also use these services to indicate how to create a shipment from a set of containers.

The error list is filled if the error mode for the session is VentureSession::Immediate.

void ShipmentOrderEntrySrvc::CancelShipment

Synopsis

```
void CancelShipment
    (in any ShipmentId,
     in VIS_V1::eShipmentId ShipmentIdType,
     in tRefNumType RefNumType,
     out VISError::DetailsErrors) raises(VISError::Immediate);
```

Description Cancels shipments.

Parameters

ShipmentList A sequence of Shipment_V2 entity structures to be cancelled.

ShipmentIdType

The method of identifying the shipments to be cancelled.

SI_NULL - No shipments are cancelled.

SI_ID - The shipments to be cancelled are identified by their IDs.

SI_NUM - The shipments to be cancelled are identified by their shipment numbers.

SI_RFRC - The shipments to be cancelled are identified by their reference numbers.

SI_RFRC_DIV - The shipments to be cancelled are identified by their reference numbers and Division ID.

If a user is not restricted to a division, then a division must be specified in the input. If a user is restricted to a division, then a division can be omitted or, if specified, should match the division to which the user is restricted.

SI_TRKG_NUM - The shipments to be cancelled are identified by their shipment tracking number. This is `Shpm_Desc` in the `Shipment_V2` structure.

SI_TRKG_NUM_DIV - The shipments to be cancelled are identified by their shipment tracking number. This is `Shpm_Desc` in the `Shipment_V2` structure.

If a user is not restricted to a division, then a division must be specified in the input. If a user is restricted to a division, then the division can be omitted or, if specified, it should match the division to which the user is restricted.

`RefNumType` The reference number type if `ShipmentIdType` is `SI_RFRC`, otherwise this value is ignored.

void ShipmentOrderEntrySvc::CreateShipment

Synopsis

```
void CreateShipment
(in any ShipmentList,
 in boolean IgnoreRfrNums,
  in VIS::vbool APRate,
  in VIS::vbool ARRate,
 in VIS_V1::eResultContents ResultContents,
 inout any OutShipmentList,
 out VISError::Details Errors)
raises(VISError::Immediate);
```

Description Creates shipments.

Parameters

`ShipmentList` A sequence of `Shipment_V2` entity structures to be created.

`IgnoreRfrNums`

Indicates whether all reference numbers provided in the input are ignored.

`APRate` Indicates whether an A/P rating is performed for the created shipments.

`ARRate` Indicates whether the A/R rating is performed for the created shipments.

`ResultContents` The kind of output that is produced:

`RES_NULL` and `RES_NONE` - Only IDs of new shipments is sent to the client.

RES_SHIPMENT_HEADER - Partial information (the shipment header) regarding created shipments is sent to the client.

RES_SHIPMENT_FULL - All the information about created shipments is sent.

RES_SHIPMENTLEG_HEADER - Partial information (the shipment leg header) regarding the shipment legs for created shipments is sent.

OutShipmentList

A sequence of output structures for created shipments. The type of output depends on the value selected for `ResultContents`, as follows:

RES_NULL, RES_NONE - The output is a list of shipment IDs.

RES_SHIPMENT_HEADER - The output is a list of `ShipmentHeader_V2` structures.

RES_SHIPMENT_FULL - The output is a list of `Shipment_V2` structures.

RES_SHIPMENTLEG_HEADER - The output is a list of `ShipmentLegHeader_V1` structures.

void ShipmentOrderEntrySvc::DeleteShipment

Synopsis

```
void DeleteShipment
    (in any ShipmentList,
     in VIS_V1::eShipmentIdShipmentIdType,
     in tRefNumType RefNumType,
     out VISError::DetailsErrors)
    raises(VISError::Immediate);
```

Description Deletes shipments.

Parameters

ShipmentList A sequence of `Shipment_V2` entity structures to be deleted.

ShipmentIdType

The method of identifying the shipments to be deleted.

SI_NULL - No shipments are deleted.

SI_ID - The shipments to be deleted are identified by their IDs.

SI_NUM - The shipments to be deleted are identified by their shipment numbers.

SI_RFRC - The shipments to be deleted are identified by their reference numbers.

SI_RFRC_DIV - The shipments to be deleted are identified by their reference numbers and Division ID.

If a user is not restricted to a division, then a division must be specified in the input. If a user is restricted to a division, then a division can be omitted or, if specified, should match the division to which the user is restricted.

SI_TRKG_NUM - The shipments to be deleted are identified by their shipment tracking number. This is `Shpm_Desc` in the `Shipment_V2` structure.

SI_TRKG_NUM_DIV - The shipments to be deleted are identified by their shipment tracking number. This is `Shpm_Desc` in the `Shipment_V2` structure.

If a user is not restricted to a division, then a division must be specified in the input. If a user is restricted to a division, then the division can be omitted or, if specified, it should match the division to which the user is restricted.

RefNumType The reference number type if `ShipmentIdType` is `SI_RFRC`, otherwise this value is ignored.

void ShipmentOrderEntrySrcv::PlanShipments

Synopsis

```
void PlanShipments
  (in any PlanningRequestList,
   inout any ShipmentPlans,
   out VISError::Details Errors)
raises(VISError::Immediate);
```

Description Creates shipment plans based on specified shipment planning requests and container plans. Indicates how input data should be grouped into shipments.

Parameters

PlanningRequestList

A list of plan requests. The supported sequence type is `PlanningRequest_V1`.

ShipmentPlans A sequence of produced shipment plans. The supported sequence type is `ShipmentPlan_V1`.

void ShipmentOrderEntrySrcv::RetrieveShipment

Synopsis

```
void RetrieveShipment
  (in any ShipmentList,
   in VIS_V1::eShipmentId ShipmentIdType,
   in VIS_V1::eResultContents ResultContents,
   in boolean ReturnContainers,
   inout any OutShipmentList,
   out VISError::Details Errors)
raises(VISError::Immediate);
```

Description Retrieves shipments.

Parameters

ShipmentList A sequence of `Shipment_V2` entity structures to be retrieved.

ShipmentIdType

The method of identifying the shipments to be retrieved.

SI_NULL - No shipments are retrieved.

SI_ID - The shipments to be retrieved are identified by their IDs.

SI_NUM - The shipments to be retrieved are identified by their shipment numbers.

SI_RFRC - The shipments to be retrieved are identified by their reference numbers.

SI_RFRC_DIV - The shipments to be retrieved are identified by their reference numbers and Division ID.

If a user is not restricted to a division, then a division must be specified in the input. If a user is restricted to a division, then a division can be omitted or, if specified, should match the division to which the user is restricted.

SI_TRKG_NUM - The shipments to be retrieved are identified by their shipment tracking number. This is `Shpm_Desc` in the `Shipment_V2` structure.

SI_TRKG_NUM_DIV - The shipments to be retrieved are identified by their shipment tracking number. This is `Shpm_Desc` in the `Shipment_V2` structure.

If a user is not restricted to a division, then a division must be specified in the input. If a user is restricted to a division, then the division can be omitted or, if specified, it should match the division to which the user is restricted.

ResultContents The type of output that is produced:

RES_NULL and **RES_NONE** - Only IDs of new shipments are sent to the client.

RES_SHIPMENT_HEADER - Partial information (the shipment header) regarding created shipments is sent to the client.

RES_SHIPMENT_FULL - All the information about created shipments is created.

RES_SHIPMENTLEG_HEADER - Partial information (the shipment leg header) regarding the shipment legs for created shipments is created.

ReturnContainers

If `True` and `ResultContents` is `RES_SHIPMENT_FULL`, then information about all containers included in this shipment is

retrieved. Otherwise, only header or shipment legs information is produced.

OutShipmentList

A sequence of output structures for created shipments. The output depends on the value in `ResultContents`, as follows:

`RES_NULL`, `RES_NONE` - The output is a list of shipment IDs.

`RES_SHIPMENT_HEADER` - The output is a list of `ShipmentHeader_V2` structures.

`RES_SHIPMENT_FULL` - The output is a list of `Shipment_V2` structures.

`RES_SHIPMENTLEG_HEADER` - The output is a list of `ShipmentLegHeader_V1` structures.

void ShipmentOrderEntrySvc::UpdateShipment

Synopsis

```
void UpdateShipment
(in any ShipmentList,
 in VIS_V1::eShipmentId ShipmentIdType,
 in tRefNumType RefNumType,
 in boolean IgnoreHeader,
 in boolean IgnoreShpmRfrcNums,
 in boolean IgnoreContainers,
 in boolean IgnoreThruPoints,
 in VIS::vbool APRate,
 in VIS::vbool ARRate,
 out VISError::DetailsErrors) raises(VISError::Immediate);
```

Description Updates shipments.

Parameters

ShipmentList A sequence of strings or numbers identifying shipments to update.

ShipmentIdType

The method of identifying the shipments to be updated.

`SI_NULL` - No shipments are updated.

`SI_ID` - The shipments to be updated are identified by their IDs.

`SI_NUM` - The shipments to be updated are identified by their shipment numbers.

`SI_RFRC` - The shipments to be updated are identified by their reference numbers.

`SI_RFRC_DIV` - The shipments to be updated are identified by their reference numbers and Division ID.

If a user is not restricted to a division, then a division must be specified in the input. If a user is restricted to a division, then a

division can be omitted or, if specified, should match the division to which the user is restricted.

SI_TRKG_NUM - The shipments to be updated are identified by their shipment tracking number. This is `Shpm_Desc` in the `Shipment_V2` structure.

SI_TRKG_NUM_DIV - The shipments to be updated are identified by their shipment tracking number. This is `Shpm_Desc` in the `Shipment_V2` structure.

If a user is not restricted to a division, then a division must be specified in the input. If a user is restricted to a division, then the division can be omitted or, if specified, it should match the division to which the user is restricted.

RefNumType The reference number type if `ShipmentIdType` is `SI_RFRC`, otherwise this parameter is ignored.

Ignore Header If False, then the shipments' header information is updated.

IgnoreShpmRfrNums

If True, no shipment reference numbers are updated and verified. If False, old reference numbers are deleted, and new reference numbers are verified and associated with shipments.

IgnoreContainers

If False, then container information can be modified.

IgnoreThruPoints

If False, then the shipment legs' content can be updated.

APRate Indicates whether A/P rating is performed for updated shipments.

ARRate Indicates whether A/R rating is performed for updated shipments.

Tariff Services

A tariff is the contract provided by a logistics provider or carrier to a customer detailing the carrier's rates, services, and options.

Use the tariff services to create, retrieve, update, and delete Transportation Manager tariffs. You can perform operations either on the entire tariff object or on the individual tariff entities. Tariff entities include the tariff service, tariff charge (condition and option), tariff lane associations, tariff rate codes, and tariff surcharges.

A condition is the primary charge types or units a carrier uses to charge for a service. An option is a miscellaneous charge for something that is not part of the base pricing structure of a tariff.

The error list is filled if the error mode for the session is `VentureSession::Deferred`.

void TariffSrvc::CreateTariff

Synopsis

```
void CreateTariff
  (in any TariffList,
   out VIS::StrIdList TariffIdList,
   out VISError::Details Errors)
raises(VISError::Immediate);
```

Description Creates the specified tariffs.

Parameters

TariffList A sequence of `Tariff_V1` entity structures to create. The supported sequence type is `TariffList_V1`.

TariffIdList A sequence of strings that identify the created tariff numbers.

void TariffSrvc::CreateTariffEntity

Synopsis

```
void CreateTariffEntity
  (in any TariffEntityList,
   out VISError::Details Errors)
raises(VISError::Immediate);
```

Description Creates tariff entities of the specified type.

Parameters

TariffEntityList A sequence of entity structures of the entity type to create. The supported sequence types are: `LaneAsscList_V1`, `RateCdList_V1`, `SchgSrvcList_V1`, `SchgChrgList_V1`, `SchgRateList_V1`, `TffChrgList_V1`, `TffSrvcEqmtList_V1`, and `TffSrvcList_V1`.

void TariffSrvc::DeleteTariff

Synopsis

```
void DeleteTariff
  (in VIS::StrIdList TariffIdList,
   out VISError::Details Errors)
raises(VISError::Immediate);
```

Description Deletes the specified tariffs.

Parameters

TariffIdList A sequence of strings that identify the tariff numbers of the tariffs to delete.

void TariffSrvc::DeleteTariffEntity

Synopsis

```
void DeleteTariffEntity
  (in any TariffEntityList,
   out VISError::Details Errors)
```

```
raises(VISError::Immediate);
```

Description Deletes the specified tariff entities.

Parameters

TariffEntityList A sequence of entity structures of the entity type to delete. The supported sequence types are the same as in CreateTariffEntity.

void TariffSrvc::RetrieveTariff

Synopsis

```
void RetrieveTariff
  (in VIS::StrIdList IdList,
   in boolean includeSrvc,
   in boolean includeRate,
   in boolean includeLane,
   in boolean includeSurChrg,
   inout any TariffList,
   out VISError::Details Errors)
raises(VISError::Immediate);
```

Description Retrieves the specified tariffs.

Parameters

IdList A sequence of strings that identify the tariff number of the tariffs to retrieve.

includeSrvc Returns services if set to True.

includeRate Returns rate information if set to True.

includeLane Returns lane associations if set to True.

includeSurchrg Returns surcharge information if set to True.

TariffList A sequence of Tariff_V1 tariff structures to retrieve. The supported sequence type is TariffList_V1.

void TariffSrvc::RetrieveTariffEntity

Synopsis

```
void RetrieveTariffEntity
  (in VIS::StrIdList TariffIdList,
   inout any TariffEntityList,
   out VISError::Details Errors)
raises(VISError::Immediate);
```

Description Retrieves the specified tariff entities.

Parameters

TariffIdList A sequence of strings that identify the tariff number of tariffs to retrieve.

TariffEntityList A sequence of entity structures of the entity type to retrieve. The supported sequence types are the same as in `CreateTariffEntity`.

void TariffSrvc::UpdateTariff

Synopsis

```
void UpdateTariff
  (in any TariffList,
   out VISError::Details Errors)
raises(VISError::Immediate);
```

Description Updates the specified tariffs.

Parameters

TariffList A sequence of `Tariff_V1` tariff structures to update. The supported sequence type is `TariffList_V1`.

void TariffSrvc::UpdateTariffEntity

Synopsis

```
void UpdateTariffEntity
  (in any TariffEntityList,
   out VISError::Details Errors)
raises(VISError::Immediate);
```

Description Updates the specified tariff entities.

Parameters

TariffEntityList A sequence of entity structures of the entity type to update. The supported sequence types are the same as in `CreateTariffEntity`.

Transport Order Services

A transport order is a request to move goods.

Use the transport order services to create, retrieve, update, and delete transport orders. You can manipulate components within transport orders, retrieve and manipulate shipments, and trigger status changes.

The error list is filled if the error mode for the session is `VentureSession::Immediate`.

void TransportOrderSrvc::CreateTO

Synopsis

```
void CreateTO
  (in any TOList,
   in boolean RouteAndRate,
   out VISError::Details Errors)
raises(VISError::Immediate);
```

Description Creates transport orders.

Parameters

- TOList** A sequence of `TransportOrder_V1` structures (`TransportOrderList_V1`).
- RouteAndRate** A boolean flag that indicates whether to automatically route and rate after successfully creating each transport order.

void TransportOrderSvc::DeleteTO**Synopsis**

```
void DeleteTO
    (in VIS::StrIDList TOIdList,
     in boolean IdsAreRefNums,
     in tRefNumType RefNumType,
     out VISError::Details Errors)
raises(VISError::Immediate);
```

Description Deletes the specified transport orders.

Parameters

- TOIdList** A sequence of strings that identify the transport orders to delete.
- IdsAreRefNums**
A boolean flag that indicates whether the supplied IDs are transport order IDs or transport order reference numbers.
- RefNumType** If the previous flag is `True`, then `RefNumType` is the reference number qualifier type on which to search.

void TransportOrderSvc::RetrieveTO**Synopsis**

```
void RetrieveTO
    (in VIS::StrIDList TOIdList,
     in boolean IdsAreRefNums,
     in tRefNumType RefNumType,
     inout any TOList,
     out VISError::Details Errors)
raises(VISError::Immediate);
```

Description Retrieves the specified transport orders.

Parameters

- TOIdList** A sequence of strings that identify the transport orders to retrieve.
- IdsAreRefNums**
A boolean flag that indicates whether the supplied IDs are transport order IDs or transport order reference numbers.
- RefNumType** If the previous flag is `True`, then `RefNumType` is the reference number qualifier type on which to search.
- TOList** A returned sequence of `TransportOrder_V1` structures (`TransportOrderList_V1`).

void TransportOrderSvc::RetrieveTOShipments

Synopsis

```
void RetrieveTOShipments
  (in string TOId,
   in boolean IdIsRefNum,
   in tRefNumType RefNumType,
   in boolean RetrieveLoads,
   out any ShipList,
   out VISError::DetailsErrors)
raises(VISError::Immediate);
```

Description Retrieves shipments from the specified transport order.

Parameters

TOId A string that identifies the transport order to retrieve the shipments from.

IdsAreRefNums A boolean flag that indicates whether the supplied IDs are transport order IDs or transport order reference numbers.

RefNumType If the previous flag is True, then `RefNumType` is the reference number qualifier type on which to search.

RetrieveLoads A boolean flag that indicates whether to return the sequence of load IDs that each shipment is on. If this information is not required, use False.

ShipList A returned sequence of `Shipment_V1` structures (`ShipmentList_V1`).

void TransportOrderSvc::SetTOUpdate

Synopsis

```
void SetTOUpdate
  (in boolean EnforceOneShipment
   in boolean MaintainContainerAssignment)
raises(VISError::Immediate);
```

Description Allows the `UpdateTO` API to be restricted in the current session for the Transportation Manager connection. Transportation Manager automatically creates a new, second shipment when the number of components in the first shipment exceeds the constraints.

However in some environments, a transport order can have only one shipment. If so, call `SetTOUpdate` with `EnforceOneShipment` set to True in the session, before calling `UpdateTO`.

Transportation Manager unassigns and reassigns components when `UpdateTO` is called, allowing you to modify any attributes. To maintain components on their shipments with only a subset of the parameters being allowed modification, call `SetTOUpdate` with `MaintainContainerAssignment` set to True. You can reset these

values at any time. Setting of the values is not permanent, and will not affect the operation of other Transportation Manager sessions.

Parameters

EnforceOneShipment

If True, only one shipment is allowed on a transport order. If a subsequent API call causes the containers to exceed the capacity of the shipment, an error is returned and no update occurs. Use False (the default value) to reverse this action.

MaintainContainerAssignment

If True, containers are not reassigned to different shipments during updating. Also, you can update only certain values of the container, namely weight by freight class, volume, length, width, height, order value, and declared value. The default value is False.

void TransportOrderSrvc::TriggerTOEvent

Synopsis

```
void TriggerTOEvent
(in VIS::StrIdList TOIdList,
 in VIS_V1::eTOEvent Event,
 in any SECInfo,
 in boolean IdsAreRefNums,
 in tRefNumType RefNumType,
 out VISError::Details Errors)
raises(VISError::Immediate);
```

Description Triggers a status change in the specified transport orders.

Parameters

- | | |
|---------------|--|
| TOIdList | A sequence of strings that identify the transport orders on which to trigger an event. |
| Event | The event to trigger. |
| SECInfo | The SEC information. |
| IdsAreRefNums | A boolean flag that indicates whether the supplied IDs are transport order IDs or transport order reference numbers. |
| RefNumType | If the previous flag is True, RefNumType is the reference number qualifier type on which to search. |

void TransportOrderSrvc::UpdateTO

Synopsis

```
void UpdateTO
(in VIS::StrIDList TOIdList,
 in any TOList,
 in boolean IdsAreRefNums,
 in tRefNumType RefNumType,
```

```

        in boolean RouteAndRate,
        out VISError::Details Errors)
raises(VISError::Immediate);

```

Description Updates the specified transport orders.

Parameters

TOIdList A sequence of strings that identify the transport orders to update.

TOList A returned sequence of `TransportOrder_V1` structures (`TransportOrderList_V1`).

IdsAreRefNums A boolean flag that indicates whether the supplied IDs are transport order IDs or transport order reference numbers.

RefNumType If the previous flag is `True`, then `RefNumType` is the reference number qualifier type on which to search.

RouteAndRate A boolean flag that indicates whether to automatically route and rate after successfully creating each transport order.

Note: When updating, the `Trpt_Odr_Id` field of the transport order is not referenced. It is assumed that the input list of transport orders correspond one to one with the provided ID list. You use a separate ID list because the transport order structure does not define an identifying reference number.

Updating Containers on a Transport Order

A container is a single physical entity within a shipment. To use the `UpdateTO` API for updating the containers on a transport order, provide the complete list of containers that should be on that transport order. After the update, the containers on the order will correspond to those that you provided as input. Any previous containers on the order that a matching component reference number cannot identify from the input will be deleted. The reference number qualifier type must be `API`.

Containers that you provide as input that cannot be matched to existing containers will be added to the transport order. If you provide exactly one `API` qualifier type reference number of a container being updated, other previous reference numbers on that container are preserved. If you provide more than one reference number, then those numbers not in the list are removed and replaced with the new list.

Note: If the container input does not contain any reference numbers, all containers and associated reference numbers that existed before the update will be removed.

Refer also to `SetTOUpdate` on page 86.

Chapter 4

API Structures

This chapter describes the structures of all Transportation Manager API services. It includes the following topics:

- [Structure Overview](#)
- [Common Structures](#)
- [Delivery Schedule Structures](#)
- [Entity Structures](#)
- [Financial Structures](#)
- [Load Structures](#)
- [Non-Operational Freight Structures](#)
- [Rate Quotation Structures](#)
- [Shipment Structures](#)
- [Shipment Order Entry Structures](#)
- [Tariff Structures](#)
- [Transport Order Structures](#)

Structure Overview

Each structure is listed alphabetically within each group of structures. The position (**P**), name, type, and description are listed for each element.

The following three columns indicate how you can use the element.

R (required) A bullet (•) in this column means that this field is required for any API operation that uses this field. These fields are key entity fields which uniquely identify the entity.

If the field is not required, the system may use a default value. Also, there are some fields that are required only during a Create operation. For example, a customer description is only required during a Create

operation, and is not required in the other operations: Retrieve, Delete and Update.

C (creatable) A bullet (•) in this column sets the value for this field in the corresponding database table.

U (updatable) A bullet (•) in this column updates this field in the corresponding database table.

If both the **C** and **U** columns are blank, then this element is read-only or is system-generated, and cannot be updated.

If no bullets apply to all the elements in a structure, the column does not appear.

Note: Structures may change in each product release. Please refer to the current .IDL files to verify the latest structure composition. Also note that you must update all of the elements listed under **Collections** even if only one of them has changed.

Values of some fields are not returned. These are indicated in the field descriptions. Some fields are no longer used, but remain here for compatibility with earlier versions. Other fields would require additional database queries to find their values. They have been omitted to avoid slowing down API operations.

Common Structures

Address_V1

A physical address.

P	Element	Type	R	C	U	Description
1	Blk	string<7>		•	•	the block (street) number
2	Str	string<32>	•	•	•	the street name
3	Unit	string<4>		•	•	the unit number
4	City	string<32>	•	•	•	the city name
5	Sta	string<4>	•	•	•	the state or province
6	Ctry	string<4>	•	•	•	the country
7	Pstl_Code	string<12>	•	•	•	the zip/postal code
8	Notes	string<70>		•	•	additional address information such as the dock or gate number
9	Latitude	VIS::num6_4		•	•	the latitude value
10	Longitude	VIS::num7_4		•	•	the longitude value

P	Element	Type	R	C	U	Description
11	GeoCodeOverriden_yn	VIS::vbool				indicates whether the latitude or longitude values have been manually overridden (see note following)
12	GeoCodeRecalculate_yn	VIS::vbool			•	indicates whether latitude and longitude values are to be recalculated using a distance engine (see note following)

Geocode Fields

GeoCodeRecalculate_yn is not a persistent value. It should only be used during an Update operation.

If either latitude or longitude is specified and GeoCodeRecalculate_yn is True, then an error message appears and the geocode is not be recalculated. Instead it is updated by the specified values.

AutoAppliedOption_V1

An option that is automatically selected for routing and rating when the entity it is attached to is selected. For all the elements listed here except Opt_Id, you can apply the option according to one of these levels: Rate, Route_Rate, and Do not apply.

Routing is the process of determining the eligible tariff and service combinations for a given origin and destination combination. Rating is the process of calculating freight charges for shipments, shipment legs, or loads.

Inbound and outbound are referred to in some of the following descriptions. Inbound refers to shipments or loads that are coming into a shipping location. Outbound refers to shipments or loads that are leaving a shipping location.

	Element	Type	R	C	U	Description
1	Opt_Id	string<4>			•	the option code to automatically apply for the associated entity for a given entity, you cannot define two auto-applied options with the same code.
2	Aply_Frm_AR_enu	VIS_V1::eOptApplyLvl	•	•	•	you can apply the option when the entity for which the Auto_applied_option is defined becomes a ShipFrom of an A/R shipment or A/R component.
3	Aply_Frm_AP_enu	VIS_V1::eOptApplyLvl	•	•	•	you can apply the option when the entity for which the Auto_applied_option is defined becomes a ShipFrom of a Load_leg/ load_leg_detail/load_leg_detail_comp
4	Aply_To_AR_enu	VIS_V1::eOptApplyLvl	•	•	•	you can apply the option when the entity for which the Auto_applied_option is defined becomes a ShipTo of an A/R shipment or A/R component

	Element	Type	R	C	U	Description
5	Aply_To_AP_enu	VIS_V1::eOptApplyLvl	•	•	•	you can apply the option when the entity for which the Auto_applied_option is defined becomes a ShipTo of a Load_leg/load_leg_detail/load_leg_detail_comp
6	Aply_To_Ibnd_AR_enu	VIS_V1::eOptApplyLvl	•	•	•	you can apply the option when a hub is inbound for a shipment apply it with the specified EnumOptApplyLvl level for the A/P side according to one of the three levels
7	Aply_To_Ibnd_AP_enu	VIS_V1::eOptApplyLvl	•	•	•	you can apply the option when a hub is inbound for a load (A/P shipment) apply it with the specified EnumOptApplyLvl level for the A/P side according to one of the three levels
8	Aply_To_Tsfr_AR_enu	VIS_V1::eOptApplyLvl	•	•	•	you can apply the option when a shipment is a hub transfer. apply it with the specified EnumOptApplyLvl level for the A/P side according to one of the three levels.
9	Aply_To_Tsfr_AP_enu	VIS_V1::eOptApplyLvl	•	•	•	you can apply the option when a load is a hub transfer apply it with the specified EnumOptApplyLvl level for the A/P side according to one of the three levels.
10	Aply_To_Obnd_AR_enu	VIS_V1::eOptApplyLvl	•	•	•	you can apply the option when a hub is outbound for a shipment apply it with the specified EnumOptApplyLvl level for the A/P side according to one of the three levels
11	Aply_To_Obnd_AP_enu	VIS_V1::eOptApplyLvl	•	•	•	you can apply the option when a hub is outbound for a load apply it with the specified EnumOptApplyLvl level for the A/P side according to one of the three levels

BusinessHours_V1

The hours of operation for an entity.

P	Element	Type	R	C	U	Description
1	DayLgt_Ofst	VIS::num2_1		•	•	the number of hours offset from the default business hours due to daylight savings time usually calculated based on the address of the object the business hours are attached to
2	Tm_Zn_Ofst	VIS::num3_1		•	•	the number of hours offset from the default business hours due to time zone differences usually calculated based on the address of the object the business hours are attached to
3	<i>Day_Frm/ To_Hrn_tm</i>	VIS::time		•	•	the time range pairs that indicate the open business hours there are two pairs for each day of the week. <i>Day</i> is one of Mon, Tue, Wed, Thu, Fri, Sat, or Sun <i>Frm/To</i> is either Frm (the time open from), or To (the time open to) <i>n</i> is the pair number, either 1 or 2 examples: Mon_Frm_Hr1_tm, Wed_To_Hr2_tm a range of 00:00 to 00:00 means the business is closed for this range
	Collections					
4	IgnoreHolidays	VIS::vbool				indicates whether the contents of Holidays (next) should be ignored
5	Holidays	HolidayList_V1		•	•	any holidays associated with the business hours

CEAConstraints_V1

The restrictions on the use of a carrier's equipment.

P	Element	Type	R	C	U	Description
1	Actv_yn	VIS::vbool	•	•	•	indicates whether the carrier equipment availability (CEA) constraints are active
2	Min_Lds	unsigned short		•	•	the minimum number of loads
3	Max_Lds	unsigned short		•	•	the maximum number of loads
4	PnlyMin_Lds_Amt_dlr	VIS::num15_2		•	•	the penalty paid for using fewer than the minimum number of loads
5	PnlyMax_Lds_Amt_dlr	VIS::num15_2		•	•	the penalty paid for using more than the maximum number of loads

Charge_V1

A general charge.

P	Element	Type	Description
1	Chrg_Lvl_enu	VIS_V1::eChargeLevel	the charge level
2	Unit_Typ_enu	VIS_V1::eUnitType	the unit type
3	Lkup_Unit_enu	VIS_V1::eUnitType	the lookup unit type
4	Lkup_Unit	VIS::num11_4	the lookup unit
5	Chgd_Unit	VIS::num11_4	the charged unit
6	Mnly_Ovrd_Unit	VIS::num11_4	the manual override unit
7	Chrg_Amt_dlr	VIS::num15_2	the charged amount
8	Dsct_Amt_dlr	VIS::num15_2	the discount amount
9	Mnly_Ovrd_dlr	VIS::num15_2	the manual override amount
10	Tax_Amt_dlr	VIS::num15_2	the total tax amount
11	Tot_Amt_dlr	VIS::num15_2	the total charged amount
12	Chrg_seq	VIS::num28_0	the charge sequence
13	Chrg_Detl_Typ_enu	VIS_V1::eChargeDetailType	the charge detail type
14	Adj_Lvl_enu	VIS_V1::eChrgDetlAdjLvl	the adjustment level
15	Shpm_Leg_seq	VIS::num28_0	the shipment leg ID
16	Chrg_Id	string<4>	the charge ID
17	Frht_Cls_Id	string<4>	the freight class identifier
18	Ratd_As_FC_Id	string<4>	the Rate As Freight class code

ChargeOverride_V1

An override to a charge.

P	Element	Type	R	C	U	Description
1	ChargeCode	string<4>	•	•		the ID code of the option or condition to override
2	APOptionActMode	VIS_V1::eOptApplyLvl		•	•	the option level applied to the charge for A/P
3	APOptionActModeSrc	VIS_V1::eOptOvrdLvlSrc		•	•	the source of the A/P charge override
4	AROptionActMode	VIS_V1::eOptApplyLvl		•	•	the option level applied to the charge for A/R
5	AROptionActModeSrc	VIS_V1::eOptOvrdLvlSrc		•	•	the source of the A/R charge override
6	ManualUnit	VIS::num11_4		•	•	the number of override units for the charge
7	ManualUnitSrc	VIS_V1::eOvrdLvlSrc		•	•	the source of the manual units

P	Element	Type	R	C	U	Description
8	CrtdUsr_Id	string<10>				the ID of the user who created the override
9	UpdtUsr_Id	string<10>				the ID of the user who last updated the override
10	Crtddtt	VIS::timestamp				the date and time that the override was created
11	Updt_dtt	VIS::timestamp				the date and time that the override was updated

ComponentType_V1

The type of component.

P	Element	Type	R	C	U	Description
1	Comp_Typ_Id	string<10>		•	•	the component type code this must be unique for a given entity
2	Comp_Typ_Group_cd	string<12>		•	•	the component type group code
3	Comp_Typ_Desc	string<70>			•	a description of the component type
4	RatgUnitTyp_enu	VIS_V1::eUnitType			•	how the component type should be rated the Enum values should be a subset of EnumUnitType
5	Actv_yn	VIS::vbool			•	indicates whether the object is active
6	Tare_Wgt	VIS::num11_4			•	the tare weight for the component type
7	Max_Wgt	VIS::num11_4			•	the maximum weight for the component type
8	Len	VIS::num9_3			•	the length of the component type
9	Wdth	VIS::num9_3			•	the width of the component type
10	Hght	VIS::num9_3			•	the height of the component type
11	Vol	VIS::num9_3			•	the volume of the component type the default is (Len x Wdth x Hght) you can override this, if required
12	Carr_Pck_typ	VIS_V1::tCarrPackageType			•	the carrier package type
13	Extl_cd	string<30>			•	an external identifier for this component type
14	LC_Comp_typ	VIS_V1::tEnumLCContainerType			•	the load configuration component type
15	Crtddtt	VIS::timestamp				the date and time that the component type was created
16	Updt_dtt	VIS::timestamp				the date and time that the component type was updated

P	Element	Type	R	C	U	Description
17	CrtdUsr_Id	string<10>				the ID of the user that created the component type
18	UpdtUsr_Id	string<10>				the ID of the user that updated the component type
19	Rfrncd_By_Cntr_yn	VIS::vbool				indicates whether the component type is referenced by at least one container
20	Mmo	Memo_V1		•	•	a memo for the component type

ComponentTypeGroup_V1

A group of container types.

P	Element	Type	R	C	U	Description
1	Comp_Typ_Group_cd	string<12>	•	•	•	the component type code
2	Comp_Typ_Group_Desc	string<69>	•	•	•	a description of the component types
3	UOM	Umsr_V1		•	•	the unit of measurement for the group
4	Crtd_dtt	VIS::timestamp				the date and time that the group was created
5	Updt_dtt	VIS::timestamp				the date and time that the group was updated
6	CrtdUsr_Id	string<10>				the ID of the user that created the group
7	UpdtUsr_Id	string<10>				the ID of the user that created the group
8	Rfrncd_By_Cust_yn	VIS::vbool				indicates whether the component type group is referenced by at least one customer
9	Mmo	Memo_V1		•	•	a memo for the group
10	IgnoreCompTypes	VIS::vbool				indicates whether the contents of CompTypes (next) should be ignored
11	CompTypes	ComponentType List_V1	•	•		the component types included in the group

Contact_V1 (Person)

A contact person within an organization.

P	Element	Type	R	C	U	Description
1	Role_typ	VIS_V1::tRole	•	•	•	the role of the contact
2	Name	string<70>	•	•	•	the name of the contact
3	Lang_typ	VIS_V1::tLanguage		•	•	the language the contact uses
4	Tele1	string<30>	•	•	•	the phone number of the primary contact
5	Tele2	string<30>		•	•	the phone number of the secondary contact

P	Element	Type	R	C	U	Description
6	Fax	string<30>		•	•	the fax number of the contact
7	Email	string<100>		•	•	the email address of the contact
8	URL	string<256>		•	•	the web URL of the contact

ContactInfo_V1

Contact information such as telephone and fax numbers.

P	Element	Type	R	C	U	Description
1	Tele1	string<30>	•	•	•	the phone number of the primary contact
2	Tele2	string<30>		•	•	the phone number of the secondary contact
3	Fax	string<30>		•	•	the fax number of the contact
4	Email	string<100>		•	•	the email address of the contact
5	URL	string<256>		•	•	the Web URL of the contact

CustCarrFrhtAudit_V1

Describes freight audit information in a customer and carrier relationship.

P	Element	Type	R	C	U	Description
1	Cust_cd	string<12>		•	•	the customer code optional if you are creating or updating a customer
2	Carr_cd	string<8>		•	•	the carrier code optional if you are creating or updating a carrier
3	UnMatch_Auth_enu	VIS_V1::eUnMatdFBAuth		•	•	if the customer is enabled for Unmatched Audit, then use a value of UNMATDFBAUTH_ALWAYS, otherwise use UNMATDFBAUTH_NEVER
4	Max_Var_Pos_pct	VIS::num5_2		•	•	the maximum positive percentage difference allowed in match pay if you omit a value, then the value in the carrier structure is used
5	Max_Var_Pos_Amt	VIS::num5_2		•	•	the maximum positive dollar difference allowed in match pay if you omit a value, then the value in the carrier structure is used

P	Element	Type	R	C	U	Description
6	Max_Var_Neg_pct	VIS::num5_2		•	•	the maximum negative percentage difference allowed in match pay if you omit a value, then the value in the carrier structure is used
7	Max_Var_Neg_Amt	VIS::num5_2		•	•	the maximum negative dollar difference allowed in match pay if you omit a value, then the value in the carrier structure is used

CustShpgLocXRef_V1

A cross-reference for a customer and shipping location.

P	Element	Type	R	C	U	Description
1	Cust_cd	string<12>		•	•	the customer code optional if you are creating or updating a customer
2	Shpg_Loc_cd	string<16>		•	•	the shipping location code optional if you are creating or updating a shipping location
3	Shpg_Loc_Typ_enu	VIS_V1::eShip PointType		•	•	the shipping location type optional if you are creating or updating a shipping location
4	Rel_Typ_enu	VIS_V1::eCustLoc RelType		•	•	the relationship type between the customer and the shipping location
5	IgnoreExternalAliases	VIS::vbool				indicates whether this customer shipping location cross-reference has an external ID list
6	ExternalAliases	ExternalAliasList_V1		•	•	a list of external IDs for this cross-reference

GeoArea_V1

The Geo_Area_c business object, which describes the geographic area covered by a zone.

P	Element	Type	R	C	U	Description
1	Usr_Pstl_cd_from	string<12>		•	•	the lower limit of the zip/postal code range of the geographic area
2	Usr_Pstl_cd_to	string<12>		•	•	the upper limit of the zip/postal code range of the geographic area
3	Dist	VIS::num5_0		•	•	the size of the radius around an origin geographic area

P	Element	Type	C	U	Description
4	CtyPrc_Zip_yn	VIS::vbool	•	•	indicates whether the system will choose lanes based on the city over lanes based on the zip/postal code, when considering lanes for routing and rating
5	City_Name	string<33>	•	•	the name of the city *ALL means multiple cities, as defined by the other fields
6	Sta_cd	string<4>	•	•	the state or province ID of the geographic area
7	GeoArea_Prcn_enu	VIS_V1::eGeoAreaPrecision	•	•	the precision of the geographic area
8	UMsr_Sys_enu	VIS_V1::eUnit MeasureSys	•	•	the measurement system for the geographic area
9	UMsr_Dst_enu	VIS_V1::eUnit MeasureDst	•	•	the unit of measure of the distances for the geographic area

DomicileEquipment_V1

The equipment used at a domicile.

P	Element	Type	R	C	U	Description
1	Eqmt_Typ_cd	string<4>	•	•		the equipment type identifier
2	Min_Lds	unsigned short		•	•	the minimum number of loads
3	Max_Lds	unsigned short		•	•	the maximum number of loads
4	PnltyMin_LdsAmt_dlr	VIS::num15_2		•	•	the penalty paid for using fewer than the minimum number of loads
5	PnltyMax_Lds_Amt_dlr	VIS::num15_2		•	•	the penalty paid for using more than the maximum number of loads
6	Max_Trvl_Dst	VIS::num15_2		•	•	the maximum travel distance allowed
7	AvailStrt_dtt	VIS::timestamp		•	•	the earliest available start date and time
8	StrtBy_dtt	VIS::timestamp		•	•	the date and time the equipment must start to be used
9	Reuse_yn	VIS::vbool	•	•	•	indicates whether the equipment can be reused

ExternalAlias_V1

An external alias.

P	Element	Type	C	U	Description
1	External_Id	string<16>	•	•	an external alias a customer can use for a shipping location
2	External_Loc_Desc	string<70>	•	•	a description of this external location
3	Preferred_yn	VIS::vbool	•	•	indicates whether a customer uses this external ID as a default

Holiday_V1

An exception to the normal hours of operation defined by a `BusinessHours_V1` structure.

P	Element	Type	R	C	U	Description
1	Day_dt	VIS::date	•	•		the date of the holiday a given <code>BusinessHours_V1</code> object cannot have two associated holidays with the same date
2	Frm_Hr_tm	VIS::time	•	•	•	the start of the time span of the holiday
3	To_Hr_tm	VIS::time	•	•	•	the end of the time span of the holiday

LaneAvailability_V1

The restrictions on the number of loads allowed on certain lanes for specific pieces of a carrier's equipment.

P	Element	Type	R	C	U	Description
1	Actv_yn	VIS::vbool	•	•	•	indicates whether the lane is active
2	Tff_Id	VIS::num28_0		•		the tariff ID
3	Srvc_Cd	string<4>		•		the service code
4	Orig_Zn_Id	string<8>		•		the origin identifier if the origin is a zone you must use either <code>Orig_Zn_Id</code> or <code>Orig_Hub_Id</code> , but not both
5	Orig_Hub_Id	string<16>		•		the origin identifier if the origin is hub you must use either <code>Orig_Zn_Id</code> or <code>Orig_Hub_Id</code> , but not both
6	Dest_Zn_Id	string<8>		•		the destination identifier if the origin is a zone you must use either <code>Dest_Zn_Id</code> and <code>Dest_Hub_Id</code> , but not both
7	Dest_Hub_Id	string<16>		•		the destination identifier if the origin is a hub you must use either <code>Dest_Zn_Id</code> and <code>Dest_Hub_Id</code> , but not both
8	Min_Lds	unsigned short		•	•	the minimum number of loads
9	Max_Lds	unsigned short		•	•	the maximum number of loads
10	Start_dtt	VIS::timestamp		•	•	the start date and time
11	End_dtt	VIS::timestamp		•	•	the end date and time
12	Carr_Cd	string<8>				the carrier code
13	Crtd_Usr_cd	string<10>				the ID of the user who created this lane availability
14	Crtd_dtt	VIS::timestamp				the date and time that this lane availability was created

P	Element	Type	R	C	U	Description
15	Updt_Usr_cd	string<10>				the ID of the user who most recently updated this lane availability
16	Updt_dtt	VIS::timestamp				the date and time that this lane availability was most recently updated

Memo_V1

Information that can be attached to objects.

P	Element	Type	R	C	U	Description
1	Prtb_Cnt	string<2000>		•	•	this string can be on printed documentation
2	Non_Prtb_Cnt	string<32000>		•	•	this string is for viewing only

NamedResult

A generic structure representing the result of an operation. Each structure element shows the name, type, and description.

`LoadSrcv::SetAllToPlanned` returns a sequence of `NamedResult` structures (`NamedResultList`) which reports the success of each load processed.

P	Element	Type	Description
1	Id	string	an identifier in <code>LoadSrcv::SetAllToPlanned</code> , this field contains the plan ID
2	Desc	string	a description or qualifier in <code>LoadSrcv::SetAllToPlanned</code> , this field contains the load ID
3	Success_yn	VIS::vbool	a boolean indicator in <code>LoadSrcv::SetAllToPlanned</code> , this field is <code>VIS::bTRUE</code> if the load was successfully set to planned status

PayableCarrOvr_V1

A payable carrier override.

P	Element	Type	R	C	U	Description
1	Chrg_Cd	string<4>	•	•	•	the charge code
2	Chrg_Desc	string<69>				a description of this charge
3	Payable_Carr_enu	VIS_V1::ePayableCarr	•	•	•	the payable carrier type
4	Carr_cd	string<8>	•	•	•	the carrier code
5	CrtdUsr_Id	string<10>				the ID of the user who created this override
6	Crted_dtt	VIS::timestamp				the date and time that this override was created

P	Element	Type	R	C	U	Description
7	UpdtUsr_Id	string<10>				the ID of the user who last updated this override
8	Updt_dtt	VIS::timestamp				the date and time that this override was last updated

POD_V1

The information about a Proof of Delivery

P	Element	Type	R	Description
1	POD_Desc	string<70>	•	a description of this POD
2	POD_dtt	VIS::timestamp	•	the delivery date and time
3	POD_Tot_Wgt	VIS::num17_4		the total weight delivered
4	POD_Tot_Pce	VIS::num7_0		the total number of pieces delivered
5	POD_Tot_Skid	VIS::num7_0		the total number of skids delivered
6	POD_Sign	string<32>		the name or ID code of the employee signing for the delivery
7	POD_Usr_cd	string<10>		the user code
8	Extl_cd	string<30>		an external identifying code

RefNumber_V1

The number that an external entity such as a customer or carrier supplies to help identify its order.

P	Element	Type	R	C	U	Description
1	Rfrnc_Num_Typ	VIS_V1::tReference NumType	•	•		a domain table value indicating the reference number qualifier type for the reference number
2	Rfrnc_Num	string<30>	•	•	•	the value of the reference number or code

RspbCustOvr_V1

A responsible customer override.

P	Element	Type	R	C	U	Description
1	Chrg_Cd	string<4>	•	•	•	the charge code
2	Chrg_Desc	string<69>				a description of this charge
3	Rspb_Cust_enu	VIS_V1::eResponsibleCust	•	•	•	the responsible customer type
4	Cust_cd	string<12>	•	•	•	the customer code
5	CrtDUsr_Id	string<10>				the ID of the user who created this override

P	Element	Type	R	C	U	Description
6	Crted_dtt	VIS::timestamp				the date and time that this override was created
7	UpdtUsr_Id	string<10>				the ID of the user who last updated this override
8	Updt_dtt	VIS::timestamp				the date and time that this override was last updated

SEC_V1

The reason code (service exception code) for an event.

P	Element	Type	R	C	U	Description
1	SEC_cd	string<5>		•	•	the service event code
2	Movmnt_dtt	VIS::timestamp		•	•	the event date and time
3	CityName	string<32>			•	the city of the event
4	State_cd	string<4>			•	the state or province of the event
5	Country_cd	string<3>			•	the country of the event
6	Appt1_From_dtt	VIS::timestamp			•	the start date and time of the appointment at the origin
7	Appt1_To_dtt	VIS::timestamp			•	the end date and time of the first appointment at the origin
8	Ovrd_Hrs_Op_Appt1_yn	VIS::vbool			•	indicates whether the appointment at the origin overrides the location's hours of operation
9	Appt2_From_dtt	VIS::timestamp			•	the start date and time of the appointment at the destination
10	Appt2_To_dtt	VIS::timestamp			•	the end date and time of the appointment at the destination
11	Ovrd_Hrs_Op_Appt2_yn	VIS::vbool			•	indicates whether the appointment at the destination overrides the location's hours of operation
12	Estmd1_dtt	VIS::timestamp			•	the estimated first date and time
13	Estmd2_dtt	VIS::timestamp			•	the estimated second date and time
14	Carr_cd	string<8>			•	the carrier code
15	Tdr_Acceptd_By_Name	string<32>			•	the name of the employee accepting the tender
16	Trailer_Num	string<24>			•	the trailer number
17	Driver	string<8>			•	the driver code or name
18	Seal_Number	string<16>			•	the seal number

P	Element	Type	R	C	U	Description
19	Team_Drvr_yn	VIS::vbool		•		indicates whether driver is a team driver
20	Sus_Func_enu	VIS_V1::eSuspendFunction		•		the suspended status
21	Ld_Schd_Cmpd_yn	VIS::vbool		•		indicates whether the load scheduling is completed

ShippingInfo_V1

Information about a shipment.

P	Element	Type	R	C	U	Description
1	SclD_Wgt	VIS::num11_4				the scaled weight of the shipment
2	Vol	VIS::num11_4		•	•	the volume of the shipment
3	Odr_Val_dlr	VIS::num15_2		•	•	the total order value of the shipment
4	DclD_Val_dlr	VIS::num15_2		•	•	the total declared value of the shipment
5	Nmnl_Wgt	VIS::num11_4				the total nominal weight of the shipment
6	Tot_Tare_Wgt	VIS::num17_4				the total tare weight of the shipment
7	Tot_Pce	VIS::num7_0				the total number of pieces in the shipment
8	Tot_Skid	VIS::num7_0				the total number of skids in the shipment
9	Ldn_Len	VIS::num9_3		•	•	the total laden length for the shipment

StopEvent_V1

The information about a stop event.

P	Element	Type	R	Description
1	Event_cd	string<5>	•	the event code
2	Movmnt_dtt	VIS::timestamp	•	the event date and time
3	EstmdStop_dtt	VIS::timestamp		the estimated delivery date and time (ETA)

UMsr_V1

A collection of units of measurement.

P	Element	Type	R	C	U	Description
1	UMsr_Sys_enu	VIS_V1::eUnitMeasureSys	•	•	•	the system of measurement units for the entity
2	UMsr_Wgt_enu	VIS_V1::eUnitMeasureWgt	•	•	•	the unit of measure for weight

P	Element	Type	R	C	U	Description
3	UMsr_Len_enu	VIS_V1::eUnitMeasureLgt	•	•	•	the unit of measure for length
4	UMsr_Dst_enu	VIS_V1::eUnitMeasureDst	•	•	•	the unit of measure for distance

Delivery Schedule Structures

An hollow bullet (°) in the R column indicates that the arrival date and time is mandatory for every point except the first. It also indicates that the departure date and time is mandatory for every point except the last.

BusinessDays_V1

The business days for the delivery schedules.

P	Element	Type	R	C	U	Description
1	Business_Days_Cd	string<5>	•	•		the business days code
2	Business_Days_Desc	string<70>	•	•	•	the business days description
3	Business_Day_Sun_yn	VIS_V1::vbool	•	•	•	indicates whether Sunday is a business day
4	Business_Day_Mon_yn	VIS_V1::vbool	•	•	•	indicates whether Monday is a business day
5	Business_Day_Tue_yn	VIS_V1::vbool	•	•	•	indicates whether Tuesday is a business day
6	Business_Day_Wed_yn	VIS_V1::vbool	•	•	•	indicates whether Wednesday is a business day
7	Business_Day_Thu_yn	VIS_V1::vbool	•	•	•	indicates whether Thursday is a business day
8	Business_Day_Fri_yn	VIS_V1::vbool	•	•	•	indicates whether Friday is a business day
9	Business_Day_Sat_yn	VIS_V1::vbool	•	•	•	indicates whether Saturday is a business day

FixedPntTimeTable_V1

A timetable point based on a fixed date.

P	Element	Type	R	C	U	Description
1	Dlvy_Schd_Cd	string<15>	•	•		the delivery schedule code
2	Itnr_TmTbl_Cd	string<15>	•	•		the itinerary timetable code
3	Itnr_TmTbl_Ent_Cd	string<15>	•	•		the itinerary timetable entry code
4	Itnr_Pnt_Seq_num	VIS::short	•	•		the itinerary point sequential number
5	Arvl_Tm_Frm_dtt	VIS::timestamp	°	•	•	the start of the arrival time range
6	Arvl_Tm_To_dtt	VIS::timestamp	°	•	•	the end of the arrival time range
7	Dptr_Tm_Frm_dtt	VIS::timestamp	°	•	•	the end of the departure time range

P	Element	Type	R	C	U	Description
8	Dptr_Tm_To_dtt	VIS::timestamp	°	•	•	the start of the departure time range
9	Arvl_dt	VIS::date	°	•	•	the arrival date
10	Dptr_dt	VIS::date	°	•	•	the departure date

DayWeekPntTimeTable_V1

A timetable point based on a day of the week.

P	Element	Type	R	C	U	Description
1	Dlvy_Schd_Cd	string<15>	•	•		the delivery schedule code
2	Itnr_TmTbl_Cd	string<15>	•	•		the itinerary timetable code
3	Itnr_TmTbl_Ent_Cd	string<15>	•	•		the itinerary timetable entry code
4	Itnr_Pnt_Seq_num	VIS::short	•	•		the itinerary point sequential number
5	Arvl_Tm_Frm_dtt	VIS::timestamp	°	•	•	the start of the arrival time range
6	Arvl_Tm_To_dtt	VIS::timestamp	°	•	•	the end of the arrival time range
7	Dptr_Tm_Frm_dtt	VIS::timestamp	°	•	•	the end of the departure time range
8	Dptr_Tm_To_dtt	VIS::timestamp	°	•	•	the start of the departure time range
9	Arvl_week	VIS::short	°	•	•	the arrival week: the number of weeks between the departure and arrival
10	ArvlDayOfWk_enu	VIS_V1::eDayOfWeek	°	•	•	the arrival day of the week
11	DptrDayOfWk_enu	VIS_V1::eDayOfWeek	°	•	•	the departure day of the week

ElapsedPntTimeTable_V1

A timetable point based on elapsed business days.

P	Element	Type	R	C	U	Description
1	Dlvy_Schd_Cd	string<15>	•	•		the delivery schedule code
2	Itnr_TmTbl_Cd	string<15>	•	•		the itinerary timetable code
3	Itnr_TmTbl_Ent_Cd	string<15>	•	•		the itinerary timetable entry code
4	Itnr_Pnt_Seq_num	VIS::short	•	•		the itinerary point sequential number
5	Arvl_Tm_Frm_dtt	VIS::timestamp	°	•	•	the start of the arrival time range
6	Arvl_Tm_To_dtt	VIS::timestamp	°	•	•	the end of the arrival time range
7	Dptr_Tm_Frm_dtt	VIS::timestamp	°	•	•	the end of the departure time range

8	Dptr_Tm_To_dtt	VIS::timestamp	°	•	•	the start of the departure time range
9	Elapsed_days	VIS::short	•	•	•	the number of elapsed business days

ItnrTimeTableEntry_V1

A timetable entry.

A hollow bullet (°) indicates the list should have at least two entries.

P	Element	Type	R	C	U	Description
1	Dlvy_Schd_Cd	string<15>	•	•		the delivery schedule code
2	Itnr_TmTbl_Cd	string<15>	•	•		the itinerary timetable code
3	Itnr_TmTblEnt_Cd	string<15>	•	•		the itinerary timetable entry code
4	TmTbl_Ent_Desc	string<70>	•	•	•	a description of the timetable entry
5	TmTbl_DateBS_enu	VIS_V1::eDlvySchdDateBs	•	•	•	the timetable entry date basis
	Collections					
6	IgnoreFixedPnt	VIS::vbool				indicates whether the contents of the following sequence should be ignored
7	Fixd_Pnt_TmTbIs	FixedPntTimeTableList_V1	•	°	•	the fixed date-based timetables of this timetable entry
8	IgnoreDayWeek	VIS::vbool				indicates whether the contents of the following sequence should be ignored
9	DayWk_PntTmTbIs	DayWeekPntTimeTableList_V1	•	°	•	the day of the week timetables of this timetable entry
10	IgnoreElapsed	VIS::vbool				indicates whether the contents of the following sequence should be ignored
11	Elpd_Pnt_TmTbIs	ElapsedPntTimeTableList_V1	•	°	•	the elapsed business days timetables of this timetable entry

ItineraryPoint_V1

An itinerary point.

P	Element	Type	R	C	U	Description
1	Itnr_Pnt_Id	VIS::num28_0				the Itinerary point ID
2	Itnr_Pnt_Desc	string<70>	•	•	•	a description of the itinerary point
3	Itnr_Pnt_Seq_num	VIS::short	•	•		the sequential number of the point in the itinerary
4	Shpg_Loc_Typ_enu	VIS_V1::eItnrPointType	•	•	•	the point type: shipping location or zone
5	Dlvy_Schd_Cd	string<15>	•	•		the code of the delivery schedule associated with the point

P	Element	Type	R	C	U	Description
6	Itnr_Cd	string<15>	•	•		the code of the itinerary associated with the point
7	ItnrPntZone_Loc_Cd	string<17>	•	•	•	the code of the itinerary zone or shipping location

ItnrTimeTable_V1

An itinerary timetable.

P	Element	Type	R	C	U	Description
1	Dlvy_Schd_Cd	string<15>	•	•		the delivery schedule code
2	Itnr_TmTbl_Cd	string<15>	•	•		the itinerary timetable code
3	Itnr_TmTbl_Desc	string<70>	•	•	•	a description of the itinerary timetable
4	Num_Of_Itnr_Points	VIS::short		•	•	the number of itinerary points
5	CrtdUsr_Id	string<10>				the ID of the user who created this delivery schedule
6	UpdtUsr_Id	string<10>				the ID of the user who last updated this delivery schedule
	Collections					
7	IgnoreTmTblEntries	VIS::vbool				indicates whether the contents of the following sequence should be ignored
8	TimeTable_Entries	ItnrTimeTableEntryList_V1		•	•	the timetable entries of this timetable

Itinerary_V1

A delivery schedule itinerary.

P	Element	Type	R	C	U	Description
1	Dlvy_Schd_Cd	string<15>	•	•		the delivery schedule code
2	Itnr_Cd	string<15>	•	•		a unique identifier for this Itinerary
3	Itnr_Desc	string<70>	•	•	•	the itinerary description
4	Status_enu	VIS_V1::eltnrStatus		•	•	the itinerary status: active or inactive
5	CrtdUsr_Id	string<10>				the ID of the user who created this delivery schedule
6	UpdtUsr_Id	string<10>				the ID of the user who last updated this delivery schedule
7	Itnr_TmTbl_Cd	string<15>		•		the code of the associated itinerary timetable
8	Itnr_TmTbl_DS_Cd	string<15>		•		the code of the Delivery schedule for the itinerary timetable

P	Element	Type	R	C	U	Description
	Collections					
9	IgnoreItnrPointsy	VIS::vbool				indicates whether the contents of the following sequence should be ignored
10	Itinerary_Points	ItineraryPointList_V1		•	•	the itinerary points associated with this itinerary
11	IgnoreLanePerfs	VIS::vbool				indicates whether the contents of the following sequence should be ignored
12	Lane_Performances	ItnrLanePerformanceList_V1		•	•	the lane performances associated with this itinerary

DeliverySchedule_V1

A delivery schedule.

P	Element	Type	R	C	U	Description
1	Dlvy_Schd_Cd	string<15>	•	•		the delivery schedule code
2	Dlvy_Schd_Desc	string<70>	•	•	•	the delivery schedule description
3	DftLoc_enu	VIS_V1::eDlvySchdLocBs	•	•	•	the basis of the default delivery schedule location: zone or location
4	PrimDate_enu	VIS_V1::eDlvySchdDateBs	•	•		the basis of the primary delivery schedule date: fixed, week day, elapsed business days
5	Enfc_Geo_Hier_yn	VIS::vbool	•	•	•	indicates whether to enforce the rules of geographic hierarchy
6	CrtdUsr_Cd	string<10>				the ID of the user who created this delivery schedule
7	UpdtUsr_Cd	string<10>				the ID of the user who last updated this delivery schedule
8	Mmo	Memo_V1		•	•	a memo associated with this delivery schedule
9	Business_Days	BusinessDays_V1		•	•	the business days mnemonic associated with this delivery schedule
	Collections					
10	IgnoreItineraries	VIS::vbool				indicates whether the contents of the following sequence should be ignored
11	Itneraries	ItineraryList_V1		•	•	the itineraries associated with this delivery schedule

P	Element	Type	R	C	U	Description
12	IgnoreItnrTmTbls	VIS::vbool				indicates whether the contents of the following sequence should be ignored
13	Itnr_TimeTables	ItnrTimeTableList_V1		•	•	itinerary timetables associated with this delivery schedule

ItnrLanePerformance_V1

An itinerary lane performance.

P	Element	Type	R	C	U	Description
1	Tff_Id	VIS::num28_0	•	•		the associated delivery schedule ID
2	Srvc_Cd	string<5>	•	•		the associated service code
3	Frm_Itnr_Pnt_Seq_num	VIS::short	•	•		the sequential number of the From point in the itinerary
4	Frm_ShpG_Loc_Typ_enu	VIS_V1::eItnrPointType	•	•		the From point type: shipping location or zone
5	Frm_ItnrPntZone_Loc_Cd	string<17>	•	•		the code of the From itinerary zone or shipping location
6	To_Itnr_Pnt_Seq_num	VIS::short	•	•		the sequential number of the To point in the itinerary
7	To_ShpG_Loc_Typ_enu	VIS_V1::eItnrPointType	•	•		the To point type: shipping location or zone
8	To_ItnrPntZone_Loc_Cd	string<17>	•	•		the code of the To itinerary zone or shipping location
9	Dlvy_Schd_Cd	string<15>	•	•		the associated delivery schedule code
10	Itnr_Cd	string<15>	•	•		the associated itinerary code
11	Performance	Perf_V1	•	•	•	the performance data

Perf_V1

Numeric data for the itinerary lane performance.

P	Element	Type	R	C	U	Description
1	Usr_Dfd_SrvcGrd	VIS::num5_2		•	•	the associated delivery schedule ID
2	Sys_Gen_SrvcGrd	VIS::num5_2		•	•	the associated service code
3	Usr_Grd_Hdl_enu	VIS_V1::eUsrDfdSrvcGrdHdl		•	•	the sequential number of the From point in the itinerary
4	Num_of_trans	Unsigned long		•	•	a counter for the number of transactions

Entity Structures

Carrier_V1

A company that arranges shipping services for customers using one or more shippers.

P	Element	Type	R	C	U	Description
1	Carr_Id	string<8>	•	•		the ID code of the carrier
2	Carr_Desc	string<70>	•	•		a description of the carrier
3	Lang_typ	VIS_V1::tLanguage		•	•	the language used by the carrier
4	Carr_typ	VIS_V1::tCarrierType		•	•	the type of the shipping service that the carrier provides
5	SCAC_typ	VIS_V1::tSCAC	•	•	•	the standard carrier alpha code assigned to the carrier by NMFTA
6	Ctrc_ID_cd	string<24>		•	•	the contract number between the company and the carrier
7	Ctrc_dt	VIS::date		•	•	the date that the above contract was made
8	Cost_Ctr_typ	VIS_V1::tCostCenter	•	•	•	the G/L cost center to be charged when transactions occur between the company and the carrier
9	Acc_Num_cd	string<12>		•	•	the account number opened for the carrier
10	AP_Vend_Num_cd	string<12>		•	•	the carrier's vendor number in the company's accounts payable system
11	Vchr_Gen_Lvl_enu	VIS::eVchr GenLvlAP		•	•	the voucher generation level
12	FB_Grp_Lvl_enu	VIS::eFrht BillGrpLvl		•	•	the freight bill group level
13	FB_Pymt_Md_enu	VIS::eFrht InvcPymtMode		•	•	the freight bill payment mode
14	Cncy_typ	VIS_V1::tCurrency		•	•	the currency used by the carrier
15	Comp_Trkg_yn	VIS::vbool		•	•	indicates whether the boxes or pieces in a shipment can be tracked if true, each component gets a separate tracking number
16	Brcd_typ	VIS_V1::tBarcodeType		•	•	the bar code type used in bar code printing for the carrier
17	Prt_Lbl_yn	VIS::vbool		•	•	indicates whether shipping labels are generated

P	Element	Type	R	C	U	Description
18	Prt_BOL_yn	VIS::vbool		•	•	indicates whether bills of lading are generated a carrier bill of lading lists the contents of an entire shipment itinerary a shipment itinerary is the path that a shipment follows from its origin to its destination
19	Prt_Mnft_yn	VIS::vbool		•	•	indicates whether a manifest is printed each time that the carrier is assigned a shipment
20	Prt_Frht_Bill_yn	VIS::vbool		•	•	indicates whether a freight bill is generated
21	Athz_Apt_yn	VIS::vbool		•	•	indicates whether the carrier is authorized to make an appointment with load-ats or consignees to pick up or deliver directly to them
22	Fax_Ebl_yn	VIS::vbool		•	•	indicates whether the carrier has online fax communication capability
23	Pkup_Led_Tm_typ	short		•	•	the type of pickup lead time always returned as 0
24	Ins_Exp_dt	VIS::date	•	•	•	the date when the carrier's insurance expires
25	Ins_Amt_dlr	VIS::num15_2		•	•	the value of the carrier's insurance policy
26	Sgl_Tdr_Need_yn	VIS::vbool		•	•	indicates whether a single tender is required for the carrier
27	Rout_Prcn_enu	VIS_V1::eRoutePrecision		•	•	indicates how shipments are routed for the carrier
28	BOL_Fmt_typ	VIS_V1::tBOLFormat		•	•	the format for the bill of lading (BOL)
29	Shp_Lbl_Fmt_typ	VIS_V1::tShipLabelFormat		•	•	the format for printing shipping labels
30	Mnft_Fmt_typ	VIS_V1::tCarrMnftFmt		•	•	the format for printing manifests
31	FrhtBill_Fmt_typ	VIS_V1::tAPInvoiceFormat		•	•	the format for printing the carrier's invoices
32	MaxPayVarPos_pct	VIS::num5_2		•	•	the maximum positive percentage difference allowed in match pay
33	MaxPayVarPos_dlr	VIS::num15_2		•	•	the maximum positive dollar difference allowed in match pay
34	MaxPayVarNeg_pct	VIS::num5_2		•	•	the maximum negative percentage difference allowed in match pay

P	Element	Type	R	C	U	Description
35	MaxPayVarNeg_dlr	VIS::num15_2		•	•	the maximum negative dollar difference allowed in match pay
36	Srvc_Grd_typ	short		•	•	the grade for the service provided by the carrier
37	Mino_Grp_typ	VIS_V1::tMinorityGroup		•	•	indicates whether the carrier is owned by a visible minority, and by which minority
38	Mnft_Num_Defd_yn	VIS::vbool		•	•	indicates whether to use the carrier manifest number
39	FB_Num_Defd_yn	VIS::vbool		•	•	indicates whether to use the carrier freight bill number
40	BOLNum_Fmt_cd	string<24>		•	•	the BOL format code
41	BOLNum_EntAut_yn	VIS::vbool		•	•	indicates validation of the BOL number
42	Slc_Tff_Md_enu	VIS_V1::eSlcTffMd		•	•	the tariff selection control
43	Rstd_Carr_yn	VIS::vbool		•	•	indicates whether the carrier is restricted
44	Enrt_Cmnt	unsigned short		•	•	the enroute communications used by the carrier
45	Stat_Vrfc_enu	VIS_V1::eStatusVrfc		•	•	the status verification control
46	AP_Trms_typ	VIS_V1::tAPterms		•	•	the A/P terms for the carrier
47	APVchrCncy_enu	VIS_V1::eAPVchrCncy	•	•	•	the currency type of the A/P voucher payment
48	Tdr_Rsps_hrs	VIS::num6_2		•	•	the maximum number of hours the carrier can take to respond to the tender offers
49	NonLive_Ld_Unld_yn	VIS::vbool	•	•	•	indicates whether non-live loading and unloading is allowed for this carrier
50	Elgb_Cnts_Mv_yn	VIS::vbool		•	•	indicates whether the carrier can use continuous moves
51	Elgb_Cnts_Mv_PInd_yn	VIS::vbool		•	•	indicates whether the carrier can use loads in planned, open or tender-rejected status when creating continuous moves
52	Elgb_Cnts_Mv_Tdrd_yn	VIS::vbool		•	•	indicates whether the carrier can use loads in tendered status when creating continuous moves
53	Elgb_Cnts_Mv_Tdr_Acpd_yn	VIS::vbool		•	•	indicates whether the carrier can use loads in tender accepted status when creating continuous moves
54	Elgb_Cnts_Mv_Cfmd_yn	VIS::vbool		•	•	indicates whether the carrier can use loads in confirmed or in-transit status when creating continuous moves

P	Element	Type	R	C	U	Description
55	Elgb_Cnts_Mv_Cpld_yn	VIS::vbool		•	•	indicates whether the carrier can use loads in completed status when creating continuous moves
56	Vchr_Gen_Lvl_Trip_enu	VIS_V1::eVchrGenLvlTrip		•	•	the voucher generation level for trips
57	FA_Detl_Chrg_yn	VIS::vbool		•		indicates whether detailed charges are required for freight audit for this carrier the system assigns a default value of false to this field
58	Umsr_Sys_enu	VIS_V1::eUnit MeasureSys		•	•	the unit of measure system of the carrier
59	Umsr_Wgt_enu	VIS_V1::eUnit MeasureWgt	•	•	•	the weight units of the carrier
60	Umsr_Len_enu	VIS_V1::eUnit MeasureLgt	•	•	•	the length units of the carrier
61	UMsr_Dst_enu	VIS_V1::eUnit MeasureDst		•	•	the unit of measure for the distances
62	Intl_Carr_yn	VIS::vbool		•	•	indicates whether the carrier is internal
63	Spot_Carr_yn	VIS::vbool		•	•	indicates whether the carrier is a spot carrier
64	Max_Alw_LwstCost_pct	VIS::num5_2		•	•	the maximum percentage of the lowest cost allowed for the carrier
65	Corp1_Id	string<30>		•	•	a corporate identifier code
66	Corp2_Id	string<30>		•	•	a second corporate identifier code
67	PyTo_Id	string<8>		•	•	the carrier that pays for the transportation services provided
68	Div_Id	string<4>		•	•	the division ID of the carrier if you do not supply an input value for this field when creating this structure, then the value will be taken from the user's profile to force this value to be blank, enter a single space (" ")
69	Extl_cd1	string<30>		•	•	an external identifying code
70	Extl_cd2	string<30>		•	•	a second external identifying code
71	Stat_enu	VIS_V1::eStatus				the status of the carrier
72	Crted_dtt	VIS::timestamp				the date and time that the carrier was created
73	Updt_dtt	VIS::timestamp				the date and time that the carrier was updated
74	CrtedUsr_Id	string<10>				the ID of the user who created the carrier
75	UpdtUsr_Id	string<10>				the ID of the user who updated the carrier

P	Element	Type	R	C	U	Description
76	DomicileIDs	VIS::StrIdList				the domicile IDs of the carrier
77	Addr	Address_V1	•	•	•	the address of the carrier
78	CntcInf	ContactInfo_V1	•	•	•	the contact information of the carrier
79	BusHours	BusinessHours_V1		•	•	the business hours of the carrier
80	Mmo	Memo_V1		•	•	a memo associated with the carrier
81	InsNote	Memo_V1		•	•	insurance notes associated with the carrier
82	CEA_Cnstrts	CEAConstraints_V1		•	•	the equipment constraints for the carrier
	Collections					
83	IgnoreConts	VIS::vbool				indicates whether the contents of Conts (next) should be ignored
84	Conts	ContactList_V1		•	•	contacts associated with the carrier
85	IgnoreCustCarrFrhtAudits	VIS::vbool				indicates whether the contents of CustCarrFrhtAudits (next) should be ignored
86	CustCarrFrhtAudits	CustCarrFrhtAuditList_V1		•	•	the customer and carrier relationship for freight audit
87	IgnoreLaneAvailabilities	VIS::vbool				indicates whether the LaneAvailabilities values should be ignored
88	LaneAvailabilities	LaneAvailabilityList_V1		•	•	the lane availabilities for this carrier

Consignee_V1

The final destination of a freight movement.

P	Element	Type	R	C	U	Description
1	Cnse_Id	string<16>	•	•		the consignee ID code
2	Cnse_Desc	string<70>	•	•	•	a description of the consignee
3	Lang_typ	VIS_V1::tLanguage		•	•	the language used at the consignee
4	Apt_Rqrd_yn	VIS::vbool		•	•	indicates whether a carrier has to make an appointment to drop off shipments at the consignee
5	Max_Shpm_Wgt	VIS::num11_4		•	•	the maximum weight of a shipment that can be unloaded at the consignee
6	Max_Shpm_Vol	VIS::num11_4		•	•	the maximum volume of a shipment that can be unloaded at the consignee

P	Element	Type	R	C	U	Description
7	Max_Eqmt_Len	VIS::num9_3		•	•	the maximum length of the equipment allowed at the consignee
8	Max_Eqmt_Wdth	VIS::num9_3		•	•	the maximum width of the equipment allowed at the consignee
9	Max_Eqmt_Hght	VIS::num9_3		•	•	the maximum height of the equipment allowed at the consignee
10	Cnse_Grp_typ	VIS_V1::tConsigneeGroup		•	•	the consignee group of the consignee
11	Stor_Num	string<10>		•	•	a store number
12	GL_Cat_typ	VIS_V1::tGLCategory		•	•	the G/L category
13	Fixd_Ld_UnLd_hrs	VIS::num6_2		•	•	the fixed load or unload hours
14	UnLd_UnitTyp_enu	VIS_V1::eLoadUnitType		•	•	the unload unit type
15	Var_UnLd_hrs	VIS::num6_2		•	•	the variable load hours
16	Max_Ld_UnLd_hrs	VIS::num6_2		•	•	the maximum load or unload hours
17	Min_Lead_hrs	VIS::num6_2		•	•	the minimum lead hours
18	Min_Ld_UnLd_hrs	VIS::num6_2		•	•	the minimum load or unload hours
19	Live_Ld_Typ_enu	VIS_V1::eLive LoadType		•	•	indicates whether live or non-live loading is allowed if non-live, then it indicates whether there can be non-live loading during business hours or off hours
20	Frst_Pick_yn	VIS::vbool		•	•	indicates whether any shipment leg starting from this location must be picked up at the first stop for the load must be false for consignees
21	Last_Drp_yn	VIS::vbool		•	•	indicates whether any shipment leg terminating at this location must be dropped off at the last stop for the load
22	Bdry_Rule_enu	VIS_V1::eBoundaryRule		•	•	the boundary rules
23	Ld_Unit_Typ_enu	VIS_V1::eLoadUnitType		•	•	the load unit type
24	Var_Ld_hrs	VIS::num6_2		•	•	the variable load hours
25	Corp1_Id	string<30>		•	•	a corporate identifier code
26	Corp2_Id	string<30>		•	•	a second corporate identifier code

P	Element	Type	R	C	U	Description
27	Lgst_Grp_Id	string<4>		•	•	the logistics group of the consignee if you do not supply an input value for this field when creating this structure, then the value will be taken from the user's profile to force this value to be blank, enter a single space (" ")
28	Div_Id	string<4>		•	•	the division of the consignee if you do not supply an input value for this field when creating this structure, then the value will be taken from the user's profile to force this value to be blank, enter a single space (" ")
29	PrefCarr_Id	string<4>		•	•	the default carrier ID
30	PrefSrvc_Id	string<4>		•	•	the default service ID
31	HubRgnZn_Id	string<8>		•	•	the hub region zone not supported
32	CsldAreaZn_Id	string<8>		•	•	the predefined consolidation area zone for the consignee not supported
33	GLRgn_Zn_cd	string<8>		•	•	the predefined G/L region zone for the consignee if you do not supply a value, then the zone is derived from the consignee's address
34	INCO_ShpG_Loc_cd	string<16>		•	•	the default INCO terms shipping location code
35	INCO_ShpG_Loc_Typ_enu	VIS_V1::eShipPointType		•	•	the default INCO terms shipping location type
36	Extl_cd1	string<30>		•	•	an external identifying code
37	Extl_cd2	string<30>		•	•	a second external identifying code
38	Stat_enu	VIS::eStatus				the status of the consignee
39	CrtD_dtt	VIS::timestamp				the date and time that the consignee was created
40	Updt_dtt	VIS::timestamp				the date and time that the consignee was updated
41	CrtDUsr_Id	string<10>				the ID of the user who created the consignee
42	UpdtUsr_Id	string<10>				the ID of the user who updated the consignee

P	Element	Type	R	C	U	Description
43	Override_enu	VIS_V1::eDTTEOverride		•	•	indicates whether to convert or delete the distance override when updating the address of the consignee
44	Addr	Address_V1	•	•	•	the address of the consignee
45	CntcInf	ContactInfo_V1	•	•	•	the contact information for the consignee
46	BusHours	BusinessHours_V1		•	•	the business hours of the consignee
47	Mmo	Memo_V1		•	•	a memo associated with the consignee
48	Directions	Memo_V1		•	•	directions to the location of the consignee
	Collections					
49	IgnoreConts	VIS::vbool				indicates whether the contents of Conts (next) should be ignored
50	Conts	ContactList_V1		•	•	contacts associated with the consignee
51	IgnoreAutoOpts	VIS::vbool				indicates whether the contents of AutoOpts (next) should be ignored
52	AutoOpts	AutoAppliedOptionList_V1		•	•	the auto-applied options for the consignee
53	IgnoreCust ShpgLocXRefs	VIS::vbool				indicates whether the contents of CustShpgLocXRefs (next) should be ignored
54	CustShpgLocXRefs	CustShpgLocXRefList_V1		•	•	the cross-reference relationship for a customer and shipping location

Customer_V1

The entity that initiates the request to move freight.

VchrCncy_enu

VchrCncy_enu indicates whether A/R vouchers are created in the tariff rating currency, the tariff payment currency, or the customer currency.

Before version 5.0, this field applies only to carrier surcharge based customers. After version 5.0, it is mandatory and applies to all customers.

The domain table values for the customer API are

- VC_TFF_RATING_CNCY
- VC_TFF_PAYMENT_CNCY
- VC_CUSTOMER_CNCY, or VC_NULL

This field is mandatory only if Chrg_Bsd_Carr_yn is True.

Note: If you use VC_NULL while creating or updating a customer, an error will occur.

P	Element	Type	R	C	U	Description
1	Cust_cd	string<12>	•	•		a unique customer code
2	Cust_Desc	string<70>	•	•	•	a description of the customer
3	Lang_typ	VIS_V1::tLanguage			•	the language the customer uses
4	Corp_Cust_yn	VIS::vbool			•	indicates whether the customer is a corporate customer
5	Prf_Ctr_typ	VIS_V1::tProfitCenter	•	•	•	the profit center to which the customer revenue is assigned
6	Cust_SrvRep_typ	VIS_V1::tCustSrvRep			•	the service representative of the customer
7	Cncy_typ	VIS_V1::tCurrency			•	the currency of the customer the default value is from the Transportation Manager user who created the customer
8	Umsr_Sys_enu	VIS_V1::eUnitMeasureSys			•	the unit of measure system used by the customer
9	Umsr_Wgt_enu	VIS_V1::eUnitMeasureWgt			•	the weight units used by the customer
10	Umsr_Len_enu	VIS_V1::eUnitMeasureLgt			•	the length units used by the customer
11	UMsr_Dst_enu	VIS_V1::eUnitMeasureDst			•	the unit of measure for the distances
12	Alw_Shpm_yn	VIS::vbool			•	indicates whether the customer is allowed to place shipments
13	Prt_Invc_yn	VIS::vbool			•	indicates whether invoices are printed for the customer
14	AR_Invc_Fmt_typ	VIS_V1::tARInvoice Format			•	the printed format of the invoices for the customer
15	Prt_Chrg_Invc_yn	VIS::vbool			•	indicates whether charges appear on the invoices for the customer
16	Fax_Ebl_yn	VIS::vbool			•	indicates whether the customer has online fax capability
17	Max_Invc_Amt_dlr	VIS::num15_2			•	the maximum dollar amount for an invoice to the customer if exceeded, then the system generates separate invoices

P	Element	Type	R	C	U	Description
18	AR_Cdt_Trm_typ	VIS_V1::tCreditTerm		•	•	the method in which the customer should pay
19	AR_Acc_Num	string<12>		•	•	the account number of the customer
20	Brcd_typ	VIS::tBarcodeType		•	•	the standard bar code type used in bar code printing
21	Rout_Prcn_typ	VIS_V1::eRouteUPrecision		•	•	the routing precision type
22	Stat_Vrfc_enu	VIS_V1::eStatusVrfc		•	•	the status verification type
23	Chrg_Bsd_Carr_yn	VIS::vbool		•	•	indicates whether this customer uses a carrier surcharge-based tariff if true, then you should set the SChg_Rate attributes the default value is false
24	VchrCncy_enu	VIS_V1::eVoucher Currency	•			indicates whether A/R vouchers are created in the tariff rating currency, the tariff payment currency, or the customer currency see " VchrCncy_enu " on page 118 for more information
25	SChg_Pct	VIS::num5_2		•	•	the surcharge or discount percentage applied to the base tariff
26	SChg_Rate	VIS::num7_2		•	•	the surcharge or discount rate applied to the base tariff
27	Mxd_Ld_yn	VIS::vbool		•	•	indicates whether the shipments from the customer can be combined with shipments from other customers the default value is true
28	Tff_Sel_Ctl_typ	VIS_V1::eTariffSel		•	•	the method used to select a tariff for A/P rating & routing
29	Mnft_Num_Defd_yn	VIS::vbool		•	•	indicates whether there is a last manifest number maintained separately for the customer
30	Mnft_Num	string<17>		•	•	the last manifest number always returned as an empty string
31	TffCtrl_Rshp_enu	VIS_V1::eTffCtrlRateUShop		•	•	determines the tariff rate shop selection

P	Element	Type	R	C	U	Description
32	Min_Odr_Val_dlr	VIS::num15_2			• •	the minimum order value for the customer
33	Max_Odr_Val_dlr	VIS::num15_2			• •	the maximum order value for the customer
34	Ship_Frm_Typ_enu	VIS_V1::eShipFromType			• •	the object type that DftFromLoc_Id refers to
35	Ship_To_Typ_enu	VIS_V1::eShipToType			• •	the object type that DftToLoc_Id refers to
36	Carr_Pymt_Rqrd_yn	VIS::vbool			• •	indicates whether a carrier payment is required
37	MaxPayVarPos_pct	VIS::num5_2			• •	the maximum pay variance positive percentage
38	MaxPayVarPos_dlr	VIS::num15_2			• •	the maximum pay variance positive amount
39	MaxPayVarNeg_pct	VIS::num5_2			• •	the maximum pay variance negative percentage
40	MaxPayVarNeg_dlr	VIS::num15_2			• •	the maximum pay variance negative amount
41	Invc_Grp_Lvl_enu	VIS_V1::eInvcGrpLvl			• •	the invoice group level
42	AccNum_WithCust	string<12>			• •	the account number
43	Intl_Cust_yn	VIS::vbool			• •	indicates whether the customer is internal
44	Invc_Num_Defd_yn	VIS::vbool			• •	indicates whether the invoice number is defined
45	Invc_Pymt_Md_enu	VIS_V1::eFrhtlnvcPymtMode			• •	the invoice payment mode
46	Tdr_Rsps_hrs	VIS::num6_2			•	the tender response hours
47	Trsp_Ebl_yn	VIS::vbool			• •	indicates whether the customer is enabled for transportation the default value is true
48	Matd_Adt_Ebl_yn	VIS::vbool			• •	indicates whether the customer is enabled for the matched audit process the default value is false
49	Matd_Adt_Olfr_id	VIS::num28_0			• •	the matched audit reference number qualifier mandatory if Matd_Adt_Ebl_yn is true

P	Element	Type	R	C	U	Description
50	Un_Matd_Adt_Ebl_yn	VIS::vbool			• •	indicates whether the customer is enabled for the unmatched audit process the default value is false
51	Un_Matd_Adt_Qlfr_id	VIS::num28_0			• •	an unmatched audit reference number qualifier Enable this field if Un_Matd_Adt_Ebl_yn is true
52	Carr_Pymt_Rblt_enu	VIS_V1::eFhgtTerms			• •	the carrier payment responsibility
53	Vldt_Itm	VIS_V1::eVldtItm			• •	the item validation type
54	ILD_Actv_enu	VIS_V1::eILDActv			• •	the item activation value
55	Itm_Grp_cd	string<12>			• •	the item group code
56	Comp_Typ_Group_cd	string<12>			• •	the component type group code
57	INCO_Ver_enu	VIS_V1::eINCOVersion			• •	the INCO terms version
58	Prepaid_Seg_Only_yn	VIS::vbool			• •	indicates whether the segment is prepaid only
59	Corp1_Id	string<30>			• •	a corporate identifier code
60	Corp2_Id	string<30>			• •	a second corporate identifier code
61	BillTo_Id	string<12>			•	the ID code of the bill-to the bill-to is the customer that pays for the freight services
62	Corp_Parn_Id	string<12>			•	the corporate parent customer code
63	SalesPers_Id	string<10>			•	the salesperson code
64	DftMstrSrvc_Id	string<4>			•	the default master service ID
65	DftTOEntVer_Id	string<10>			•	the default transport order entry version ID
66	Cdty_Id	string<12>			•	the default commodity code
67	FC_Id	string<4>			•	the default freight class code
68	DftFromLoc_Id	string<16>			•	the default From Location code
69	DftToLoc_Id	string<16>			•	the default To Location code

P	Element	Type	R	C	U	Description
70	Lgst_Grp_Id	string<4>			• •	the logistics group of the customer if you do not supply an input value for this field when creating this structure, then the value will be taken from the user's profile to force this value to be blank, enter a single space (" ")
71	Div_Id	string<4>			• •	the division ID of the customer if you do not supply an input value for this field when creating this structure, then the value will be taken from the user's profile to force this value to be blank, enter a single space (" ")
72	Shpm_Ent_Typ_Id	string<2>			• • •	the shipment entry type ID
73	Extl_cd1	string<30>			• •	an external identifying code
74	Extl_cd2	string<30>			• •	a second external identifying code
75	Stat_enu	VIS_V1::eStatus				the status of the customer
76	CrtdUsr_Id	string<10>				the ID of the user who created the customer
77	UpdtUsr_Id	string<10>				the ID of the user who updated the customer
78	Crtddtt	VIS::timestamp				the date and time that the customer was created
79	Updtdtt	VIS::timestamp				the date and time that the customer was updated
80	Addr	Address_V1			• • •	the customer address
81	Cntclnf	ContactInfo_V1			• • •	the contact information for the customer
82	BusHours	BusinessHours_V1			• •	the business hours for the customer
83	Mmo	Memo_V1			• •	a memo associated with the customer
	Collections					
84	IgnoreConts	VIS::vbool				indicates whether the contents of Conts (next) should be ignored
85	Conts	ContactListU_V1			• •	contacts associated with the customer

P	Element	Type	R	C	U	Description
86	IgnoreAutoOpts	VIS::vbool				indicates whether the contents of AutoOpts (next) should be ignored
87	AutoOpts	AutoAppliedOptionList_V1			• •	the auto-applied options for the customer
88	IgnoreCust CarrFrhtAudits	VIS::vbool				indicates whether the contents of the CustCarrFrhtAudits (next) should be ignored
89	CustCarrFrht Audits	CustCarrFrhtAuditList_V1			• •	the customer/carrier freight audit relationship
90	IgnoreCust ShpgLocXRefs	VIS::vbool				indicates whether the contents of CustShpgLocXRefs (next) should be ignored
91	CustShpgLoc XRefs	CustShpgLocXRefList_V1			• •	the cross-reference relationship for a customer and shipping location

DistributionCenter_V1

A physical location where merchandise is stored, shipped, and received.

P	Element	Type	R	C	U	Description
1	DC_Id	string<16>	•	•		a unique distribution center code
2	DC_Desc	string<70>	•	•	•	a description of the distribution center
3	Lang_typ	VIS_V1::UtLanguage			• •	the language used at the distribution center
4	Cncy_typ	VIS_V1::tCurrency			• •	the currency used at the distribution center
5	UMsr_Sys_enu	VIS_V1::eUnitMeasureSys			• •	the unit of measure system used at the distribution center
6	UMsr_Wgt_enu	VIS_V1::eUnit MeasureWgt			• •	the weight units used at the distribution center
7	UMsr_Len_enu	VIS_V1::eUnit MeasureLgt			• •	the length units used at the distribution center
8	Apt_Rqrd_yn	VIS::vbool			• •	indicates whether a carrier has to make an appointment to pick up or drop off shipments at the distribution center
9	Max_Shpm_Wgt	VIS::num11_4			• •	the maximum weight of a shipment that can be loaded or unloaded at the distribution center

P	Element	Type	R	C	U	Description
10	Max_Shpm_Vol	VIS::num11_4		•	•	the maximum volume of a shipment that can be loaded or unloaded at the distribution center
11	Max_Eqmt_Len	VIS::num9_3		•	•	the maximum length of the equipment allowed at the distribution center
12	Max_Eqmt_Width	VIS::num9_3		•	•	the maximum width of the equipment allowed at the distribution center
13	Max_Eqmt_Hght	VIS::num9_3		•	•	the maximum height of the equipment allowed at the distribution center
14	Fixd_Ld_UnLd_hrs	VIS::num6_2		•	•	the fixed load or unload hours
15	Ld_Unit_Typ_enu	VIS_V1::eLoadUnitType		•	•	the load unit type
16	Var_Ld_hrs	VIS::num5_2		•	•	the variable load hours
17	UnLd_UnitTyp_enu	VIS_V1::eLoadUnitType		•	•	the unload unit type
18	Var_UnLd_hrs	VIS::num6_2		•	•	the variable unload hours
19	Max_Ld_UnLd_hrs	VIS::num6_2		•	•	the maximum load or unload hours
20	Min_Lead_hrs	VIS::num6_2		•	•	the minimum lead hours
21	Min_Ld_UnLd_hrs	VIS::num6_2		•	•	the minimum load or unload hours
22	Live_Ld_Typ_enu	VIS_V1::eLiveLoadType		•	•	indicates whether live or non-live loading is allowed if non-live, then it indicates whether there can be non-live loading during business hours or off hours
23	Frst_Pick_yn	VIS::vbool		•	•	indicates whether any shipment leg starting from this location must be picked up at the first stop for the load
24	Last_Drp_yn	VIS::vbool		•	•	indicates whether any shipment leg terminating at this location must be dropped off at the last stop for the load
25	Bdry_Rule_enu	VIS_V1::eBoundaryRule		•	•	the boundary rules
26	GL_Cat_typ	VIS_V1::tGLCategory		•	•	the G/L category
27	Corp1_Id	string<30>		•	•	a corporate identifier code
28	Corp2_Id	string<30>		•	•	a second corporate identifier code

P	Element	Type	R	C	U	Description
29	Div_Id	string<4>		•	•	the division ID of the distribution center if you do not supply an input value for this field when creating this structure, then the value will be taken from the user's profile to force this value to be blank, enter a single space (" ")
30	Lgst_Grp_Id	string<4>		•	•	the logistics group of the distribution center if you do not supply an input value for this field when creating this structure, then the value will be taken from the user's profile to force this value to be blank, enter a single space (" ")
31	HubRgnZn_Id	string<8>		•	•	the hub region zone not supported
32	CsldAreaZn_Id	string<8>		•	•	the predefined consolidation area zone for the distribution center not supported
33	GLRgnZn_Id	string<8>		•	•	the predefined G/L region zone if you do not supply a value, then the zone is derived from the hub's address
34	INCO_ShpG_Loc_cd	string<16>		•	•	the default INCO terms shipping location code
35	INCO_ShpG_Loc_Typ_enu	VIS_V1::eShipPointType		•	•	the default INCO terms shipping location type
36	Extl_cd1	string<30>		•	•	an external identifying code
37	Extl_cd2	string<30>		•	•	a second external identifying code
38	Stat_enu	VIS_V1::eStatus				the status of the distribution center
39	CrtD_dtt	VIS::timestamp				the date and time that the distribution center was created
40	Updt_dtt	VIS::timestamp				the date and time that the distribution center was updated
41	CrtDUsr_Id	string<10>				the ID of the user who created the distribution center
42	UpdtUsr_Id	string<10>				the ID of the user who updated the distribution center

P	Element	Type	R	C	U	Description
43	Override_enu	VIS_V1::eDTTEOverride		•	•	indicates whether to convert or delete the distance override when updating the distribution center's address
44	Addr	Address_V1	•	•	•	the address of the distribution center
45	CntcInf	ContactInfo_V1	•	•	•	the contact information for the distribution center
46	BusHours	BusinessHours_V1		•	•	the business hours for the distribution center
47	Mmo	Memo_V1		•	•	a memo associated with the distribution center
	Collections					
48	IgnoreConts	VIS::vbool				indicates whether the contents of Conts (next) should be ignored
49	Conts	ContactList_V1		•	•	contacts associated with the distribution center
50	IgnoreCompTyps	VIS::vbool				always returned as VIS::bTRUE reserved for backward compatibility
51	CompTyps	ComponentTypeList_V1		•	•	always empty because component types are no longer associated with distribution centers reserved for backward compatibility
52	IgnoreCust ShpgLocXRefs	VIS::vbool				indicates whether the contents of CustShpgLocXRefs (next) should be ignored
53	CustShpgLoc XRefs	CustShpgLocXRefList_V1		•	•	the cross-reference relationship for a customer and shipping location

Domicile_V1

A physical location from which carriers manage their equipment.

P	Element	Type	R	C	U	Description
1	Dmcl_Cd	string<16>	•	•		the domicile ID code
2	Dmcl_Name	string<69>	•	•	•	the name or description
3	Carr_Id	string<8>	•	•		the ID of the carrier that owns this domicile
4	ShpgPnt_enu	VIS_V1::eShipPointType		•	•	the type of location associated with this domicile
5	Shpg_Loc_Cd	string<16>		•	•	the location associated with this domicile

P	Element	Type	R	C	U	Description
6	Dispatch_Zn_Cd	string		•	•	the dispatch zone code
7	Crted_Usr_cd	string<10>				the ID of the user who created this domicile
8	Crted_dtt	VIS::timestamp				the date and time that the domicile was created
9	Updt_Usr_cd	string<10>				the ID of the user who most recently updated the domicile
10	Updt_dtt	VIS::timestamp				the date and time that the domicile was most recently updated
11	Addr	Address_V1		•	•	the address of this domicile if a shipping location is specified, the domicile's address is used as the shipping location's address an override address can be specified as mandatory if no shipping location is specified
12	BusHours	BusinessHours_V1		•	•	the business hours of the domicile
13	IgnoreConts	VIS::vbool				indicates whether the contents of Conts (next) should be ignored
14	Conts	ContactList_V1		•	•	the contacts associated with the domicile
15	IgnoreDomicileEquipment	VIS::vbool				indicates whether DomicileEquipment is processed
16	DomicileEquipment	DomicileEquipmentList_V1		•	•	the list of equipment for this domicile

EquipmentType_V1

The type of equipment used to move the load.

P	Element	Type	R	C	U	Description
1	Eqmt_Typ_cd	string<4>	•	•	•	the equipment type code
2	Eqmt_Typ_desc	string<69>	•	•	•	a description of the equipment type
3	Len	VIS::num9_3		•	•	the equipment type length
4	Wdth	VIS::num9_3		•	•	the equipment type width
5	Hght	VIS::num9_3		•	•	the equipment type height
6	Tare_wgt	VIS::num11_4		•	•	the equipment type tare weight
7	Usable_len	VIS::num9_3		•	•	the usable length of the equipment type
8	Usable_wdth	VIS::num9_3		•	•	the usable width of the equipment type
9	Usable_hght	VIS::num9_3		•	•	the usable height of the equipment type

P	Element	Type	R	C	U	Description
10	Max_Ld_wgt	VIS::num11_4		•	•	the maximum load weight
11	UOM	UMsr_V1		•	•	the unit of measurement for the equipment types
12	Ldpnt1_loc	VIS::num9_3		•	•	the location of the first axle
13	Ldpnt1_Tare_wgt	VIS::num11_4		•	•	the tare weight on the first axle
14	Ldpnt1_Max_Ld_wgt	VIS::num11_4		•	•	the maximum load weight on of the first axle
15	Ldpnt2_loc	VIS::num9_3		•	•	the location of the second axle
16	Ldpnt2_Tare_wgt	VIS::num11_4		•	•	the tare weight on the second axle
17	Ldpnt2_Max_Ld_wgt	VIS::num11_4		•	•	the maximum load weight on the second axle
18	Ld_UnitTyp_enu	VIS_V1::eLoadUnitType		•	•	the load unit type
19	UnLd_UnitTyp_enu	VIS_V1::eLoadUnitType		•	•	the unload unit type
20	Fixd_Ld_UnLd_hrs	VIS::num6_2		•	•	the fixed loading and unloading time
21	Var_Ld_hrs	VIS::num6_2		•	•	the variable loading time
22	Var_UnLd_hrs	VIS::num6_2		•	•	the variable unloading time
23	Max_Ld_UnLd_hrs	VIS::num6_2		•	•	the maximum loading and unloading time
24	Min_Ld_UnLd_hrs	VIS::num6_2		•	•	the minimum loading and unloading time
25	Extl_cd	string<30>				an external identifying code

HarmTariff_V1

A harmonized tariff.

P	Element	Type	R	C	U	Description
1	Hmn_Tff_cd	string<30>	•	•	•	the ID code of the harmonized tariff
2	Hmn_Tff_Desc	string<69>	•	•	•	a description of the harmonized tariff
3	Rfrncd_By_ItmMstr_yn	VIS::vbool				indicates whether the tariff is referenced by the item master
4	Mmo	Memo_V1		•	•	a memo for the tariff
5	Crted_dtt	VIS::timestamp				the date and time that the tariff was created
6	Updt_dtt	VIS::timestamp				the date and time that the tariff was updated
7	Crted_Usr_cd	string<10>				the ID of the user who created the tariff
8	Updt_Usr_cd	string<10>				the ID of the user who updated the tariff

Hub_V1

A center for load consolidation or deconsolidation, also called a cross-dock or pool-point. Consolidation is the combining of several shipments into a single load or multiple loads. Deconsolidation is the separation of shipments that are grouped together in a load for delivery to their final destinations.

P	Element	Type	R	C	U	Description
1	Hub_Id	string<16>	•	•		a unique hub code
2	Hub_Desc	string<70>	•	•	•	a description of the hub
3	Lang_typ	VIS_V1::tLanguage		•	•	the language used at the hub
4	Apt_Rqrd_yn	VIS::vbool		•	•	indicates whether a carrier has to make an appointment to pick up or drop off shipments at the hub
5	Max_Shpm_Wgt	VIS::num11_4		•	•	the maximum weight of a shipment that can pass through the hub
6	Max_Shpm_Vol	VIS::num11_4		•	•	the maximum volume of a shipment that can pass through the hub
7	Max_Eqmt_Len	VIS::num9_3		•	•	the maximum length of the equipment allowed at this hub
8	Max_Eqmt_Wdth	VIS::num9_3		•	•	the maximum width of the equipment allowed at this hub
9	Max_Eqmt_Hght	VIS::num9_3		•	•	the maximum height of the equipment allowed at this hub
10	Drpf_enu	VIS_V1::eDrop		•	•	the degree to which drop-offs of shipments at the hub are allowed
11	Pkup_enu	VIS_V1::ePick		•	•	the degree to which pickups of shipments at the hub are allowed
12	Alw_Dcon_yn	VIS::vbool		•	•	indicates whether the hub allows deconsolidation
13	Min_Dcon_Wgt	VIS::num11_4		•	•	the minimum deconsolidation weight allowed for the hub
14	Min_Csld_Wgt	VIS::num11_4		•	•	the minimum consolidation weight allowed for the hub
15	Min_Drp_Wgt	VIS::num11_4		•	•	the minimum drop-off weight allowed for the hub
16	Lgst_Free_yn	VIS::vbool		•	•	indicates whether the carrier who owns the hub does not charge for the largest drop
17	Ebl_Alrt_yn	VIS::vbool		•	•	indicates whether the delivery alert process is active

P	Element	Type	R	C	U	Description
18	Fixd_Ld_UnLd_hrs	VIS::num6_2		•	•	the fixed load or unload hours
19	Ld_Unit_Typ_enu	VIS_V1::eLoadUnitType		•	•	the load unit type
20	Var_Ld_hrs	VIS::num6_2		•	•	the variable load hours
21	UnLd_UnitTyp_enu	VIS_V1::eLoadUnitType		•	•	the unload unit type
22	Var_UnLd_hrs	VIS::num6_2		•	•	the variable unload hours
23	Max_Ld_UnLd_hrs	VIS::num6_2		•	•	the maximum load or unload hours
24	Min_Lead_hrs	VIS::num6_2		•	•	the minimum lead time hours
25	Min_Ld_UnLd_hrs	VIS::num6_2		•	•	the minimum load or unload hours
26	Live_Ld_Typ_enu	VIS_V1::eLiveLoadType		•	•	indicates whether live or non-live loading is allowed if non-live, then it indicates whether there can be non-live loading during business hours or off hours
27	Frst_Pick_yn	VIS::vbool		•	•	indicates whether any shipment leg starting from this location must be picked up at the first stop for the load
28	Last_Drp_yn	VIS::vbool		•	•	indicates whether any shipment leg terminating at this location must be dropped off at the last stop for the load
29	Bdry_Rule_enu	VIS_V1::eBoundaryRule		•	•	the boundary rules
30	Full_Srvc_yn	VIS::vbool		•	•	indicates whether the hub provides full service
31	GL_Cat_typ	VIS_V1::tGLCategory		•	•	the G/L category type
32	Corp1_Id	string<30>		•	•	a corporate identifier code
33	Corp2_Id	string<30>		•	•	a second corporate identifier code
34	Lgst_Grp_Id	string<4>		•	•	the logistics group of the hub if you do not supply an input value for this field when creating this structure, then the value will be taken from the user's profile to force this value to be blank, enter a single space (" ")
35	Div_Id	string<4>		•	•	the division ID of the hub if you do not supply an input value for this field when creating this structure, then the value will be taken from the user's profile to force this value to be blank, enter a single space (" ")

P	Element	Type	R	C	U	Description
36	TransSrvc_Id	string<4>		•	•	the charge associated with the hub transfer
37	HubOnr_Id	string<8>		•	•	the carrier that owns the hub
38	HubRgnZn_Id	string<8>		•	•	the zone in which the hub is located not supported
39	CsldAreaZn_Id	string<8>		•	•	the predefined consolidation area zone for the hub not supported
40	HubExclRgnGrp_Id	string<12>		•	•	the group ID of the hub's excluded region
41	GLRgnZn_Id	string<8>		•		the G/L region zone ID
42	INCO_ShpG_Loc_cd	string<16>		•	•	the default INCO terms shipping location code
43	INCO_ShpG_Loc_Typ_enu	VIS_V1::eShipPointType		•	•	the default INCO terms shipping location type
44	Extl_cd1	string<30>		•	•	an external identifying code
45	Extl_cd2	string<30>		•	•	a second external identifying code
46	Stat_enu	VIS_V1::eStatus				the status of the hub
47	CrtD_dtt	VIS::timestamp				the date and time that the hub was created
48	Updt_dtt	VIS::timestamp				the date and time that the hub was updated
49	CrtDUsr_Id	string<10>				the ID of the user who created the hub
50	UpdtUsr_Id	string<10>				the ID of the user who updated the hub
51	Override_enu	VIS_V1::eDTTEOverride		•	•	indicates whether to convert or delete the distance override when updating the hub's address
52	Addr	Address_V1	•	•	•	the address of the hub
53	CntcInf	ContactInfo_V1	•	•	•	the contact information for the hub
54	BusHours	BusinessHours_V1		•	•	the business hours for the hub
55	Mmo	Memo_V1		•	•	a memo associated with the hub
	Collections					
56	IgnoreConts	VIS::vbool				indicates whether the contents of Conts (next) should be ignored
57	Conts	ContactList_V1		•	•	contacts associated with the hub

P	Element	Type	R	C	U	Description
58	IgnoreAutoOpts	VIS::vbool				indicates whether the contents of the AutoOpts (next) should be ignored
59	AutoOpts	AutoApplied OptionList_V1		•	•	the auto-applied options for the hub
60	IgnoreCust ShpgLocXRefs	VIS::vbool				indicates whether the contents of CustShpgLocXRefs (next) should be ignored
61	CustShpgLoc XRefs	CustShpgLocXRefList_V1		•	•	the cross-reference relationship for a customer and shipping location

INCOTerms_V1

INCO terms are a set of standard international freight terms defined by the International Chamber of Commerce. They bring consistency to the description of freight movements, and help eliminate uncertainty regarding which party is responsible for paying the cost of the freight movement, particularly for international shipments.

P	Element	Type	R	C	U	Description
1	INCO_Terms_cd	string<4>		•	•	the INCO terms code
2	INCO_Terms_Desc	string<70>			•	a description of these INCO terms
3	INCO_Ver_enu	VIS_V1::eINCOVersion		•	•	the version of these INCO terms
4	Div_Id	string<4>		•	•	the division code
5	Shpg_Loc_Src_enu	VIS_V1::eINCOShpgLocSrc			•	the INCO terms Shipping Location Source
6	Crtddtt	VIS::timestamp				the date and time these INCO terms were created
7	Updt_dtt	VIS::timestamp				the date and time these INCO terms was last updated
8	Crtddusr_cd	string<10>				the ID of the user who create these INCO terms
9	Updtusr_cd	string<10>				the ID of the user who last updated these INCO terms

ItemGroup_V1

An item group.

P	Element	Type	R	C	U	Description
1	itm_Grp_cd	string<12>		•	•	the Item Group code
2	itm_Grp_Desc	string<69>		•	•	the Item Group description
3	NMFC_Rqrd_yn	VIS::vbool		•	•*	indicates whether a NMFC is required by the item group

P	Element	Type	R	C	U	Description
4	Hmn_Tff_Rqrd_yn	VIS::vbool			• •*	indicates whether a harmonized tariff is required by the item group
5	UOM	Umsr_V1			• •*	the unit of measurement for the tariff
6	Crtd_dtt	VIS::timestamp				the date and time that the item group was created
7	Updt_dtt	VIS::timestamp				the date and time that the item group was updated
8	CrtdUsr_Id	string<10>				the ID of the user who created the item group
9	UpdtUsr_Id	string<10>				the ID of the user who updated the item group
10	Rfrncd_By_Cust_yn	VIS::vbool				indicates whether the item group is referenced by at least one customer
11	Mmo	Memo_V1			• •	a memo for the item group
12	IgnoreItmMstrs	VIS::vbool				indicates whether the contents of ItmMstrs (next) should be ignored
13	ItmMstrs	ItemMaster List_V1			• •	the list of item masters included in the item group

* You cannot update this element if there is at least one item master included in this item group.

ItemMaster_V1

An item master.

P	Element	Type	R	C	U	Description
1	Itm_Num	string<30>			• • •	the item master code
2	Itm_Grp_cd	string<12>			• • •	the item master group code
3	Itm_Desc	string<69>			• • •	a description of the item master
4	Itm_typ	VIS_V1::tItemType			• •	the item type
5	Nmnl_Wgt	VIS::num11_4			• •	the nominal weight
6	Stat_enu	VIS_V1::eltmMstrStatus			• •	the item master status
7	SerialLotCtrl_enu	VIS_V1::eSerialLotCtrl			• •	the serial lot control type
8	Excl_cat	unsigned short				the exclusion category: not currently used
9	Itm_GL_cat	VIS_V1::tGL Category				the G/L category of the item not currently used
10	HazMat_cd	string<8>				the hazardous material code not currently used
11	NMFC_cd	string<30>			• •	the NMFC code

P	Element	Type	R	C	U	Description
12	Hmn_Tff_cd	string<30>		•	•	the harmonized tariff code
13	Frht_Cls_cd	string<4>		•	•*	the freight class code
14	Orig_Ctry_cd	string<3>		•	•	the country of origin code
15	Cdty_cd	string<12>		•	•	the commodity code
16	Mmo	Memo_V1		•	•	a memo for the item master
17	Crted_dtt	VIS::timestamp				the date and time that the item master was created
18	Updt_dtt	VIS::timestamp				the date and time that the item master was updated
19	CrtedUsr_Id	string<10>				the ID of the user who created the item master
20	UpdtUsr_Id	string<10>				the ID of the user who updated the item master

* You can update this element only if it is enabled.

LoadAt_V1

A location where freight is picked up that is neither a hub nor a distribution center. If a transport order references a load-at, then it is the origin point for the freight movement.

P	Element	Type	R	C	U	Description
1	Ldat_Id	string<16>		•	•	the ID code of the load-at
2	Ldat_Desc	string<70>		•	•	a description of the load-at
3	Lang_typ	VIS_V1::tLanguage		•	•	the language used at the load-at
4	Apt_Rqrd_yn	VIS::vbool		•	•	indicates whether a carrier has to make an appointment to pick up shipments at the load-at
5	Max_Shpm_Wgt	VIS::num11_4		•	•	the maximum weight of a shipment that can be loaded at the load-at
6	Max_Shpm_Vol	VIS::num11_4		•	•	the maximum volume of a shipment that can be loaded at the load-at
7	Max_Eqmt_Len	VIS::num9_3		•	•	the maximum length of the equipment allowed at the load-at
8	Max_Eqmt_Wdth	VIS::num9_3		•	•	the maximum width of the equipment allowed at the load-at
9	Max_Eqmt_Hght	VIS::num9_3		•	•	the maximum height of the equipment allowed at the load-at
10	Stor_Num	string<10>		•	•	a store number
11	GL_Cat_typ	VIS_V1::tGLCategory		•	•	the G/L category

P	Element	Type	R	C	U	Description
12	Ld_Unit_Typ_enu	VIS_V1::eLoadUnitType		•	•	the load unit type
13	Fixd_Ld_UnLd_hrs	VIS::num6_2		•	•	the fixed load or unload hours
14	Var_Ld_hrs	VIS::num6_2		•	•	the variable load hours
15	Max_Ld_UnLd_hrs	VIS::num6_2		•	•	the maximum load or unload hours
16	Min_Lead_hrs	VIS::num6_2		•	•	the minimum lead hours
17	Min_Ld_UnLd_hrs	VIS::num6_2		•	•	the minimum load or unload hours
18	Live_Ld_Typ_enu	VIS_V1::eLiveLoadType		•	•	indicates whether live or non-live loading is allowed if non-live, then it indicates whether there can be non-live loading during business hours or off hours
19	Frst_Pick_yn	VIS::vbool		•	•	indicates whether any shipment leg starting from this location must be picked up at the first stop for the load
20	Last_Drp_yn	VIS::vbool		•	•	indicates whether any shipment leg terminating at this location must be dropped off at the last stop for the load must be false for load-ats
21	Bdry_Rule_enu	VIS_V1::eBoundryRule		•	•	the boundary rules
22	Var_UnLd_hrs	VIS::num6_2		•	•	the variable load hours
23	UnLd_UnitTyp_enu	VIS_V1::eLoadUnitType		•	•	the unload unit type
24	Corp1_Id	string<30>		•	•	a corporate identifier code
25	Corp2_Id	string<30>		•	•	a second corporate identifier code
26	Lgst_Grp_Id	string<4>		•	•	the logistics group of the load-at if you do not supply an input value for this field when creating this structure, then the value will be taken from the user's profile to force this value to be blank, enter a single space (" ")
27	Div_Id	string<4>		•	•	the division ID of the load-at if you do not supply an input value for this field when creating this structure, then the value will be taken from the user's profile to force this value to be blank, enter a single space (" ")
28	PrefCarr_Id	string<10>		•	•	the default carrier ID

P	Element	Type	R	C	U	Description
29	PrefSrvc_Id	string<10>		•	•	the default service ID
30	HubRgnZn_Id	string<8>		•	•	the hub region zone not supported
31	CsldAreaZn_Id	string<8>		•	•	the predefined consolidation area zone for the load-at not supported
32	GLRgnZnId	string<8>		•	•	the predefined G/L region zone for the load-at if you do not supply a value, then the zone is derived from the load-at address
33	INCO_Shpg_Loc_cd	string<16>		•	•	the default INCO terms shipping location code
34	INCO_Shpg_Loc_Typ_enu	VIS_V1::eShipPointType		•	•	the default INCO terms shipping location type
35	Extl_cd1	string<30>		•	•	an external identifying code
36	Extl_cd2	string<30>		•	•	a second external identifying code
37	Stat_enu	VIS_V1::eStatus				the status of the load-at
38	Crted_dtt	VIS::timestamp				the date and time that the load-at was created
39	Updt_dtt	VIS::timestamp				the date and time that the load-at was updated
40	CrtedUsr_Id	string<10>				the ID of the user who created the load-at
41	UpdtUsr_Id	string<10>				the ID of the user who updated the load- at
42	Override_enu	VIS_V1::eDTTEOverride		•	•	indicates whether to convert or delete the distance override when updating the load-at's address
43	Addr	Address_V1		•	•	the address of the load-at
44	CntcInf	ContactInfo_V1		•	•	the contact information for the load-at
45	BusHours	BusinessHours_V1		•	•	the business hours for the load-at
46	Mmo	Memo_V1		•	•	a memo associated with the load-at
47	Directions	Memo_V1		•	•	directions to the location of the load-at
	Collections					
48	IgnoreConts	VIS::vbool				indicates whether the contents of Conts (next) should be ignored
49	Conts	ContactList_V1		•	•	contacts associated with the load-at

P	Element	Type	R	C	U	Description
50	IgnoreAutoOpts	VIS::vbool				indicates whether the contents of AutoOpts (next) should be ignored
51	AutoOpts	AutoApplied OptionList_V1		•	•	the auto-applied options for the load-at
52	IgnoreCust ShpgLocXRefs	VIS::vbool				indicates whether the contents of CustShpgLocXRefs (next) should be ignored
53	CustShpgLoc XRefs	CustShpgLocXRefList_V1		•	•	the cross-reference relationship for a customer and shipping location

Zone_V1

A geographic area that represents the origin or destination point of a shipment. Examples include a country, a state or province, a city, and a zip code range.

P	Element	Type	R	C	U	Description
1	Zn_cd	string<8>	•	•		the zone ID
2	Zn_Desc	string<70>	•	•	•	a description of the zone
3	Ctry_cd	string<3>	•	•	•	the ID of the zone's country
4	Sta_cd	string<4>		•	•	the ID of the zone's state or province *ALL means any state or province in this zone's country
	Related Shared Object IDs					
5	HubRgn_Grp_cd	string<12>		•	•	the hub region zone group of the zone not supported
6	CsldArea_Grp_cd	string<12>		•	•	the consolidation area zone group of the zone not supported
7	GLRgn_Grp_cd	string<12>		•	•	the G/L region zone group of the zone
8	TaxRgn_Grp_cd	string<12>		•	•	the tax region zone group of the zone
	Collections					
9	GeoAreas	GeoAreaList (of GeoArea_V1)	•	•	•	the details of the geographic area of the zone

Entity Structures - CIS

These structures apply to CIS (Common Integration Services) only. For details on the service that uses this structures, refer to

[“void CEntityProcessor::RetrieveTMEntityIds” on page 36](#)

All fields are optional.

CXX_CarrierXRef_V1

The carrier cross-reference.

Element	Type	Description
SCAC_typ	String<12>	the SCAC code
Carr_Cd	String<8>	the carrier code

CXX_ComponentTypeXRef_V1

The component type cross-reference.

Element	Type	Description
Extl_Cd	String<30>	the external code
Comp_Typ_Cd	String<20>	the component type code

CXX_CustomerXRef_V1

The customer cross-reference.

Element	Type	Description
Extl_Cd	String<30>	the external code
Cust_Cd	String<12>	the customer ID code
CustShpgLocRefs	CXX_CustomerLocationXRefList_V1	a list of customer shipping location cross-references
CompTypeRefs	CXX_ComponentTypeXRefList_V1	a list of component type cross-references

CXX_CustomerLocationXRef_V1

The customer location cross-reference.

Element	Type	Description
Extl_Cd	String<30>	the external code

Shpg_Loc_Cd	String<16>	the shipping location point code
Shpg_Loc_Typ_enu	VIS:: eShipPointType	the shipping location point type

CXX_EquipmentTypeXRef_V1

The equipment type cross-reference.

Element	Type	Description
Extl_Cd	String<30>	the external code
Eqmt_Typ_Cd	String<4>	the equipment type code

CXX_RefNumberTypeXRef_V1

The reference number type cross-reference.

Element	Type	Description
Extl_Cd	String<30>	the external code
Rfrc_Num_Typ	String<12>	the reference number type code

CXX_RetrieveTMEntityId_V1

The retrieved Transportation Manager entity ID.

Element	Type	Description
CarrierRefs	CXX_CarrierXRefList_V1	a list of carrier cross-references
ServiceRefs	CXX_ServiceXRefList_V1	a list of service cross-references
EqmtTypeRefs	CXX_EquipmentTypeXRefList_V1	a list of equipment type cross-references
CustomerRefs	CXX_CustomerXRefList_V1	a list of customer cross-references
RefNumberTypeRefs	CXX_RefNumberTypeXRefList_V1	a list of reference number type cross-references

CXX_ServiceXRef_V1

The service cross-reference.

Element	Type	Description
Extl_Cd	String<30>	the external code
Srvc_Cd	String<4>	the service code

Financial Structures

APTransaction_V1

General ledger accounts payable (A/P) transactions.

P	Element	Type	Description
1	Trns_id	VIS::num28_0	the ID code of the transaction used in the commit operation
2	Carr_Name	string<70>	the carrier name
3	Carr_Vend_Num	string<12>	the carrier vendor number
4	Carr_Acc_Num	string<12>	the carrier account number
5	PyTo_Carr_Name	string<70>	the Pay To carrier name
6	PyTo_Vend_Num	string<12>	the Pay To vendor number
7	PyTo_Acc_Num	string<12>	the Pay To account number
8	Frht_Bill_Num	string<12>	the freight bill number
9	Frht_Bill_Typ_typ	VIS_V1::tFrhtBillType	the freight bill type
10	FrhtBill_Src_enu	VIS_V1::eFrhtBillSource	the freight bill source type
11	Ori_FrhtBill_Num	string<12>	the original freight bill number for post charges
12	Sm_ApvdFBDtl_dlr	VIS::num21_2	the sum of the approved freight bill detail amounts
13	SmApvdDtlTax_dlr	VIS::num21_2	the sum of the approved freight bill tax amounts
14	Tot_FrhtChrg_dlr	VIS::num21_2	the total non-tax freight bill charges reported by the carrier
15	Tot_Tax_Amt_dlr	VIS::num21_2	the total freight bill tax charges reported by the carrier
16	Pymt_Trms_typ	VIS_V1::tAPTerms	the type of payment terms
17	Cncy_typ	VIS_V1::tCurrency	the currency type of the freight bill charge
18	Echg_Rate	VIS::num11_6	the exchange rate during freight bill generation
19	Stat_enu	VIS_V1::eStatus	the transaction status
20	Pymt_Due_dt	VIS::date	the payment due date
21	Invc_dt	VIS::date	the date quoted by the carrier on the invoice or freight bill form
22	Rcvd_dt	VIS::date	the date the freight bill was received
23	Crted_dtt	VIS::timestamp	the date and time that the transaction was created
24	Updt_dt	VIS::timestamp	the date and time that the transaction was updated
25	CrtedUsr_Id	string<10>	the ID of the user who created the transaction
26	UpdtUsr_Id	string<10>	the ID of the user who updated the transaction

P	Element	Type	Description
27	APTrnsVer_Id	string<10>	the A/P transaction version ID
28	FrhtBill_Id	VIS::num28_0	the freight bill ID
29	OriFrhtBill_Id	VIS::num28_0	the original freight bill ID for post charges
30	Carr_Id	string<8>	the carrier ID
31	PyToCarr_Id	string<8>	the Pay To carrier ID
32	Div_Id	string<4>	the division ID
33	LgstGrp_Id	string<4>	the logistics group ID
34	FBBtchCtl_Id	VIS::num28_0	the freight bill batch control ID
35	Fscl_yr	VIS::num4_0	the accounting fiscal year
36	Accg_Prid	VIS::num2_0	the accounting period of the fiscal year
37	TrnsRun_Id	VIS::num28_0	the transaction run ID
38	Mmo	Memo_V1	a freight bill memo field

ARTransaction_V1

General ledger accounts receivable (A/R) transactions.

P	Element	Type	Description
1	Trns_id	VIS::num28_0	the ID code of the transaction used in the commit operation
2	Cust_Name	string<70>	the customer name
3	Cust_Acc_Num	string<12>	the customer account number
4	Acc_Num_WithCust	string<12>	the account number with the customer
5	BillTo_Cust_Name	string<70>	the bill-to customer name
6	BillTo_Acc_Num	string<12>	the bill-to customer number
7	AccNumWithBillTo	string<12>	the bill-to account number
8	InvNum	string<12>	the invoice number
9	Inv_typ	VIS_V1::tInvType	the invoice type
10	Inv_Src_enu	VIS_V1::eInvSource	the type of invoice source
11	Pymt_Trms_typ	VIS_V1::tARTerms	the payment term type
12	Ori_Inv_Num	string<12>	the original invoice number
13	Sm_ApvdFBDtl_dlr	VIS::num21_2	the sum of the approved invoice detail amounts
14	SmApvdDtITax_dlr	VIS::num21_2	the sum of the approved invoice tax amounts

P	Element	Type	Description
15	Tot_FrhtChrg_dlr	VIS::num21_2	the total non-tax invoice charges reported by the carrier
16	Tot_Tax_Amt_dlr	VIS::num21_2	the total invoice tax charges amounts reported by the carrier
17	Cncy_typ	VIS_V1::tCurrency	the currency type of the invoice charge
18	Echg_Rate	VIS::num11_6	the exchange rate at the time of invoice creation
19	Stat_enu	VIS_V1::eStatus	the transaction status
20	Pymt_Due_dt	VIS::date	the payment due date
21	Invc_dt	VIS::date	the invoice date established by the logistics provider
22	Rcvd_dt	VIS::date	the date the freight invoice was received
23	Crted_dtt	VIS::timestamp	the date and time that the transaction was created
24	Updt_dt	VIS::timestamp	the date and time that the transaction was updated
25	CrtedUsr_Id	string<10>	the ID of the user who created the transaction
26	UpdtUsr_Id	string<10>	the ID of the user who updated the transaction
27	ARTrnsVer_Id	string<10>	the A/R transaction version ID
28	Frht_Invc_Id	VIS::num28_0	the freight invoice ID
29	Ori_Frht_Invc_Id	VIS::num28_0	the original freight invoice ID for post charges
30	Cust_Id	string<12>	the customer ID
31	BillToCust_Id	string<12>	the bill-to customer ID
32	Div_Id	string<4>	the division ID
33	LgstGrp_Id	string<4>	the logistics group ID
34	FscI_yr	VIS::num4_0	the accounting fiscal year
35	Accg_Prid	VIS::num2_0	the accounting period of the fiscal year
36	TrnsRun_Id	VIS::num28_0	the ID of the run that created the transaction

FreightBill_V1

A document issued by a carrier requesting payment for freight services.

P	Element	Type	R	C	U	Description
1	Frht_Bill_Id	VIS::num28_0				the freight bill ID
2	Frht_Bill_Num	string<12>	•	•	•	the freight bill number
3	Frht_Bill_typ	VIS_V1::tFrhtBillType		•	•	the freight bill type
4	Frht_Bill_Ver_Cd	string<11>				the freight bill version code
5	Div_Id	string<4>		•		the division ID

P	Element	Type	R	C	U	Description
6	Lgst_Grp_Id	string<4>		•		the logistics group ID
7	Carr_Id	string<8>		•		the carrier ID
8	Cust_Id	string<12>		•		the customer ID
9	BillTo_Cust_cd	string<12>		•		the Bill To customer code
10	Curr_Stat_cd	VIS_V1::eStatus				the current status
11	Match_Exist_Vchr_yn	VIS::vbool				indicates whether the freight bill can be matched to an existing voucher
12	InvDate	VIS::timestamp		•	•	the invoice date
13	PymtDueDate	VIS::timestamp		•	•	the payment due date
14	Orig_Frht_Inv_Id	VIS::num28_0				the invoice ID of the origin freight
15	FB_Auto_Pymt_Run_Id	VIS::num28_0				the freight bill auto payment run ID
16	Pymt_Trms_typ	VIS_V1::tPymtTerms		•	•	the payment terms type
17	CarrierAccNum	string		•		the carrier account number
18	Cncy_typ	VIS_V1::tCurrency		•		the currency type
19	Echg_Rate	VIS::num11_6				the exchange rate
20	Fiscal_Year	VIS::num4_0				the fiscal year
21	Accg_Period	VIS::num2_0				the accounting period
22	Carr_Pymt_Rblt_enu	VIS_V1::eFhgtTerms		•	•	the carrier payment responsibility
23	Tot_Amt	VIS::num15_2		•	•	the total amount
24	Tot_Tax_Amt	VIS::num15_2		•	•	the total tax amount
25	Sum_FB_Dtl_Tot_Amt	VIS::num15_2				the sum of the freight bill detail total amounts
26	Sum_FB_Dtl_Tot_Tax_Amt	VIS::num15_2				the sum of the freight bill detail total tax amounts
27	Sum_Appr_FB_Dtl_Tot_Amt	VIS::num15_2				the sum of the freight bill detail total approved amounts
28	Sum_Appr_FB_Dtl_Tot_Tax_Amt	VIS::num15_2				the sum of the approved freight bill detail total tax amounts
29	Sum_Vchr_Tot_Amt	VIS::num15_2				the sum of the voucher total amounts
30	Sum_Vchr_Tot_Tax_Amt	VIS::num15_2				the sum of the voucher total tax amounts
31	FrhtBill_Src_enu	VIS_V1::eFrhtBillSource		•		the freight bill source

P	Element	Type	R	C	U	Description
32	Hold_yn	VIS::vbool				indicates whether the freight bill is on hold
33	Held_yn	VIS::vbool		•		indicates whether the freight bill was held
34	Valid_yn	VIS::vbool		•		indicates whether the freight bill is valid
35	RcvdDate	VIS::timestamp		•	•	the received date
36	Pymt_Stat_enu	VIS_V1::tPymtStatusType		•	•	the payment status
37	Crted_Usr_cd	string<10>				the user that created the freight bill
38	Updt_Usr_cd	string<10>				the user that last updated the freight bill
39	Crted_dtt	VIS::timestamp				the date and time the freight bill was created
40	Updt_dtt	VIS::timestamp				the date and time the freight bill was updated
41	Memo	Memo_V1		•	•	a memo for this freight bill
42	IgnoreRefNums	VIS::vbool				indicates whether the contents of RefNums (next) should be ignored
43	RefNums	RefNumberList_V1		•	•	the reference numbers
44	IgnoreFrhtDets	VIS::vbool				indicates whether the contents of FrhtBillDetails (next) should be ignored
45	FrhtBillDetails	VISFreightBillDetailList_V1		•	•	the freight bill details

FreightBillDetail_V1

A specific entry on a freight bill.

P	Element	Type	R	C	U	Description
1	Frht_Bill_Detl_Id	VIS::num28_0				the freight bill detail ID
2	Frht_Bill_Detl_Seq_Num	unsigned long			•*	the freight bill detail sequence number
3	Curr_Stat_cd	VIS_V1::eStatus				the current status
4	Frht_Bill_Id	VIS::num28_0				the freight bill ID
5	Frht_Bill_Num	string<12>				the freight bill number
6	Vchr_Num	string<12>				the voucher number

P	Element	Type	R	C	U	Description
7	User_Rfrnc_Num	string<30>	•	•	•	the reference number supplied by the user
8	Intl_Rfrnc_Num_Typ	VIS_V1::tReferenceNumType				the internal reference number type
9	Intl_Rfrnc_Num	string<30>				the internal reference number
10	Frht_Bill_Detl_Src_enu	VIS_V1::eFrhtBillSource				the freight bill detail source
11	Frht_Audit_Mode_enu	VIS_V1::eFrhtAuditMode				the freight audit mode
12	Aprvd_By_Usr_cd	string				the user that approved the freight bill detail
13	Tot_Amt	VIS::num15_2		•	•	the total amount
14	Tot_Tax_Amt	VIS::num15_2		•	•	the total tax amount
15	Aprvd_Tot_Amt	VIS::num15_2				the total approved amount
16	Aprvd_Tot_Tax_Amt	VIS::num15_2				the total approved tax amount
17	IgnoreCharges	VIS::vbool				indicates whether the contents of Charges (next) should be ignored
18	Charges	VISFreightBillDetailChargeList_V1	•	•		the freight bill detail charges

* This field should be specified in the Update input for the existing details but cannot be modified.

FreightBillDetailCharge_V1

A specific charge in a freight bill detail.

P	Element	Type	R	C	U	Description
1	Frht_Bill_Detl_Chrg_Id	VIS::num28_0				the freight bill detail charge ID
2	Frht_Bill_Detl_Id	VIS::num28_0				the freight bill detail ID
3	Srvc_Lvl_Chrg_yn	VIS::vbool		•		indicates whether charge is at the service level
4	Mstr_Chrg_Cd	string<4>		•		the master charge code
5	Frht_Cls_Cd	string<4>		•		the freight class code
6	Curr_Stat_cd	VIS_V1::eStatus		•		the current status code
7	Chrgd_Units	VIS::num11_4		•	•	the charged units
8	Chrgd_Unit_Rate	VIS::num11_4		•		the charged unit rate
9	Extd_Amt	VIS::num15_2		•	•	the extended amount
10	Discount_Amt	VIS::num15_2		•		the discounted amount
11	Net_Amt	VIS::num15_2		•		the net amount

P	Element	Type	R	C	U	Description
12	Aprvd_Amt	VIS::num15_2		•		the approved amount
13	Crted_Usr_cd	string<10>		•		the user that created the freight bill detail charge
14	Updt_Usr_cd	string<10>				the user that last updated the freight bill detail charge
15	Crted_dtt	VIS::timestamp		•		the date and time the freight bill detail charge was created
16	Updt_dtt	VIS::timestamp				the date and time the freight bill detail charge was updated

GLTransaction_V1

A general ledger transaction. An accrual transaction recognizes a current expense before a freight bill is received, and recognizes current revenue before an invoice is issued.

P	Element	Type	Description
1	GL_Trns_id	string<29>	the ID for the G/L transaction used in the commit operation
2	GL_Trns_Typ_enu	VIS_V1::eGLTrnsType	the type of G/L transaction (accrual, posting)
3	GL_Clsc_enu	VIS_V1::eGLClsc	the G/L transaction classification (expense, liability, A/R, or A/P)
4	GL_typ	VIS_V1::tGLType	the G/L transaction type
5	GL_Acc_Num	string<12>	the G/L account number
6	GLAccNum_Src_enu	VIS_V1::eGLAccount Source	the G/L account number source
7	Cdt_Dbt_enu	VIS::eCreditDebit	the transaction type (credit or debit)
8	Cost_Ctr_typ	VIS_V1::tCostCenter	the cost center
9	Prf_Ctr_typ	VIS_V1::tProfitCenter	the profit center
10	OriLocGL_Cat_typ	VIS_V1::tGLCategory	the G/L category of the origin shipping location
11	DestLocGLCat_typ	VIS_V1::tGLCategory	the G/L category of the destination shipping location
12	Cust_Name	string<70>	the customer name
13	Carr_Name	string<70>	the carrier name
14	Frht_Bill_Num	string<12>	the freight bill number (A/P only)
15	Ori_FrhtBill_Num	string<12>	the original freight bill number if the transaction is a post charge (A/P only)
16	Rfrc_Num_Typ_enu	VIS::eVchrRfrcNumType	the type of reference number (A/R, A/P)
17	Rfrc_Num	string<30>	the reference number for the charged entity

P	Element	Type	Description
18	Vchr_Num	string<12>	the voucher from which the transaction charge is derived (A/R, A/P)
19	ChrgDetl_Typ_enu	VIS::eChargeDetailType	the charge detail type (post charge, adjustment)
20	Amt_dlr	VIS::num15_2	the transaction amount
21	Cncy_tpy	VIS_V1::tCurrency	the currency type of the transaction
22	Echg_Rate	VIS::num11_6	the exchange rate of the base currency to the transaction currency
23	Btch_Seq_Num	Unsigned long	the batch sequence number
24	Stat_enu	VIS_V1::eStatus	the transaction status
25	Fscl_yr	VIS::num4_0	the accounting fiscal year for the transaction
26	Accg_Prid	VIS::num2_0	the accounting period within the fiscal year
27	Crted_dtt	VIS::timestamp	the date and time that the transaction was created
28	Updt_dtt	VIS::timestamp	the date and time that the transaction was updated
29	CrtedUsr_Id	string<10>	the ID of the user who created the transaction
30	UpdtUsr_Id	string<10>	the ID of the user who updated the transaction
31	Div_Id	string<4>	the division ID
32	Lgst_Grp_Id	string<4>	the logistics group ID
33	Cdty_Id	string<12>	the commodity code
34	Carr_Id	string<8>	the carrier code returned as an empty string for A/R transactions
35	Cust_Id	string<12>	the customer code
36	OriGLRgnZn_Id	string<8>	the origin G/L region zone ID
37	DestGLRgnZn_Id	string<8>	the destination G/L region zone ID
38	TOEntTyp_Id	string<2>	the transport order entry type ID
39	Shpm_Id	VIS::num28_0	the shipment ID
40	AcrlRun_Id	VIS::num28_0	the accrual run ID
41	TrnsRun_Id	VIS::num28_0	the transaction run ID
42	Invn_Num	string<12>	the invoice number (A/R only) returned as an empty string for A/P transactions
43	Ori_Invn_Num	string<12>	the original invoice number (A/R only) returned as an empty string for A/P transactions
44	Container_Id	VIS::num28_0	the component ID

P	Element	Type	Description
45	Load_Id	VIS::num28_0	the load ID for the A/P type or A/R type of a carrier surcharge based customer returned as an empty string for A/R transactions
46	Shpm_Leg_Id	VIS::num28_0	the shipment leg ID for the A/P type or A/R type of a carrier surcharge based customer
47	Chrg_Cd	string<5>	the charge code returned as an empty string if no charges are associated with the transaction
48	Srvc_Cd	string<5>	the service code returned as an empty string if no charges are associated with the transaction, or if the associated charge has no load, shipment, or shipment leg as a rate owner
49	Mmo	Memo_V1	a freight bill memo field empty for an A/R transaction, or for an A/P transaction with no associated freight bill otherwise, the memo of the freight bill is returned
50	Trip_Id	VIS::num28_0	the trip ID

Load Structures

Load_V1

A freight service provided by a carrier to move goods from one or more points to one or more points. A load is built assuming that one item, such as a truck or rail car, moves a set of shipment legs from their respective origins to their respective destination.

Note: Only use this structure to retrieve load information.

A shipment leg is considered relevant to a condensed load if it meets the following conditions:

- the shipment leg's transport order was input through the API (the leg's `Inpt_Src_enu` is `INPUT_SOURCE_EXTERNAL_API`) and,
- the Release/Return is active under Global Defaults and:
 - if the `Out_ERP_Crted_dtt` of the load is valid (the load is being updated), and the `Rls_Func_enu` of the shipment leg is `RLS_VIA_SAME_NLLD_LLD` or,
 - if the Load's `Out_ERP_Crted_dtt` is invalid (the load is being created), and the Shipment Leg's `Rls_Func_Enu` is `NOT_RELEASED`

- otherwise, if the Release/Return is not active under Global Defaults), then the first shipment leg for each shipment is returned

P	Element	Type	Description
1	Load_Id	VIS::num28_0	the ID code of the load
2	Load_Desc	string<70>	a description of the load
3	Frht_Trm_enu	VIS_V1::eFhgtTerms	the freight terms for the load
4	Cncy	VIS_V1::tCurrency	the currency of the load
5	Echg_Rate	VIS::num11_6	the currency exchange rate
6	Umsr_Sys_enu	VIS_V1::eUnitMeasureSys	the unit of measure system of the load
7	Umsr_Wgt_enu	VIS_V1::eUnitMeasureWgt	the weight units of the load
8	UMsr_Len_enu	VIS_V1::eUnitMeasureLgt	the length units of the load
9	UMsr_Dst_enu	VIS_V1::eUnitMeasureDst	the unit of measure for the distances
10	Sys_Calc_Amt_dlr	VIS::num15_2	the system-calculated charge amount for the load
11	Dsct_Amt_dlr	VIS::num15_2	the discount amount for the load
12	Chgd_Amt_dlr	VIS::num15_2	the charged amount for the load
13	Fedl_Tax_Amt_dlr	VIS::num15_2	the federal taxes for the load
14	Sta_Tax_Amt_dlr	VIS::num15_2	the state or provincial taxes for the load
15	Loc_Tax_Amt_dlr	VIS::num15_2	the local regional taxes for the load
16	Adtn_Chrg_dlr	VIS::num15_2	the additional charges for the load
17	PreCslD_Amt_dlr	VIS::num15_2	the total charge before pre-consolidation for the load
18	Scld_Wgt	VIS::num11_4	the scaled weight of the load
19	Vol	VIS::num11_4	the total volume of the load
20	Odr_Val_dlr	VIS::num15_2	the order value of the load
21	DclD_Val_dlr	VIS::num15_2	the declared value of the load
22	Nmnl_Wgt	VIS::num11_4	the nominal weight of the load
23	Tot_Tare_Wgt	VIS::num17_4	the total tare weight of the load
24	Tot_Pce	VIS::num7_0	the total number of pieces in the load
25	Tot_Skid	VIS::num7_0	the total number of skids in the load
26	CurOptlStat	VIS_V1::eStatus	the current operational status of the load
27	CurFnclStat	VIS_V1::eStatus	the current financial status of the load
28	CrtD_dtt	VIS::timestamp	the date and time that the load was created

P	Element	Type	Description
29	Scdd_dtt	VIS::timestamp	the date and time that the load was scheduled
30	Updt_dtt	VIS::timestamp	the date and time that the load was updated
31	Shpd_dtt	VIS::timestamp	the date and time that the load was shipped
32	Cpld_dtt	VIS::timestamp	the date and time that the load was completed
33	Ratd_dtt	VIS::timestamp	the date and time that the load was rated
34	Elgb_Cnts_Mv_yn	VIS::vbool	indicates whether the customer can participate in continuous moves
35	Seq_num	unsigned short	the trip sequence number
36	Max_Wait_At_Stop_hrs	VIS::num6_2	the maximum wait time at a stop
37	Tot_Wait_Ld_hrs	VIS::num6_2	the total wait time for a load
38	Elpd_hrs	VIS::num6_2	the maximum elapsed time in hours for the load leg
39	Tot_Driving_hrs	VIS::num6_2	the total driving hours
40	Tot_On_Duty_hrs	VIS::num6_2	the total hours on duty
41	Tot_Off_Duty_hrs	VIS::num6_2	the total hours off duty
42	Tot>Loading_hrs	VIS::num6_2	the total loading hours
43	Tot_Unloading_hrs	VIS::num6_2	the total unloading hours
44	Strd_dtt	VIS::timestamp	the start date and time
45	End_dtt	VIS::timestamp	the end (or Return to base) date and time
46	Eqmt_Typ	string<4>	the equipment type code
47	Eqmt_Typ_Cmtd_yn	VIS::vbool	indicates whether the equipment type is committed
48	Shpm_Lg_Eqmt_Cmtd_Num	unsigned short	the number of shipment legs with a committed equipment type
49	Trip_Id	VIS::num28_0	the ID number of the trip this load is on
50	Bill_To_Cust_cd	string<12>	the ID of the bill-to customer
51	Far_Pnt_Stop_id	VIS::num28_0	the ID of the far-point stop
52	Drct_dist	VIS::num5_0	the direct distance
53	OutOfRout_dist	VIS::num5_0	the out-of-route distance
54	Ldd_dist	VIS::num5_0	the loaded distance
55	Tot_Unldd_dist	VIS::num5_0	the total unloaded distance
56	In_Tnst_Unldd_dist	VIS::num5_0	the in-transit unloaded distance
57	Init_Reps_dist	VIS::num5_0	the initial repositioning distance
58	Ret_To_Orig_dist	VIS::num5_0	the return-to-origin distance

P	Element	Type	Description
59	Tot_dist	VIS::num5_0	the total distance for this load leg
60	Fixd_ltnr_dist	VIS::num5_0	the fixed itinerary distance based on the distance unit of measure defined in the global defaults other distance fields in this structure are based on the one defined in UMSr_Dst_enu
61	UMsr_Fixd_ltnr_dist_enu	VIS_V1::eUnitMeasureDst	the unit of measure that the Fixd_ltnr_dist field is expressed in
62	Non_Live_Pkup_yn	VIS::vbool	indicates whether the load can have an off-hours pickup
63	Non_Live_Drpf_yn	VIS::vbool	indicates whether the load can have an off-hours drop-off
64	Actl_CarrSrv	string<29>	the actual carrier and service for this load, if using a virtual carrier such as NTE
65	Actl_Chgd_Amt_dlr	VIS::num15_2	the actual charged amount for this load, if using a virtual carrier such as NTE
66	Actl_Cncy_typ	VIS_V1::tCurrency	the actual currency type for this load, if using a virtual carrier such as NTE
67	Div_Id	string<4>	the division ID of the load
68	Lgst_Grp_Id	string<4>	the logistics group ID of the load
69	Carr_Id	string<12>	the carrier ID of the load
70	Srvc_Id	string<4>	the code service ID of the load
71	Crted_Usr_Id	string<10>	the ID of the user who created the load
72	Updt_Usr_Id	string<10>	the ID of the user who updated the load
73	Tdrd_Usr_Id	string<10>	the ID of the user who tendered the load
74	Cfmng_Usr_Id	string<10>	the ID of the user who confirmed the load
75	Rutd_Ori_Zn_cd	string<8>	the route origin zone ID
76	Rutd_Dest_Zn_cd	string<8>	the route destination zone ID
77	Cust_cd	string<12>	the customer code associated with the first shipment assigned to this load
78	Cost_Ctr_typ	VIS_V1::tCostCenter	the cost center type
79	Out_ERP_Crted_dtt	VIS::timestamp	the first time that the load was sent to the enterprise resource planning (ERP) system returned as an empty string unless the load is retrieved as a condensed load

P	Element	Type	Description
80	Out_ERP_Updt_dtt	VIS::timestamp	the last time that the load was sent to the ERP system returned as an empty string unless the load is retrieved as a condensed load
81	Mstr_BOL_Qlfr_Id	VIS::num28_0	the qualifier type of the master BOL returned as an empty string unless the load is retrieved as a condensed load
82	Mstr_BOL_Num	string<30>	the master BOL number returned as an empty string unless the load is retrieved as a condensed load
83	Ld_Cmps_Aprd_yn	VIS::vbool	indicates whether the load composition is approved
84	Tff_Id	VIS::num28_0	the tariff ID used in the load rating, if any
85	Rate_Cd	string<6>	the rate code used in the load rating, if any
86	SndrLdTdrCntc	LoadTenderContact_V1	the tender contact of the sender
87	CarrLdTdrCntc	LoadTenderContact_V1	the tender contact of the carrier
88	ShpLegs	ShipmentLegList_V1	the shipment legs attached to the load these are returned if the load is not retrieved as a condensed load otherwise only relevant Shipment Legs are returned see note at the beginning of this table
89	Stops	StopList_V1	the stops the load makes these are returned in delivery order
90	RfrcNums	RefNumberList_V1	the reference numbers attached to the load not returned unless the load is retrieved as a condensed load
91	APChrgs	Charge_V1	the set of charges associated with the load not returned if the load is retrieved as a condensed load not returned if the caller has not requested charges and the compute charges argument is false

LoadBuildPlan_V1

A load plan.

P	Element	Type	R	C	Description
1	Plan_id	VIS::num28_0	•	•	the ID code of the plan
2	Plan_Desc	string<70>	•	•	a description of the plan

P	Element	Type	R	C	Description
3	Div_Id	string<4>	•	•	the division ID of the plan
4	Lgst_Grp_Id	string<4>	•	•	the logistics group of the plan

LoadConfirmData_V1

The load to be confirmed.

P	Element	Type	R	Description
1	Load_Id	VIS::num28_0	•	the load ID to be confirmed
2	Carr_cd	string<8>		the carrier code for the load
3	Srvc_cd	string<4>		the service code for the load
4	Eqmt_typ	string<5>		the equipment type code
5	MBOL_Num	string<30>		the master BOL number to be used when generating the master BOL
6	Confirm_dtt	VIS::timestamp		the confirm date and time the default value is the current date and time
7	ARrating	VIS::vbool		indicates whether the A/R rating is needed the default value is false
8	PrintBOL	VIS::vbool		indicates whether to print a master BOL
9	SECInfo	SEC_V1		the service event code information

LoadCreateData_V1

Information about a newly created load.

P	Element	Type	R	C	U	Description
1	Load_Id	VIS::num28_0		•		the ID of the load to be created must be in the range 80000000 to 99999999
2	Load_Desc	string<70>		•		a description of the load
3	Plan_Id	VIS::num28_0	•	•		the plan ID for the load
4	Carr_cd	string<8>		•	•	the carrier code
5	Srvc_cd	string<4>		•	•	the service code
6	Eqmt_Typ	string<4>		•	•	the equipment type
7	Scdd_dtt	VIS::timestamp		•	•	the load scheduled date
8	Ld_Cmps_Aprd_yn	VIS::vbool				indicates whether the load composition is approved
9	Trlr_Num	string<24>		•		the trailer number

P	Element	Type	R	C	U	Description
10	Drvr	string<8>		•		the driver code
11	Seal_Num	string<16>		•		the seal number
12	LD_Schd_Cmpd_yn	VIS::vbool		•		indicates whether the load schedule is completed (default value is True)
13	Strd_Dtt	timestamp		•		the start date
14	End_dtt	timestamp		•		the end date

LoadInstruction_V1 Structure

Detailed instructions for handling a load. This is a secondary structure for `Stop_V1` (refer to page 164). Only pick stops can have loading instructions, drop stops cannot.

P	Element	Type	R	C	U	Description
1	Instruction	string<2000>	•	•	•	the loading instructions
2	Crted_dtt	VIS::timestamp				the date and time these instructions were created
3	Updt_dtt	VIS::timestamp				the date and time these instructions were updated
4	Crted_Usr_Cd	string<10>				the user that created these instructions
5	Updt_Usr_Cd	string<10>				the user that updated these instructions

LoadManifestData_V1

A particular load to be manifested.

P	Element	Type	R	Description
1	Load_Id	VIS::num28_0	•	the ID of the manifest load
2	Shpm_dtt	VIS::timestamp		the date and time that the load was shipped the default is the current date
3	Manifest_num	string<30>		the manifest number if not entered, the system will try to generate a default value
4	PrintDoc_yn	VIS::vbool		indicates whether the manifest document should be printed
5	PrintParcelManifest_yn	VIS::vbool		indicates whether to print a carrier-compliant manifest available only for manifest loads containing parcel shipment legs
6	ParcelManifestPrinterName	string<14>		the name of the printer that will produce the report, if a carrier-compliant manifest report has been requested the printer must be configured as a parcel printer under the parcel mappings

LoadPlan_V1

A load ID and plan ID pair.

P	Element	Type	R	Description
1	Load_Id	VIS::num28_0	•	the load ID
2	Plan_Id	VIS::num28_0		the corresponding plan ID

LoadSelectionCriteria_V1

The criteria used to identify a set of loads.

P	Element	Type	Description
1	Load_Cat_enu	VIS_V1::eLoadCategory	identifies built loads or manifest loads
2	Frht_Trm_enu	VIS_V1::eFhgtTerms	identifies loads paid for by a shipper or a carrier
3	CurOptlStat	VIS_V1::eStatus	identifies loads by their current operational status
4	Load_Date_enu	VIS_V1::eRetrieveLoadDate	identifies the type of date range on which to search
5	Frm_dtt	VIS::timestamp	the earliest acceptable date and time stamp
6	To_dtt	VIS::timestamp	the latest acceptable date and time stamp
7	Div_Id	string<4>	identifies loads by their assigned division
8	Carr_Id	string<8>	identifies loads by their assigned carrier
9	Excl_Carr_Ids	VIS::StrIdList	identifies loads not assigned to any carrier in the list

LoadSpotRateData_V1

The system uses the data from this table to update the load table, `Ld_Leg_t`.

P	Element	Type	R	Description
1	Load_Id	VIS::num28_0	•	a system-generated load ID must belong to an existing load with valid rating results that is eligible for rating.
2	SpotRate	VIS::num15_2	•	the spot rate amount cannot be negative
3	SpotRate_Cncy_tpy	VIS_V1::tCurrency	•	the spot rate currency must match the load currency

LoadTenderContact_V1

A load tender contact.

P	Element	Type	R	C	U	Description
1	LdTdr_Cntc_enu	VIS_V1::eLdTdrCntcType	•	•	•	indicates the tender contact type: carrier or sender
2	Name	String		•	•	the name of the contact
3	Lang_typ	VIS_V1::tLanguage		•	•	the language of the contact
4	Tele1	String		•	•	the phone number of the contact
5	Tele2	String		•	•	the alternative phone number of the contact
6	Fax	String		•	•	the fax number of the contact
7	Email	String		•	•	the email address of the contact
8	URL	String		•	•	the web URL of the contact

LoadTenderData_V1

The tender information for a load.

`Carr_cd` and `SrvC_cd` ensure that the caller is accepting or rejecting the correct tender information. That is, they ensure that the carrier and service have not changed since the caller retrieved the tender information. This is especially important if auto-tendering is active, because the automatic tendering process could time out a carrier and service. It would then proceed to the next eligible pair for a load between the time that the caller retrieves tender information, and calls the tender accept or tender reject service.

`Cur_Opt1Stat_id` is mandatory for tender reject and tender cancel operations to ensure that a tender is not accidentally rejected or cancelled after it has already been accepted by someone else.

For fax and email notifications, you can override the default settings in `Carr_Notn`, with the following restrictions:

- if the load must be re-rated, the user override information supplied in `Carr_Notn` is disregarded, and the information for the new tariff service defaults is used instead

The remaining restrictions assume the Load has not been re-rated.

- if any of autofax or autoemail notifications are enabled, the caller may disable the notification using `Carr_Notn` field
- if autofax or autoemail is enabled in global defaults level, but disabled for a given carrier and service combination, you can override this setting to enable the notification
- note that a fax number or email address must be available either from your overrides, or from the tariff service setup

The following abbreviations appear in the column headings of this table:

- T - Tender
- TA - Tender Accept
- TR - Tender Reject
- TC - Tender Cancel

A hollow bullet (°) indicates the field can be set during an operation, but is not mandatory.

A solid bullet (•) indicates the field is mandatory for an operation.

P	Element	Type	T	TA	TR	TC	Description
1	Load_id	VIS::num28_0	•	•	•	•	the load ID
2	Carr_cd	string<8>	°	•	•	•	the ID of the carrier for the load
3	Carr_Cmtd_yn	VIS::vbool	°				indicates whether the load has been committed to its carrier
4	Srvc_cd	string<4>	°	•	•	•	the ID of the master service for the load
5	Srvc_Cmtd_yn	VIS::vbool	°				indicates whether the load has been committed to its master service
6	Eqmt_Typ	string<4>	°				the equipment type code
7	Eqmt_Typ_Cmtd_yn	VIS::vbool	°				indicates whether the equipment type is committed
8	Tdr_Acpd_By_Name	string<32>	°	•			the name of the employee who accepted the tender
9	Cur_OptlStat_id	VIS_V1::eStatus			•	•	the current operational status of the load
10	Actl_CarrSrvc	string<29>		°			the actual carrier and service for this load tender, if using a virtual carrier such as NTE
11	ActlCost	VIS::num15_2		°			the actual cost of this load tender, if using a virtual carrier such as NTE
12	Mmo	Memo_V1	°	°	°	°	the memo field of the load
13	SECInfo	SEC_V1			°	°	the service event code information
14	Carr_Notn	NotificationData_V1	°			°	the carrier notification information
15	Trip_Id	VIS::num28_0					the trip that this load is on read-only
16	Trip_Seq_Num	unsigned short					the sequence number of this load on its trip read-only
17	ReviseAccept_yn	VIS::vbool	°	°	°	°	indicates if there was a revision to an accepted load tender
18	Trlr_Num	string	°	°	°	°	the trailer number

P	Element	Type	T	TA	TR	TC	Description
19	Trlr_Onr_Carr_Id	string	°	°	°	°	the carrier that owns the trailer
20	Trlr_Lic_Num	string	°	°	°	°	the trailer license number
21	Trctr_Onr_Carr_Id	string	°	°	°	°	the carrier that owns the tractor
22	Trctr_Lic_Num	string	°	°	°	°	the tractor license number
23	Drvr	string	°	°	°	°	the ID of the driver
24	Drvr_Lic_Num	string	°	°	°	°	the license number of the driver
25	Seal_Num	string	°	°	°	°	the seal number

LoadUpdateProgress_V1

An event that describes the progress of a load.

P	Element	Type	R	Description
1	Carr_cd	string<8>	•	the carrier code
2	Load_Id	string<28>		the load ID or reference number
3	Load_Desc	string<70>		the load tracking number
4	MBOL_Num	string<30>		the master BOL number
5	Mmo	Memo_V1		a memo for the load update progress
6	SECInfo	SEC_V1		the service event code information

NotificationData_V1

The notification data including sender and destination information.

P	Element	Type	R	C	U	Description
1	Print_yn	VIS::vbool				indicates whether to print a report
2	Tdr_Rsps_By_dtt	VIS::timestamp		•	•	the date and time that the tender response is required by
3	SndrLdTdrCntc	LoadTenderCo ntact_V1		•	•	the tender contact of the sender
4	CarrLdTdrCntc	LoadTenderCo ntact_V1		•	•	the tender contact of the carrier

ShipmentKeyInfo_V1

Shipment identifying information.

P	Element	Type	R	Description
1	Load_Id	VIS::num28_0	•	the load ID
2	Shpm_Id	VIS::num28_0	•	the shipment ID

ShipmentLeg_V1

A shipment leg. Use this structure only to retrieve shipment leg information.

The carrier and service code in the shipment leg (`Carr_cd` and `Srvc_cd`) are not synchronized with their corresponding values (`Carr_Id` and `Srvc_Id`) in the load. The carrier and service for a load apply to all the shipment legs on that load. However, individual shipment legs can retain older values that are different from those in the load.

For example, if a shipment is rated before its shipment legs are assigned to a load, the shipment legs keep the old carrier and service values from the first rating. The original values are kept for the several reasons.

For audit purposes, carrier commitments are preserved at the shipment leg level. If the carrier at the load level is overridden during load confirmation, the previous values still exist at the shipment leg level.

To improve performance, results derived when the shipment leg is routed and rated as a distinct entity are preserved for as long as possible. The shipment leg may be added to a load and then removed from it.

By keeping the routing and rating results for the shipment leg as a distinct entity, this often avoids routing and rating for the independent shipment leg. This improves overall performance by reducing the time that the routing and rating server has to run.

There would be extra overhead if the carrier at the shipment leg was kept in synchronization with the carrier at the load level. Any carrier changes at the load level would mean replication of this data to each attached shipment leg.

P	Element	Type	Description
1	Shpg_Leg_Id	VIS::num28_0	the ID code of the shipment leg
2	Trpt_Odr_Id	string<12>	the transport order of the shipment leg
3	Shp_Id	VIS::num28_0	the shipment ID of the shipment leg
4	Load_Id	VIS::num28_0	the ID of the load that the shipment leg is on
5	Seq_Num	unsigned short	the position of the shipment leg within the sequence of legs of the shipment
6	Sp_Csld_Cls	string<12>	the shipment consolidation class
7	Frht_Trm_enu	VIS_V1::eFhgtTerms	the payment freight terms of the shipment leg

P	Element	Type	Description
8	Pick_Seq_Num	unsigned short	the sequence number of the Pick (Ship From) stop
9	ShipFromLoc_id	string<16>	the ID of the From shipping point
10	Delvy_Seq_Num	unsigned short	the sequence number of the Delivery (Ship To) stop
11	Ship_Frm_enu	VIS_V1::eShipPointType	the From shipping point type
12	Ship_To_enu	VIS_V1::eShipPointType	the To shipping point type
13	ShipToLoc_id	string<16>	the ID of the To shipping point
14	Cncy	VIS_V1::tCurrency	the currency used
15	Echg_Rate	VIS::num11_6	the currency exchange rate
16	Umsr_Sys_enu	VIS_V1::eUnitMeasureSys	the unit of measure system of the shipment leg
17	Umsr_Wgt_enu	VIS_V1::eUnitMeasureWgt	the weight units of the shipment leg
18	Umsr_Len_enu	VIS_V1::eUnitMeasureLgt	the length units of the shipment leg
19	UMsr_Dst_enu	VIS_V1::eUnitMeasureDst	the unit of measure for the distances
20	Sys_Calc_Amt_dlr	VIS::num15_2	the system-calculated charge amount for the shipment leg
21	Dsct_Amt_dlr	VIS::num15_2	the discount amount for the shipment leg
22	Chgd_Amt_dlr	VIS::num15_2	the total charged for the shipment leg
23	Fedl_Tax_Amt_dlr	VIS::num15_2	the federal taxes for the shipment leg
24	Sta_Tax_Amt_dlr	VIS::num15_2	the state or provincial taxes for the shipment leg
25	Loc_Tax_Amt_dlr	VIS::num15_2	the local or regional taxes for the shipment leg
26	Adtn_Chrg_dlr	VIS::num15_2	the additional charges for the shipment leg
27	PreCslid_Amt_dlr	VIS::num15_2	the total charge of the shipment leg before pre-consolidation
28	SclD_Wgt	VIS::num11_4	the scaled weight of the shipment leg
29	Vol	VIS::num11_4	the total volume of the shipment leg
30	Odr_Val_dlr	VIS::num15_2	the total order of the shipment leg
31	DclD_Val_dlr	VIS::num15_2	the total declared value of the shipment leg
32	Nmnl_Wgt	VIS::num11_4	the total nominal weight of the shipment leg
33	Tot_Tare_Wgt	VIS::num17_4	the total tare weight of the shipment leg
34	Tot_Pce	VIS::num7_0	the total number of pieces of the shipment leg
35	Tot_Skid	VIS::num7_0	the total number of skids of the shipment leg
36	CurOptlStat	VIS_V1::eStatus	the current operational status of the shipment leg
37	Ratd_dtt	VIS::timestamp	the date and time that the load containing the shipment leg was rated

P	Element	Type	Description
38	Rutd_Ori_Zn_cd	string<8>	the route origin zone ID
39	Rutd_Dest_Zn_cd	string<8>	the route destination zone ID
40	Carr_cd	string<8>	the ID code of the carrier that this shipment leg is using when a shipment leg is attached to a planned load, the carrier specified here may be incorrect for the correct carrier, see the Carr_Id in the Load_V1 structure identified in the Load_Id of this shipment's leg
41	Srvc_Cd	string<4>	the service code of this shipment leg when a shipment leg is attached to a planned load, the service specified here may be incorrect For the correct service, see the Srvc_Id in the Load_V1 structure identified in the Load_Id of this shipment's leg
42	CmpdPkupArvl_dtt	VIS::timestamp	the computed pickup arrival date and time of this shipment leg
43	CmpdPkupDptr_dtt	VIS::timestamp	the computed pickup departure date and time of this shipment leg
44	CmpdDropArvl_dtt	VIS::timestamp	the computed drop arrival date and time of this shipment leg
45	CmpdDropDptr_dtt	VIS::timestamp	the computed drop departure date and time of this shipment leg
46	Div_cd	string<4>	the ID code of the division of the shipment leg
47	Lgst_Grp_Id	string<4>	the ID code of the logistics group of the shipment leg
48	Tot_dist	VIS::num5_0	the total distance for this shipment leg
49	Non_Live_Pkup_yn	VIS::vbool	indicates whether the shipment leg can have an off-hours pickup
50	Non_Live_Drpf_yn	VIS::vbool	indicates whether the shipment leg can have an off-hours drop-off
51	Eqmt_Typ	string<4>	the equipment type code
52	Eqmt_Typ_Cmtd_yn	VIS::vbool	indicates whether the equipment type is committed
53	Comp_Trkg_yn	VIS::vbool	indicates whether component tracking is required for this shipment leg
54	Cdty_Pick_Seq_num	unsigned short	the commodity pick sequence number
55	Ordr_Grp_cd	string<4>	the order group code
56	Tnst_Md_enu	VIS_V1::eTransitMode	the transit mode
57	Tnst_Md_Cmtd_yn	VIS::vbool	indicates whether the transit mode has been committed
58	FromAddr	Address_V1	the address of the start point of the shipment leg

P	Element	Type	Description
59	ToAddr	Address_V1	the address of the destination point of the shipment leg not returned for a shipment legs that are returned on a condensed load
60	Shpm_Info	Shipment_V1	the shipment information structure
61	RfrcNums	RefNumberList_V1	any reference numbers attached to the shipment leg
62	APChrgs	Charge_V1	the set of load level charges associated with the shipment leg this field is always blank the Charge_V1 substructure is not returned at the shipment level

ShipmentLegHeader_V1

The header information of a shipment leg.

P	Element	Type	Description
1	Shpm_Id	VIS::num28_0	the shipment ID
2	Shpm_Num	string<12>	the shipment number
3	Shpg_Leg_Id	VIS::num28_0	the shipment leg ID
4	NullLdLegDetl_yn	VIS::vbool	indicates whether the load leg detail has a null value
5	Load_Id	VIS::num28_0	the load ID
6	Seq_Num	unsigned short	the sequence number of the shipment
7	Ship_Frm_enu	VIS_V1::eShipFromType	a description of the shipment's origin point
8	ShipFromLoc_id	string<16>	the ID of the shipment's origin point
9	Dlvy_Seq_Num	unsigned short	the delivery sequence number
10	Ship_To_enu	VIS_V1::eShipToType	a description of the shipment's destination point
11	ShipToLoc_id	string<16>	the ID of the shipment's destination point
12	CurOptlStat	VIS_V1::eStatus	the current operational status of the shipment
13	Carr_cd	string<8>	the ID code of the carrier
14	Carr_Cmtd_yn	VIS::vbool	indicates whether the carrier is committed
15	Srvc_cd	string<4>	the service code
16	Srvc_Cmtd_yn	VIS::vbool	indicates whether the service code is committed
17	Eqmt_Typ	string<4>	the equipment type
18	Eqmt_Typ_Cmtd_yn	VIS::vbool	indicates whether the equipment type has been committed

P	Element	Type	Description
19	Tnst_Md_enu	VIS_V1::eTransitMode	the transit mode
20	Tnst_Md_Cmtnd_yn	VIS::vbool	indicates whether the transit mode has been committed

ShipmentUpdateProgress_V1

The data for updating a shipment's progress or for a shipment's proof of delivery.

P	Element	Type	R	Description
1	Carr_cd	string<8>	•	the carrier code
2	Shpm_Id	string<28>		the shipment ID
3	Shpm_Num	string<12>		the shipment number
4	Shpm_Desc	string<12>		the shipment tracking number
5	Div_cd	string<4>		the division code
6	BOL_Num	String		the bill of lading number
7	Shpm_Leg_Id	String		the shipment leg ID
8	Shpm_Seq_Num	unsigned short		the shipment leg sequence number
9	Mmo	Memo_V1		a memo for this update
10	SECInfo	SEC_V1		the service event code information
11	POD	POD_V1		the proof of delivery information

Stop_V1

A load has two or more stops. It starts at one shipping location and ends at another. Each of these locations is a stop for the load. Multi-stop loads have intermediate stops to pick up or drop off additional shipments.

P	Element	Type	R	C	U	Description
1	Stop_Id	VIS::num28_0	•			the ID of the stop
2	Seq_Num	unsigned short				the position of the stop within the sequence of stops of the load
3	Num_Picks	unsigned short				the total number of picks for the stop
4	Num_Drops	unsigned short				the total number of drops for the stop
5	Shpg_Pnt_enu	VIS_V1::eShipPointType				the type of shipping point for the stop
6	Act_Arvl_dtt	VIS::timestamp				the date and time of the actual arrival
7	Act_Dptr_dtt	VIS::timestamp				the date and time of the actual departure
8	Pick_Cfmd_yn	VIS::vbool				indicates whether the pick stop is confirmed

P	Element	Type	R	C	U	Description
9	Pick_Cfmd_dtt	VIS::Timestamp				the date and time of the pick stop confirmation
10	Pick_Cfmd_Usr_Cd	string<10>				the user who confirmed the pick stop
11	Addr	Address_V1				the address of the stop
12	ShipLoc_Id	string<16>				the ID of the shipping point of the stop
13	PickShipmentLegIDs	VIS::StrIdList				the IDs of the shipment legs for a pick stop
14	DropShipmentLegIDs	VIS::StrIdList				the IDs of the shipment legs for a drop stop
15	Ldng_Inst_Exst_yn	VIS::vbool	•	•	•	indicates whether there is loading Instruction data only pick stops can have loading instructions; drop stops cannot
16	LdngInst	LoadInstruction_V1		•	•	the loading instructions

StopConfirmData_V1

The stop to be confirmed. You can identify the stop using one of the following parameters (the second item listed is a parameter set):

- Stop_Id
- OR-**
- Shpg_Loc_cd and Shpg_Loc_Typ_enu (shipping location code and shipping location type)
- OR-**
- Stop_Seq_Num (stop sequence number)

Ensure that only one of these parameters/parameter sets are provided, otherwise you will receive an error message because the stop will not be uniquely identifiable.

P	Element	Type	R	Description
1	Load_Id	VIS::num28_0	•	the load ID to be confirmed
2	Stop_Id	VIS::num28_0	•	the stop ID to be confirmed
3	Shpg_Loc_cd	string	•	the shipping location code
4	Shpg_Loc_Typ_enu	VIS_V1::eShipPointType	•	the shipping location type
5	Stop_Seq_Num	unsigned short	•	the stop sequence number
6	Carr_cd	string<8>		the carrier code for the load
7	Srvc_cd	string<4>		the service code for the load
8	Eqmt_typ	string<5>		the equipment type code
9	MBOL_Num	string<30>		the master BOL number to be used when generating the master BOL

P	Element	Type	R	Description
10	Confirm_dtt	VIS::timestamp		the confirm date and time the default value is the scheduled date of the stop
11	PrintBOL	VIS::vbool		indicates whether to print a master BOL
12	SECInfo	SEC_V1		the service event code information

StopUpdateProgress_V1

An event that describes the progress of a stop.

P	Element	Type	R	Description
1	Carr_cd	string<8>	•	the carrier code
2	Load_Id	string<28>		the load ID or reference number
3	Load_Desc	string<70>		the load tracking number
4	MBOL_Num	string<30>		the master BOL number
5	Shpg_Loc_cd	string<16>		the shipping location code
6	Shpg_Loc_Typ_enu	VIS_V1::eShipPointType		the shipping location type
7	Stop_Seq_Num	unsigned short		the stop sequential number
8	Mmo	Memo_V1		a memo for the stop update progress
9	EventInfo	StopEvent_V1		information about the event
10	POD	POD_V1		proof of delivery information

Non-Operational Freight Structures

NonOperationalFreight_V1

An entity that reflects freight transactions that are not handled operationally by Transportation Manager, for example, in situations where only freight auditing is

being performed. A hollow bullet (°) indicates the field is mandatory if the user's profile does not have the appropriate default values.

P	Element	Type	R	C	U	Description
1	Cust_Id	string<12>		•		the customer ID
2	Non_Op_Frht_Num	string<30>		•		the NOF number
3	Non_Op_Frht_Desc	string<30>		•	•	a description of the NOF
4	Div_Id	string<4>	•	•	•	the division ID
5	LgstGrp_Id	string<4>	•	•	•	the logistics group ID
6	Frst_ShipLoc_Id	string<16>	°	•	•	the location ID of the first stop
7	Frst_ShpgPnt_enu	VIS_V1::eShip PointType	°	•	•	the location type of the first stop
8	Frst_Addr	Address_V1		•	•	the override address of the first stop
9	Last_ShipLoc_Id	string<16>	°	•	•	the location ID of the last stop
10	Last_ShpgPnt_enu	VIS_V1::eShip PointType	°	•	•	the location type of the last stop
11	Last_Addr	Address_V1		•	•	the override address of the last stop
12	TdPy_Loc_Id	string<16>		•	•	the 3PL location ID
13	TdPy_Loc_Typ_enu	VIS_V1::eShip PointType		•	•	the location type of the 3PL
14	TdPy_Addr	Address_V1		•	•	the override address of the 3PL
15	Carr_Frht_Trms_enu	VIS_V1::eCarrFrhtTrms		•	•	the carrier freight terms
16	Carr_Pymt_Rblt_enu	VIS_V1::eFhgt Terms		•	•	the carrier payment responsibility
17	Shpd_dtt	VIS::timestamp	•	•	•	the shipment date
18	Divd_dtt	VIS::timestamp		•	•	the delivery date
19	Rvnu_Trpt_Odr_yn	VIS::vbool		•	•	the revenue/non-revenue transport order
20	Carr_Id	string<8>	•	•		the carrier ID
21	Cncy	VIS_V1::tCurrency	•	•	•	the currency type
22	UMsr_Sys_enu	VIS_V1::eUnitMeasureSys	•	•	•	the unit of measurement
23	UMsr_Wgt_enu	VIS_V1::eUnitMeasureWgt	•	•	•	the unit of weight
24	UMsr_Len_enu	VIS_V1::eUnit MeasureLgt	•	•	•	the unit of length
25	UMsr_Dst_enu	VIS_V1::eUnit MeasureDst	•	•	•	the unit of distance
26	Cdty_Id	string<12>	•	•	•	the commodity code
27	AP_Srvc_Id	string<4>	•	•	•	the A/P service code
28	AR_Srvc_Id	string<4>		•	•	the A/R service code
29	Rate_Typ_enu	VIS_V1::eRateType		•	•	the A/P rating type

P	Element	Type	R	C	U	Description
30	Eqmt_Typ_Id	string<5>		•	•	the equipment type
31	Prf_Ctr_typ	VIS_V1::tProfit Center		•	•	the profit center type
32	Tot_Scld_Wgt	VIS::num17_4		•	•	the total scaled weight
33	Tot_Vol	VIS::num11_4		•	•	the total volume for a shipment
34	Tot_Odr_Val_dlr	VIS::num15_2		•	•	the total order value for a shipment
35	Tot_Dcld_Val_dlr	VIS::num15_2		•	•	the total declared value for a shipment
36	Tot_Nmnl_Wgt	VIS::num17_4				the total nominal weight for a shipment
37	Tot_Tot_Tare_Wgt	VIS::num17_4		•	•	the total tare weight for a shipment
38	Tot_Tot_Pce	VIS::num7_0		•	•	the total number of pieces for a shipment
39	Tot_Tot_Skid	VIS::num7_0		•	•	the total number of skids for a shipment
40	Tot_Ldn_Len	VIS::num9_3		•	•	the total laden length for a shipment
41	Tot_Num_Shpm	unsigned short				the total number of shipments
42	Ldn_Len	VIS::num9_3				the laden length of the container
43	Len	VIS::num9_3				the length of the container
44	Wdth	VIS::num9_3				the width of the container
45	Hght	VIS::num9_3				the height of the container
46	CompTyp_Id	string<10>		•	•	the container type
47	Max_Scld_Wgt	VIS::num17_4				the maximum scaled weight for a shipment
48	Max_Vol	VIS::num11_4				the maximum volume for a shipment
49	Max_Odr_Val_dlr	VIS::num15_2				the maximum order value for a shipment
50	Max_Dcld_Val_dlr	VIS::num15_2				the maximum declared value for a shipment
51	Max_Nmnl_Wgt	VIS::num17_4				the maximum nominal weight for a shipment
52	Max_Tot_Tare_Wgt	VIS::num17_4				the maximum total tare for a shipment
53	Max_Tot_Pce	VIS::num7_0				the maximum number of pieces for a shipment
54	Max_Tot_Skid	VIS::num7_0				the maximum number of total skids for a shipment
55	Max_Ldn_Len	VIS::num9_3				the maximum laden length for a shipment
56	Max_Num_Shpm	unsigned short				the maximum number of shipments
57	Elpd_hrs	VIS::num6_2		•	•	the elapsed time
58	Lnst_Stop_Wait_Tm	VIS::num6_2		•	•	the maximum waiting time between stops in hours

P	Element	Type	R	C	U	Description
59	Memo_V1	Mmo		•	•	a memo
60	Non_Op_Frht_Typ_enu	VIS_V1::eNon OpFrhtType		•	•	the non-operational freight type
61	Non_Op_Frht_Src_enu	VIS_V1::eTrns Src		•	•	the transaction source
62	Bill_To_Cust_Id	string<12>	•	•	•	the Bill-to customer ID
63	Cust_Ver_Id	VIS::num28_0				the customer version
64	Bill_To_Ver_Id	VIS::num28_0				the Bill-to customer version
65	Echg_Rate	VIS::num11_6				the exchange rate
66	Eqmt_Typ_Cmdt_yn	VIS::vbool				indicates whether the equipment type is committed
67	Num_Stop	unsigned short				the number of stops
68	Vchr_Num	string<12>				the A/P voucher number
69	InvStat	VIS_V1::eStatus				the NOF invoice status
70	Cost_Ctr_typ	VIS_V1::tCost Center				the cost center
71	Crtd_dtt	VIS::timestamp				the creation date
72	CrtdUsr_Id	string<10>				the ID of the user that created this NOF
73	Updt_dtt	VIS::timestamp				the date this NOF was updated
74	CrtdUsr_Id	string<10>				the ID of the user that updated this NOF
75	APRatgInfo	RatingInfo_V1				A/P rating information
76	AP_Tot_Dist	VIS::num5_0				the total A/P distance
77	Tnst_Md_Enu	VIS_V1::eTransitMode		•	•	the transit mode possible values are TSM_NULL, TSM_AIR, TSM_RAIL, TSM_ROAD, TSM_MARINE, and TSM_PARCEL only TSM_PARCEL is active; if selected, the Kewill engine is called for rating
78	Drct_dist	VIS::num5_0				the direct distance
79	OutOfRout_dist	VIS::num5_0				the out-of-route distance
80	Ldd_dist	VIS::num5_0				the loaded distance
81	Tot_Unldd_dist	VIS::num5_0				the total unloaded distance
82	Ret_To_Orig_dist	VIS::num5_0				the return to origin distance
83	ARRatgInfo	RatingInfo_V1				the A/R rating information
84	AR_Tot_Dist	VIS::num5_0				the total A/R distance
85	Tnst_Md_Enu	VIS_V1::eTransitMode				the transit mode

P	Element	Type	R	C	U	Description
86	APChrgs	ChargeDetail List_V1				the A/P charges
87	ARChrgs	ChargeDetail List_V1				the A/R charges
88	WgtByFCs	NOFWeightByFreight ClassList_V1		•	•	the weight by freight classes
	Collections					
89	IgnoreRfrNums	VIS::vbool				indicates whether the contents of RfrNums should be ignored
90	RfrNums	RefNumberList_V1		•	•	the NOF reference numbers refer to additional information below
91	IgnoreStops	VIS::vbool				indicates whether the contents of Stops should be ignored
92	Stops	NOFStopList_V1		•	•	a list of all stops on this NOF
93	IgnoreChargeOverrides	VIS::vBool				indicates whether the contents of ChargeOverrides should be ignored
94	ChargeOverrides	ChargeOverrideList_V1		•	•	the charge overrides

RfrNums

If you define an NOF type reference number for input into the create or update function, the system ignores it. The system instead generates its own NOF reference number with the reference number field equal to the NOF object ID.

RatingInfo_V1

The rating information.

P	Element	Type	Description
1	UMsr_Sys_enu	VIS_V1::eUnitMeasureSys	the unit of measurement
2	UMsr_Wgt_enu	VIS_V1::eUnitMeasureWgt	the unit of weight
3	UMsr_Len_enu	VIS_V1::eUnitMeasureLgt	the unit of length
4	UMsr_Dst_enu	VIS_V1::eUnitMeasureDst	the unit of distance
5	Cncy	VIS_V1::tCurrency	the currency type
6	Echg_Rate	VIS::num11_6	the exchange rate
7	Spot_Rate_yn	VIS::vbool	indicates whether to use a spot rate
8	Spot_Rate_dtt	VIS::timestamp	the spot rate date and time
9	Ratd_dtt	VIS::timestamp	the rated date and time
10	Cost_Ctr_typ	VIS_V1::tCostCenter	the cost center type

P	Element	Type	Description
11	Sys_Calc_Amt_dlr	VIS::num15_2	the system calculated amount
12	Dsct_Amt_dlr	VIS::num15_2	the discount amount
13	Chgd_Amt_dlr	VIS::num15_2	the charged amount
14	Fedl_Tax_Amt_dlr	VIS::num15_2	the federal tax
15	Sta_Tax_Amt_dlr	VIS::num15_2	the state or provincial tax
16	Loc_Tax_Amt_dlr	VIS::num15_2	the local tax
17	Adtn_Chrg_dlr	VIS::num15_2	an additional charge
18	PreCslid_Amt_dlr	VIS::num15_2	the pre-consolidated amount
19	Rate_Cd	string<6>	the rate code
20	Rate_Cd_Tff_Id	VIS::num28_0	the rate code tariff ID
21	SpotRateUsr_Id	string<10>	the spot rate user ID
22	Tff_Srv_Cd	string<4>	the tariff service
23	Tff_Id	VIS::num28_0	the tariff ID

NOFWeightByFreightClass_V1

Non-operational weight by freight class identifiers.

P	Element	Type	R	C	U	Description
1	Sclid_Wgt	VIS::num11_4	•	•	•	the scaled weight of the freight class
2	FC_Id	string<5>	•	•	•	the freight class ID

NOF_Stop_V1

A non-operational freight stop.

P	Element	Type	R	C	U	Description
1	Seq_Num	unsigned short	•	•	•	the sequence number of this stop
2	ShpgPnt_enu	VIS_V1::eShip PointType		•	•	the location type of this stop
3	Shpg_Loc_Id	string<16>		•	•	the location ID of this stop
4	Shpg_Addr	Address_V1		•	•	the override address of this stop
5	Shpm_Pick	unsigned short	•	•	•	the number of pickups at this stop
6	Pick_Sclid_Wgt	VIS::num17_4	•	•	•	the scaled weight of the pickup
7	Pick_Vol	VIS::num11_4		•	•	the volume of the pickup
8	Pick_Odr_Val_dlr	VIS::num15_2		•	•	the order value of the pickup

P	Element	Type	R	C	U	Description
9	Pick_Dclد_Val_dlr	VIS::num15_2		•	•	the declared value of the pickup
10	Pick_Nmnl_Wgt	VIS::num17_4				the nominal weight of the pickup
11	Pick_Tot_Tare_Wgt	VIS::num17_4		•	•	the total tare weight of the pickup
12	Pick_Tot_Pce	VIS::num7_0		•	•	the total number of pieces of the pickup
13	Pick_Tot_Skid	VIS::num7_0		•	•	the total number of skids of the pickup
14	Pick_Ldn_Len	VIS::num9_3		•	•	the laden length of the pickup
15	Shpm_Drop	unsigned short	•	•	•	the number of drop-offs at this stop
16	Drop_Scld_Wgt	VIS::num17_4	•	•	•	the scaled weight of the drop-off
17	Drop_Vol	VIS::num11_4		•	•	the volume of the drop-off
18	Drop_Odr_Val_dlr	VIS::num15_2		•	•	the order value of the drop-off
19	Drop_Dclد_Val_dlr	VIS::num15_2		•	•	the declared value of the drop-off
20	Drop_Nmnl_Wgt	VIS::num17_4				the nominal weight of the drop-off
21	Drop_Tot_Tare_Wgt	VIS::num17_4		•	•	the total tare weight of the drop-off
22	Drop_Tot_Pce	VIS::num7_0		•	•	the total number of pieces of the drop-off
23	Drop_Tot_Skid	VIS::num7_0		•	•	the total number of skids of the drop-off
24	Drop_Ldn_Len	VIS::num9_3		•	•	the laden length of the drop-off
25	Pick_dtt	VIS::timestamp		•	•	the date of the pickup
26	Drop_dtt	VIS::timestamp		•	•	the date of the drop-off
27	Cdty_Note	string<70>		•	•	Additional information about the stop
28	IgnoreRfrNums	VIS::vbool				indicates whether RfrNums should be ignored
29	RfrNums	RefNumberList_V1		•	•	the NOF stop reference numbers

Rate Quotation Structures

ChargeCriteria_V1

The applicable entities for the potential charge details of a rate calculator request.

ChargeCriteria is a parameter of the `GetPossibleCharges` service (refer to “void RateQuotationSvc::GetPossibleCharges” in “API Services” on page 68).

P	Element	Type	R	Description
1	CustCarrCode	string<12>	•	the customer or carrier code
2	SrvCode	string<4>		the A/R or A/P service

P	Element	Type	R	Description
3	CommodityCode	string<4>		the commodity code
4	LoadAtCode	string<16>		the load-at code
5	HubFromCode	string<16>		the origin hub code leave this field blank if LoadAtCode has a value
6	HubToCode	string<16>		the destination hub code
7	ConsigneeCode	string<16>		the consignee code leave this field blank if HubToCode has a value

ChargeDetail_V1

Describes all valid lanes and associated costs, and provides input for the required options. You can overwrite manual input units and service level charge amounts where applicable. If you manually override an input unit for a container-based charge, then identify the container by a reference number.

P	Element	Type	R	Description
1	ItemId	VIS::num28_0		the ID of the charge detail always returned as an empty string
2	ChargeCode	string<4>	•	the code for the service, condition, or option charged
3	FreightClass	string<4>		the freight class of the charged entity
4	RatedAsFC	string<4>		the rated freight class
5	ChargeSequence	unsigned short		the charge detail number in a sequence
6	ChargeLevel	VIS_V1::eChargeLevel	•	the service, condition, or option
7	ChargeUnit	VIS::num11_4		the charge units
8	LookupUnit	VIS::num11_4		the lookup units
9	RateUnit	VIS::num11_4		the number of units at which this charge is rated
10	RateRange	VIS::num11_4		the rate range for calculating the charge
11	RatedAs	VIS::num11_4		the rate at which this charge is rated
12	ChargeAmount	VIS::num15_2		the charge amount including any discount or surcharge
13	Discount	VIS::num15_2		the discount or surcharge amount if negative, it is a discount
14	OptionApplyLevel	VIS_V1::eOptApplyLvl		the option level applied to this charge this field only applies if the ChargeLevel is Option
15	ManualOverrideUnit	VIS::num11_4		the manual override units for a charge
16	ManualOverride Amount	VIS::num11_4		the manual override amount for a service

FreightDetail_V1

The item to be rated. Freight All Kinds refers to tariff rates that do not incorporate different freight classes.

P	Element	Type	R	Description
1	FreightClassInfo	VIS_V1::WeightBy FreightClassList_V1	•	the freight class of the item being rated: FAK indicates Freight All Kinds freight classes under 100 are preceded by a blank padded with trailing blanks to make it 4 characters long
2	Cube	VIS::num11_4	•	the volume of the shipment
3	Order	VIS::num15_2	•	the total value of the shipment
4	Declared	VIS::num15_2	•	the declared value of the shipment
5	Pieces	unsigned short	•	the number of pieces of the shipment
6	Skids	unsigned short	•	the number of skids of the shipment
7	Scld_Wgt	VIS::num11_4	•	the scaled weight of the shipment

RateCriteria_V1

The rate criteria describe what is being shipped and the constraints for selecting eligible lanes.

Specify the origin and destination shipping points by doing one of the following actions:

- provide a shipping location type and shipping location code, with the shipping address implied
- explicitly provide an address

P	Element	Type	R	C	U	Description
1	ProcessARSide	VIS::vbool	•			indicates whether to find lanes for the A/R side rating (if true), or for the A/P side rating (if false)
2	OriginShpgPtType	VIS_V1::eShipPointType	•			the origin shipping point type if SPT_NULL, then the OriginShpgLocCd is blank, and the origin address fields are mandatory if not SPT_NULL, then the address of the shipping point is used for routing
3	OriginShpgLocCd	string<16>				the shipping location code for the corresponding origin shipping location type if OriginShpgPtType is not SPT_NULL, then this field is mandatory

P	Element	Type	R	C	U	Description
4	OriginCity	string<32>				the origin city for the routing set field and other origin address fields only if OriginShpgPtType is SPT_NULL
5	OriginState	string<4>				the origin state or province for the routing
6	OriginPostalCode	string<12>				the origin zip/postal code
7	OriginCountry	string<4>				the domain table string value for country
8	OriginLatitude	VIS::num6_4 type		•	•	the latitude of the origin's shipping point
9	OriginLongitude	VIS::num6_4 type		•	•	the longitude of the origin's shipping point
10	DestShpgPtType	VIS_V1::eShipPointType	•			the destination shipping point type if set to SPT_NULL, then DestShpgLocCd is blank, and the destination address fields are mandatory
11	DestShpgLocCd	string<16>				the shipping location code for the corresponding destination shipping location type if DestShpgPtType is not SPT_NULL, then this field is mandatory
12	DestinationCity	string<32>				the destination city for the routing Set this field and other destination address fields only if OriginShpgPtType is SPT_NULL
13	DestinationState	string<4>				the destination state or province for the routing
14	DestinationPostalCode	string<12>				the destination zip/postal code
15	DestinationCountry	string<4>				corresponds to the domain table string value for country
16	DestinationLatitude	VIS::num6_4 type		•	•	the latitude of the destination's shipping point
17	DestinationLongitude	VIS::num6_4 type		•	•	the longitude of the destination's shipping point
18	CustomerCode	string<12>				the customer that the rating is for required for A/R rating may not be relevant for A/P rating if, for example, there are multiple customers for the items being shipped
19	CarrierCode	string<8>				if specified, the best lane chosen will have this carrier
20	Service	string<4>				if specified, the best lane chosen will provide this service

P	Element	Type	R	C	U	Description
21	ProjectCode	string<12>				the project code: only relevant for A/R rating its use depends upon the customer settings
22	ConsigneeGroup	VIS_V1::tConsigneeGroup				the consignee group: only relevant for A/R rating its use depends upon the customer settings
23	UseTariffHierarchy	VIS::vbool				indicates whether to use the tariff hierarchy only relevant for A/R rating
24	ShipmentDivision	string<4>	•			the division of the shipment
25	CommodityCode	string<4>				if specified, only lanes providing this commodity code are eligible for selection
26	EarliestDepartureTime	VIS_V1::timestamp				if specified, rate shopping is done for the provided date otherwise this is the current date
27	LatestDepartureTime	VIS_V1::timestamp				if not specified, this is the current date plus two months
28	EarliestArrivalTime	VIS_V1::timestamp				if specified, rate shopping is done for the provided date, otherwise this is the current date
29	LatestArrivalTime	VIS_V1::timestamp				if not specified, this is the current date plus two months
30	ShipUMsrSys	VIS_V1::eUnitMeasureSys	•			the unit of measure for the shipment
31	ShipUMsrWgt	VIS_V1::eUnitMeasureWgt	•			the weight unit of measure for the shipment
32	ShipUMsrLen	VIS_V1::eUnitMeasureLen	•			the length unit of measure for the shipment
33	FreightDetails	FreightDetail_V1	•			unit and freight class information describing the contents of the shipment
34	ChargeOverwrites	ChargeDetailList_V1				a set of charge details used for charge overwrites
35	Currency	VIS_V1::tCurrency	•			the currency code for the shipment, corresponding to a domain table currency code
36	Stops	Long				the number of stops for the freight
37	Distance	VIS::num15_2				the distance that the freight is being shipped

Schedule_V1

A delivery schedule for shipping on a lane, and the times and stops for this schedule.

P	Element	Type	Description
1	Dlvy_Schd_cd	string<15>	the delivery schedule code always returned as an empty string
2	Itnr_cd	string<15>	the itinerary code
3	Itnr_TmTbl_cd	string<15>	the itinerary timetable code
4	Itnr_TmTbl_Ent_cd	string<15>	the itinerary timetable entry code
5	Stop_Schd_Lst	VIS_V1::StopScheduleList_V1	the stops and their associated times
6	Tot_Elpd_tm	VIS::time	the total elapsed time in hours
7	Tot_Wait_tm	VIS::time	the total departure time in hours
8	Default_Schd	VIS::vbool	the best elapsed time for the schedules

StopSchedule_V1

The schedule information of a stop for the freight being shipped.

P	Element	Type	Description
1	Stop_Seq_num	unsigned short	the sequence number of the stop in the stop list required by the rate quotation
2	Arrival_dtt	VIS_V1::timestamp	the arrival date and time
3	Departure_dtt	VIS_V1::timestamp	the departure date and time
4	Wait_tm	VIS::num5_2	the wait time
5	Load_tm	VIS::num5_2	the load time

ValidLane_V1

A valid candidate lane for shipping the specified items.

P	Element	Type	Description
1	TariffCode	string<12>	the tariff number of a valid lane
2	MasterTariffCode	string<12>	the master tariff number of a valid lane
3	Service	string<4>	the service of a valid lane
4	RateCode	string<6>	the rate code of a valid lane
5	ChargeAmount	VIS::num15_2	the charge for the shipping
6	ChargeAmountTffc	VIS::num15_2	the charge in the currency of the tariff

P	Element	Type	Description
7	Currency	VIS_V1::tCurrency	the currency of the tariff
8	ExchangeRate	VIS::num11_6	the exchange rate of the tariff to the system currency
9	Carrier	string<8>	the carrier for the lane
10	SchedulingValid	VIS::vbool	indicates whether shipment information is available
11	BestTotalHrsElapsedTime	VIS::num5_2	the hours of elapsed time for delivery
12	Srvc_Grd_typ	short	the grade for the service provided by the carrier
13	Schedule	Schedule_V1	schedule and stop information
14	ChargeDetails	VIS_V1::ChargeDetailList_V1	the charge details for the lane

Shipment Structures

CompTrackingNum_V1

A component tracking number override used in shipment confirmation.

P	Element	Type	R	Description
1	Rfrc_Num_Typ	VIS_V1::tReferenceNumType	•	the reference number type identifying the component to which the tracking number is applied must be "API"
2	Rfrc_Num	string<30>	•	the reference number identifying the component to be applied
3	Trkg_Num	string<30>	•	the overriding tracking number for the specified component

SelectionCriteria_V1

The criteria for selecting shipments to be sent to Transportation Optimizer for Manager.

You can use this function in the following ways. In all cases, you must enter the plan ID.

Enter only the plan ID - All the other fields are blank. This will send all shipments that are already attached to the current plan to Optimizer.

Enter the optional criteria - This will send only shipments in the current plan that match the optional criteria to Optimizer.

Enter the shipment ID field - You enter IDs of shipments that are not yet attached to the current plan. The system checks that these shipments have same division ID and logistics ID as the current plan. They are then attached to the current plan, and sent with the other shipments in the plan to Optimizer.

Frm_Pkup_dtts

Frm_Pkup_dtts requires a corresponding To_Pkup_dtts value, for example:

Frm_Pkup_dtts (3/3/1999, 4/27/1999)

To_Pkup_dtts (3/15/1999, 4/30/1999)

Shipments with Pickup From date greater than or equal 3/3/1999 and a Pickup To date less than or equal to 3/15/1999 will be selected.

Shipments with a Pickup From date greater than or equal to 4/27/1999 and a Pickup To date less than or equal to 4/30/1999 will also be selected.

Frm_Dlvy_dtts

Frm_Dlvy_dtts requires a corresponding To_Dlvy_dtts value, for example:

Frm_Dlvy_dtts (3/3/1999)

To_Dlvy_dtts (3/15/1999)

Shipments with Delivery From date greater than or equal 3/3/1999 and a Delivery To date less than or equal to 3/15/1999 will be selected.

P	Element	Type	R	Description
1	PlanId	VIS::num28_0	•	the current plan ID Listed next are the optional criteria which the system uses when entered to select shipments
2	ShipmentIds	VIS::StrIdList		the list of shipment IDs
3	OriginCities	VIS::StrIdList		the list of origin cities
4	DestCities	VIS::StrIdList		the list of destination cities
5	Frm_Pkup_dtts	VIS::StrIdList		the list of Pickup From dates
6	To_Pkup_dtts	VIS::StrIdList		the list of Pickup To dates see Frm_Pkup_dtts
7	Frm_Dlvy_dtts	VIS::StrIdList		the list of Delivery From dates
8	To_Dlvy_dtts	VIS::StrIdList		the list of DeliveryTo dates see Frm_Dlvy_dtts

ShipmentCompChangeData_V1

The shipment composition to be changed.

P	Element	Type	R	Description
1	Trpt_Odr_Id	string<12>	•	the transport order ID of the shipment

P	Element	Type	R	Description
2	Shpm_Id	VIS::num28_0		the shipment ID if not provided and the given transport order contains only one shipment, then this shipment is assumed to be the one you want to change
3	Comps	ComponentList_V1		the list of components to be changed

ShipmentConfirmData_V1

The shipment leg to be confirmed.

P	Element	Type	R	Description
1	Shpg_Leg_Id	VIS::num28_0	•	the ID of the shipment leg to be confirmed
2	Carr_cd	string<8>		the carrier code to be used for the shipment leg
3	Srvc_cd	string<4>		the service code to be used for the shipment leg
4	Eqmt_typ_cd	string<5>		the equipment type code
5	MBOL_Num	string<30>		the BOL number to be used when generating the BOL
6	Confirm_dtt	VIS::timestamp		the confirm date and time the default value is the current date time
7	PrintBOL	VIS::vbool		indicates whether to print a BOL
8	ARrating	VIS::vbool		indicates whether the A/R rating is needed the default value is false
9	IgnoreComponentTrackingNums	VIS::vbool		indicates whether ComponentTrackingNums is processed
10	ComponentTrackingNums	CompTrackingNumList_V1		the component tracking number list
11	PrintParcelLabel_yn	VIS::vbool		indicates whether to print package labels for the shipment leg available only for parcel shipment legs
12	ParcelLabelPrinterName	string<14>		the name of the printer that will produce the labels, if parcel labels have been requested the printer must be configured as a parcel printer under the parcel mappings

Shipment Order Entry Structures

Container_V1

A container of a shipment.

P	Element	Type	R	C	U	Description
1	Elmt_Id	VIS::num28_0				the element ID
2	Comp_Typ_Id	string<20>	•	•	•	the component type ID
3	Qnty	VIS::num7_0	•	•	•	the number of identical containers
4	Len	VIS::num9_3		•	•	the length of the container if NULL, then the default value is based on the component type that CompTyp_Id specifies
5	Wdth	VIS::num9_3		•	•	the width of the container if NULL, then the default value is based on the component type that CompTyp_Id specifies
6	Hght	VIS::num9_3		•	•	the height of the container if NULL, then the default value is based on the component type that CompTyp_Id specifies
7	Comp_ShpG_Info	ShippingInfo_V1		•	•	the component shipping information structure
8	Comp_Desc	string<69>		•	•	a description of the component
9	Stck_cd	unsigned short		•	•	the stacking mode ID
10	Stck_fctr	unsigned short		•	•	the stacking factor: the amount of weight that can be stacked on top of this container
11	Stck_grp	unsigned short		•	•	the stacking group: indicates the relative durability of this container
12	Nest_Dimn_enu	VIS_V1::eCntrDimn		•	•	the container dimension along which the nesting occurs if no value is assigned, then nesting is not allowed
13	Nest_Val	VIS::num9_3		•	•	the nesting value: the increase along the nesting dimension with each additional nested unit if no value is assigned, then nesting is not allowed
14	Max_Nest_Size	unsigned short		•	•	the maximum number of allowed nested units if no value is assigned, then nesting is not allowed

P	Element	Type	R	C	U	Description
15	Ocrn	unsigned short				the container occurrence used as a factor when calculating the total scaled weight, volume, order value, and declared value for the shipment having this container equal to 1 when the summarized entry mode is false equal to the container quantity when the summarized entry mode is true
16	LC_Comp_typ	unsigned short				the load configuration component type
17	RatgUnitTyp_enu	VIS_V1::eUnitType				the rating unit type
18	Max_Scld_Wgt	VIS::num11_4				the maximum scaled weight
19	Shpm_Cntr_Num	string<30>		•	•	the shipment container number
20	Shpm_Cntr_Xref	string<30>		•	•	the cross-reference of the shipment container
21	Itm_GL_cat	unsigned short				the G/L category for items for this container
22	Inpt_Src_enu	VIS_V1::eTrnsSrc		•		the source of data input, such as API or through the GUI
23	MrkedForLoc_enu	VIS_V1::eShipPointType		•	•	a location to which the container is taken after the shipment is delivered
24	MrkedForLoc_cd	string<16>		•	•	the ID for the location to which the container is taken after the shipment is delivered
25	IgnoreCntrOrtns	VIS::vbool				indicates whether the contents of CntrOrtns (next) should be ignored
26	CntrOrtns	CntrOrtnList_V1		•	•	the container orientation objects attached to the container
27	IgnoreWgtByFCs	VIS::vbool				indicates whether the contents of WgtByFCs (next) should be ignored
28	WgtByFCs	WeightByFreight ClassList_V1		•	•	the weight by freight classes associated with the container
29	IgnoreShpmlItems	VIS::vbool				indicates whether the contents of ShpmlItems (next) should be ignored
30	ShpmlItems	ShipmentItemList_V1		•	•	shipment items associated with the container
31	IgnoreRfrNums	VIS::vbool				indicates whether the contents of RfrNums (next) should be ignored
32	RfrNums	CompRefNumberList_V1		•	•	reference numbers associated with the container

ContainerPlan_V1

Container information that is used in the PlanShipment service.

P	Element	Type	R	C	U	Description
1	Comp_Typ_Id	string<20>	•	•	•	the component type code
2	Qty	VIS::num7_0	•	•	•	the number of identical containers
3	Comp_Shpgr_Info	ShippingInfo_V1	•	•	•	the container shipping information structure
4	Cdty_cd	string<12>	•	•	•	the commodity code
5	Frm_Pkup_dtt	VIS::timestamp	•	•	•	the start of the date and time range in which the container should be picked up
6	To_Pkup_dtt	VIS::timestamp	•	•	•	the end of the date and time range in which the container should be picked up
7	Frm_Dlvy_dtt	VIS::timestamp	•	•	•	the start of the date and time range in which the container should be delivered
8	To_Dlvy_dtt	VIS::timestamp	•	•	•	the end of the date and time range in which the container should be delivered
9	Container_Id	string	•	•	•	the container ID
10	Sp_Csld_Cls	string<12>		•	•	the shipment consolidation class
11	Mrge_Csld_Cls_Cd	string<30>		•	•	the merge consolidation class code
12	Mrge_Csld_Seq_N um	unsigned short		•	•	the merge consolidation sequence number
13	Len	VIS::num9_3		•	•	the length of this container
14	Wdth	VIS::num9_3		•	•	the width of this container
15	Hght	VIS::num9_3		•	•	the height of this container
16	Allow_Split_yn	VIS::vbool		•	•	indicates whether this container can be split
17	Stck_Hght	unsigned short		•	•	the stack height
18	Stck_Grp	unsigned short		•	•	the durability of the container's contents relative to the other containers
19	Stck_Fctr	unsigned short		•	•	the amount of weight that can be stacked on top of this container
20	IgnoreCntrOrtns	VIS::vbool				indicates whether the contents of the next field (CntrOrtns) can be ignored
21	CntrOrtns	CntrOrtnList_V1		•	•	the container orientation objects attached to the container
22	Nest_Dimn_enu	VIS_V1::eCntrDi mn		•	•	the nesting container dimension: Height, Length or Width

P	Element	Type	R	C	U	Description
23	Nest_Val	VIS::num9_3		•	•	used with the nesting dimension to indicate how a container increases in size with each additional nested unit
24	Max_Nest_Size	unsigned short		•	•	the maximum number of units in a nested stack

NMFC_V1

A national motor carrier freight classification (NMFC).

P	Element	Type	R	C	U	Description
1	NMFC_cd	string<30>		•	•	the ID code of the NMFC
2	NMFC_Desc	string<69>		•	•	a description of the NMFC
3	Frht_Cls_cd	string<4>		•	•	the freight class code
4	Rfrncd_By_ItmMstr_yn	VIS::vbool				indicates whether the NMFC is referenced by item master
5	Mmo	Memo_V1		•	•	a memo for the NMFC
6	Crted_dtt	VIS::timestamp				the date and time that the NMFC was created
7	Updt_dtt	VIS::timestamp				the date and time that the NMFC was updated
8	Crted_Usr_cd	string<10>				the ID of the user who created the NMFC
9	Updt_Usr_cd	string<10>				the ID of the user who updated the NMFC

PlanningRequest_V1

A request to plan shipments.

P	Element	Type	R	C	U	Description
1	Cust_cd	string<12>		•	•	the customer code
2	TO_Ent_Ver_cd	string<10>		•	•	the transport order entry version code
3	TO_Ent_Typ_cd	string<2>		•	•	the transport order entry type code
4	Ship_Frm_enu	VIS_V1::eShipFromType		•	•	the type of origin shipping point
5	Frm_Shpg_Loc_cd	string<16>		•	•	the ID code of the shipment origin point
6	Ship_To_enu	VIS_V1::eShipToType		•	•	the type of destination shipping point
7	To_Shpg_Loc_cd	string<16>		•	•	the ID code of the shipment destination point
8	Jrny_Tplt_cd	string<8>		•	•	the journey template code
9	Allow_Split_yn	VIS::vbool		•	•	indicates whether shipments can be split
10	Containers	ContainerPlanList_V1		•	•	the list of container plans

P	Element	Type	R	C	U	Description
11	Orig_Shpg_Loc_Addr	Address_V1				the shipping address of the origin point
12	Dest_Shpg_Loc_Addr	Address_V1				the shipping address of the destination point
13	Override_Carr_Cd	string<8>				the carrier override code
14	Override_Srv_Cd	string<4>				the service override code

Shipment_V2

A shipment entity.

P	Element	Type	R	C	U	Description
1	Shpm_Id	VIS::num28_0				an automatically generated shipment system identifier
2	Shpm_Num	string<12>				the shipment number
3	Cust_cd	string<12>		•	•	the customer ID code
4	TO_Ent_Ver_cd	string<10>			•	the transport order entry version code
5	TO_Ent_Typ_cd	string<2>			•	the transport order entry type code
6	Lgst_Grp_cd	string<4>			•	the logistics group code
7	Div_cd	string<4>			•	the division code
8	Jrny_Tplt_cd	string<8>			•	the journey template code
9	Inpt_Src_enu	VIS_V1::eInputSource			•	the source from which the shipment was created
10	Prj_Id	string<12>			•	the project ID
11	Hold_yn	VIS::vbool			•	indicates whether the shipment is on hold or is to be placed on hold later
12	Shpm_Entry_Md_enu	VIS_V1::eTranspOrderMode			•	the mode in which the shipment was created
13	Cnse_Grp_typ	VIS_V1::tConsigneeGroup			•	the type of consignee group
14	PreferedCarr_Id	string<8>			•	the default carrier ID
15	PreferedAPSvc_Id	string<4>			•	the default A/P service ID
16	EqmtType	string<4>			•	the equipment type

P	Element	Type	R	C	U	Description
17	CarrSrvcEqmtRule	VIS_V1::eCarr SrvcEqmtRule		•		the carrier service equipment rule propagates the carrier, service and equipment type defined in the shipment structure to the first leg, last leg, or all legs of the itinerary this rule only applies to a journey template, and does not apply if the shipment throughpoints are defined values are CSE_NULL, CSE_FIRSTLEG, CSE_LASTLEG, and CSE_ALLLEGS
18	AR_Srvc_cd	string<4>		•	•	the A/R service code
19	SalesPerson_Id	string<10>		•	•	the sales person responsible for the shipment
20	Plan_Id	VIS::num28_0		•	•	the plan ID
21	Shpm_Desc	string<12>		•	•	a description of the shipment
22	Frht_Trm_enu	VIS_V1::eFhgtTerms		•	•	the freight terms for the transport order
23	BillTo_Cust_cd	string<12>		•	•	the customer to whom the bill will be sent
24	Drct_Frht_yn	VIS::vbool		•	•	indicates whether the shipment is to be shipped directly
25	Urgt_yn	VIS::vbool		•	•	indicates whether the shipment is urgent
26	Csld_Cls	string<30>		•	•	the shipment consolidation class
27	Mrge_Csld_Cls_id	string<30>		•	•	the merge in transit consolidation class (MITCC)
28	Mrge_Csld_Cls_Num	unsigned short		•	•	the merge in transit consolidation class (MITCC) sequence number
29	Memo	Memo_V1		•	•	a memo associated with the shipment
30	Ship_Frm_enu	VIS_V1::eShipFromType		•	•	the type of shipment origin point
31	Frm_Shpg_Loc_cd	string<16>		•	•	the code of the shipment origin point
32	Frm_Shpg_Desc	string<69>		•	•	a description of the shipment origin point
33	Frm_Shpg_Addr	Address_V1		•	•	the address of the shipment origin point
34	SpFm_Apt_Rqrd_yn	VIS::vbool		•	•	indicates whether an appointment is required at the shipment origin point
35	Ship_To_enu	VIS_V1::eShipToType		•	•	the type of shipment destination point

P	Element	Type	R	C	U	Description
36	To_Shpg_Loc_cd	string<16>		•	•	the shipment destination point code
37	To_Shpg_Desc	string<69>		•	•	a description of the shipment destination point
38	To_Shpg_Addr	Address_V1		•	•	the address of the shipment destination
39	SpTo_Apt_Rqrd_yn	VIS::vbool		•	•	indicates whether an appointment is required for the shipment destination
40	Frm_Pkup_dtt	VIS::timestamp	•	•	•	the start of the date and time range in which the shipment should be picked up
41	To_Pkup_dtt	VIS::timestamp	•	•	•	the end of the date and time range in which the shipment should be picked up
42	Frm_Dlvy_dtt	VIS::timestamp	•	•	•	the start of the date and time range in which the shipment should be delivered
43	To_Dlvy_dtt	VIS::timestamp	•	•	•	the end of the date and time range in which the shipment should be delivered
44	ReComputeDates_yn	VIS::vbool		•		indicates whether to re-calculate the pickup and delivery dates using the default dates
45	Cdty_cd	string<12>		•	•	the commodity code
46	Prof_Ctr_typ	VIS_V1::tProfitCenter		•	•	the profit center type
47	Cust_SrvRep_typ	VIS_V1::tCustSrvRep		•	•	the customer service representative type
48	PickSeqReq_yn	VIS::vbool		•	•	indicates whether a pick sequence is required for the shipment
49	Batch_Number	unsigned short		•	•	the batch number the maximum value is 9999
50	Crted_Usr_cd	string<10>				the date and time that the shipment was created
51	Updt_Usr_cd	string<10>				the date and time that the shipment was updated
52	Crted_dtt	VIS::timestamp				the ID of the user who created the shipment
53	Updt_dtt	VIS::timestamp				the ID of the user who updated the shipment
54	Cncy	VIS_V1::tCurrency				the currency type

P	Element	Type	R	C	U	Description
55	Echg_Rate	VIS::num11_6				the exchange rate
56	Curr_OptlStat_cd	VIS_V1::eStatus				the current operational status
57	Curr_FncIStat_cd	VIS_V1::eStatus				the current financial status
58	UOM	UMsr_V1				the unit of measurement structure
59	TariffUOM	UMsr_V1				the tariff unit of measurement
60	ShippingInfo	ShippingInfo_V1				the shipping information structure
61	INCO_Terms_cd	string<4>		•	•	the INCO terms code
62	Buyer_Seller_enu	VIS_V1::eINCOBuyerSeller		•	•	a description of the buyer and seller relationship
63	INCO_ShpG_Loc_cd	string<16>		•	•	the default INCO terms shipping location code
64	INCO_ShpG_Loc_Typ_enu	VIS_V1::eShipPointType		•	•	the default INCO terms shipping location type
65	Prepaid_Seg_Only_yn	VIS::vbool		•	•	indicates whether the segment is prepaid only
66	Odr_Val_Rqrd_yn	VIS::vbool				the order value
67	Dcl_Val_Rqrd_yn	VIS::vbool				the declared value
68	Defr_AP_Ratg_yn	VIS::vbool				indicates whether the A/P rating is deferred
69	Defr_AR_Ratg_yn	VIS::vbool				indicates whether the A/R rating is deferred
70	Rvnu_Trpt_Odr_yn	VIS::vbool				indicates whether the shipment produces revenue
71	MaxOrderValue	VIS::num15_2				the maximum order value
72	Max_Wgt	VIS::num11_4				the maximum weight
73	Max_Vol	VIS::num11_4				the maximum volume
74	Inelgb_Sel_Dltd_yn	VIS::vbool				indicates whether the shipment is ineligible for selective deletion
75	OriginDefault_yn	VIS::vbool			•	indicates whether the origin shipping location profile is refreshed
76	DestinationDefault_yn	VIS::vbool			•	indicates whether the destination shipping location profile is refreshed
77	Itm_Group_cd	string<12>				the item group code
78	Comp_Typ_Grp_cd	string<12>				the component type group code

P	Element	Type	R	C	U	Description
79	Tnst_Md_Enu	VIS_V1::eTransitMode		•	•	the transit mode possible values are TSM_NULL, TSM_AIR, TSM_RAIL, TSM_ROAD, TSM_MARINE, and TSM_PARCEL only TSM_PARCEL is active; if selected, the Kewill engine is called for rating
80	IgnoreRefNums	VIS::vbool				indicates whether the contents of RefNums (next) should be ignored
81	RefNums	RefNumberList_V1		•	•	reference numbers attached to the shipment
82	IgnoreContainers	VIS::vbool				indicates whether the contents of Containers (next) should be ignored
83	Containers	ContainerList_V1		•	•	containers associated with the shipment
84	IgnoreThruPoints	VIS::vbool				indicates whether the contents of ThruPoints (next) should be ignored
85	ThruPoints	ShipmentThruPointList_V1		•	•	the intermediate points of the shipment

ShipmentHeader_V2

The essential information about a shipment. This structure is used for output only.

P	Element	Type	Description
1	Shpm_Id	VIS::num28_0	an automatically generated system identifier for the shipment
2	Shpm_Num	string<12>	the shipment number
3	Shpm_Desc	string<12>	a description of the shipment
4	BillTo_Cust_cd	string<12>	the customer to whom the bill will be sent
5	Drct_Frht_yn	VIS::vbool	indicates whether the shipment is to be shipped directly
6	Urgt_yn	VIS::vbool	indicates whether the shipment is urgent
7	Csld_Cls	string<30>	the shipment consolidation class
8	Hold_yn	VIS::vbool	indicates whether the shipment is on hold or will be on hold
9	AR_Srvc_cd	string<4>	the A/R service code
10	Mrge_Csld_Cls_id	string<30>	the merge in transit consolidation class (MITCC)
11	Mrge_Csld_Cls_Num	unsigned short	the MITCC sequence number
12	Ship_Frm_enu	VIS_V1::eShipFromType	the type of shipment origin point

P	Element	Type	Description
13	Frm_Shpg_Loc_cd	string<16>	the code of the shipment origin point
14	Frm_Shpg_Desc	string<69>	a description of the shipment origin point
15	Frm_Shpg_Addr	Address_V1	the shipment origin address
16	SpFm_Apt_Rqrd_yn	VIS::vbool	indicates whether an appointment is required for the shipment origin location
17	Ship_To_enu	VIS_V1::eShipToType	the type of shipment destination point
18	To_Shpg_Loc_cd	string<16>	the code of the shipment destination point
19	To_Shpg_Desc	string<69>	a description of the shipment destination point
20	To_Shpg_Addr	Address_V1	the shipment destination address
21	SpTo_Apt_Rqrd_yn	VIS::vbool	indicates whether an appointment is required for the shipment destination
22	Frm_Pkup_dtt	VIS::timestamp	the start of the date and time range in which the shipment should be picked up
23	To_Pkup_dtt	VIS::timestamp	the end of the date and time range in which the shipment should be picked up
24	Frm_Dlvy_dtt	VIS::timestamp	the start of the date and time range in which the shipment should be delivered
25	To_Dlvy_dtt	VIS::timestamp	the end of the date and time range in which the shipment should be delivered
26	Cdty_cd	string<12>	the commodity code
27	Cnse_Grp_typ	VIS_V1::tConsigneeGroup	the consignee group type
28	PreferedCarr_Id	string<8>	the default carrier ID
29	PreferedAPSvc_Id	string<4>	the default A/P service ID
30	EqmtType	string<4>	the equipment type
31	Tnst_Md_Enu	VIS_V1::eTransitMode	the transit mode possible values are TSM_NULL, TSM_AIR, TSM_RAIL, TSM_ROAD, TSM_MARINE, and TSM_PARCEL only TSM_PARCEL is active; if selected, the Kewill engine is called for rating

ShipmentPlan_V1

The output structure that the PlanShipments service creates.

P	Element	Type	Description
1	ShipmentGroup	unsigned long	a group of planned shipments
2	Containers	ContainerPlanList_V1	a list of containers

ShipmentThruPoint_V1

The intermediate point of a shipment.

P	Element	Type	R	C	U	Description
1	Ship_Thru_enu	VIS_V1::eShipThruPointType	•	•	•	the intermediate point type
2	ShipThruLoc_id	string<16>	•	•	•	the ID code of the point
3	Carr_Id	string<8>		•	•	the carrier to use for the journey step this can be blank if you do not need a specific carrier
4	Srvc_Id	string<4>		•	•	the service to use for the journey step this can be blank if you do not need a specific service
5	ReqEqmtType	string<4>		•	•	the equipment type code

Tariff Structures

FC_Crs_Rfrc_V1

A cross-reference for a freight class.

P	Element	Type	R	C	U	Description
1	FC_cd	string<4>	•	•	•	the original freight class
2	RatedAs_FC_cd	string<4>	•	•	•	the Rate As freight class

LaneAssc_V1

A lane for a tariff.

P	Element	Type	R	C	U	Description
1	Tff_Id	VIS::num28_0	•	•		the associated tariff ID
2	Tff_Srvc_Cd	string<4>	•	•		the associated tariff service *ALL means apply to all services

P	Element	Type	R	C	U	Description
3	Base_Div_Id	string<4>	•	•		the division code
4	Orig_Zn_Id	string<8>		•		the origin zone Specify either this field or Orig_Hub_Id during creation
5	Dest_Zn_Id	string<8>		•		the destination zone Specify either this field or Dest_Hub_Id during creation
6	Orig_Hub_Id	string<16>		•		the origin hub
7	Dest_Hub_Id	string<16>		•		the destination hub
8	Dest_DC_Id	string<16>				the destination distribution center
9	Rate_Cd	string<6>		•	•	the rate code used for the lane
10	Mstr_Rate_Cd_yn	VIS::vBool	•	•	•	indicates whether the rate code is defined within the tariff or obtained from the master tariff
11	Cdty_cd	string<12>	•	•	•	the commodity code *ALL means apply to all commodities
12	Stat_enu	VIS_V1::eLaneStatus		•	•	the current status of the lane
13	Orig_sta	string<4>				the origin state or province
14	Orig_Ctry	string<4>				the origin country
15	Dest_Sta	string<4>				the destination state or province
16	Dest_Ctry	string<4>				the destination country
17	CrtdUsr_Id	string<10>				the ID of the user who created the lane
18	UpdtUsr_Id	string<10>				the ID of the user who updated the lane
19	Crtd_dtt	VIS::timestamp				the date and time that the lane was created
20	Updt_dtt	VIS::timestamp				the date and time that the lane was updated
	Collections					
21	Rstc_Exst_yn	VIS::vbool				indicates whether to consider the contents of the restriction
22	Restriction	Rstc_V1		•	•	the restriction associated with the lane

Rate_V1

A rate.

P	Element	Type	R	C	U	Description
1	Tff_Id	VIS::num28_0	•	•		the associated tariff number
2	Rate_Cd	string<6>	•	•		the associated rate code

P	Element	Type	R	C	U	Description
3	Tff_Srvc_Cd	string<4>	•	•	•	the associated tariff service
4	Tff_Chrg_Cd	string<4>	•	•	•	the associated tariff charge: condition/option
5	Frht_Cls_Cd	string<4>	•	•	•	the associated freight class
6	Rate_for_Shpm_yn	VIS::vbool	•	•	•	indicates whether this rate applies to shipment rating
7	Bs_Chrg_dlr	VIS::num15_2		•	•	the base charge amount
8	Min_Chrg_dlr	VIS::num15_2		•	•	the minimum charge amount
9	Max_Chrg_dlr	VIS::num15_2		•	•	the maximum charge amount
10	Num_of_Rng_Rate	unsigned short				the number of associated ranges
11	Rng_Id	string<4>				the associated range code
12	Collections					
13	IgnoreRange Rates	VIS::vbool				indicates whether the contents of the RangeRates (next) should be ignored
14	RangeRates	RngRateList_V1		•	•	the ranges associated with this rate

RateCd_V1

A rate code for a tariff.

P	Element	Type	R	C	U	Description
1	Tff_Id	VIS::num28_0	•	•		the associated tariff number
2	Rate_Cd	string<6>	•	•		the ID code of the rate code
3	Desc	string<70>	•	•	•	a description of this rate code
4	CrtdUsr_Id	string<10>				the ID of the user who created this rate code
5	UpdtUsr_Id	string<10>				the ID of the user who updated this rate code
6	Crted_dtt	VIS::timestamp				the date and time that this rate code was created
7	Updt_dtt	VIS::timestamp				the date and time that this rate code was updated
	Collections					
8	IgnoreRates	VIS::vbool				indicates whether the contents of Rates (next) should be ignored
9	Rates	RateList_V1		•	•	the rates associated with this rate code

RngRate_V1

A rate range. The fourth element refers to clipping. Clip is a method of calculating charges. The system calculates an amount for the portion of the rating unit greater than the previous range break. This is then added to the full charge associated with the previous range break as defined using the base charge.

P	Element	Type	R	C	U	Description
1	Rng_Cd	string<4>	•	•		a unique range rate code
2	Rng_To	VIS::num11_4	•	•		a description of this range
3	Chrg_Typ	VIS_V1::eChargeType				the charge type
4	Clip_yn	VIS::vbool				indicates whether the rate should be clipped
5	Brk_Bs_dlr	VIS::num15_2	•	•	•	the base charge
6	Brk_Amt_dlr	VIS::num11_4	•	•	•	the base rate
7	Vlid_yn	VIS::vbool				indicates whether the range is valid

SchgChrg_V1

A surcharge for a tariff.

P	Element	Type	R	C	U	Description
1	Tff_Id	VIS::num28_0	•	•		the associated tariff number
2	Srv_Cd	string<4>	•	•		the associated tariff service
3	Chrg_Cd	string<4>	•	•		the associated tariff charge: condition or option
4	Schg_Pct	VIS::num5_2		•	•	the surcharge percentage
5	Schg_Rate	VIS::num7_2		•	•	the surcharge rate

SchgRate_V1

A surcharge rate for a tariff.

P	Element	Type	R	C	U	Description
1	Tff_Id	VIS::num28_0	•	•		the associated tariff number
2	Rate_Cd	string<6>	•	•		the associated tariff rate code
3	Srv_Cd	string<4>	•	•		the associated tariff service
4	Chrg_Cd	string<4>	•	•		the associated tariff charge: condition or option
5	Frht_Cls_Cd	string<4>		•		the associated freight class

P	Element	Type	R	C	U	Description
6	Schg_Pct	VIS::num5_2		•	•	the surcharge percentage
7	Schg_Rate	VIS::num7_2		•	•	the surcharge rate

SchgSrvc_V1

A surcharge service for a tariff.

P	Element	Type	R	C	U	Description
1	Tff_Id	VIS::num28_0		•	•	the associated tariff number
2	Srvc_Cd	string<4>		•	•	the associated tariff service
3	Schg_Pct	VIS::num5_2		•	•	the surcharge percentage
4	Schg_Rate	VIS::num7_2		•	•	the surcharge rate

Rstc_V1

A tariff restriction. Restrictions are constraints that you can set for a tariff service or tariff service lane, for the shipments, shipment legs, or loads that are returned during routing and rating.

P	Element	Type	R	C	U	Description
1	Rstc_tlpt_Id	string<8>		•		the restriction template ID which provides default values before applying any changes
2	Rstc_Id	VIS::num28_0				the ID code of this restriction
3	Rstc_Desc	string<70>	•	•	•	a description of this restriction
4	Max_Pces	unsigned short	•	•	•	the maximum number of shipment pieces
5	Min_Pces	unsigned short	•	•	•	the minimum number of shipment pieces
6	Max_Comp_Wgt	VIS::num11_4	•	•	•	the maximum container weight
7	Max_Wgt	VIS::num11_4	•	•	•	the maximum shipment weight
8	Min_Wgt	VIS::num11_4	•	•	•	the minimum shipment weight
9	Max_Comp_Len	VIS::num6_3	•	•	•	the maximum container length
10	Max_Comp_Wdth	VIS::num6_3	•	•	•	the maximum container width
11	Max_Comp_Hght	VIS::num6_3	•	•	•	the maximum container height
12	Max_Vol	VIS::num7_4	•	•	•	the maximum shipment volume
13	Min_Vol	VIS::num11_4	•	•	•	the minimum shipment volume
14	Max_Grth_Size	VIS::num6_3	•	•	•	the maximum parcel size
15	Max_Stop	unsigned short	•	•	•	the maximum number of shipment stops

P	Element	Type	R	C	U	Description
16	Max_Skid	unsigned short	•	•	•	the maximum shipment skid
17	Min_Skid	unsigned short	•	•	•	the minimum shipment skid
18	Max_Shpm	unsigned short	•	•	•	the maximum number of shipments
19	Min_Shpm	unsigned short	•	•	•	the minimum number of shipments
20	Max_Load	unsigned short	•	•	•	the maximum number of shipment loads
21	Min_Load	unsigned short	•	•	•	the minimum number of shipment loads
22	Max_Shpm_Val_dlr	VIS::num15_2	•	•	•	the maximum shipment value
23	Min_Shpm_Val_dlr	VIS::num15_2	•	•	•	the minimum shipment value
24	Max_Otrt_Pct	VIS::num5_2	•	•	•	the maximum out-of-route percentage
25	Max_Otrt_Dist	VIS::num5_0	•	•	•	the maximum out-of-route distance
26	Max_Dhed_Dist	VIS::num5_2	•	•	•	the maximum deadhead distance
27	Max_Dhed_Pct	VIS::num3_2	•	•	•	the maximum deadhead percentage
28	Max_DhedLeg_Dist	VIS::num5_0	•	•	•	the maximum deadhead leg distance
29	Max_Wait_Stop_hrs	VIS::num6_2	•	•	•	the maximum waiting time at a stop
30	Max_Ret_To_Orig_Dist	VIS::num5_0	•	•	•	the maximum return to origin distance
31	Max_Wait_Btwn_Lds_hrs	VIS::num6_2	•	•	•	the maximum waiting in hours between loads
32	Max_Elpd_hrs	VIS::num6_2	•	•	•	the maximum elapsed time for the load
33	Max_Odr_Val_dlr	VIS::num15_2	•	•	•	the maximum shipment order value
34	Min_Odr_Val_dlr	VIS::num15_2	•	•	•	the minimum shipment order value
35	Max_Ldn_Len	VIS::num9_3	•	•	•	the maximum laden length
36	Min_Ldn_Len	VIS::num9_3	•	•	•	the minimum laden length
37	Max_Driving_hrs	VIS::num6_2	•	•	•	the maximum driving hours
38	Max_Tot_Wait_hrs	VIS::num6_2	•	•	•	the maximum total waiting hours
39	Max_On_Duty_hrs	VIS::num6_2	•	•	•	the total hours on duty
40	Max_Off_Duty_hrs	VIS::num6_2	•	•	•	the total hours off duty
41	Max>Loading_hrs	VIS::num6_2	•	•	•	the total loading hours
42	Max_Unloading_hrs	VIS::num6_2	•	•	•	the total unloading hours

Tariff_V1

A tariff. Values of some elements vary with the type of tariff as follows:

- carrier-related tariff - the value is the same as the carrier's
- customer tariff - the value is the same as the customer's

- referential tariff - the value is the same as the master tariff

Tariff Charges

During the creation of a referential tariff, charges are set to the same value as in the master tariff and cannot be modified using the Create tariff service. You can modify these default values using the Update tariff service after the tariff has been created.

However, you may have a Customer master tariff with conditions or options that have the “Charge based on carrier” field set to true. If so, then only the Responsible Customer fields can be updated for these conditions or options. Other conditions and options in this tariff in which this field is false can be updated.

P	Element	Type	R	C	U	Description
1	Tff_Id	VIS::num28_0				a system-generated tariff number
2	Tff_Cd	string<12>	•	•		an identifier for the tariff
3	Tff_Desc	string<70>	•	•	•	a description of the tariff
4	Tff_Grp_typ	VIS_V1::tTariffGroup	•	•	•	the tariff group
5	Tff_Attr_enu	VIS_V1::eTariffAttribute	•	•		the tariff attribute
6	Tff_Stat_enu	VIS_V1::eTariffStatus				the current status of the tariff
7	Cnse_Grp_typ	VIS_V1::tConsigneeGroup	•	•	•	the consignee group of the tariff
8	Prj_cd	string<12>		•	•	the project code of the tariff
9	Effective_dt	VIS::date	•	•	•	the effective date for the tariff the default value is today's date
10	Expired_dt	VIS::date	•	•	•	the expiry date for the tariff the default value is one year from today
11	Cncy_typ	VIS_V1::eCurrency	•	•		the rating currency code the value varies with the type of tariff
12	Pymnt_Cncy_typ	VIS_V1::eCurrency		•		the payment currency code
13	UMsr_Sys_enu	VIS_V1::eUnitMeasureSys	•	•		the unit of measure system of the tariff the value varies with the type of tariff
14	UMsr_Wgt_enu	VIS_V1::eUnit MeasureWgt	•	•		the weight units of the tariff the value varies with the type of tariff
15	UMsr_Len_enu	VIS_V1::eUnitMeasureLgt	•	•		the length units of the tariff the value varies with the type of tariff
16	UMsr_Dst_enu	VIS_V1::eUnit MeasureDst	•	•		the distance units for the tariff
17	Rout_Prcn_enu	VIS_V1::eRoutePrecision		•	•	the routing precision the value varies with the type of tariff

P	Element	Type	R	C	U	Description
18	Rstd_Use_yn	VIS::vbool		•	•	indicates whether this tariff is restricted from being used in rateshop the default value is false
19	Vlmt_yn	VIS::vbool		•	•	indicates whether this tariff is volumetric the default value is false
20	Sav_yn	VIS::vbool		•	•	indicates whether this tariff is savings-based the default value is false
21	RfrnLane_Exst_yn	VIS::vbool				indicates whether referential lanes exist
22	RfrnRate_Exst_yn	VIS::vbool				indicates whether referential rates exist
23	RfrnSchg_Exst_yn	VIS::vbool				indicates whether referential surcharges exist
24	FI_Pymt_Md_enu	VIS_V1::eFrhtIncPymtMode	•	•		the freight invoice payment method cannot be updated for generic tariffs the value varies with the type of tariff values are FIPM_NULL, FIPM_AUTO_PAY, and FIPM_AUDIT FIPM_NULL is the default value if the field is blank
25	Rate_Typ_enu	VIS_V1::eRateType	•	•	•	the rate type values are RteT_POINT_TO_POINT RteT_ROUND_TRIP RteT_NULL the last value is the default for a blank field
26	Mstr_Tff_id	VIS::num28_0		•		the master tariff number
27	Cust_Id	string<12>		•		the customer ID mandatory for a customer tariff or a customer specific carrier tariff
28	Carr_Id	string<8>		•		the carrier ID mandatory for a carrier-related tariff
29	CrtDUsr_Id	string<10>				the ID of the user who created the tariff
30	UpdtUsr_Id	string<10>				the ID of the user who updated the tariff
31	CrtD_dtt	VIS::timestamp				the date and time that the tariff was created
32	Updt_dtt	VIS::timestamp				the date and time that the tariff was updated
33	Mmo	Memo_V1		•	•	a memo associated with the tariff
	Collections					
34	IgnoreServices	VIS::vbool				indicates whether the contents of Services (next) should be ignored

P	Element	Type	R	C	U	Description
35	Services	TffSrvclst_V1		•	•	the services associated with the tariff
36	IgnoreRateCodes	VIS::vbool				indicates whether the contents of RateCodes (next) should be ignored
37	RateCodes	RateCdList_V1		•	•	the rate codes associated with the tariff
38	IgnoreLaneAssocs	VIS::vbool				indicates whether the contents of LaneAssocs (next) should be ignored
39	LaneAssocs	LaneAsscList_V1		•	•	the lanes associated with the tariff
40	IgnoreSchgSrvcs	VIS::vbool				indicates whether the contents of SchgSrvcs (next) should be ignored
41	SchgSrvcs	SchgSrvclst_V1		•	•	the surcharge services associated with the tariff
42	IgnoreSchgChrgs	VIS::vbool				indicates whether the contents of SchgChrgs (next) should be ignored
43	SchgChrgs	SchgChrgList_V1		•	•	the surcharges associated with the tariff
44	IgnoreSchgRates	VIS::vbool				indicates whether the contents of SchgRates (next) should be ignored
45	SchgRates	SchgRateList_V1		•	•	the surcharge rates associated with the tariff

TffChrg_V1

A tariff charge (condition or option).

P	Element	Type	R	C	U	Description
1	Tff_Id	VIS::num28_0	•	•		the associated tariff number
2	Srvc_Cd	string<4>	•	•		the associated tariff service
3	Chrg_Cd	string<4>	•	•		the ID code of this charge
4	Chrg_Desc	string<70>	•	•	•	a description of this charge
5	Chrg_Cond_yn	VIS::vbool	•	•		if true, then this charge is a condition, otherwise it is an option
6	Max_RUnt_Aply	VIS::num7_11	•	•	•	the maximum rate units
7	Min_RUnt_Aply	VIS::num7_11	•	•	•	the minimum rate units
8	Max_RUnt_Slc	VIS::num7_11	•	•	•	the maximum rate units Rate As
9	Min_RUnt_Slc	VIS::num7_11	•	•	•	the minimum rate units Rate As
10	Bs_Chrg_dlr	VIS::num15_2	•	•	•	the base charge
11	Dsc_Min_yn	VIS::vbool		•	•	the minimum discount charge
12	Min_Chrg_dlr	VIS::num15_2	•	•	•	the minimum charge amount

P	Element	Type	R	C	U	Description
13	Max_Chrg_dlr	VIS::num15_2	•	•	•	the maximum charge amount
14	GL_typ	VIS_V1::tGLType		•	•	the G/L type
15	OvrdAPGL_AcuExp_Acc	string<30>		•	•	the accrued expense account
16	OvrdAPGL_AcuLiab_Acc	string<30>		•	•	the accrued expense liability account
17	OvrdAPGL_Exp_Acc	string<30>		•	•	the expense account
18	OvrdAPGL_Liab_Acc	string<30>		•	•	the liability account
19	Pre_Pchd_yn	VIS::vbool		•	•	indicates whether charge is pre-purchased by customer
20	Tax_Chrg_yn	VIS::vbool		•	•	indicates whether charges from this tariff are taxable
21	Alert	VIS::vbool		•	•	indicates whether an alert is required when using this charge
22	Max_Min_Pce_yn	VIS::vbool		•	•	indicates whether the maximum or minimum is applicable to each piece of the shipment for this charge
23	Amt_dlr	VIS::num15_2		•	•	the discount percentage
24	Prtty	unsigned short	•	•	•	the priority for this charge
25	Dim_Wgt_Fctr	VIS::num11_6		•	•	the dimension weight factor
26	Min_Dim_wgt	VIS::num11_4		•	•	the dimension weight minimum
27	Eqiv_Ovsz_Wgt	VIS::num11_4		•	•	the equivalent oversize weight
28	Min_Ovsz_wgt	VIS::num11_4		•	•	the oversize minimum weight
29	Dim_Wgt_Mlt_yn	VIS::vbool		•	•	if true, the dimensional weight factor is a multiplication factor if false, the dimensional weight factor is a division factor
30	Spsd_Chrg_Id	string<4>		•	•	the supersede charge ID
31	Rng_Cd	string<4>		•	•	the range code
32	Crtd_dtt	VIS::timestamp				the date and time that this charge was created
33	Updt_dtt	VIS::timestamp				the date and time that this charge was updated
34	CrtdUsr_Id	string<10>				the ID of the user who created this charge
35	UpdtUsr_Id	string<10>				the ID of the user who updated this charge

P	Element	Type	R	C	U	Description
36	Mdy_Rate_yn	VIS::vbool		•	•	indicates whether this charge is mandatory during rating
37	Unit_Typ_enu	VIS_V1::eUnitType	•	•	•	the unit type
38	Lkup_Unit_enu	VIS_V1::eUnitType	•	•	•	the lookup unit type
39	Unit_Divd_Fctr	VIS::num7_3	•	•	•	the unit divide factor
40	Rndg_Rule_enu	VIS_V1::eRoundRule	•	•	•	the rounding rule
41	Chrg_By_Rng_yn	VIS::vbool		•	•	indicates whether this charge is calculated using ranges
42	Spsd_Rule_enu	VIS_V1::eSupeSedeRule	•	•	•	a supersede rule
43	Lgst_Free_yn	VIS::vbool		•	•	indicates whether the largest item is free when using this charge
44	Min_Lgst_Free	unsigned short	•	•	•	the minimum weight for the largest free item
45	NEft_Unt_Mul_enu	VIS_V1::eUnitType		•	•	the net effect calculation on the charge unit (multiply factor)
46	NEftUnt_Divd_enu	VIS_V1::eUnitType		•	•	the net effect calculation on the charge unit (divide factor)
47	NEft_Unt_Add_enu	VIS_V1::eUnitType		•	•	the net effect calculation on the charge unit (add factor)
48	NEft_Unt_Sub_enu	VIS_V1::eUnitType		•	•	the net effect calculation on the charge unit (subtract factor)
49	NEft_Amt_Mul_enu	VIS_V1::eUnitType		•	•	the net effect calculation on the charge amount (multiply factor)
50	NEftAmt_Divd_enu	VIS_V1::eUnitType		•	•	the net effect calculation on the charge amount (divide factor)
51	NEft_Amt_Add_enu	VIS_V1::eUnitType		•	•	the net effect calculation on the charge amount (add factor)
52	NEft_Amt_Sub_enu	VIS_V1::eUnitType		•	•	the net effect calculation on the charge amount (subtract factor)
53	Mlti_Frgt_Cls_yn	VIS::vbool				indicates whether this charge is multi-freight class
54	MltiComp_Lvl_enu	VIS_V1::eMulti CompLvl	•	•	•	the multi-container level
55	Lkup_UnitLvl_enu	VIS_V1::eLkupUnitLvl	•	•	•	the lookup unit level
56	Extl_Rate_Egin_Ver_Cd	string<15>	•	•	•	the external rating engine version code

P	Element	Type	R	C	U	Description
57	TffLkAhd_enu	VIS_V1::eTffLookAhead		•	•	the tariff rating look ahead value values are LA_AHEAD, LA_BACK, LA_DISABLED the default value is LA_AHEAD
58	Unit_Divd_Mlt_yn	VIS::vbool		•	•	indicates whether to multiply by the unit factor while rating shipments
59	Apply_Unit_Divd_Lk_up_yn	VIS::vbool		•	•	determines when the unit factor is applied against the rate if true, the unit factor is applied against the weight first, then the rate is applied if false, the rate is applied against the weight, then the unit factor is applied
60	Chrg_Bsd_Carr_yn	VIS::vbool	•	•	•	indicates whether the carrier is surcharge based
61	Rspb_Cust_enu	VIS_V1::eResponsibleCust	•	•	•	the customer responsible for paying the charges associated with this tariff charge
62	Rspb_Cust_cd	string<12>		•	•	the responsible customer code available only for customer tariffs
63	Payable_Carr_enu	VIS_V1::ePayableCarr	•	•	•	the carrier that will be paid for the charges based on this tariff charge values are PAY_CARR_TRNS_CARR and PAY_CARR_OTHER_CARR
64	Payable_Carr_cd	string<8>		•	•	the payable carrier code available for carrier and customer specified carrier tariffs
65	Trip_Lvl_Rtng_yn	VIS::vbool		•	•	if true, then trip rating applies to this tariff condition if false, then load rating applies to this tariff condition this field is always false for tariff options
	Collections					
66	IgnoreFCCross Rfrcs	VIS::vbool				indicates whether the contents of FCCrossRfrcs (next) should be ignored
67	FCCrossRfrcs	FC_Crs_Rfrc List_V1		•	•	the freight class cross- reference list associated with the tariff charge

TffSrvc_V1

A tariff service.

P	Element	Type	R	C	U	Description
1	Tff_Id	VIS::num28_0	•	•		the associated tariff number
2	Srvc_Cd	string<4>	•	•		the ID code of this service
3	Tff_Srvc_Desc	string<70>	•	•	•	a description of this service
4	Cust_Chrg_Cd	string<24>		•		the customer charge code
5	GL_typ	VIS_V1::tGL Type		•	•	the G/L type
6	OvrAPGL_Acu Exp_Acc	string<30>		•	•	the accrued expense account
7	OvrAPGL_Acu Liab_Acc	string<30>		•	•	the accrued expense liability account
8	OvrAPGL_Exp_Acc	string<30>		•	•	the expense account
9	OvrAPGL_Liab_Acc	string<30>		•	•	the liability account
10	Bs_Chrg_dlr	VIS::num15_2	•	•	•	the base charge
11	Min_Chrg_dlr	VIS::num15_2	•	•	•	the minimum charge
12	Max_Chrg_dlr	VIS::num15_2	•	•	•	the maximum charge
13	Tax_Chrg_yn	VIS::vbool	•	•	•	indicates whether this service is taxable
14	Rstd_Use_yn	VIS::vbool	•	•	•	indicates whether this service is restricted from being used in rateshop
15	CarrCslD_Ctl_enu	VIS_V1::eCarr ConsoFreq	•	•	•	the carrier consolidation frequency
16	CslD_PrcnLvl_enu	VIS_V1::e ManifConso Level	•	•	•	the carrier consolidation level
17	CustCslD_Ctl_enu	VIS_V1::eCustConsoFreq	•	•	•	the method of consolidating customer shipments
18	Prnt_Lbl_yn	VIS::vbool	•	•	•	indicates whether to print a label during confirmation when using this tariff service
19	Ship_Lbl_Fmt_typ	VIS_V1::tShipLabelFormat		•	•	a shipment label format
20	Prnt_BOL_yn	VIS::vbool		•	•	indicates whether to print a BOL during confirmation when using this tariff service
21	BOL_Num_Fmt	string<24>		•	•	the BOL format
22	BOL_Fmt_typ	VIS_V1::tBOLFormat		•	•	the BOL format type
23	BOL_Num_Ent_yn	VIS::vbool	•	•	•	indicates whether a BOL number entry is required
24	Mnft_Fmt_typ	VIS_V1::tCarrMnftFmt		•	•	the manifest format

P	Element	Type	R	C	U	Description
25	Invc_Fmt_typ	VIS_V1::tAPIInvoiceFormat		•	•	the invoice format
26	Alw_Stop_yn	VIS::vbool	•	•	•	indicates whether to allow stops
27	Team_Drvr_yn	VIS::vbool	•	•	•	indicates whether team drivers are needed for this service
28	Pre_PChd_yn	VIS::vbool	•	•	•	indicates whether the service is pre-purchased by the customer
29	Dsc_Min_yn	VIS::vbool	•	•	•	indicates whether the minimum charge can be discounted, if applicable, when using this tariff service
30	Alrt_Apt_yn	VIS::vbool	•	•	•	indicates whether the system generates an appointment alert for this service
31	Alrt_Carr_yn	VIS::vbool	•	•	•	indicates whether the system generates an carrier alert for this service
32	Assn_To_Ld_yn	VIS::vbool	•	•	•	indicates whether shipments using this service can be assigned to a load
33	Amt_dlr	VIS::num15_2		•	•	the discount percentage
34	Tnst_Md_enu	VIS_V1::eTransitMode	•	•	•	the transit mode
35	Auto_Acpt_Tdr_yn	VIS::vbool	•	•	•	indicates whether to automatically accept the tender when using this service
36	Lkup_UnitLvl_enu	VIS_V1::eLkupUnitLvl		•	•	the lookup unit level
37	Tdr_Rqrd_yn	VIS::vbool	•	•	•	indicates whether a tender is required when for this carrier and service
38	Trkg_Lvl_enu	VIS_V1::eTrkgLvl		•	•	the tracking level mandatory if the tariff is not a customer tariff
39	Comp_Trkg_yn	VIS::vbool	•	•	•	indicates whether container tracking is required for this service
40	Mnft_Ld_Grp_cd	string<4>	•	•	•	the manifest group ID required for carrier and customer specific carrier tariffs only
41	Num_of_Chrg	unsigned short				the number of associated charges for this service
42	Schd_Mthd_enu	VIS_V1::eSchdMethod		•	•	the scheduling method
43	Dlvy_Schd_Cd	string<15>		•	•	the delivery schedule code
44	Extl_DS_Egin_Ver	string<15>		•	•	the external delivery engine version code

P	Element	Type	R	C	U	Description
45	Ctrc_Carr_yn	VIS::vbool		•	•	indicates whether the carrier of this tariff is a contracted carrier
46	Crtddtt	VIS::timestamp				the date and time that this service was created
47	Updt_dtt	VIS::timestamp				the date and time that this service was updated
48	Crtddusr_id	string<10>				the ID of the user who created this service
49	Updtusr_id	string<10>				the ID of the user who updated this service
50	IgnoreRestriction	VIS::vbool				indicates whether to ignore the restrictions of this service
51	Restriction	Rstc_V1		•	•	the restrictions of this service
52	CEA_Cnstrts	CEAConstraints_V1		•	•	the equipment constraints for the carrier associated with this service
	Collections					
53	IgnoreCharges	VIS::vbool				indicates whether the contents of Charges (next) should be ignored
54	Charges	TffChgList_V1		•	•	the charges associated with this service
55	IgnoreEquipment	VIS::vbool				indicates whether the contents of Equipment (next) should be ignored
56	Equipment	TffSrvceqmtList_V1		•	•	the equipment associated with this service

TffSrvceqmt_V1

The equipment for the tariff service.

P	Element	Type	R	C	U	Description
1	TffSrvceqmt_id	VIS::num28_0				a system-generated tariff service equipment number
2	Eqmt_Typ_Cd	string<4>	•	•	•	the equipment type code
3	Tff_id	VIS::num28_0				the tariff ID number
4	Srvce_Cd	string<4>	•	•		the tariff service code
5	Prtty	unsigned short	•	•	•	the priority of the tariff service equipment
6	Op_Zn_Cd	string<8>		•	•	the operation zone code
7	Crtddtt	VIS::timestamp				the date and time that this tariff service equipment was created

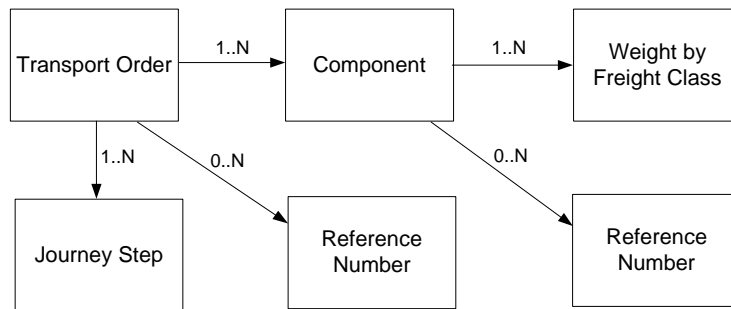
P	Element	Type	R	C	U	Description
8	Updt_dtt	VIS::timestamp				the date and time that this tariff service equipment was last updated
9	CrtdUsr_Id	string<10>				the ID of the user who created this tariff service equipment
10	UpdtUsr_Id	string<10>				the ID of the user who last updated this tariff service equipment
11	Rstc_Exst_yn	VIS::vbool				indicates whether there is a restriction associated with this tariff service equipment
12	Restriction	Rstc_V1	•	•	•	a restriction structure associated with this tariff service equipment

Transport Order Structures

A transport order is a request by a customer to a third party logistics provider (TPL) to move goods. A TPL is a separate company that arranges shipping services for customers using one or more shippers.

The following diagram shows the mandatory and optional information that you pass when creating a transport order using the API. The set of components of the transport order defines its contents. Each transport order has least one component.

The freight class entries define the weight of each component. For a single freight class, you only need to define the value for *FAK. Optionally, you can attach one or more reference numbers to each component. One or more itinerary steps define the origin, destination, and any intermediate itinerary steps for the transport order.



Specifying Origin, Destination and Throughpoints

For a transport order that originates from load-at (A) and has a destination of consignee (B), a single itinerary step is defined as:

Ship_Frm_enu	ShipFromLoc_id	Ship_To_enu	ShipToLoc_id
SHIPPOINTTYPE_LDAT	A	SHIPPOINTTYPE_CNSE	B

In the more complex case where a transport order is routed through a hub or DC, two itinerary steps are defined as:

Ship_Frm_enu	ShipFromLoc_id	Ship_To_enu	ShipToLoc_id
SHIPPOINTTYPE_LDAT	A	SHIPPOINTTYPE_HUB	C
SHIPPOINTTYPE_HUB	C	SHIPPOINTTYPE_CNSE	B

Normally, the origin and destination addresses for the transport order are set directly from the specified load-at and consignee entities. However, you can override the addresses of these entities (for example, if the address of the entity has changed, but the entity has not been updated.)

Another advantage of address overriding is that it can reduce or eliminate the need to define a load-at or consignee for every unique origin and destination. You can define and reuse one or more generic load-ats and consignees, simply by overriding their address in the transport order.

Consolidating Multiple Components

Transport orders are typically split into one or more shipments based on the following criteria:

- the shipping commodity must be unique for a shipment
- the requested pickup and delivery times must be identical for all components of a shipment
- the physical limitations associated with the origin and destination locations, with respect to the maximum weight or volume, are a factor; these limitations may mean that the transport order has to be divided into smaller shipments below the maximum amounts.
- the system parameters can dictate the maximum cube or weight of any shipment created.

Example 1: Transport Order A

Note the resulting shipments.

Item Type	Quantity	Component Type	Result
chair A	10	crate	S-1
chair B	10	crate	S-1
table X	5	skid	S-2

Consolidating Multiple Shipments

Shipments with the same Shipment Consolidation Class (SCC) identifier are combined during the load consolidation process (freight optimization). They will not be consolidated with other shipments that have a different SCC.

During load consolidation, Transportation Manager tries to form as many full truckloads as possible out of the shipments in the set. Any shipments in this set that cannot be put on a full truckload are shipped individually, as LTL (less than truckload). The table below shows an example load consolidation for a set of shipments.

Shipment ID	SCC	Load ID
S-001	B	L-1
S-002	B	L-1
S-003	B	L-1
S-004		L-2
S-005		L-3

During the creation of a transport order, the shipments do not yet exist. Therefore, you cannot directly set the value of the SCC for the resultant shipments. Instead, do this indirectly, by specifying the SCC at the transport order level. All shipments that are created from a transport order will have the same SCC. The following table shows this method.

Item Type	Quantity	Component Type	Resultant Shipment	Resultant Shipment SCC
chair A	10	crate	S-1	X
chair B	10	crate	S-1	X
table X	5	skid	S-2	X

In this table, transport order A has an SCC of 'X', so all shipments from this transport order will have this SCC value.

You can use the same SCC value in more than one transport order. Several shipments from different transport orders, but with the same SCC, are considered a group. Transportation Optimizer for Manager treats these shipments as a set and creates loads out of them.

The following transport order structures are listed alphabetically.

CntrOrtn_V1

The container orientation structure. This is associated with `Component_V1`.

P	Element	Type	R	C	U	Description
1	Cntr_Ortn_enu	VIS_V1::eCntrOrtn	•	•	•	the container orientation type

P	Element	Type	R	C	U	Description
2	Ortn_Alwd_enu	VIS_V1::eCntrOrtnAlwd	•	•	•	indicates whether the orientation is allowed, disallowed, or undefined
3	Prfn	unsigned short	•	•	•	the container orientation preference

Component_V1

A single physical entity within a shipment. A component has measurable attributes such as weight, length, width, height, and volume.

P	Element	Type	R	C	U	Description
1	Qty	VIS::num7_0	•	•	•	if <n> is greater than one, there are <n> identical components
2	Frm_Pkup_dtt	VIS::timestamp		•	•	the start of the date and time range in which the component should be picked up
3	To_Pkup_dtt	VIS::timestamp		•	•	the end of the date and time range in which the component should be picked up
4	Frm_Dlvy_dtt	VIS::timestamp		•	•	the start of the date and time range in which the component should be delivered
5	To_Dlvy_dtt	VIS::timestamp		•	•	the end of the date and time range in which the component should be delivered
6	Len	VIS::num9_3		•	•	the length of the component if NULL, then the default value is based on the component type specified by CompTyp_Id
7	Wdth	VIS::num9_3		•	•	the width of the component if NULL, then the default value is based on the component type specified by CompTyp_Id
8	Hgh	VIS::num9_3		•	•	the height of the component if NULL, then the default value is based on the component type specified by CompTyp_Id
9	Vol	VIS::num11_4		•	•	if the previous three fields are supplied and Vol is NULL, then the value is (Len x Wdth x Hgh) you can also specify a value
10	Nmnl_Wgt	VIS::num11_4		•	•	the nominal weight of the component
11	Odr_Val_dlr	VIS::num15_2		•	•	the order value of the component
12	DclD_Val_dlr	VIS::num15_2		•	•	the declared value of the component
13	CompTyp_Id	string<10>		•	•	the type of the component
14	Cdty_Id	string<12>		•	•	the commodity code for the component

P	Element	Type	R	C	U	Description
15	Sclد_Wgt	VIS::num11_4				the scaled weight of the component (the nominal weight + the tare weight)
16	Tot_Tare_Wgt	VIS::num17_4				the total tare weight of the component and any sub components
17	Tot_Pce	VIS::num7_0				the total number of pieces of the component
18	Tot_Skid	VIS::num7_0				the total number of skids of the component
19	Trkg_Qlfr_Id	VIS::num28_0				the tracking number qualifier ID
20	Trkg_Num	string<30>				the tracking number
21	Dlvy_Sort_Seq	unsigned short				the delivery sorting sequence always returned as zero
22	Comp_Desc	string<70>			• •	the component description
23	Stck_Hght	unsigned short			• •	the maximum number of containers of this type that can be stacked
24	Stck_Fctr	unsigned short			• •	the stacking factor: the amount of weight that can be stacked on top of this container
25	Stck_Grp	unsigned short			• •	the stacking group: indicates the relative durability of this container
26	Nest_Dimn_enu	VIS_V1::eCntrDimn			• •	the container dimension along which the nesting occurs if no value is assigned, then nesting is not allowed
27	Nest_Val	VIS::num9_3			• •	the nesting value: the increase along the nesting dimension with each additional nested unit if no value is assigned, then nesting is not allowed
28	MaxNestingSize	unsigned short			• •	the maximum number of allowed nested units if no value is assigned, then nesting is not allowed
	Collections					
29	IgnoreWgtByFCs	VIS::vbool				indicates whether the contents of WgtByFCs (next) should be ignored
30	WgtByFCs	WeightByFreightClassList_V1			• •	the weight by freight class of the component
31	IgnoreRfrNums	VIS::vbool				indicates whether the contents of RfrNums (next) should be ignored
32	RfrNums	CompRefNumberList_V1			• •	the reference numbers attached to the component

P	Element	Type	R	C	U	Description
33	IgnoreCntrOrtns	VIS::vbool				indicates whether contents of CntrOrtns (next) should be ignored
34	CntrOrtns	CntrOrtnList_V1		•	•	the container orientation objects attached to the component

CompRefNumber_V1

The reference number of the component.

P	Element	Type	R	C	U	Description
1	Item	Long	•	•	•	always returned as zero
2	Rfrnc_Num_Typ	VIS_V1::tReferenceNumType	•	•		a domain table value that indicates the reference number qualifier type for this reference number
3	Rfrnc_Num	string<30>	•	•	•	the value of the reference number or code

ItineraryStep_V1

A journey step.

P	Element	Type	R	C	U	Description
1	Ship_Frm_enu	VIS_V1::eShipPointType	•	•		the type of entity ShipFromLoc_id (next) refers to
2	ShipFromLoc_id	string<16>	•	•		identifies the shipping point
3	Ship_To_enu	VIS_V1::eShipPointType	•	•		the type of entity the ShipToLoc_id (next) refers to
4	ShipToLoc_id	string<16>	•	•		the shipping point ID
5	Carr_Id	string<8>		•	•	the carrier to use for the journey step this can be "" if you do not need a specific carrier
6	Carr_Cmtd_yn	VIS::vbool		•	•	indicates whether the carrier has been committed for the journey step
7	Srvc_Id	string<4>		•	•	the service to use for the journey step this can be "" if you do not need a specific service
8	Srvc_Cmtd_yn	VIS::vbool		•	•	indicates whether the service has been committed for the journey step

Shipment_V1

The physical collection of components that moves between any two given locations.
The system creates shipments from a single transport order.

P	Element	Type	Description
1	Shpm_Id	VIS::num7_0	an automatically generated shipment system identifier
2	Shpm_Num	string<12>	the shipment number
3	Sp_Csld_Cls	string<12>	the consolidation class
4	Drct_Frht_yn	VIS::vbool	indicates whether this shipment is to be shipped direct (no intermediate stops are made)
5	Frm_Pkup_dtt	VIS::timestamp	the start of the date and time range in which the shipment should be picked up
6	To_Pkup_dtt	VIS::timestamp	the end of the date and time range in which the shipment should be picked up
7	Frm_Dlvy_dtt	VIS::timestamp	the start of the date and time range in which the shipment should be delivered
8	To_Dlvy_dtt	VIS::timestamp	the end of the date and time range in which the shipment should be delivered
9	Prpd_Amt_dlr	VIS::num15_2	the prepaid amount
10	COD_To_Clct_dlr	VIS::num15_2	the COD amount
11	Tot_Mile_dist	VIS::num15_0	the total distance in miles
12	Cncy	VIS_V1::tCurrency	the currency type
13	Echg_Rate	VIS::num15_6	the exchange rate
14	UMsr_Sys_enu	VIS_V1::eUnit MeasureSys	the unit of measure enumerated identifier
15	UMsr_Wgt_enu	VIS_V1::eUnit MeasureWgt	the weight unit of measure identifier
16	UMsr_Len_enu	VIS_V1::eUnit MeasureLgt	the length unit of measure identifier
17	UMsr_Dst_enu	VIS_V1::eUnit MeasureDst	the distance units for the shipment
18	Sys_Calc_Amt_dlr	VIS::num15_2	the system-calculated amount
19	Dsct_Amt_dlr	VIS::num15_2	the discount amount
20	Chgd_Amt_dlr	VIS::num15_2	the charge amount
21	Fedl_Tax_Amt_dlr	VIS::num15_2	the federal tax amount
22	Sta_Tax_Amt_dlr	VIS::num15_2	the state or provincial tax amount
23	Loc_Tax_Amt_dlr	VIS::num15_2	the local tax amount
24	Adtn_Chrg_dlr	VIS::num15_2	the additional charge amount
25	PreCsld_Amt_dlr	VIS::num15_2	the pre-consolidation amount

P	Element	Type	Description
26	Sclد_Wgt	VIS::num11_4	the scaled weight
27	Vol	VIS::num11_4	the volume
28	Odr_Val_dlr	VIS::num15_2	the order value
29	Dclد_Val_dlr	VIS::num15_2	the declared value
30	Nmnl_Wgt	VIS::num11_4	the nominal weight
31	Tot_Tare_Wgt	VIS::num17_4	the total tare weight
32	Tot_Pce	VIS::num7_0	the total number of pieces
33	Tot_Skid	VIS::num7_0	the total number of skids
34	Urgt_yn	VIS::vbool	indicates whether this shipment is urgent
35	AR_Srvع_Cmtd_yn	VIS::vbool	indicates whether the A/R service has been committed
36	Ratg_Vlid_yn	VIS::vbool	indicates whether the rating is valid
37	Hold_yn	VIS::vbool	indicates whether the shipment is on hold
38	Cfmd_Tms	unsigned short	the number of times the shipment has been confirmed
39	CurOptStat	VIS_V1::eStatus	the current operational status
40	CurFnclStat	VIS_V1::eStatus	the current financial status
41	Crtد_dtt	VIS::timestamp	the date and time that the shipment was created
42	Updд_dtt	VIS::timestamp	the date and time that the shipment was updated
43	Ratд_dtt	VIS::timestamp	the rated date and time
44	Crtд_Usr_Id	string<10>	the ID of the user who created the shipment
45	Updд_Usr_Id	string<10>	the ID of the user who updated the shipment
46	Cfmg_Usr_Id	string<10>	the ID of the user who confirmed the shipment
47	Trpt_Odr_Id	string<4>	the transport order ID
48	Cdty_Id	string<12>	the commodity ID
49	AR_Srvع_Id	string<4>	the A/R service ID
50	Div_Id	string<4>	the division ID
51	Lgst_Grp_Id	string<4>	the logistics group ID
52	Rutд_Ori_Zn_cd	string<8>	the routing origin zone ID
53	Rutд_Dest_Zn_cd	string<8>	the routing destination zone ID
54	PickStop_Seq_num	unsigned short	the pick stop sequence number always returned as VIS::NULLushort

P	Element	Type	Description
55	DropStop_Seq_num	unsigned short	the drop stop sequence number always returned as VIS::NULLushort
56	DlvyCarr_Cd	string<8>	the delivery carrier code always returned as an empty string
57	DlvyCarrSCAC_typ	VIS_V1::tSCAC	the SCAC of the delivery carrier always returned as an empty string
58	DlvyCarrSrvC_Cd	string<4>	the service code of the delivery carrier always returned as an empty string
59	Ship_To_enu	VIS_V1::eShipToType	the Ship To location type always returned as VIS_V1::STT_NULL
60	ShipToLoc_cd	string<16>	the Ship To location ID always returned as an empty string
61	ShipToLoc_Name	string<70>	the Ship To location name always returned as an empty string
62	ShipToLoc_Addr	Address_V1	the Ship To location address
63	Mrge_Csld_Cls_Id	string<30>	the merge in transit consolidation class (MITCC) ID
64	Mrge_Csld_Seq_Num	long	the MITCC sequence number always returned as zero
	Collections		
65	IgnoreRfrCnums	VIS::vbool	indicates whether the contents of RfrCnums (next) should be ignored
66	RfrCnums	RefNumberList_V1	the reference numbers attached to the shipment
67	IgnoreComps	VIS::vbool	indicates whether the contents of Comps (next) should be ignored
68	Comps	ComponentList_V1	the components associated with the shipment specify at least one component
69	LoadIDs	VIS::StrIdList	the load IDs associated with the shipment these are not returned

ShipmentItem_V1

An object inside a container, which is in turn part of a shipment that is on a load.

P	Element	Type	R	C	U	Description
1	Shpm_Itm_Id	VIS::num28_0				the shipment item ID assigned by the system
2	Shpm_Itm_Num	string<30>		•	•	the shipment item number
3	Shpm_Itm_Xref	string<30>		•	•	a cross-reference for the shipment item
4	Itm_Desc	string<69>		•	•	a description of the item
5	Qnty	VIS::num7_2		•	•	the quantity
6	Nmnl_Wgt	VIS::num11_4		•	•	the nominal weight
7	Extd_Nmnl_Wgt	VIS::num11_4				the extended nominal weight
8	ProRatd_SclD_Wgt	VIS::num11_4				the prorated scaled weight
9	Odr_Val_dlr	VIS::num19_6		•	•	the order value
10	Extd_Odr_Val_dlr	VIS::num15_2				the extended order value
11	DclD_Val_dlr	VIS::num19_6		•	•	the declared value
12	Extd_DclD_Val_dlr	VIS::num15_2				the extended declared value
13	Inpt_Src_enu	VIS_V1::eTrnsSrc	•*	•		the enumerated input source
14	SerialLotCtrl_enu	VIS_V1::eSerialLotCtrl		•	•	the enumerated serial lot control number
15	Cur_OptlStat_id	VIS_V1::eStatus				the current operational status ID
16	Shpm_Id	VIS::num28_0				the shipment ID
17	Comp_Id	VIS::num28_0				the component ID
18	Itm_Num	string<30>	•	•	•	the item number
19	Frht_Cls_Cd	string<4>		•	•	the freight class code
20	NMFC_Cd	string<30>		•	•	the NMFC code
21	Hmn_Tff_Cd	string<30>		•	•	the harmonized tariff code
22	Orig_Ctry_Cd	string<3>		•	•	the origin country code
23	SLCtrlNums	string<2000>		•	•	the serial lot control number
24	IgnoreRfrCnums	VIS::vbool				indicates whether the contents of RfrCnums (next) should be ignored
25	RfrCnums	RefNumberList_V1		•	•	the reference numbers attached to the shipment item

* Mandatory only if the shipment item is created.

TransportOrder_V1

A transport order.

P	Element	Type	R	C	U	Description
1	Trpt_Odr_Id	string<12>	•	•		the ID code of the transport order
2	Urgt_yn	VIS::vbool		•		indicates whether the transport order is urgent
3	Cur_Loc_Typ_enu	VIS_V1::eCurrLocType				the current location type
4	Sp_Csld_Cls	string<12>		•	•	the shipment consolidation class for the shipments created from the transport order always returned as an empty string
5	LB_Plan_Id	VIS::num28_0		•		the plan ID always returned as an empty string
6	SpFm_Apt_Rqrd_yn	VIS::vbool		•		indicates whether a shipping point requires a carrier to make an appointment before picking up
7	SpTo_Apt_Rqrd_yn	VIS::vbool		•		indicates whether a shipping point requires a carrier to make an appointment before dropping off
8	Frht_Trm_enu	VIS_V1::eFhgtTerms		•		the freight terms for the transport order
9	COD_yn	VIS::vbool		•		indicates whether the transport order is cash on delivery (COD)
10	Drct_Frht_yn	VIS::vbool		•		indicates whether the transport order is to be shipped directly
11	Prj_Id	string<12>		•		the project to associate the transport order with
12	Div_Id	string<4>		•		the division ID of the transport order
13	LgstGrp_Id	string<4>		•		the logistics group of the transport order
14	Cust_Id	string<12>	•	•		the customer the transport order is for
15	TOEntVer_Id	string<10>		•		the order entry version of the transport order
16	TOEntType_Id	string<2>		•		the order entry type of the transport order
17	JrnyTplt_Id	string<8>		•		the journey template to use
18	SalesPerson_Id	string<10>		•	•	the salesperson responsible for the transport order
19	PreferedARSvc_Id	string<4>		•	•	the default service ID
20	FromAddr	Address_V1		•		the origin address -optional the default value is the address of the origin shipping point

P	Element	Type	R	C	U	Description
21	ToAddr	Address_V1		•		the destination address optional the default value is the address of the destination shipping point
22	Mmo	Memo_V1		•	•	a memo associated with the transport order
23	CurOptStat	VIS_V1::eStatus				the current operational status
24	Trpt_Odr_Md_enu	VIS_V1::eTransp OrderMode				the mode in which the transport order was created
25	Inpt_Src_enu	VIS_V1::eInput Source				the source from which the transport order was created
26	Prof_Ctr_typ	TProfitCenter		•	•	the profit center type
27	Cust_SrvRep_typ	TCustSrvRep		•	•	the customer service representative type
28	Cncy	VIS_V1::tCurrency				the currency used
29	Echg_Rate	VIS::num11_6				the exchange rate
30	UMsr_Sys_enu	VIS_V1::eUnit MeasureSys				the measurement system
31	UMsr_Wgt_enu	VIS_V1::eUnit MeasureWgt				the weight units
32	UMsr_Len_enu	VIS_V1::eUnit MeasureLgt				the length units
33	UMsr_Dst_enu	VIS_V1::eUnitMeasureDst				the distance units for the transport order
34	Sys_Calc_Amt_dlr	VIS::num15_2				the system-calculated charge amount
35	Dsct_Amt_dlr	VIS::num15_2				the discount amount
36	Chgd_Amt_dlr	VIS::num15_2				the total charged amount
37	Fedl_Tax_Amt_dlr	VIS::num15_2				the federal taxes amount
38	Sta_Tax_Amt_dlr	VIS::num15_2				the state or provincial tax amount
39	Loc_Tax_Amt_dlr	VIS::num15_2				the local or regional tax amount
40	Adtn_Chrg_dlr	VIS::num15_2				the additional charges
41	PreCslD_Amt_dlr	VIS::num15_2				the total charge amount before pre-consolidation
42	SclD_Wgt	VIS::num11_4				the scaled weight
43	Vol	VIS::num11_4				the total volume
44	Odr_Val_dlr	VIS::num15_2				the order value
45	DclD_Val_dlr	VIS::num15_2				the declared value
46	Nmnl_Wgt	VIS::num11_4				the nominal weight
47	Tot_Tare_Wgt	VIS::num17_4				the total tare weight

P	Element	Type	R	C	U	Description
48	Tot_Pce	VIS::num7_0				the total number of pieces
49	Tot_Skid	VIS::num7_0				the total number of skids
50	Crtd_dtt	VIS::timestamp				the date and time that the transport order was created
51	Updt_dtt	VIS::timestamp				the date and time that the transport order was updated
52	CrtdUsr_Id	string<10>				the ID of the user who created the transport order
53	UpdtUsr_Id	string<10>				the ID of the user who updated the transport order
54	ASN_Prted_Tms	unsigned short				the number of times that the advanced shipping notice (ASN) has been printed
	Collections					
55	IgnoreComps	VIS::vbool				indicates whether the contents of Comps (next) should be ignored
56	Comps	ComponentList_V1		•	•	components associated with the transport order specify at least one component
57	IgnoreItinerarySteps	VIS::vbool				indicates whether the contents of ItinerarySteps (next) should be ignored
58	ItinerarySteps	ItineraryStep_V1		•	•	the origin, destination and any intermediate hubs/DCs that the shipments for the transport order should travel through provide at least an origin and a destination
59	IgnoreRfrcNums	VIS::vbool				indicates whether the contents of RfrcNums (next) should be ignored
60	RfrcNums	RefNumberList_V1		•	•	any reference numbers attached to the transport order
	MITCC Attributes					
61	Mrge_Csld_Cls_Id	string<30>				the merge in transit consolidation class (MITCC) ID always returned as an empty string
62	Mrge_Csld_Seq_Num	long				the MITCC sequence number always returned as zero
63	ReqEquipmentType	string<5>		•	•	the required equipment type for continuous moves

WeightByFreightClass_V1

The weight by freight class.

P	Element	Type	R	C	U	Description
1	Nmnl_Wgt	VIS::num11_4	•	•	•	the nominal weight of the freight class
2	FC_Id	string<5>	•	•	•	the freight class

Chapter 5

Flat File Driver

This chapter describes the flat file driver extension of the Transportation Manager CORBA integration services API. It is divided into the following topics:

- [Overview](#)
- [System Interfaces](#)
- [Functions and Facilities](#)
- [Flat File Driver Daemon](#)

For description and examples of the API services for the flat file driver, refer to “[Flat File Driver API Services](#)” on page 229.

Overview

Using the flat file driver, the communications API is a flat file, or ASCII text file. This driver is a client of the API server, and you can use it as an alternative to providing your own CORBA client. You can do both inbound and outbound transactions using this API.

Terms and Definitions

API server A server that implements the CORBA API. You use this to direct data to and from the Transportation Manager database.

CORBA object A data object in the API server. The flat file driver requests operations that this object will do.

Delimiter The group of symbols which differentiate fields in the flat file API. You set this through a parameter switch. A space, then a double vertical bar, then another space, is the delimiter used in the example codes and definitions. This appears as " || ". The default delimiter for the driver application is two vertical bars.

Bold type indicates keywords in the file format definition boxes. These are read and used exactly as shown. *Italic* type indicates data values that you provide or define. It represents placeholders for the actual data.

System Interfaces

The flat file API is an executable program, `vffder.exe`, which reads and processes data files (flat files). The data input files are created by you in a readable, meta-format. For further information, see “Functions and Facilities” on page 226.

`vffder.exe` handles any number of input files. It uses the command line options listed in the next section. Some options are for debugging, while others control where data is coming from and going.

Flat File Driver Command Line Options

If the `-G` switch is present, then the flat file driver runs in GUI mode and the user can open and run flat files using the menus.

If the `-G` switch is not present, then at least one `-i` switch must be supplied to specify an input flat file for processing.

If the `-ior` switch is present, it takes precedence, and the IOR string in that file is used. Otherwise, if `-nsname`, `-nshost`, and `-nsport` are all present, then they are used to obtain the API server's IOR from the naming service. If neither of these is available, then the flat file driver looks for the IOR string file, `VentureFactory.ior`, in the current working directory.

Option: c

Supplied data value contents: True/False

Description

Indicates whether `vffder.exe` continues after a transaction raises an exception or has an error.

If True, `vffder.exe` continues after an exception or error. If False, `vffder.exe` terminates with an error after the first error or exception. The default value is True.

This condition does not control the inability to bind to a server object. If this happens, `vffder.exe` terminates, regardless of the option setting.

Option: C

Supplied data value contents: Username string and password string

Description

Allows `vffder.exe` to access a Transportation Manager database for transactions. The username string is the username in the database, and the password string validates the user's entry.

The database determines any restrictions on service availability. These restrictions are limited in the server, based on the username and password. The client side does not control or configure user privileges.

This option occurs after the host and bind options, if they are specified and not bound locally. This ensures the users is using the correct services.

You can omit this option from the command line, but you have to specify it in each data input file as a connect object.

Option: dcache

Supplied data value contents: True/False

Enables caching for distance calculation engines. It is identical to the `CACHE` registry key for the API server.

Option: def

Supplied data value contents: Default text string name

Description

The default text string for each structure. If you do not supply this option, the system searches for `defaults.txt` in the current directory.

Option: droute

Supplied data value contents: Route type string

The routing type for the distance calculation engines. It is identical to the `ROUTE` registry key for the API Server.

Option: e

Supplied data value contents: Error file name string

Description

The output file for errors. If you do not supply a value, errors are sent to `errorlog.txt` in the current directory.

Option: F

Supplied data value contents: none

Description

Enables the forward-declaring of objects and their use. If you do not supply this option, then you have to define objects before using them in a data file.

Data structures that contain aggregates have, by default, the aggregates defined before the use of the aggregate inside another data structure. Using this option removes this restriction.

Option: G

Supplied data value contents: none

Description

Enables a graphical interface for interactively diagnosing problems and running multiple flat files.

If you do not specify `G`, then you must specify `i`. That is, you must provide an input file to process on the command line if you are not running the GUI. You may still specify `i` even if you have specified `G`.

Option: i

Supplied data value contents: Input file name string

Description

The input file to open. This option can appear many times to process many input files. Each input file will retain variables locally during processing, starting from the first input file to the last on the command line.

See usage note in the description of `G`.

Option: ior

Supplied data value contents: IOR string filename

Description

A path to a file holding the API server's IOR string.

Option: m

Supplied data value contents: Comment symbol string

Description

The symbol for indicating comments in the input file. The default value is the pound sign (#). It is a sequence of characters and is in the first column of a row.

Option: nshost

Supplied data value contents: TCP/IP hostname for the VisiBroker naming service

Description

The TCP/IP hostname on which the naming service is running. This corresponds to the API server parameter set entry `NamingServiceHost`

Option: nsname

Supplied data value contents: API server for the VisiBroker naming service

Description

The identifier with which the API server will identify itself in the naming service. This corresponds to the server parameter set entry `IDInNamingService`

Option: nsport

Supplied data value contents: TCP/IP port for the VisiBroker naming service

Description

The TCP/IP port on which the naming service is running. This corresponds to the API server parameter set entry `NamingServicePort`.

Option: o

Supplied data value contents: Output file name string

Description

The output data file. If you do not supply a value, then the output data is sent to `output.txt` in the current directory.

Option: P

Supplied data value contents: none

Description

Indicates whether the defined data structures are purged between files. If True, the data read from an input file, and the data retrieved from a server, are both purged when another file is processed.

This value stops the data in more than one file from combining, allowing files to contain mnemonics which map to different data structures. The default setting is False.

Note: The server does not purge data from the database. Only data created from parsing an input file and data that was retrieved and sent to an output file is purged.

Option: t

Supplied data value contents: Token delimiter string

Description

The method in which the input files are parsed based on the supplied token delimiter string. It is a reference for splicing a line into fields. The default value is “ | | ”.

Using the Flat File driver

With the flat file driver running on the same machine as the API server, a typical shortcut command is:

```
C:\tm\Vffder.exe /G /C MyUsername MyPassword /ior
VentureFactory.ior
```

For a flat file driver running on a different machine than the API server, a typical shortcut command is:

```
c:\tm\Vffder.exe /G /C MyUsername MyPassword /ior \\hostname\
tm\VentureFactory.ior
```

where:

- `C:\tm\Vffder.exe` is the location of `vffder.exe`. The correct release of `vffder.exe` must be in the `C:\TM` folder on the local computer.
- `/G` allow you to work with the GUI version
- `MyUsername MyPassword` are the logon and password for the API user

- `/ior \\hostname\tm\VentureFactory.ior` identifies the machine and folder where the IOR is located

Functions and Facilities

You supply the input files containing data structures and service calls to `vffder.exe`. These structures and calls allow your system to access and populate a Transportation Manager system with data through the API server.

Common Services

Use the common services to specify locations for output and error data other than those in the default or command line directives. You can also specify common data structures and lists. You typically use the common data types as aggregate data types within an entity or transport order service type.

The following are the common services that support output file generation.

Note: [| | *Mnemonic*]* indicates one or more occurrences of *Mnemonic*.

```
OutputFile || Append/Create || Filename || Mnemonic || [ | |
Mnemonic]*
ErrorFile || Append/Create || Filename || Mnemonic
[ | | Mnemonic]*
```

Both the `OutputFile` and `ErrorFile` services allow you to create or append a file. Ensure that the field `Filename` is a full file path. The generated data can be one or more data structures that are represented by the mnemonic fields. These service lines are run immediately.

Flat File Structures

Not all structures are mandatory in the flat files. For these optional structures, you must still define a default object using the `Defaults.txt` file.

Omitted structures must contain at least the name of empty object that will be used. For example, you have the following structure:

```
Event_V1 || NULL_Event_V1
```

If you omitted the `Event_V1` structure in the flat file script, then the default `Event_V1` object with the name `NULL_EVENT_V1` will be created and passed to the API server.

Specify any default fields that are defined in the empty object in the `Defaults.txt`. For example, the following default values are defined in this `BusinessHours_V1` structure:

```
BusinessHours_V1|NULL_BusinessHours_V1|| |
|9:0:0|12:30:0|13:00:0|17:0:0|9:0:0|12:30:0|13:0:0|17:0
:0|9:0:0|12:30:0|13:0:0|17:0:0|9:0:0|12:30:0|13:0:0|17:
0:0|9:0:0|12:30:0|13:0:0|17:0:0|9:0:0|12:30:0|13:0:0|17
:0:0|9:0:0|12:30:0|13:0:0|17:0:0|bTRUE|
```

If you did not define the business hours in the input file, for example, in a shipping location, then a default set of business hours will be created with the values specified in the previous example.

Flat File Driver Daemon

The flat file driver daemon is a utility for the flat file driver. It starts the flat file driver in regular intervals and automatically detects the input file.

The general format is:

```
FFDdaemon.exe options
```

Daemon Example

```
FFDdaemon.exe /ior VenturFactory.ior -C*DFT/*DFT -w5 -iWork -a"Done" -k*.dat -e"c:\ffderror -oc:\ffdout -P -c"True"
```

Flat File Driver Daemon Command Line Options

Note: The following are the command line options for the flat file driver daemon. All options are optional except where indicated as mandatory.

Option string	Parameter	Description
a	directory name	the archive directory to which the input files are copied after the invocation default is <code>current directory\ARCHIVE</code>
c	true or false	the exception/error handling mode that the flat file driver uses
C	string/string	the username/password defined in Transportation Manager and that the flat file driver uses mandatory
def	filename	the name of the existing default data text file that the flat file driver uses
dt	date-time stamp	input files that have a date and time of their last modification greater than or equal to the specified date/time stamp are selected for processing format is <code>MMDDYY@HHMM</code>
e	directory name	the error directory in which the API flat file driver stores the error files default is <code>current directory\ERROR</code>
i	directory name	the source directory that the daemon searches for the input files for the API flat file driver default is <code>current directory</code>
ior	IOR string filename	a path to a file holding the API Server's IOR string
k	file selection mask	the file mask for the input file selection default is <code>*.*</code>

Option string	Parameter	Description
l	log filename	the file or full pathname for the run log default is <code>FFDdaemon.log</code>
m	string	the comment symbol string that the flat file driver uses
nshost	TCP/IP hostname for the VisiBroker naming service	the TCP/IP hostname on which the naming service is running corresponds to the API Server Parameter Set entry <code>NamingServiceHost</code>
nsname	API server for the VisiBroker naming service	the identifier with which the API Server will identify itself in the naming service corresponds to the API Server Parameter Set entry <code>IDInNamingService</code>
nsport	TCP/IP port for the VisiBroker naming service	the TCP/IP port on which the naming service is running corresponds to the API Server Parameter Set entry <code>NamingServicePort</code>
o	directory name	the output directory in which the API flat file driver stores output files default is <code>current directory\OUT</code>
P	none	indicates whether to purge defined data structures between files
RR	string	the rating routing server group that the flat file driver uses
t	string	the token delimiter string that the flat file driver uses
w	number	the interval in minutes in which the daemon checks on the input source directory default is 15 minutes

Daemon Run Log

A daemon run log has the following information when running the flat file driver.

```
>***** Start of API Flat FileDriver Invocation *****
>* Input Source files from directory : C:\Daemon\Work
>*           to archive directory : C:\Daemon\Done
>*           file1.dat
>*           file2.dat
>*           file3.dat
>*
>*Flat File Driver output to directory : C:\Daemon\Out
>*Output_06181998_121010.dat
>*
>*Flat File Driver error log to directory: C:\Daemon\Error
>*Error_06181998_121010.dat
>***** End of API Flat File Driver Invocation *****
```

Chapter 6

Flat File Driver API Services

This chapter describes the format of the API services for the flat file driver, and gives examples of each service. It includes the following topics:

- [Address Service](#)
- [Delivery Schedule Services](#)
- [Entity Services](#)
- [Financial Services](#)
- [Load Services](#)
- [Rate Quotation Services](#)
- [Shipment Services](#)
- [Shipment Order Entry Services](#)
- [Tariff Services](#)
- [Tariff Entity Services – Service Equipment](#)
- [Transport Order Services](#)

For complete descriptions of these services, refer to “API Services” on page 29.

Address Service

The only service for addresses is Address Validation.

Validate Example

```
Address_V1 || i2_AddressList_1 || 81 || WhiteHall Drive || || Markham ||  
ON || CAN || L3R 0P3  
AddressList_V1 || i2_AddressList || i2_AddressList_1  
AddrSrvc || Validate || i2_AddressList || STREET
```

Delivery Schedule Services

Use the delivery schedule services to create, retrieve, update, and delete delivery schedules, and to detach itineraries from timetables.

Create Delivery Schedule-Example 1

The following creates a delivery schedule containing the business days only, with no itineraries or timetables.

```
Memo_V1 || NULL_Memo_V1 || ""
BusinessDays_V1 || ALL_BDays_V1 || 7DAYW
DeliverySchedule_V1 || TS0908-1 || TS0908-1 || "Delivery Schedule with
Business days only" || DSLB_ZONE || DSDB_ELAPSED_DAYS || bFALSE || || ||
|| ALL_BDays_V1 || bTRUE || || bTRUE
DeliveryScheduleList_V1 || DSList || TS0908-1
DeliveryScheduleSrcv || CreateDeliverySchedule || DSList
```

Create Delivery Schedule-Example 2

The following creates a delivery schedule with two itineraries and one timetable. The timetable consists of two fixed point timetable entries and two Day of week point timetable entries.

```
Memo_V1 || NULL_Memo_V1 || ""
ItineraryPoint_V1 || PT511 || || Point 51 #1 || 1 || IPT_LA || TS0908-7
|| IT_TS_51 || TORONTO
ItineraryPoint_V1 || PT512 || || Point 51 #2 || 2 || IPT_DC || TS0908-7
|| IT_TS_51 || BARRIE
ItineraryPointList_V1 || Pnt51List || PT511 || PT512
Perf_V1 || PRFLP511 || 4 || || USE_USRDEFD_SERV_GRD_WHEN_INSUF_TRANS ||
3
Perf_V1 || PRFLP512 || 6 || || USE_USRDEFD_SERV_GRD_WHEN_INSUF_TRANS ||
2
ItnrLanePerformance_V1 || LP511 || 2143 || LTL || 1 || IPT_LA || TORONTO
|| 2 || IPT_DC || BARRIE || TS0908-7 || IT_TS_51 || PRFLP511
ItnrLanePerformance_V1 || LP512 || 2162 || LTL || 1 || IPT_LA || TORONTO
|| 2 || IPT_DC || BARRIE || TS0908-7 || IT_TS_51 || PRFLP512
ItnrLanePerformanceList_V1 || Lane51List || LP511 || LP512
Itinerary_V1 || ITTS51 || TS0908-7 || IT_TS_51 || Itinerary 1 for DS
TS0908-7 || ITNR_ACTIVE || || || || bFALSE || Pnt51List || bFALSE ||
Lane51List
ItineraryPoint_V1 || PT521 || || Point 52 #1 || 1 || IPT_LA || TS0908-7
|| IT_TS_52 || BRAMPTON
ItineraryPoint_V1 || PT522 || || Point 52 #2 || 2 || IPT_HUB || TS0908-7
|| IT_TS_52 || IYTORONTO
ItineraryPoint_V1 || PT523 || || Point 52 #3 || 3 || IPT_DC || TS0908-7
|| IT_TS_52 || TORONTO
ItineraryPointList_V1 || Pnt52List || PT521 || PT522 || PT523
Perf_V1 || PRFLP521 || 4 || || USE_USRDEFD_SERV_GRD_WHEN_INSUF_TRANS ||
3
Perf_V1 || PRFLP522 || 6 || || USE_USRDEFD_SERV_GRD_WHEN_INSUF_TRANS ||
2
ItnrLanePerformance_V1 || LP521 || 2143 || LTL || 1 || IPT_LA || BRAMPTON
|| 3 || IPT_DC || TORONTO || TS0908-7 || IT_TS_52 || PRFLP521
ItnrLanePerformance_V1 || LP522 || 2162 || LTL || 1 || IPT_LA || BRAMPTON
|| 3 || IPT_DC || TORONTO || TS0908-7 || IT_TS_52 || PRFLP522
ItnrLanePerformanceList_V1 || Lane52List || LP521 || LP522
Itinerary_V1 || ITTS52 || TS0908-7 || IT_TS_52 || Itinerary 2 for DS
TS0908-7 || ITNR_ACTIVE || || || TMTBL5_1 || || bFALSE || Pnt52List ||
bFALSE || Lane52List
```

```

ItineraryList_V1 || IT5List || ITTS51 || ITTS52
FixedPntTimeTable_V1 || Fixed51_1_1 || TS0908-7 || TMTBL5_1 ||
TmTblEnt51_1 || 1 || || || 10:10:00 || 10:30:00 || || 12/12/98
FixedPntTimeTable_V1 || Fixed51_1_2 || TS0908-7 || TMTBL5_1 ||
TmTblEnt51_1 || 2 || 15:10:00 || 15:30:00 || 16:10:00 || 16:30:00 || 12/
12/98 || 12/15/98
FixedPntTimeTable_V1 || Fixed51_1_3 || TS0908-7 || TMTBL5_1 ||
TmTblEnt51_1 || 3 || 15:10:00 || 15:30:00 || || || 12/16/98 ||
FixedPntTimeTableList_V1 || Fixed51_1 || Fixed51_1_1 || Fixed51_1_2 ||
Fixed51_1_3
ItnrTimeTableEntry_V1 || TmTblEnt51_1 || TS0908-7 || TMTBL5_1 ||
TmTblEnt51_1 || Entry #1 for timetable TMTBL5_1 || DSDB_FIXED_DATE ||
bFALSE || Fixed51_1 || bTRUE || || bTRUE
DayWeekPntTimeTable_V1 || Daywk51_2_1 || TS0908-7 || TMTBL5_1 ||
TmTblEnt51_2 || 1 || || || 10:10:00 || 10:30:00 || 0 || || DW_MON
DayWeekPntTimeTable_V1 || Daywk51_2_2 || TS0908-7 || TMTBL5_1 ||
TmTblEnt51_2 || 2 || 09:10:00 || 09:30:00 || 08:10:00 || 08:30:00 || 1 ||
DW_TUE || DW_WED
DayWeekPntTimeTable_V1 || Daywk51_2_3 || TS0908-7 || TMTBL5_1 ||
TmTblEnt51_2 || 3 || 09:10:00 || 09:30:00 || || || 1 || DW_WED
DayWeekPntTimeTableList_V1 || Daywk51_2 || Daywk51_2_1 || Daywk51_2_2 ||
Daywk51_2_3
ItnrTimeTableEntry_V1 || TmTblEnt51_2 || TS0908-7 || TMTBL5_1 ||
TmTblEnt51_2 || Entry #2 for timetable TMTBL5_1 || DSDB_DAY_OF_WK ||
bTRUE || || bFALSE || Daywk51_2 || bTRUE
ItnrTimeTableEntryList_V1 || TmTblEntList_51 || TmTblEnt51_1 ||
TmTblEnt51_2
ItnrTimeTable_V1 || TMTBL5_1 || TS0908-7 || TMTBL5_1 || TimeTable 1 for
TS0908-7 || 3 || || || bFALSE || TmTblEntList_51
ItnrTimeTableList_V1 || TmTblList || TMTBL5_1
FixedPntTimeTable_V1 || Fixed52_1_1 || TS0908-7 || TMTBL5_2 ||
TmTblEnt52_1 || 1 || || || 10:10:00 || 10:30:00 || || 12/12/98
FixedPntTimeTable_V1 || Fixed52_1_2 || TS0908-7 || TMTBL5_2 ||
TmTblEnt52_1 || 2 || 15:10:00 || 15:30:00 || 16:10:00 || 16:30:00 || 12/
12/98 || 12/15/98
FixedPntTimeTable_V1 || Fixed52_1_3 || TS0908-7 || TMTBL5_2 ||
TmTblEnt52_1 || 3 || 15:10:00 || 15:30:00 || || || 12/16/98 ||
FixedPntTimeTableList_V1 || Fixed52_1 || Fixed52_1_1 || Fixed52_1_2 ||
Fixed52_1_3
ItnrTimeTableEntry_V1 || TmTblEnt52_1 || TS0908-7 || TMTBL5_2 ||
TmTblEnt52_1 || Entry #1 for timetable TMTBL5_2 || DSDB_FIXED_DATE ||
bFALSE || Fixed52_1 || bTRUE || || bTRUE
ItnrTimeTableEntryList_V1 || TmTblEntList_52 || TmTblEnt52_1
ItnrTimeTable_V1 || TMTBL5_2 || TS0908-7 || TMTBL5_2 || TimeTable 2 for
TS0908-7 || 3 || || || bFALSE || TmTblEntList_52
ItnrTimeTableList_V1 || TmTblList || TMTBL5_1 || TMTBL5_2
DeliverySchedule_V1 || TSDS5 || TS0908-7 || Test Delivery Schedule ||
DSL_B_SHPG_LOC || DSDB_DAY_OF_WK || bFALSE || || || || bFALSE ||
IT5List || bFALSE || TmTblList
DeliveryScheduleList_V1 || DSLList || TSDS5
DeliveryScheduleSrvc || CreateDeliverySchedule || DSLList

```

Create Delivery Schedule Entity

The following example creates a shipment itinerary containing two points.

```
ItineraryPoint_V1 || PT541 || || Point 54 #1 || 1 || IPT_LA || DSAPI030-
13 || TOR-MIL || HAMILTON
ItineraryPoint_V1 || PT542 || || Point 54 #2 || 2 || IPT_DC || DSAPI030-
13 || TOR-MIL || MILTON
ItineraryPointList_V1 || Pnt54List || PT541 || PT542
Itinerary_V1 || ITMF54 || DSAPI030-13 || TOR-MIL || Itinerary 4 for DS
DSAPI030-13 || ITNR_ACTIVE || || || || bFALSE || Pnt54List || bTRUE
ItineraryList_V1 || IT5List || ITMF54
DeliveryScheduleSrcv || CreateDeliveryScheduleEntity || IT5List
```

Detach All Itineraries from Timetable Example

The following example detaches all itineraries from timetable TMTBLW_1 for TS_DS_WEEK.

```
ItnrTimeTable_V1 || TMTBLW_1 || TS_DS_WEEK || TMTBLW_1 || TimeTable 1 for
TS_DS_WEEK || 3 || || || bTRUE
ItnrTimeTableList_V1 || TmTblList || TMTBLW_1
DeliveryScheduleSrcv || DetachAllItineraries || TmTblList
```

Delete Delivery Schedule Example

```
StrIdList || DelList || TS0908-7
DeliveryScheduleSrcv || DeleteDeliverySchedule || DelList
```

Delete Delivery Schedule Entity Example

```
Itinerary_V1 || ITTS54 || TS0908-6 || IT_TS0908-61
ItineraryList_V1 || IT2List || ITTS54
DeliveryScheduleSrcv || DeleteDeliveryScheduleEntity || IT2List
```

Retrieve Delivery Schedule Entity Example

```
Perf_V1 || Perf1 || || || ||
ItnrLanePerformance_V1 || LP541 || 2143 || LTL || 1 || IPT_ZONE ||
WINNIPEG || 2 || IPT_ZONE || TIMMINS || TS0908-6 || IT_TS0908-62 || Perf1
ItnrLanePerformanceList_V1 || Lane54List || LP541
DeliveryScheduleSrcv || RetrieveDeliveryScheduleEntity || Lane54List
```

Retrieve Delivery Schedules Example

```
DeliveryScheduleList_V1 || aList
StrIdList || IdList1 || TS0908-1 || TS0908-2 || TS0908-3 || TS0908-4 ||
TS0908-5 || TS0908-6
DeliveryScheduleSrcv || RetrieveDeliverySchedule || IdList1 || TRUE ||
TRUE || aList
```

Update Delivery Schedule Example

The following example updates a delivery schedule with two itineraries and one timetable. The timetable consists of two fixed point timetable entries and two Day of week point timetable entries.


```

Memo_V1 || NULL_Memo_V1 || ""
ItineraryPoint_V1 || PT511 || || Point 51 #1 || 1 || IPT_LA || TS0908-7
|| IT_TS_51 || TRENTON
ItineraryPoint_V1 || PT512 || || Point 51 #2 || 2 || IPT_DC || TS0908-7
|| IT_TS_51 || ATLANTA
ItineraryPointList_V1 || Pnt51List || PT511 || PT512
Perf_V1 || PRFLP511 || 4 || || USE_USRDEFD_SERV_GRD_WHEN_INSUF_TRANS ||
3
Perf_V1 || PRFLP512 || 6 || || USE_USRDEFD_SERV_GRD_WHEN_INSUF_TRANS ||
2
ItnrLanePerformance_V1 || LP511 || 2143 || LTL || 1 || IPT_LA || TRENTON
|| 2 || IPT_DC || ATLANTA || TS0908-7 || IT_TS_51 || PRFLP511
ItnrLanePerformance_V1 || LP512 || 2162 || LTL || 1 || IPT_LA || TRENTON
|| 2 || IPT_DC || ATLANTA || TS0908-7 || IT_TS_51 || PRFLP512
ItnrLanePerformanceList_V1 || Lane51List || LP511 || LP512
Itinerary_V1 || ITTS51 || TS0908-7 || IT_TS_51 || Updated Itinerary 1 for
DS TS0908-7 || ITNR_INACTIVE || || || || bFALSE || Pnt51List || bFALSE
|| Lane51List
ItineraryPoint_V1 || PT521 || || Updated Point 52 #1 || 1 || IPT_LA ||
TS0908-7 || IT_TS_52 || BRAMPTON
ItineraryPoint_V1 || PT522 || || Updated Point 52 #2 || 2 || IPT_HUB ||
TS0908-7 || IT_TS_52 || IYTORONTO
ItineraryPoint_V1 || PT523 || || Updated Point 52 #3 || 3 || IPT_DC ||
TS0908-7 || IT_TS_52 || MIAMI
ItineraryPointList_V1 || Pnt52List || PT521 || PT522 || PT523
Perf_V1 || PRFLP521 || 4 || || USE_USRDEFD_SERV_GRD || 31
Perf_V1 || PRFLP522 || 6 || || USE_USRDEFD_SERV_GRD || 21
ItnrLanePerformance_V1 || LP521 || 2143 || LTL || 1 || IPT_LA || BRAMPTON
|| 3 || IPT_DC || TRENTON || TS0908-7 || IT_TS_52 || PRFLP521
ItnrLanePerformance_V1 || LP522 || 2162 || LTL || 1 || IPT_LA || BRAMPTON
|| 3 || IPT_DC || MIAMI || TS0908-7 || IT_TS_52 || PRFLP522
ItnrLanePerformanceList_V1 || Lane52List || LP521 || LP522
Itinerary_V1 || ITTS52 || TS0908-7 || IT_TS_52 || Updated Itinerary 2 for
DS TS0908-7 || ITNR_INACTIVE || || || TMTBL5_2 || || bFALSE || Pnt52List
|| bFALSE || Lane52List
ItineraryList_V1 || IT5List || ITTS51 || ITTS52
FixedPntTimeTable_V1 || Fixed52_1_1 || TS0908-7 || TMTBL5_2 ||
TmTblEnt52_1 || 1 || || || 20:10:00 || 20:30:00 || || 12/12/99
FixedPntTimeTable_V1 || Fixed52_1_2 || TS0908-7 || TMTBL5_2 ||
TmTblEnt52_1 || 2 || 22:10:00 || 22:30:00 || 08:10:00 || 08:30:00 || 12/
12/99 || 12/15/99
FixedPntTimeTable_V1 || Fixed52_1_3 || TS0908-7 || TMTBL5_2 ||
TmTblEnt52_1 || 3 || 18:10:00 || 18:30:00 || || || 12/16/99 ||
FixedPntTimeTableList_V1 || Fixed52_1 || Fixed52_1_1 || Fixed52_1_2 ||
Fixed52_1_3
ItnrTimeTableEntry_V1 || TmTblEnt52_1 || TS0908-7 || TMTBL5_2 ||
TmTblEnt52_1 || Entry #1 for timetable TMTBL5_2 || DSDB_FIXED_DATE ||
bFALSE || Fixed52_1 || bTRUE || || bTRUE
DayWeekPntTimeTable_V1 || DaywkW1_2_1 || TS0908-7 || TMTBL5_2 ||
TmTblEnt52_2 || 1 || || || 10:10:00 || 10:30:00 || 0 || || DW_MON
DayWeekPntTimeTable_V1 || DaywkW1_2_2 || TS0908-7 || TMTBL5_2 ||
TmTblEnt52_2 || 2 || 09:10:00 || 09:30:00 || 08:10:00 || 08:30:00 || 1 ||
DW_TUE || DW_WED

```

```

DayWeekPntTimeTable_V1 || DaywkW1_2_3 || TS0908-7 || TMTBL5_2 ||
TmTblEnt52_2 || 3 || 09:10:00 || 09:30:00 || || || 1 || DW_WED
DayWeekPntTimeTableList_V1 || DaywkW1_2 || DaywkW1_2_1 || DaywkW1_2_2 ||
DaywkW1_2_3
ItnrTimeTableEntry_V1 || TmTblEnt52_2 || TS0908-7 || TMTBL5_2 ||
TmTblEnt52_2 || Entry #2 for timetable TMTBL5_2 || DSDB_DAY_OF_WK ||
bTRUE || || bFALSE || DaywkW1_2 || bTRUE
ItnrTimeTableEntryList_V1 || TmTblEntList_52 || TmTblEnt52_1 ||
TmTblEnt52_2
ItnrTimeTable_V1 || TMTBL5_2 || TS0908-7 || TMTBL5_2 || TimeTable 2 for
TS0908-7 || 3 || || || bFALSE || TmTblEntList_52
ItnrTimeTableList_V1 || TmTblList || TMTBL5_2
DeliverySchedule_V1 || TSDS5 || TS0908-7 || Updated Test Delivery
Schedule TS0908-7 || DSLB_SHPG_LOC || || bFALSE || || || || bFALSE ||
IT5List || bFALSE || TmTblList
DeliveryScheduleList_V1 || DSLList || TSDS5
DeliveryScheduleSrcv || UpdateDeliverySchedule || DSLList

```

Update Delivery Schedule Entity Example

```

ItineraryPoint_V1 || PT541 || || Point 54 #1 || 1 || IPT_LA || DSAPI030-
13 || TOR-MIL || HAMILTON
ItineraryPoint_V1 || PT542 || || Point 54 #2 || 2 || IPT_DC || DSAPI030-
13 || TOR-MIL || MILTON
ItineraryPointList_V1 || Pnt54List || PT541 || PT542
Itinerary_V1 || ITMF54 || DSAPI030-13 || TOR-MIL || Itinerary 4 for DS
DSAPI030-13 || ITNR_INACTIVE || || || || bFALSE || Pnt54List || bTRUE
ItineraryList_V1 || IT5List || ITMF54
DeliveryScheduleSrcv || UpdateDeliveryScheduleEntity || IT5List

```

Update Itinerary Points and Performance Example

```

ItineraryPoint_V1 || PT511 || || Point 42 #1 || 1 || IPT_ZONE || TS0908-
6 || IT_TS0908-62 || WINNIPEG
ItineraryPoint_V1 || PT512 || || Point 42 #2 || 2 || IPT_ZONE || TS0908-
6 || IT_TS0908-62 || TIMMINS
ItineraryPoint_V1 || PT513 || || Point 42 #3 || 3 || IPT_ZONE || TS0908-
6 || IT_TS0908-62 || NIAGARA
ItineraryPointList_V1 || Pnt51List || PT511 || PT512 || PT513
Perf_V1 || PRFLP511 || 4 || || USE_USRDEFD_SERV_GRD || 103
Perf_V1 || PRFLP512 || 6 || || USE_USRDEFD_SERV_GRD || 102
Perf_V1 || PRFLP513 || 8 || || USE_USRDEFD_SERV_GRD || 104
ItnrLanePerformance_V1 || LP511 || 2143 || LTL || 1 || IPT_ZONE ||
WINNIPEG || 2 || IPT_ZONE || TIMMINS || TS0908-6 || IT_TS0908-62 ||
PRFLP511
ItnrLanePerformance_V1 || LP512 || 2143 || LTL || 2 || IPT_ZONE ||
TIMMINS || 3 || IPT_ZONE || NIAGARA || TS0908-6 || IT_TS0908-62 ||
PRFLP511
ItnrLanePerformance_V1 || LP513 || 2162 || LTL || 1 || IPT_ZONE ||
WINNIPEG || 2 || IPT_ZONE || TIMMINS || TS0908-6 || IT_TS0908-62 ||
PRFLP512
ItnrLanePerformance_V1 || LP514 || 2162 || LTL || 2 || IPT_ZONE ||
TIMMINS || 3 || IPT_ZONE || NIAGARA || TS0908-6 || IT_TS0908-62 ||
PRFLP512

```

```

ItnrLanePerformance_V1 || LP515 || 2120 || LTL || 1 || IPT_ZONE ||
WINNIPEG || 2 || IPT_ZONE || TIMMINS || TS0908-6 || IT_TS0908-62 ||
PRFLP511
ItnrLanePerformance_V1 || LP516 || 2120 || LTL || 2 || IPT_ZONE ||
TIMMINS || 3 || IPT_ZONE || NIAGARA || TS0908-6 || IT_TS0908-62 ||
PRFLP512
ItnrLanePerformanceList_V1 || Lane51List || LP511 || LP512 || LP513 ||
LP514 || LP515 || LP516
Itinerary_V1 || ITTS51 || TS0908-6 || IT_TS0908-62 || Updated Itinerary 2
for DS TS0908-6 || ITNR_INACTIVE || || || || || bFALSE || Pnt51List ||
bFALSE || Lane51List
ItineraryList_V1 || IT5List || ITTS51
DeliveryScheduleSrvc || UpdateDeliveryScheduleEntity || IT5List

```

Entity Services

Entities have a data structure format that is passed between your system and the server. Define the data structures in the input files as single line entries using the following format. For details of all API structures, see “API Structures” on page 89.

```

Data_Structure_Name || Mnemonic || value || value || value
...

```

`Data_Structure_Name` is the type of entity structure object you are creating, for example, `Carrier_V1` or `Customer_V1`.

`Mnemonic` is a tag which allows other lines to refer to this data structure object. This value must be unique throughout the file so that the definition of the data values is not lost.

The values in this file are specific to the data structure object you are defining. Specify all fields for each type of structure before the end of the line. Otherwise, an error will be sent to the error file or the error window.

To omit specifying a data field, leave it blank by placing two field separators back to back in a row. In the previous example, this is:

“ || || ”, which is a space, then a double vertical bar, then another space.

Data Structure Contents

A data structure can contain other data structures or lists. For ease of definition and to provide some flexibility, you can use references to undefined data structures or lists. Having undefined data structures or lists within the declaration of the data structure or list is known as forward-declaring a mnemonic. To do this, start `vffder.exe` with the `F` option. See “System Interfaces” on page 222.

You can define the undefined data structures after the parent data structure, but before using the parent data structure in a data transfer.

To do an entity transaction, package each defined data structure inside a data structure list. The following definition is for a data structure list.

```
Data_Structure_List || Mnemonic || value || value || value
...
```

`Data_Structure_List` is the type of entity sequence list you are creating, for example, `CarrierList_V1` or `CustomerList_V1`. `Mnemonic` is a tag for referring to this list in later transactions.

The `values` in this list are the mnemonics associated with the data structures. The mnemonics indicate the list you are creating, and you do not have to predefine them.

Define all the data elements of a data structure or list before using them in a data transfer operation. Otherwise, the attempted data transfer will fail. Also, if you try to insert a data structure that is inconsistent with its type, this will generate an error in the error file. For example, adding a `Hub_V1` object to a `LoadAtList_V1`.

Data Transfer

You use the entity services to create, retrieve, update, delete, and set the status for the supported entity types.

`EntitySrvc` means that you are defining an entity service transaction. The following field is the operation. The appropriate information is retrieved and bundled based on the specific operation. You use a list to transfer data in bulk between your system and the server.

The system checks the list type for the mnemonic in the given operation. If there is a mismatch, an error is sent to the error file and the operation stops.

All entity service transactions create a list of errors from the server side. The `Retrieve` service can also create a list of entity structures. These two lists are the only ones that you do not have to define before using.

Destination of Retrieved Data and Errors

Another feature unique to these two lists is the destination of the retrieved data and error list data. To control the destination of the retrieved data, the following rules apply for calls to the retrieve entity service.

In these rules, the data is the entity data and the transaction call is the entity transaction call. However, these rules also apply to other services.

- if you do not enter the return list in the call to retrieve the data, the return list is sent to the default output location; the default output is controlled by the command line parameter `/i`
- if you enter the mnemonic list in the call to retrieve the data, the returned data is captured and placed into this list

The same logic applies to error lists in all calls to entity transactions.

- if you do not enter the error list in the transaction call, any returned errors are sent to the default error location

- if you enter the error list in the transaction call, the errors are captured and placed into this list

This approach gives greater flexibility in implementing systems that use `vffder.exe`. For example, by not supplying an error list mnemonic in the calls to perform (in this case) entity transactions, all errors are consolidated inside either the file `errorlog.txt`, or the file named in the command line parameter `e`.

Alternatively, if you supply the error list mnemonic, the errors are captured for each transaction, and then sent to the error file.

Create

Define the top-level (or parent) data structure along with the necessary supporting aggregate data structures. The top-level data structures are any data structures that can contain an aggregate.

All of the entity structures are top-level and all of the common data structures are aggregate data structures. However, a few of the common data structures are also top-level because they contain aggregates. For example, the `BusinessHours_V1` structure can contain a `HolidayList_V1`.

The following are examples of the Create entity service operation for creating each of the possible Transportation Manager entity data elements.

Note: Some of examples throughout this guide are database dependent.

Create Carrier Example

```
Address_V1 || carr_list.[0].Addr || 15 || TANGREEN CRT || || NORTH YORK
|| ON || CAN || M2M3Z2 ||
ContactInfo_V1 || carr_list.[0].CntcInf || (416) 225-7013 [ ] || || || ||
Holiday_V1 || carr_list.[0].BusHours.Holidays.[0] || 07/01/1998 ||
00:00:00 || 23:59:00
Holiday_V1 || carr_list.[0].BusHours.Holidays.[1] || 08/04/1998 ||
00:00:00 || 23:59:00
Holiday_V1 || carr_list.[0].BusHours.Holidays.[2] || 09/01/1998 ||
00:00:00 || 23:59:00
Holiday_V1 || carr_list.[0].BusHours.Holidays.[3] || 10/13/1998 ||
00:00:00 || 23:59:00
Holiday_V1 || carr_list.[0].BusHours.Holidays.[4] || 12/25/1998 ||
00:00:00 || 23:59:00
Holiday_V1 || carr_list.[0].BusHours.Holidays.[5] || 12/26/1998 ||
00:00:00 || 23:59:00
Holiday_V1 || carr_list.[0].BusHours.Holidays.[6] || 12/31/1998 ||
00:00:00 || 23:59:00
HolidayList_V1 || carr_list.[0].BusHours.Holidays ||
carr_list.[0].BusHours.Holidays.[0] ||
carr_list.[0].BusHours.Holidays.[1] ||
carr_list.[0].BusHours.Holidays.[2] ||
carr_list.[0].BusHours.Holidays.[3] ||
carr_list.[0].BusHours.Holidays.[4] ||
carr_list.[0].BusHours.Holidays.[5] ||
carr_list.[0].BusHours.Holidays.[6]
```

```

BusinessHours_V1 || carr_list.[0].BusHours || 1 || 0 || 08:00:00 ||
23:59:00 || 00:00:00 || 00:00:00 || 08:00:00 || 23:59:00 || 00:00:00 ||
00:00:00 || 08:00:00 || 23:59:00 || 00:00:00 || 00:00:00 || 08:00:00 ||
23:59:00 || 00:00:00 || 00:00:00 || 08:00:00 || 23:59:00 || 00:00:00 ||
00:00:00 || 08:00:00 || 23:59:00 || 00:00:00 || 00:00:00 || 08:00:00 ||
23:59:00 || 00:00:00 || 00:00:00 || bFALSE ||
carr_list.[0].BusHours.Holidays
Memo_V1 || carr_list.[0].Mmo || ||
Memo_V1 || carr_list.[0].InsNote || ||
Contact_V1 || carr_list.[0].Conts.[0] || SALES REP || Irene Yeschin ||
ENG2 || (905) 944-8088 [8351 ] || (416) 225-7013 [ ] || (905) 944-0364 [
] || Irene.Yeschin@i2.com || www.i2.com
ContactList_V1 || carr_list.[0].Conts || carr_list.[0].Conts.[0]
CustCarrFrhtAudit_V1 || carr_list.[0].CustCarrFrhtAudits.[0] || FBAP_CU1
|| FBAP_CR5 || UNMATDFBAUTH_NEVER || 5 || 5 || 5 || 5
CustCarrFrhtAudit_V1 || carr_list.[0].CustCarrFrhtAudits.[1] || FBAP_CU0
|| FBAP_CR5 || UNMATDFBAUTH_NEVER || 5 || 5 || 5 || 5
CustCarrFrhtAudit_V1 || carr_list.[0].CustCarrFrhtAudits.[2] || FBAP_CU2
|| FBAP_CR5 || UNMATDFBAUTH_OTHER || 5 || 5 || 5 || 5
CustCarrFrhtAudit_V1 || carr_list.[0].CustCarrFrhtAudits.[3] || FBAP_CU9
|| FBAP_CR5 || UNMATDFBAUTH_NEVER || 5 || 5 || 5 || 5
CustCarrFrhtAuditList_V1 || carr_list.[0].CustCarrFrhtAudits ||
carr_list.[0].CustCarrFrhtAudits.[0] ||
carr_list.[0].CustCarrFrhtAudits.[1] ||
carr_list.[0].CustCarrFrhtAudits.[2] ||
carr_list.[0].CustCarrFrhtAudits.[3]
Carrier_V1 || carr_list.[0] || FBAP_CR5 || IY's Carrier || ENG2 || ROAD
|| *DFT || Contract ID || 06/20/1998 || C004 || Account ID || A/P Vendor
# || VGL_LD_LEG || FBGL_ONE_FB_PER_DAY || FIPM_AUTO_PAY || USD || bFALSE
|| 39 || bFALSE || bTRUE || bFALSE || bFALSE || bFALSE || bFALSE || 0 ||
06/20/2003 || 10000 || bFALSE || ROP_GEOGRAPHY || CarrBOL || || CarrMnfst
|| || 5 || 5 || 5 || 5 || 0 || || bFALSE || bFALSE || #####
|| bTRUE || STM_COMPARE_CARR_SPEC_TO_REFERENTIAL || bFALSE || 0 || 0 ||
SV_POD_OR_DELVY_NOTIFICATION_NOT_REQUIRED || 01 || || bFALSE || bFALSE
|| || || || || VGL_TRIP_LOAD || bFALSE || UMS_IMPERIAL || UMW_LB ||
UML_FT || UMD_MILES || bFALSE || || S_ACTIVE || 03/07/2000@15:35:15 || ||
VENTURE || || carr_list.[0].Addr || carr_list.[0].CntcInf ||
carr_list.[0].BusHours || carr_list.[0].Mmo || carr_list.[0].InsNote ||
bFALSE || carr_list.[0].Conts || bFALSE ||
carr_list.[0].CustCarrFrhtAudits
CarrierList_V1 || carr_list || carr_list.[0]
EntitySrvc || Create || carr_list

```

Create Component Type Group Example

```

UMsr_V1 || CompGroupList_Out.[0].UMsr || UMS_IMPERIAL || UMW_LB ||
UML_FT || UMD_MILES
Memo_V1 || Memo11 || Memo 11 A || Memo 11 B
ComponentType_V1 || CompGroupList_Out.[0].CompTypes.[0] || SKIDS ||
MF0913-01 || skids || UT_SKIDS || bTRUE || 0 || 9999999.9999 || 0 || 0 ||
0 || 0 || BOX || || 0 || 01/01/ 2000@00:00:00 || || VENTURE || || Memo11
Memo_V1 || Memo12 || Memo 12 A || Memo 12 B
ComponentType_V1 || CompGroupList_Out.[0].CompTypes.[1] || PIECES ||
MF0913-01 || Pieces || UT_PIECES || bTRUE || 0 || 9999999.9999 || 0 || 0

```

```

|| 0 || 0 || || || 0 || 01/01/ 2000@00:00:00 || || VENTURE || || ||
Memo12
ComponentTypeList_V1 || CompGroupList_Out.[0].CompTypes ||
CompGroupList_Out.[0].CompTypes.[0] ||
CompGroupList_Out.[0].CompTypes.[1]
Memo_V1 || Memo1 || Memo 1 A || Memo 1 B
ComponentTypeGroup_V1 || CompGroupList_Out.[0] || MF0913-01 || Default
|| CompGroupList_Out.[0].UMsr || 05/25/ 2000@11:31:14 || 06/08/
2000@15:21:16 || VENTURE || || Memo1 || bFALSE ||
CompGroupList_Out.[0].CompTypes
ComponentTypeGroupList_V1 || CompGroupList_Out || CompGroupList_Out.[0]
EntitySrvc || Create || CompGroupList_Out

```

Create Consignee Example

```

Address_V1 || CNSE_list.[0].Addr || 100 || ROWENA DR || 902 || NORTH YORK
|| ON || CAN || M3A1R1 ||
ContactInfo_V1 || CNSE_list.[0].CntcInf || 416-321-9782 || || || ||
Holiday_V1 || CNSE_list.[0].BusHours.Holidays.[0] || 07/01/1998 ||
00:00:00 || 23:59:00
Holiday_V1 || CNSE_list.[0].BusHours.Holidays.[1] || 08/04/1998 ||
00:00:00 || 23:59:00
Holiday_V1 || CNSE_list.[0].BusHours.Holidays.[2] || 09/01/1998 ||
00:00:00 || 23:59:00
Holiday_V1 || CNSE_list.[0].BusHours.Holidays.[3] || 10/13/1998 ||
00:00:00 || 23:59:00
Holiday_V1 || CNSE_list.[0].BusHours.Holidays.[4] || 12/25/1998 ||
00:00:00 || 23:59:00
Holiday_V1 || CNSE_list.[0].BusHours.Holidays.[5] || 12/26/1998 ||
00:00:00 || 23:59:00
Holiday_V1 || CNSE_list.[0].BusHours.Holidays.[6] || 09/09/1999 ||
00:00:00 || 23:59:00
Holiday_V1 || CNSE_list.[0].BusHours.Holidays.[7] || 12/31/1999 ||
00:00:00 || 23:59:00
Holiday_V1 || CNSE_list.[0].BusHours.Holidays.[8] || 01/01/2000 ||
00:00:00 || 23:59:00
Holiday_V1 || CNSE_list.[0].BusHours.Holidays.[9] || 02/29/2000 ||
00:00:00 || 23:59:00
Holiday_V1 || CNSE_list.[0].BusHours.Holidays.[10] || 12/31/2000 ||
00:00:00 || 23:59:00
Holiday_V1 || CNSE_list.[0].BusHours.Holidays.[11] || 01/01/2001 ||
00:00:00 || 23:59:00
HolidayList_V1 || CNSE_list.[0].BusHours.Holidays ||
CNSE_list.[0].BusHours.Holidays.[0] ||
CNSE_list.[0].BusHours.Holidays.[1] ||
CNSE_list.[0].BusHours.Holidays.[2] ||
CNSE_list.[0].BusHours.Holidays.[3] ||
CNSE_list.[0].BusHours.Holidays.[4] ||
CNSE_list.[0].BusHours.Holidays.[5] ||
CNSE_list.[0].BusHours.Holidays.[6] ||
CNSE_list.[0].BusHours.Holidays.[7] ||
CNSE_list.[0].BusHours.Holidays.[8] ||
CNSE_list.[0].BusHours.Holidays.[9] ||

```

```

CNSE_list.[0].BusHours.Holidays.[10] ||
CNSE_list.[0].BusHours.Holidays.[11]
BusinessHours_V1 || CNSE_list.[0].BusHours || 1 || 0 || 00:00:00 ||
23:59:00 || 00:00:00 || 00:00:00 || 00:00:00 || 23:59:00 || 00:00:00 ||
00:00:00 || 00:00:00 || 23:59:00 || 00:00:00 || 00:00:00 || 00:00:00 ||
23:59:00 || 00:00:00 || 00:00:00 || 00:00:00 || 23:59:00 || 00:00:00 ||
00:00:00 || 00:00:00 || 23:59:00 || 00:00:00 || 00:00:00 || 00:00:00 ||
23:59:00 || 00:00:00 || 00:00:00 || bFALSE ||
CNSE_list.[0].BusHours.Holidays
Memo_V1 || CNSE_list.[0].Mmo || ||
Memo_V1 || CNSE_list.[0].Directions || ||
ExternalAlias_V1 ||
CNSE_list.[0].CustShpgLocXRefs.[0].ExternalAliases.[0] || EMAIL || ||
bTRUE
ExternalAliasList_V1 ||
CNSE_list.[0].CustShpgLocXRefs.[0].ExternalAliases ||
CNSE_list.[0].CustShpgLocXRefs.[0].ExternalAliases.[0]
CustShpgLocXRef_V1 || CNSE_list.[0].CustShpgLocXRefs.[0] || FBAP_CU2 ||
FBAP_CN1 || SPT_CONSIGNEE || bFALSE ||
CNSE_list.[0].CustShpgLocXRefs.[0].ExternalAliases
CustShpgLocXRefList_V1 || CNSE_list.[0].CustShpgLocXRefs ||
CNSE_list.[0].CustShpgLocXRefs.[0]
Consignee_V1 || CNSE_list.[0] || FBAP_CN1 || SM Consignee - 1 || ENG2 ||
bFALSE || 9999999.9999 || 9999999.9999 || || || 0 || LUT_NULL || 0 ||
0 || 0 || 0 || LLT_LIVE || bFALSE || bFALSE || BDRYRULES_ON_TOP_ONE_MOVE
|| LUT_NULL || 0 || SH01 || SH01 || || || ONT || ONT || ONT || S_ACTIVE
|| 03/10/2000@19:58:33 || 03/11/2000@13:15:31 || VENTURE || VENTURE ||
DTTE_CONVERT_ALL_OVRDS || CNSE_list.[0].Addr || CNSE_list.[0].CntcInf ||
CNSE_list.[0].BusHours || CNSE_list.[0].Mmo || CNSE_list.[0].Directions
|| bTRUE || || bTRUE || || bFALSE || CNSE_list.[0].CustShpgLocXRefs
ConsigneeList_V1 || CNSE_list || CNSE_list.[0]
EntitySrvc || Create || CNSE_list

```

Create Customer Example

```

Address_V1 || cust_list.[0].Addr || 55 || AUDREY ST || || THUNDER BAY ||
ON || CAN || P7B5E4 ||
ContactInfo_V1 || cust_list.[0].CntcInf || 807-543-9357 || || || ||
Holiday_V1 || cust_list.[0].BusHours.Holidays.[0] || 07/01/1998 ||
00:00:00 || 23:59:00
Holiday_V1 || cust_list.[0].BusHours.Holidays.[1] || 08/04/1998 ||
00:00:00 || 23:59:00
Holiday_V1 || cust_list.[0].BusHours.Holidays.[2] || 09/01/1998 ||
00:00:00 || 23:59:00
Holiday_V1 || cust_list.[0].BusHours.Holidays.[3] || 10/13/1998 ||
00:00:00 || 23:59:00
Holiday_V1 || cust_list.[0].BusHours.Holidays.[4] || 12/25/1998 ||
00:00:00 || 23:59:00
Holiday_V1 || cust_list.[0].BusHours.Holidays.[5] || 12/26/1998 ||
00:00:00 || 23:59:00
HolidayList_V1 || cust_list.[0].BusHours.Holidays ||
cust_list.[0].BusHours.Holidays.[0] ||
cust_list.[0].BusHours.Holidays.[1] ||
cust_list.[0].BusHours.Holidays.[2] ||

```



```

cust_list.[0].BusHours.Holidays.[3] ||
cust_list.[0].BusHours.Holidays.[4] ||
cust_list.[0].BusHours.Holidays.[5]
BusinessHours_V1 || cust_list.[0].BusHours || 1 || 0 || 00:00:00 ||
23:59:00 || 00:00:00 || 00:00:00 || 00:00:00 || 23:59:00 || 00:00:00 ||
00:00:00 || 00:00:00 || 23:59:00 || 00:00:00 || 00:00:00 || 00:00:00 ||
23:59:00 || 00:00:00 || 00:00:00 || 00:00:00 || 23:59:00 || 00:00:00 ||
00:00:00 || 00:00:00 || 23:59:00 || 00:00:00 || 00:00:00 || 00:00:00 ||
23:59:00 || 00:00:00 || 00:00:00 || bFALSE ||
cust_list.[0].BusHours.Holidays
Memo_V1 || cust_list.[0].Mmo || ||
AutoAppliedOptionList_V1 || cust_list.[0].CompTyps.[0].AutoOpts
ComponentType_V1 || cust_list.[0].CompTyps.[0] || CASES || Cases ||
UT_SKIDS || bTRUE || 0 || 9999999.9999 || 2 || 2 || 2 || 8 || || 32767 ||
" 50 " || bTRUE || cust_list.[0].CompTyps.[0].AutoOpts
AutoAppliedOptionList_V1 || cust_list.[0].CompTyps.[1].AutoOpts
ComponentType_V1 || cust_list.[0].CompTyps.[1] || BOXES || Boxes ||
UT_PIECES || bTRUE || 0 || 9999999.9999 || 0 || 0 || 0 || 0 || || 0 || ||
bTRUE || cust_list.[0].CompTyps.[1].AutoOpts
ComponentTypeList_V1 || cust_list.[0].CompTyps ||
cust_list.[0].CompTyps.[0] || cust_list.[0].CompTyps.[1]
CustCarrFrhtAudit_V1 || cust_list.[0].CustCarrFrhtAudits.[0] || ||
FBAP_CR2 || UNMATDFBAUTH_NEVER || 5 || 5 || 5 || 5
CustCarrFrhtAuditList_V1 || cust_list.[0].CustCarrFrhtAudits ||
cust_list.[0].CustCarrFrhtAudits.[0]
ExternalAlias_V1 ||
cust_list.[0].CustShpgLocXRefs.[0].ExternalAliases.[0] || MICKY ||
Description || bTRUE
ExternalAlias_V1 ||
cust_list.[0].CustShpgLocXRefs.[0].ExternalAliases.[1] || SNOOPY ||
Description || bFALSE
ExternalAliasList_V1 ||
cust_list.[0].CustShpgLocXRefs.[0].ExternalAliases ||
cust_list.[0].CustShpgLocXRefs.[0].ExternalAliases.[0] ||
cust_list.[0].CustShpgLocXRefs.[0].ExternalAliases.[1]
CustShpgLocXRef_V1 || cust_list.[0].CustShpgLocXRefs.[0] || || FBAP_H1
|| SPT_HUB || bFALSE ||
cust_list.[0].CustShpgLocXRefs.[0].ExternalAliases
CustShpgLocXRef_V1 || cust_list.[0].CustShpgLocXRefs.[1] || || FBAP_LA1
|| SPT_LA || bTRUE ||
CustShpgLocXRefList_V1 || cust_list.[0].CustShpgLocXRefs ||
cust_list.[0].CustShpgLocXRefs.[0] || cust_list.[0].CustShpgLocXRefs.[1]
Customer_V1 || cust_list.[0] || FBAP_CU9 || Load Confirmation - Customer
1 || ENG2 || bFALSE || *DFT || || USD || UMS_IMPERIAL || UMW_LB || UML_FT
|| UMD_MILES || bTRUE || bTRUE || || bFALSE || bFALSE || 9999999999999.99
|| || || 39 || ROP_GEOGRAPHY || SV_POD_OR_DELVY_NOTIFICATION_REQUIRED ||
bFALSE || VC_NULL || 0 || 0 || bTRUE || TS_BEST || bTRUE || ||
TCRS_ALL_AR_TARIFFS || 0 || 99999999999999.99 || SFT_NULL || STT_NULL ||
bFALSE || 5 || 5 || 5 || 5 || IGL_ONE_INVC_SAME_DAY_VCHRS || || bFALSE
|| bFALSE || FIPM_AUTO_PAY || 0 || bFALSE || bFALSE || || bFALSE || ||
FT_CUSTOMER || || || || || LCOEVER || *DFT || " 50 " || || || LC01 ||
LC01 || S_ACTIVE || VENTURE || VENTURE || 03/11/2000@13:10:15 || 03/11/
2000@17:11:18 || cust_list.[0].Addr || cust_list.[0].CntcInf ||
cust_list.[0].BusHours || cust_list.[0].Mmo || bTRUE || || bTRUE || ||

```

```

bFALSE || cust_list.[0].CompTyps || bFALSE ||
cust_list.[0].CustCarrFrhtAudits || bFALSE ||
cust_list.[0].CustShpgLocXRefs
CustomerList_V1 || cust_list || cust_list.[0]
EntitySrvc || Create || cust_list

```

Create Distribution Center Example

```

Address_V1 || DC_list.[0].Addr || 515 || PORTAGE AVE || || WINNIPEG || MB
|| CAN || R3B2E9 ||
ContactInfo_V1 || DC_list.[0].CntcInf || 204-786-7811 || || || ||
Holiday_V1 || DC_list.[0].BusHours.Holidays.[0] || 07/01/1998 ||
00:00:00 || 23:59:00
Holiday_V1 || DC_list.[0].BusHours.Holidays.[1] || 08/04/1998 ||
00:00:00 || 23:59:00
Holiday_V1 || DC_list.[0].BusHours.Holidays.[2] || 09/01/1998 ||
00:00:00 || 23:59:00
Holiday_V1 || DC_list.[0].BusHours.Holidays.[3] || 10/13/1998 ||
00:00:00 || 23:59:00
Holiday_V1 || DC_list.[0].BusHours.Holidays.[4] || 12/25/1998 ||
00:00:00 || 23:59:00
Holiday_V1 || DC_list.[0].BusHours.Holidays.[5] || 12/26/1998 ||
00:00:00 || 23:59:00
HolidayList_V1 || DC_list.[0].BusHours.Holidays ||
DC_list.[0].BusHours.Holidays.[0] || DC_list.[0].BusHours.Holidays.[1]
|| DC_list.[0].BusHours.Holidays.[2] ||
DC_list.[0].BusHours.Holidays.[3] || DC_list.[0].BusHours.Holidays.[4]
|| DC_list.[0].BusHours.Holidays.[5]
BusinessHours_V1 || DC_list.[0].BusHours || 1 || -1 || 08:00:00 ||
17:00:00 || 00:00:00 || 00:00:00 || 08:00:00 || 17:00:00 || 00:00:00 ||
00:00:00 || 08:00:00 || 17:00:00 || 00:00:00 || 00:00:00 || 08:00:00 ||
17:00:00 || 00:00:00 || 00:00:00 || 08:00:00 || 17:00:00 || 00:00:00 ||
00:00:00 || 00:00:00 || 00:00:00 || 00:00:00 || 00:00:00 || 00:00:00 ||
00:00:00 || 00:00:00 || 00:00:00 || bFALSE ||
DC_list.[0].BusHours.Holidays
Memo_V1 || DC_list.[0].Mmo || ||
ExternalAlias_V1 || DC_list.[0].CustShpgLocXRefs.[0].ExternalAliases.[0]
|| DISNEY || Disney Land || bTRUE
ExternalAlias_V1 || DC_list.[0].CustShpgLocXRefs.[0].ExternalAliases.[1]
|| BOTTLE || Like a bottle || bFALSE
ExternalAliasList_V1 || DC_list.[0].CustShpgLocXRefs.[0].ExternalAliases
|| DC_list.[0].CustShpgLocXRefs.[0].ExternalAliases.[0] ||
DC_list.[0].CustShpgLocXRefs.[0].ExternalAliases.[1]
CustShpgLocXRef_V1 || DC_list.[0].CustShpgLocXRefs.[0] || FBAP_CU1 ||
FBAP_DC5 || SPT_DC || bFALSE ||
DC_list.[0].CustShpgLocXRefs.[0].ExternalAliases
CustShpgLocXRef_V1 || DC_list.[0].CustShpgLocXRefs.[1] || FBAP_CU7 ||
FBAP_DC5 || SPT_DC || bTRUE ||
CustShpgLocXRefList_V1 || DC_list.[0].CustShpgLocXRefs ||
DC_list.[0].CustShpgLocXRefs.[0] || DC_list.[0].CustShpgLocXRefs.[1]
DistributionCenter_V1 || DC_list.[0] || FBAP_DC5 || SH Winnipeg DC 1 ||
ENG2 || USD || UMS IMPERIAL || UMW_LB || UML_FT || bFALSE || 9999999.9999
|| 9999999.9999 || 0 || LUT_NULL || 0 || LUT_NULL || 0 || 0 || 0 || 0 ||
LLT_LIVE || bTRUE || bTRUE || BDRYRULES_STRT_FLR_TO_CLNG || || SH01 ||

```

```

SH01 || ALLCAN || ALLCAN || ALLCAN || S_ACTIVE || 03/10/2000@19:47:30 ||
03/12/2000@19:10:14 || VENTURE || VENTURE || DTTE_CONVERT_ALL_OVRDS ||
DC_list.[0].Addr || DC_list.[0].CntcInf || DC_list.[0].BusHours ||
DC_list.[0].Mmo || bTRUE || || bFALSE || DC_list.[0].CustShpgLocXRefs
DistributionCenterList_V1 || DC_list || DC_list.[0]
EntitySrvc || Create || DC_list

```

Create Equipment Type Example

```

EquipmentType_V1 || equip || 50FT || 50 FT || 40FTCONT || TB
EquipmentTypeList_V1 || equipList || equip
EntitySrvc || Create || equipList

```

Create Harmonized Tariff Example

```

Memo_V1 || HarmTariffList_Out.[0].Mmo || ||
HarmTariff_V1 || HarmTariffList_Out.[0] || MF0913-1 || MF0913-1
1234567890 1234567890 1234567890 1234567890 1234567890 1234567890
1234567890 1234567890 || bFALSE || HarmTariffList_Out.[0].Mmo || 06/09/
2000@15:35:12 || 06/13/2000@11:28:34 || VENTURE || VENTURE
HarmTariffList_V1 || HarmTariffList_Out || HarmTariffList_Out.[0]
EntitySrvc || Create || HarmTariffList_Out

```

Create Hub Example

```

Address_V1 || hub_list.[0].Addr || 19 || BALACLAVA ST || || KINGSTON ||
ON || CAN || K7K1J4 ||
ContactInfo_V1 || hub_list.[0].CntcInf || 613-872-8305 || || || ||
Holiday_V1 || hub_list.[0].BusHours.Holidays.[0] || 07/01/1998 ||
00:00:00 || 23:59:00
Holiday_V1 || hub_list.[0].BusHours.Holidays.[1] || 08/04/1998 ||
00:00:00 || 23:59:00
Holiday_V1 || hub_list.[0].BusHours.Holidays.[2] || 09/01/1998 ||
00:00:00 || 23:59:00
Holiday_V1 || hub_list.[0].BusHours.Holidays.[3] || 10/13/1998 ||
00:00:00 || 23:59:00
Holiday_V1 || hub_list.[0].BusHours.Holidays.[4] || 12/25/1998 ||
00:00:00 || 23:59:00
Holiday_V1 || hub_list.[0].BusHours.Holidays.[5] || 12/26/1998 ||
00:00:00 || 23:59:00
HolidayList_V1 || hub_list.[0].BusHours.Holidays ||
hub_list.[0].BusHours.Holidays.[0] || hub_list.[0].BusHours.Holidays.[1]
|| hub_list.[0].BusHours.Holidays.[2] ||
hub_list.[0].BusHours.Holidays.[3] || hub_list.[0].BusHours.Holidays.[4]
|| hub_list.[0].BusHours.Holidays.[5]
BusinessHours_V1 || hub_list.[0].BusHours || 1 || 0 || 00:00:00 ||
23:59:00 || 00:00:00 || 00:00:00 || 00:00:00 || 23:59:00 || 00:00:00 ||
00:00:00 || 00:00:00 || 23:59:00 || 00:00:00 || 00:00:00 || 00:00:00 ||
23:59:00 || 00:00:00 || 00:00:00 || 00:00:00 || 23:59:00 || 00:00:00 ||
00:00:00 || 00:00:00 || 23:59:00 || 00:00:00 || 00:00:00 || 00:00:00 ||
23:59:00 || 00:00:00 || 00:00:00 || bFALSE ||
hub_list.[0].BusHours.Holidays
Memo_V1 || hub_list.[0].Mmo || ||
ExternalAlias_V1 ||
hub_list.[0].CustShpgLocXRefs.[0].ExternalAliases.[0] || MICKY ||
Default external alias || bTRUE

```

```

ExternalAliasList_V1 ||
hub_list.[0].CustShpgLocXRefs.[0].ExternalAliases ||
hub_list.[0].CustShpgLocXRefs.[0].ExternalAliases.[0]
CustShpgLocXRef_V1 || hub_list.[0].CustShpgLocXRefs.[0] || FBAP_CU1 ||
|| || bFALSE || hub_list.[0].CustShpgLocXRefs.[0].ExternalAliases
CustShpgLocXRefList_V1 || hub_list.[0].CustShpgLocXRefs ||
hub_list.[0].CustShpgLocXRefs.[0]
Hub_V1 || hub_list.[0] || FBAP_H13 || SH Hub - Kingston 1 || ENG2 ||
bFALSE || 9999999.9999 || 9999999.9999 || D_ALLOWED || P_ALLOWED || bTRUE
|| 0 || 0 || 0 || bFALSE || bFALSE || 0 || LUT_NULL || 0 || LUT_NULL || 0
|| 0 || 0 || 0 || LLT_LIVE || bFALSE || bFALSE ||
BDRYRULES_ON_TOP_ONE_MOVE || bFALSE || || SH01 || SH01 || || SHCARR01 ||
LCKIN1 || ONT || || ONT || S_ACTIVE || 03/07/2000@15:38:22 || || VENTURE
|| || DTTE_CONVERT_ALL_OVRDS || hub_list.[0].Addr ||
hub_list.[0].CntcInf || hub_list.[0].BusHours || hub_list.[0].Mmo ||
bTRUE || || bTRUE || || bFALSE || hub_list.[0].CustShpgLocXRefs
HubList_V1 || hub_list || hub_list.[0]
EntitySrvc || Create || hub_list

```

Create Item Group List with Item Masters Example

```

UMsr_V1 || ItemGroupList_Out.[0].UMsr || UMS_IMPERIAL || UMW_LB ||
UML_FT || UMD_MILES
Memo_V1 || ItemGroupList_Out.[0].Memo || ||
Memo_V1 || ItemGroupList_Out.[0].ItmMstrs.[0].Mmo || Item 11A || Item
11B
ItemMaster_V1 || ItemGroupList_Out.[0].ItmMstrs.[0] || MF0705-03-01 ||
MF-TEST1 || Group 1 item 1 || || 10 || ITM_ACTIVE || SLC_NOT_REQRD || 0
|| || || || || " 50 " || ARE || GROC ||
ItemGroupList_Out.[0].ItmMstrs.[0].Mmo || ||
ItemMasterList_V1 || ItemGroupList_Out.[0].ItmMstrs ||
ItemGroupList_Out.[0].ItmMstrs.[0]
ItemGroup_V1 || ItemGroupList_Out.[0] || MF-TEST1 || required data for
Harmonized Tariff Testing || bFALSE || bFALSE ||
ItemGroupList_Out.[0].UMsr || 07/21/2000@13:37:52 || 07/21/2000@13:49:09
|| VENTURE || || || ItemGroupList_Out.[0].Memo || bFALSE ||
ItemGroupList_Out.[0].ItmMstrs
ItemGroupList_V1 || ItemGroupList_Out || ItemGroupList_Out.[0]
EntitySrvc || Create || ItemGroupList_Out

```

Create Load-at Example

```

Address_V1 || LA_list.[0].Addr || 420 || COUCHICHING POINT RD || ||
ORILLIA || ON || CAN || L3V6P6 ||
ContactInfo_V1 || LA_list.[0].CntcInf || 705-462-9431 || || || ||
Holiday_V1 || LA_list.[0].BusHours.Holidays.[0] || 07/01/1998 ||
00:00:00 || 23:59:00
Holiday_V1 || LA_list.[0].BusHours.Holidays.[1] || 08/04/1998 ||
00:00:00 || 23:59:00
Holiday_V1 || LA_list.[0].BusHours.Holidays.[2] || 09/01/1998 ||
00:00:00 || 23:59:00
Holiday_V1 || LA_list.[0].BusHours.Holidays.[3] || 10/13/1998 ||
00:00:00 || 23:59:00
Holiday_V1 || LA_list.[0].BusHours.Holidays.[4] || 12/25/1998 ||
00:00:00 || 23:59:00

```

```

Holiday_V1 || LA_list.[0].BusHours.Holidays.[5] || 12/26/1998 ||
00:00:00 || 23:59:00
Holiday_V1 || LA_list.[0].BusHours.Holidays.[6] || 09/09/1999 ||
00:00:00 || 23:59:00
Holiday_V1 || LA_list.[0].BusHours.Holidays.[7] || 12/31/1999 ||
00:00:00 || 23:59:00
Holiday_V1 || LA_list.[0].BusHours.Holidays.[8] || 01/01/2000 ||
00:00:00 || 23:59:00
Holiday_V1 || LA_list.[0].BusHours.Holidays.[9] || 02/29/2000 ||
00:00:00 || 23:59:00
Holiday_V1 || LA_list.[0].BusHours.Holidays.[10] || 12/31/2000 ||
00:00:00 || 23:59:00
Holiday_V1 || LA_list.[0].BusHours.Holidays.[11] || 01/01/2001 ||
00:00:00 || 23:59:00
HolidayList_V1 || LA_list.[0].BusHours.Holidays ||
LA_list.[0].BusHours.Holidays.[0] || LA_list.[0].BusHours.Holidays.[1]
|| LA_list.[0].BusHours.Holidays.[2] ||
LA_list.[0].BusHours.Holidays.[3] || LA_list.[0].BusHours.Holidays.[4]
|| LA_list.[0].BusHours.Holidays.[5] ||
LA_list.[0].BusHours.Holidays.[6] || LA_list.[0].BusHours.Holidays.[7]
|| LA_list.[0].BusHours.Holidays.[8] ||
LA_list.[0].BusHours.Holidays.[9] || LA_list.[0].BusHours.Holidays.[10]
|| LA_list.[0].BusHours.Holidays.[11]
BusinessHours_V1 || LA_list.[0].BusHours || 1 || 0 || 00:00:00 ||
23:59:00 || 00:00:00 || 00:00:00 || 00:00:00 || 23:59:00 || 00:00:00 ||
00:00:00 || 00:00:00 || 23:59:00 || 00:00:00 || 00:00:00 || 00:00:00 ||
23:59:00 || 00:00:00 || 00:00:00 || 00:00:00 || 23:59:00 || 00:00:00 ||
00:00:00 || 00:00:00 || 23:59:00 || 00:00:00 || 00:00:00 || 00:00:00 ||
23:59:00 || 00:00:00 || 00:00:00 || bFALSE ||
LA_list.[0].BusHours.Holidays
Memo_V1 || LA_list.[0].Mmo || ||
Memo_V1 || LA_list.[0].Directions || ||
CustShpgLocXRef_V1 || LA_list.[0].CustShpgLocXRefs.[0] || FBAP_CU9 ||
FBAP_LA5 || SPT_LA || bTRUE ||
ExternalAlias_V1 || LA_list.[0].CustShpgLocXRefs.[1].ExternalAliases.[0]
|| HOLLYWOOD || || bTRUE
ExternalAlias_V1 || LA_list.[0].CustShpgLocXRefs.[1].ExternalAliases.[1]
|| BAYVIEW || || bFALSE
ExternalAliasList_V1 || LA_list.[0].CustShpgLocXRefs.[1].ExternalAliases
|| LA_list.[0].CustShpgLocXRefs.[1].ExternalAliases.[0] ||
LA_list.[0].CustShpgLocXRefs.[1].ExternalAliases.[1]
CustShpgLocXRef_V1 || LA_list.[0].CustShpgLocXRefs.[1] || FBAP_CU7 ||
FBAP_LA5 || SPT_LA || bFALSE ||
LA_list.[0].CustShpgLocXRefs.[1].ExternalAliases
ExternalAlias_V1 || LA_list.[0].CustShpgLocXRefs.[2].ExternalAliases.[0]
|| Alias1 || description || bTRUE
ExternalAliasList_V1 || LA_list.[0].CustShpgLocXRefs.[2].ExternalAliases
|| LA_list.[0].CustShpgLocXRefs.[2].ExternalAliases.[0]
CustShpgLocXRef_V1 || LA_list.[0].CustShpgLocXRefs.[2] || FBAP_CU2 ||
FBAP_LA5 || SPT_LA || bFALSE ||
LA_list.[0].CustShpgLocXRefs.[2].ExternalAliases
CustShpgLocXRefList_V1 || LA_list.[0].CustShpgLocXRefs ||
LA_list.[0].CustShpgLocXRefs.[0] || LA_list.[0].CustShpgLocXRefs.[1] ||
LA_list.[0].CustShpgLocXRefs.[2]

```

```

LoadAt_V1 || LA_list.[0] || FBAP_LA5 || Shipment Manifest Load-at 1 ||
ENG2 || bFALSE || 9999999.9999 || 9999999.9999 || || || LUT_NULL || 0 ||
0 || 0 || 0 || 0 || LLT_LIVE || bFALSE || bFALSE ||
BDRYRULES_ON_TOP_ONE_MOVE || 0 || LUT_NULL || SH01 || SH01 || || || ONT
|| ONT || ONT || S_ACTIVE || 03/10/2000@19:55:22 || 03/12/2000@19:37:32
|| VENTURE || VENTURE || DTTE_CONVERT_ALL_OVRDS || LA_list.[0].Addr ||
LA_list.[0].CntcInf || LA_list.[0].BusHours || LA_list.[0].Mmo ||
LA_list.[0].Directions || bTRUE || || bTRUE || || bFALSE ||
LA_list.[0].CustShpgLocXRefs
LoadAtList_V1 || LA_list || LA_list.[0]
EntitySrvc || Create || LA_list

```

Create NMFC Example

```

Memo_V1 || NMFCList_Out.[0].Mmo || ||
NMFC_V1 || NMFCList_Out.[0] || MF0802-1 || test MF0724-1 || " 50 " ||
bFALSE || NMFCList_Out.[0].Mmo || 06/08/2000@18:59:30 || || VENTURE ||
Memo_V1 || NMFCList_Out.[1].Mmo || asdf || asdf
NMFC_V1 || NMFCList_Out.[1] || MF0724-2 || MF0724-2 || " 50 " || bFALSE
|| NMFCList_Out.[1].Mmo || 06/08/2000@19:00:26 || 06/08/2000@19:01:56 ||
VENTURE || VENTURE
Memo_V1 || NMFCList_Out.[2].Mmo || MF0705-3A || MF0705-3B
NMFC_V1 || NMFCList_Out.[2] || MF0724-3 || MF0724-3 || " 55 " || bFALSE
|| NMFCList_Out.[2].Mmo || 06/08/2000@19:01:40 || 06/09/2000@15:34:39 ||
VENTURE || VENTURE
NMFCList_V1 || NMFCList_Out || NMFCList_Out.[0]
# || NMFCList_Out.[1] || NMFCList_Out.[2]
EntitySrvc || Create || NMFCList_Out

```

Delete

Place the relevant entity IDs into a `StrIdList`. All the IDs provided represent the ID for the same entity type as specified in the delete operation. If the operation is successful, the error list will be empty. Ensure that all the entities exist or the operation will fail.

The following are examples of the delete entity service operation for deleting each of the possible Transportation Manager entity data elements.

Delete Carrier Example

```

StrIdList || IdList_1 || STEVE_1
EntitySrvc || Delete || CARRIER || IdList_1

```

Delete Component Type Group Example

```

StrIdList || MyList || MF0913-01
EntitySrvc || Delete || COMPONENT_TYPE_GROUP || MyList

```

Delete Consignee Example

```

StrIdList || IdList_1 || STEVE_1
EntitySrvc || Delete || CONSIGNEE || IdList_1

```

Delete Customer Example

```

StrIdList || IdList_1 || STEVE_1

```

```
EntitySrvc || Delete || CUSTOMER || IdList_1
```

Delete Distribution Center Example

```
StrIdList || IdList_1 || STEVE_1
EntitySrvc || Delete || DC || IdList_1
```

Delete Equipment Type Example

```
StrIdList || IdList || 50FT
EntitySrvc || Delete || EQUIPMENT_TYPE || IdList
```

Delete Harmonized Tariff Example

```
StrIdList || MyList || MF0913-1
EntitySrvc || Delete || HARM_TARIFF || MyList
```

Delete Item Group Example

```
StrIdList || MyList || MF-TEST1
EntitySrvc || Delete || ITEM_GROUP || MyList
```

Delete Hub Example Example

```
StrIdList || IdList_1 || STEVE_1
EntitySrvc || Delete || HUB || IdList_1
```

Delete Load-at Example

```
StrIdList || IdList_1 || STEVE_1
EntitySrvc || Delete || LOADAT || IdList_1
```

Delete NMFC Example

```
StrIdList || MyList || MF0802-1
EntitySrvc || Delete || NMFC || MyList
```

Remove

Remove Item Master Example

```
ItemMaster_V1 || ItemGroupList_Out.[0].ItmMstrs.[0] || MF1101-01 || MF-
TEST2 || || || || || || || || || || || || || || || || || || || ||
ItemMasterList_V1 || ItemGroupList_Out.[0].ItmMstrs ||
ItemGroupList_Out.[0].ItmMstrs.[0]
EntitySrvc || RemoveItems || ITEM_MASTER ||
ItemGroupList_Out.[0].ItmMstrs
```

Remove Component Type Example

```
ComponentType_V1 || CompTypes.[0] || ROLLS || MF-TST-6 || || || || || || ||
|| || || || || || || || || || || || || || || || || || || || || || || ||
ComponentTypeList_V1 || CompTypes || CompTypes.[0]
EntitySrvc || RemoveItems || COMPONENT_TYPE || CompTypes
```

Retrieve

Place the relevant entity IDs into a `StrIdList`. All the IDs provided represent the ID for the same entity type. This is because the returned list can only contain data structures indicative of the same entity type.

Ensure the returned list is previously defined but empty. If the operation is successful, the error list will be empty and the returned entity list will contain at least one item. The items will be packaged in the same format as when they were created.

The following are examples of the Retrieve entity service operation for retrieving each of the possible Transportation Manager entity data elements.

Retrieve Carrier Example

```
CarrierList_V1 || CarrierList_Out  
StrIdList || CarrList_1 || STEVE_1  
EntitySrvc || Retrieve || IdList_1 || CarrierList_Out
```

Retrieve Component Type Group

```
ComponentTypeGroupList_V1 || CompGroupList_Out  
StrIdList || IdList_1 || MF0913-01  
EntitySrvc || Retrieve || IdList_1 || CompGroupList_Out
```

Retrieve Consignee Example

```
ConsigneeList_V1 || ConsList_Out  
StrIdList || IdList_1 || STEVE_1  
EntitySrvc || Retrieve || IdList_1 || ConsList_Out
```

Retrieve Customer Example

```
CustomerList_V1 || CustList_Out  
StrIdList || IdList_1 || STEVE_1  
EntitySrvc || Retrieve || IdList_1 || CustList_Out
```

Retrieve Distribution Center Example

```
DistributionCenterList_V1 || DCList_Out  
StrIdList || IdList_1 || STEVE_1  
EntitySrvc || Retrieve || IdList_1 || DCList_Out
```

Retrieve Equipment Type Example

```
StrIdList || IdList || 50FT  
EquipmentTypeList_V1 || equipList  
EntitySrvc || Retrieve || IdList || equipList
```

Retrieve Harmonized Tariff Example

```
HarmTariffList_V1 || HarmTariffList_Out  
StrIdList || IdList_1 || MF0913-1  
EntitySrvc || Retrieve || IdList_1 || HarmTariffList_Out
```

Retrieve Hub Example

```
HubList_V1 || HubList_Out  
StrIdList || IdList_1 || STEVE_1  
EntitySrvc || Retrieve || IdList_1 || HubList_Out
```

Retrieve Item Group List Example

```
ItemGroupList_V1 || ItemGroupList_Out  
StrIdList || IdList_1 || MF-TEST1  
EntitySrvc || Retrieve || IdList_1 || ItemGroupList_Out
```


Retrieve Load-at Example

```
LoadAtList_V1 || LAList_Out
StrIdList || IdList_1 || STEVE_1
EntitySrvc || Retrieve || IdList_1 || LAList_Out
```

Retrieve NMFC Example

```
NMFCList_V1 || NMFCList_Out
StrIdList || IdList_1 || MF0802-1
EntitySrvc || Retrieve || IdList_1 || NMFCList_Out
```

Set Status

Place the relevant entity IDs into a `StrIdList`. All the IDs provided represent the ID for the same entity type as specified in the `SetStatusV1` operation. If the operation is successful, the error list will be empty. Ensure that all the entities exist or the operation will fail.

The following are examples of the `SetStatusV1` entity service operation for setting the status for each of the possible Transportation Manager entity data elements.

SetStatusV1 Carrier Example

```
StrIdList || IdList_1 || STEVE_1
EntitySrvc || SetStatusV1 || CARRIER || IdList_1 || S_INACTIVE
```

SetStatusV1 Consignee Example

```
StrIdList || IdList_1 || STEVE_1
EntitySrvc || SetStatusV1 || CONSIGNEE || IdList_1 || S_INACTIVE
```

SetStatusV1 Customer Example

```
StrIdList || IdList_1 || STEVE_1
EntitySrvc || SetStatusV1 || CUSTOMER || IdList_1 || S_INACTIVE
```

SetStatusV1 Distribution Center Example

```
StrIdList || IdList_1 || STEVE_1
EntitySrvc || SetStatusV1 || DC || IdList_1 || S_INACTIVE
```

SetStatusV1 Hub Example

```
StrIdList || IdList_1 || STEVE_1
EntitySrvc || SetStatusV1 || HUB || IdList_1 || S_INACTIVE
```

SetStatusV1 Load-at Example

```
StrIdList || IdList_1 || STEVE_1
EntitySrvc || SetStatusV1 || LOADAT || IdList_1 || S_INACTIVE
```

Update

Define the top-level data structure along with the supporting aggregate data structures. For additional details, see “Create” on page 237.

All entities in the data structure list must be in the Transportation Manager database before updating, otherwise the updates will fail.

In the examples given, only one field is updated. The remaining fields are blank.

Update Carrier Example

```
Carrier_V1 || Carr.[0] || CARDUF5 || DUFFY TRANSPORT INC. || FRE || ROAD
|| *DFT || || C001 || || VGL_LD_LEG || FBGL_ONE_FB_FOR_ALL ||
FIPM_AUTO_PAY || USD || bFALSE || 39 || bFALSE || bTRUE || bFALSE ||
bTRUE || bFALSE || bFALSE || 0 || 12/12/2000 || 100000000 || bFALSE ||
ROP_GEOGRAPHY || || CarrMnfst || 5 || 5 || 5 || 5 || 0 || bFALSE
|| bFALSE || ##### || bTRUE ||
STM_COMPARE_CARR_SPEC_TO_REFERENTIAL || bFALSE || 0 || 0 ||
SV_POD_OR_DELVY_NOTIFICATION_REQUIRED || 0 || bNULL || UMS_IMPERIAL
|| UMW_LB || UML_FT || bFALSE || S_ACTIVE || 05/21/1999@16:13:07 ||
JO-ANNE || Carr.[0].Addr || Carr.[0].CntcInf || Carr.[0].BusHours ||
Carr.[0].Mmo || Carr.[0].InsNote || bTRUE || Carr.[0].Conts || bTRUE ||
CarrierList_V1 || Carr || Carr.[0]
EntitySrvc || Update || Carr
```

Update Component Type Group Example

```
ComponentType_V1 || CompGroupList_Out.[0].CompTypes.[0] || SKIDS ||
MF0913-01 || Updated skids || || || || || || || || || || || || || || ||
|| || || bTRUE
ComponentType_V1 || CompGroupList_Out.[0].CompTypes.[1] || PIECES ||
MF0913-01 || Updated Pieces || || || || || || || || || || || || || || ||
|| || || bTRUE
ComponentTypeList_V1 || CompGroupList_Out.[0].CompTypes ||
CompGroupList_Out.[0].CompTypes.[0] ||
CompGroupList_Out.[0].CompTypes.[1]
ComponentTypeGroup_V1 || CompGroupList_Out.[0] || MF0913-01 || Updated
Default || || || || || || bFALSE || CompGroupList_Out.[0].CompTypes
ComponentTypeGroupList_V1 || CompGroupList_Out || CompGroupList_Out.[0]
EntitySrvc || Update || CompGroupList_Out
```

Update Consignee Example

```
Consignee_V1 || Cons.[0] || CONJUNE || JUNE WAREHOUSE || ENG1 || bFALSE
|| 9999999.9999 || 9999999.9999 || || || || 0 || LUT_NULL || 0 || 0 || 0
|| LUT_NULL || 0 || CST1 || CST1 || || || ONT || ONT || ONT || S_ACTIVE
|| 05/22/1999@16:10:44 || JO-ANNE || Cons.[0].Addr ||
Cons.[0].CntcInf || Cons.[0].BusHours || Cons.[0].Mmo ||
Cons.[0].Directions || bTRUE || Cons.[0].Conts || bTRUE || || bTRUE ||
ConsigneeList_V1 || Cons || Cons.[0]
EntitySrvc || Update || Cons
```

Update Customer Example

```
Customer_V1 || Custs.[0] || HAYES || HAYES BOOKS || FRE || bFALSE || P001
|| || USD || UMS_IMPERIAL || UMW_LB || UML_FT || bTRUE || bTRUE || ||
bFALSE || bFALSE || 99999999999999.99 || || 39 || ROP_GEOGRAPHY ||
SV_POD_OR_DELVY_NOTIFICATION_REQUIRED || bFALSE || 0 || 0 || bTRUE ||
TS_SPEC_FIRST || bTRUE || || TCRS_AR_TARIFF_HIEARCHY || 0 ||
99999999999999.99 || SFT_NULL || STT_NULL || bFALSE || 5 || 5 || 5 || 5 ||
IGL_ONE_INVC_FOR_ALL_VCHRS || || bFALSE || bFALSE || FIPM_AUTO_PAY || 15
|| bTRUE || bFALSE || || bFALSE || || FT-CUSTOMER || || || *DFT ||
*DFT || " 50 " || || CST1 || CST1 || S_ACTIVE || JO-ANNE || || 05/22/
1999@14:58:04 || 05/22/1999@14:58:04 || Custs.[0].Addr ||
```

```

Custs.[0].CntcInf || Custs.[0].BusHours || Custs.[0].Mmo || bTRUE ||
Custs.[0].Conts || bTRUE || || bFALSE || Custs.[0].CompTyps || bTRUE ||
|| bTRUE ||
CustomerList_V1 || Custs || Custs.[0]
EntitySrvc || Update || Custs

```

Update Distribution Center Example

```

DistributionCenter_V1 || DC_List.[0] || JUNE8 || JUNE DC || FRE || CAD ||
UMS_IMPERIAL || UMW_LB || UML_FT || bFALSE || 9999999.9999 ||
9999999.9999 || 0 || LUT_NULL || 0 || LUT_NULL || 0 || 0 || 0 || || CST1
|| CST1 || ONT || ONT || ONT || S_ACTIVE || 05/22/1999@15:29:58 || || JO-
ANNE || || DC_List.[0].Addr || DC_List.[0].CntcInf ||
DC_List.[0].BusHours || DC_List.[0].Mmo || bTRUE || DC_List.[0].Conts ||
bTRUE ||
|| bTRUE || ||
DistributionCenterList_V1 || DC_List || DC_List.[0]
EntitySrvc || Update || DC_List

```

Update Equipment Type Example

```

EquipmentType_V1 || equip || 50FT || 50 FT || 36FTCONT || TB
EquipmentTypeList_V1 || equipList || equip
EntitySrvc || Update || equipList

```

Update Harmonized Tariff

```

Memo_V1 || HarmTariffList_Out.[0].Mmo || ||
HarmTariff_V1 || HarmTariffList_Out.[0] || MF0913-1 || Upd MF0913-1 desc
|| bFALSE || HarmTariffList_Out.[0].Mmo || 06/09/2000@15:35:12 || 06/13/
2000@11:28:34 || VENTURE || VENTURE
Memo_V1 || HarmTariffList_Out.[1].Mmo || Desc 2A || Desc 2B
HarmTariffList_V1 || HarmTariffList_Out || HarmTariffList_Out.[0]
EntitySrvc || Update || HarmTariffList_Out

```

Update Hub Example

```

Hub_V1 || Hubs.[0] || JEFF8 || JEFFS HUB || bFALSE || 9999999.9999 ||
9999999.9999 || D_ALLOWED || P_ALLOWED || bTRUE || 0 || 0 || 0 || bFALSE
|| bFALSE || 0 || LUT_NULL || 0 || LUT_NULL || 0 || 0 || 0 || bFALSE ||
|| CST1 || CST1 || || JLMAY21 || BC || BC || || ONT || S_ACTIVE || 05/26/
1999@15:09:31 || || JO-ANNE || || Hubs.[0].Addr || Hubs.[0].CntcInf ||
Hubs.[0].BusHours || Hubs.[0].Mmo || bTRUE || Hubs.[0].Conts || bTRUE ||
|| bTRUE || ||
HubList_V1 || Hubs || Hubs.[0]
EntitySrvc || Update || Hubs

```

Update Item Master

```

Memo_V1 || ItemGroupList_Out.[0].ItmMstrs.[0].Mmo || Upd A || Upd B
ItemMaster_V1 || ItemGroupList_Out.[0].ItmMstrs.[0] || MF0705-03-01 ||
MF-TEST1 || Upd Group 1 item 5 || || 20 || ITM_ACTIVE || SLC_NOT_REQRD ||
0 || || || || || 55 || ARE || GROC ||
ItemGroupList_Out.[0].ItmMstrs.[0].Mmo || ||
ItemMasterList_V1 || ItemGroupList_Out.[0].ItmMstrs ||
ItemGroupList_Out.[0].ItmMstrs.[0]
EntitySrvc || Update || ItemGroupList_Out.[0].ItmMs

```

Update Load-at Example

```

LoadAt_V1 || LAs.[0] || LEN8 || LEN LA || ENG1 || bFALSE || 9999999.9999
|| 9999999.9999 || || || LUT_NULL || 0 || 0 || 0 || 0 || 0 || LUT_NULL ||
CST1 || CST1 || || || || || || S_ACTIVE || 05/22/1999@16:07:14 || || JO-
ANNE || || LAs.[0].Addr || LAs.[0].CntcInf || LAs.[0].BusHours ||
LAs.[0].Mmo || LAs.[0].Directions || bTRUE || LAs.[0].Conts || bTRUE ||
|| bTRUE || ||
LoadAtList_V1 || LAs || LAs.[0]
EntitySrvc || Update || LAs

```

Update NMFC

```

Memo_V1 || NMFCList_Out.[0].Mmo || ||
NMFC_V1 || NMFCList_Out.[0] || MF0802-1 || Updated test MF0724-1 || " 50
" || bFALSE || NMFCList_Out.[0].Mmo || 06/08/2000@18:59:30 || || VENTURE
||
NMFCList_V1 || NMFCList_Out || NMFCList_Out.[0]
EntitySrvc || Update || NMFCList_Out

```

Financial Services

The financial services retrieve and commit Transportation Manager accounting transactions.

Examples**Commit AP Transaction Example**

```

StrIdList || APTrnsID || 2081
APTransactionList_V1 || APRetData
FinancialsSrvc || CommitAPTransaction || APTrnsID

```

Commit AR Transaction Example

```

StrIdList || ARTrnsID || 2081
ARTransactionList_V1 || ArRetData
FinancialsSrvc || CommitARTransaction || ARTrnsID

```

Commit GLTransaction Example

```

StrIdList || GLTrnsID || X2081
GLTransactionList_V1 || GLRetData
FinancialsSrvc || CommitGLTransaction || GLTrnsID

```

where x is P for A/P transactions, and R is for A/P transactions.

Create NOF

```

UMsr_V1 || OutputSHList.[0].UMsr || UMS_IMPERIAL || UMW_LB || UML_FT ||
UMD_MILES
UMsr_V1 || OutputSHList.[0].TariffUMsr || UMS_IMPERIAL || UMW_LB ||
UML_FT || UMD_MILES
ShippingInfo_V1 || OutputSHList.[0].ShipInfo || 120 || 1000 || 2000 || 0
|| 100 || 20 || 0 || 1 || 0
Address_V1 || OutputSHList.[0].FromAddr || 5 || HAHN PL || 316 || TORONTO
|| ON || CAN || M5A4G1 ||

```

```

Address_V1 || OutputSHList.[0].ToAddr || 6161 || BATHURST ST || 1202 ||
NORTH YORK || ON || CAN || M2R1Z5 ||
Memo_V1 || OutputSHList.[0].Memo || ||
ShippingInfo_V1 || OutputSHList.[0].Containers.[0].ShipInfo || 120 ||
1000 || 2000 || 0 || 100 || 20 || 0 || 1 || 0
ShipmentItem_V1 || OutputSHList.[0].Containers.[0].ShpmItems.[0] || 9166
|| || || 1234567890 || 1 || 100 || 100 || 100 || 2000 || 2000 || || ||
IS_EXTERNAL_API || SLC_NOT_REQRD || S_NULL || 33345 || 48861 || IYCUST1
|| " 50 " || || || CAN || 1,1,1 || bTRUE ||
ShipmentItemList_V1 || OutputSHList.[0].Containers.[0].ShpmItems ||
OutputSHList.[0].Containers.[0].ShpmItems.[0]
Container_V1 || OutputSHList.[0].Containers.[0] || 48861 || SKID || 1 ||
10 || 10 || 10 || OutputSHList.[0].Containers.[0].ShipInfo || Skid 1 || 0
|| 0 || 0 || DIMN_NULL || || 0 || 1 || 0 || UT_SKIDS || 9999999.9999 ||
MF-717631234567 || || 0 || TRNSSRC_API || SPT_NULL || || bTRUE || ||
bTRUE || || bFALSE || OutputSHList.[0].Containers.[0].ShpmItems || bTRUE
||
ContainerList_V1 || OutputSHList.[0].Containers ||
OutputSHList.[0].Containers.[0]
Shipment_V2 || OutputSHList.[0] || 33345 || SH-000033345 || SI-CUST5 ||
SI-1 || S1 || *DFT || *DFT || || IS_EXTERNAL_API || || bFALSE ||
TOM_ITEM_LEVEL_DETAIL || KC || || || CSE_NULL || || || || ||
FT_PRE_PAID || SI-CUST1 || bFALSE || bFALSE || DRY || || 0 ||
OutputSHList.[0].Memo || SFT_LA || SI-LA1 || SHIPMENT INTERFACE - LOAD-
AT 1 || OutputSHList.[0].FromAddr || bFALSE || STT_CONSIGNEE || SI-CN1 ||
SHIPMENT INTERFACE - CONSIGNEE 1 || OutputSHList.[0].ToAddr || bFALSE ||
02/02/2000@08:00:00 || 02/02/2000@10:00:00 || 02/02/2000@09:00:00 || 02/
02/ 2000@10:01:00 || bFALSE || *DFT || *DFT || || bFALSE || 0 || VENTURE
|| VENTURE || || || || S_PROCESSED || S_F_INELIGIBLE ||
OutputSHList.[0].UMsr || OutputSHList.[0].TariffUMsr ||
OutputSHList.[0].ShipInfo || bFALSE || bFALSE || bTRUE || bTRUE || bTRUE
|| 10000 || 45000 || 4500 || bFALSE || bFALSE || || SI-CUST5 || bTRUE ||
|| bFALSE || OutputSHList.[0].Containers || bTRUE ||
ShipmentList_V2 || OutputSHList || OutputSHList.[0]
ItineraryPoint_V1 || ITList.[0].ItineraryPoint.[0] || 1754 || Point 54
#1 || 1 || IPT_LA || DSAPI030-13 || TOR-MIL || HAMILTON ||
ItineraryPoint_V1 || ITList.[0].ItineraryPoint.[1] || 1755 || Point 54
#2 || 2 || IPT_DC || DSAPI030-13 || TOR-MIL || MILTON ||
ItineraryPointList_V1 || ITList.[0].ItineraryPoint ||
ITList.[0].ItineraryPoint.[0] || ITList.[0].ItineraryPoint.[1]
Itinerary_V1 || ITList.[0] || DSAPI030-13 || TOR-MIL || Itinerary 4 for
DS DSAPI030-13 || ITNR_ACTIVE || VENTURE || || || bFALSE ||
ITList.[0].ItineraryPoint || bTRUE ||
ItineraryList_V1 || ITList || ITList.[0]
Address_V1 || aList.[0].FrstAddr || 5 || HAHN PL || 316 || TORONTO || ON
|| CAN || M5A4G1 ||
Address_V1 || aList.[0].LastAddr || 6161 || BATHURST ST || 1202 || NORTH
YORK || ON || CAN || M2R1Z5 ||
Address_V1 || aList.[0].TdPyAddr || || || || || || || || ||
Memo_V1 || aList.[0].Mmo || ||
RatingInfo_V1 || aList.[0].APRatingInfo || UMS_IMPERIAL || UMW_LB ||
UML_FT || UMD_MILES || USD || 1 || bFALSE || || 08/ 30/2000@15:22:09 ||
C004 || 100 || 0 || 100 || || || || 0 || || R1 || || 3DAY || 1846

```

```

RatingInfo_V1 || aList.[0].ARRatingInfo || UMS_NULL || UMW_NULL ||
UML_NULL || UMD_NULL || || || bNULL || || || || || || || || || || || ||
|| || || ||
ChargeDetail_V1 || aList.[0].APChargeDetail.[0] || || || || || 0 ||
CHL_SERVICE || 0 || 0 || 0 || 0 || 0 || 0 || 0 || 0 || OAL_NULL || 0 || 0
ChargeDetail_V1 || aList.[0].APChargeDetail.[1] || || WGT || *FAK ||
*FAK || 1 || CHL_CONDITION || 100 || 100 || 100 || 499.9999 || 1 || 100
|| 0 || OAL_NULL || 1 || 100
ChargeDetail_V1 || aList.[0].APChargeDetail.[2] || || || || || 3 ||
CHL_TOTAL || 0 || 0 || 0 || 0 || 0 || 100 || 0 || OAL_NULL || 0 || 100
ChargeDetailList_V1 || aList.[0].APChargeDetail ||
aList.[0].APChargeDetail.[0] || aList.[0].APChargeDetail.[1] ||
aList.[0].APChargeDetail.[2]
NOFWeightByFreightClass_V1 || aList.[0].NOFWeightByFreight.[0] || 100 ||
" 45 "
NOFWeightByFreightClassList_V1 || aList.[0].NOFWeightByFreight ||
aList.[0].NOFWeightByFreight.[0]
RefNumber_V1 || aList.[0].RefNumber.[0] || CR || 1234567
RefNumberList_V1 || aList.[0].RefNumber || aList.[0].RefNumber.[0]
Address_V1 || aList.[0].NOFStop.[0].ShpgAddr || 5 || HAHN PL || 316 ||
TORONTO || ON || CAN || M5A4G1 ||
NOF_Stop_V1 || aList.[0].NOFStop.[0] || 1 || SPT_LA || IYLA ||
aList.[0].NOFStop.[0].ShpgAddr || 1 || 100 || 0 || 0 || 0 || 100 || 0 ||
0 || 0 || 0 || 0 || 0 || 0 || 0 || 0 || 0 || 0 || 0 || 0 || 0 || 0 ||
bTRUE ||
Address_V1 || aList.[0].NOFStop.[1].ShpgAddr || 6161 || BATHURST ST ||
1202 || NORTH YORK || ON || CAN || M2R1Z5 ||
NOF_Stop_V1 || aList.[0].NOFStop.[1] || 2 || SPT_CONSIGNEE || IYCON ||
aList.[0].NOFStop.[1].ShpgAddr || 0 || 0 || 0 || 0 || 0 || 0 || 0 || 0 ||
0 || 0 || 1 || 100 || 0 || 0 || 0 || 100 || 0 || 0 || 0 || 0 || 0 || 0 ||
bTRUE ||
NOFStopList_V1 || aList.[0].NOFStop || aList.[0].NOFStop.[0] ||
aList.[0].NOFStop.[1]
ChargeOverride_V1 || aList.[0].ChargeOverride.[0] || BLOK ||
OAL_RATE_ROUTE || OVS_API || OAL_DO_NOT_APPLY || OVS_NULL || || OVS_NULL
|| || || ||
ChargeOverrideList_V1 || aList.[0].ChargeOverride ||
aList.[0].ChargeOverride.[0]
NonOperationalFreight_V1 || aList.[0] || IYCUST || MF-0913-02 || || IYD
|| IYLG || IYLA || SPT_LA || aList.[0].FrstAddr || IYCON || SPT_CONSIGNEE
|| aList.[0].LastAddr || || SPT_NULL || aList.[0].TdPyAddr ||
CARRFRHTTRMS_PREPAID || FT_PRE_PAID || 08/30/2000@08:00:08 || || bFALSE
|| IYCAR || USD || UMS_IMPERIAL || UMW_LB || UML_FT || UMD_MILES || GROC
|| 3DAY || || RteT_POINT_TO_POINT || || P001 || 100 || 0 || 0 || 0 || 100
|| 0 || 0 || 0 || 0 || 1 || || || || || BOX || 100 || 0 || 0 || 0 || 100
|| 0 || 0 || 0 || 0 || 1 || || || || aList.[0].Mmo || NOFTYPE_AUDIT_NO_MATCH
|| TRNSSRC_GUI || IYCUST || || 1 || bFALSE || 2 || || NOF_INELIGIBLE
|| || 08/30/2000@15:22:41 || VENTURE || 08/30/2000@15:50:15 || VENTURE
|| aList.[0].APRatingInfo || 7 || 7 || 0 || 7 || 0 || 0 ||
aList.[0].ARRatingInfo || || aList.[0].APChargeDetail || ||
aList.[0].NOFWeightByFreight || bFALSE || aList.[0].RefNumber || bFALSE
|| aList.[0].NOFStop || bFALSE || aList.[0].ChargeOverride
NonOperationalFreightList_V1 || aList || aList.[0]
StrIdList || Output

```

```
FinancialsSrvc || CreateNOF || aList || Output
```

Delete NOF Example

```
StrIdList || Idlist || clover2
StrIdList || Idlist || MF-0913-02
FinancialsSrvc || DeleteNOF || IdList
```

Responsible Customer Override Example

```
RspbCustOvr_V1 || cust || BULK || || || RSPB_CUST_ORIG_SHPG_LOC ||
RspbCustOvrList_V1 || custList || cust
FinancialsSrvc || ResponsibleCustomerOverride || 1016 || custList
```

Retrieve AP Transaction Example

```
APTransactionList_V1 || transactions
FinancialsSrvc || RetrieveAPTransaction || transactions
```

Retrieve AP Transaction Quantity Example

```
APTransactionList_V1 || transactions
FinancialsSrvc || RetrieveAPTransactionQty || transactions || 200
```

Retrieve AR Transaction Example

```
ARTransactionList_V1 || transactions
FinancialsSrvc || RetrieveARTransaction || transactions
```

Retrieve AR Transaction Quantity Example

```
ARTransactionList_V1 || transactions
FinancialsSrvc || RetrieveARTransactionQty || transactions || 200
```

Retrieve GL Transaction Example

```
GLTransactionList_V1 || transactions
FinancialsSrvc || RetrieveGLTransaction || transactions || GLTT_POSTED
|| GLC_AP_EXPS_RVNU_SHPM
```

Retrieve GL Transaction Quantity Example

```
GLTransactionList_V1 || transactions
FinancialsSrvc || RetrieveGLTransactionQty || transactions ||
GLTT_POSTED || GLC_AP_EXPS_RVNU_SHPM || || || || || 200
```

Retrieve NOF Example

```
NonOperationalFreightList_V1 || aList
StrIdList || IdList1 || MF-0913-02
FinancialsSrvc || RetrieveNOF || IdList1 || aList
```

Retrieve Responsible Customer Override Example

```
RspbCustOvrList_V1 || custList ||
FinancialsSrvc || RetrieveResponsibleCustomerOverride || 1016 ||
custList
```

Update NOF Example

```
UMsr_V1 || OutputSHList.[0].UMsr || UMS_IMPERIAL || UMW_LB || UML_FT ||
UMD_MILES
```

```

UMsr_V1 || OutputSHList.[0].TariffUMsr || UMS_IMPERIAL || UMW_LB ||
UML_FT || UMD_MILES
ShippingInfo_V1 || OutputSHList.[0].ShipInfo || 120 || 1000 || 2000 || 0
|| 100 || 20 || 0 || 1 || 0
Address_V1 || OutputSHList.[0].FromAddr || 5 || HAHN PL || 316 || TORONTO
|| ON || CAN || M5A4G1 ||
Address_V1 || OutputSHList.[0].ToAddr || 6161 || BATHURST ST || 1202 ||
NORTH YORK || ON || CAN || M2R1Z5 ||
Memo_V1 || OutputSHList.[0].Memo || ||
ShippingInfo_V1 || OutputSHList.[0].Containers.[0].ShipInfo || 120 ||
1000 || 2000 || 0 || 100 || 20 || 0 || 1 || 0
ShipmentItem_V1 || OutputSHList.[0].Containers.[0].ShpmItems.[0] || 9166
|| || 1234567890 || 1 || 100 || 100 || 100 || 2000 || 2000 || ||
IS_EXTERNAL_API || SLC_NOT_REQRD || S_NULL || 33345 || 48861 || IYCUST1
|| " 50 " || || CAN || 1,1,1 || bTRUE ||
ShipmentItemList_V1 || OutputSHList.[0].Containers.[0].ShpmItems ||
OutputSHList.[0].Containers.[0].ShpmItems.[0]
Container_V1 || OutputSHList.[0].Containers.[0] || 48861 || SKID || 1 ||
10 || 10 || 10 || OutputSHList.[0].Containers.[0].ShipInfo || Skid 1 || 0
|| 0 || 0 || DIMN_NULL || 0 || 1 || 0 || UT_SKIDS || 9999999.9999 ||
MF-717631234567 || 0 || TRNSSRC_API || SPT_NULL || bTRUE ||
bTRUE || bFALSE || OutputSHList.[0].Containers.[0].ShpmItems || bTRUE
||
ContainerList_V1 || OutputSHList.[0].Containers ||
OutputSHList.[0].Containers.[0]
Shipment_V2 || OutputSHList.[0] || 33345 || SH-000033345 || SI-CUST5 ||
SI-1 || S1 || *DFT || *DFT || IS_EXTERNAL_API || bFALSE ||
TOM_ITEM_LEVEL_DETAIL || KC || CSE_NULL ||
FT_PRE_PAID || SI-CUST1 || bFALSE || bFALSE || DRY || 0 ||
OutputSHList.[0].Memo || SFT_LA || SI-LA1 || SHIPMENT INTERFACE - LOAD-
AT 1 || OutputSHList.[0].FromAddr || bFALSE || STT_CONSIGNEE || SI-CN1 ||
SHIPMENT INTERFACE - CONSIGNEE 1 || OutputSHList.[0].ToAddr || bFALSE ||
02/02/2000@08:00:00 || 02/02/2000@10:00:00 || 02/02/2000@09:00:00 || 02/
02/2000@10:01:00 || bFALSE || *DFT || *DFT || bFALSE || 0 || VENTURE
|| VENTURE || S_PROCESSED || S_F_INELIGIBLE ||
OutputSHList.[0].UMsr || OutputSHList.[0].TariffUMsr ||
OutputSHList.[0].ShipInfo || bFALSE || bFALSE || bTRUE || bTRUE || bTRUE
|| 10000 || 45000 || 4500 || bFALSE || bFALSE || SI-CUST5 || bTRUE ||
|| bFALSE || OutputSHList.[0].Containers || bTRUE ||
ShipmentList_V2 || OutputSHList || OutputSHList.[0]
ItineraryPoint_V1 || ITList.[0].ItineraryPoint.[0] || 1754 || Point 54
#1 || 1 || IPT_LA || DSAPI030-13 || TOR-MIL || HAMILTON ||
ItineraryPoint_V1 || ITList.[0].ItineraryPoint.[1] || 1755 || Point 54
#2 || 2 || IPT_DC || DSAPI030-13 || TOR-MIL || MILTON ||
ItineraryPointList_V1 || ITList.[0].ItineraryPoint ||
ITList.[0].ItineraryPoint.[0] || ITList.[0].ItineraryPoint.[1]
Itinerary_V1 || ITList.[0] || DSAPI030-13 || TOR-MIL || Itinerary 4 for
DS DSAPI030-13 || ITNR_ACTIVE || VENTURE || bFALSE ||
ITList.[0].ItineraryPoint || bTRUE ||
ItineraryList_V1 || ITList || ITList.[0]
Address_V1 || aList.[0].FrstAddr || 5 || HAHN PL || 316 || TORONTO || ON
|| CAN || M5A4G1 ||
Address_V1 || aList.[0].LastAddr || 6161 || BATHURST ST || 1202 || NORTH
YORK || ON || CAN || M2R1Z5 ||

```



```

Address_V1 || aList.[0].TdPyAddr || || || || || || || || ||
Memo_V1 || aList.[0].Mmo || ||
RatingInfo_V1 || aList.[0].APRatingInfo || UMS_IMPERIAL || UMW_LB ||
UML_FT || UMD_MILES || USD || 1 || bFALSE || || 08/30/2000@15:22:09 ||
C004 || 100 || 0 || 100 || || || || 0 || || R1 || || || 3DAY || 1846
RatingInfo_V1 || aList.[0].ARRatingInfo || UMS_NULL || UMW_NULL ||
UML_NULL || UMD_NULL || || || bNULL || || || || || || || || || || ||
|| || || || ||
ChargeDetail_V1 || aList.[0].APChargeDetail.[0] || || || || || || 0 ||
CHL_SERVICE || 0 || 0 || 0 || 0 || 0 || 0 || 0 || 0 || OAL_NULL || 0 || 0
ChargeDetail_V1 || aList.[0].APChargeDetail.[1] || || WGT || *FAK ||
*FAK || 1 || CHL_CONDITION || 100 || 100 || 100 || 499.9999 || 1 || 100
|| 0 || OAL_NULL || 1 || 100
ChargeDetail_V1 || aList.[0].APChargeDetail.[2] || || || || || || 3 ||
CHL_TOTAL || 0 || 0 || 0 || 0 || 0 || 100 || 0 || OAL_NULL || 0 || 100
ChargeDetailList_V1 || aList.[0].APChargeDetail ||
aList.[0].APChargeDetail.[0] || aList.[0].APChargeDetail.[1] ||
aList.[0].APChargeDetail.[2]
NOFWeightByFreightClass_V1 || aList.[0].NOFWeightByFreight.[0] || 100 ||
" 45 "
NOFWeightByFreightClassList_V1 || aList.[0].NOFWeightByFreight ||
aList.[0].NOFWeightByFreight.[0]
RefNumber_V1 || aList.[0].RefNumber.[0] || CR || 1234568
RefNumberList_V1 || aList.[0].RefNumber || aList.[0].RefNumber.[0]
Address_V1 || aList.[0].NOFStop.[0].ShpgAddr || 5 || HAHN PL || 316 ||
TORONTO || ON || CAN || M5A4G1 ||
NOF_Stop_V1 || aList.[0].NOFStop.[0] || 1 || SPT_LA || IYLA ||
aList.[0].NOFStop.[0].ShpgAddr || 1 || 100 || 0 || 0 || 0 || 100 || 0 ||
0 || 0 || 0 || 0 || 0 || 0 || 0 || 0 || 0 || 0 || 0 || 0 || || || ||
bTRUE ||
Address_V1 || aList.[0].NOFStop.[1].ShpgAddr || 6161 || BATHURST ST ||
1202 || NORTH YORK || ON || CAN || M2R1Z5 ||
NOF_Stop_V1 || aList.[0].NOFStop.[1] || 2 || SPT_CONSIGNEE || IYCON ||
aList.[0].NOFStop.[1].ShpgAddr || 0 || 0 || 0 || 0 || 0 || 0 || 0 || 0 ||
0 || 0 || 1 || 100 || 0 || 0 || 0 || 100 || 0 || 0 || 0 || 0 || || || ||
bTRUE ||
NOFStopList_V1 || aList.[0].NOFStop || aList.[0].NOFStop.[0] ||
aList.[0].NOFStop.[1]
NonOperationalFreight_V1 || aList.[0] || || MF-0913-02 || || IYD || IYLG
|| IYLA || SPT_LA || aList.[0].FrstAddr || IYCON || SPT_CONSIGNEE ||
aList.[0].LastAddr || || SPT_NULL || aList.[0].TdPyAddr ||
CARRFRHTTRMS_PREPAID || FT_PRE_PAID || 08/30/2000@08:00:08 || || bFALSE
|| || USD || UMS_IMPERIAL || UMW_LB || UML_FT || UMD_MILES || GROC ||
3DAY || || RteT_POINT_TO_POINT || || P001 || 100 || 0 || 0 || 0 || 100 ||
0 || 0 || 0 || 0 || 1 || || || || || BOX || 100 || 0 || 0 || 0 || 100 ||
0 || 0 || 0 || 0 || 1 || || || aList.[0].Mmo || NOFTYPE_AUDIT_NO_MATCH ||
TRNSSRC_GUI || IYCUST || || 1 || bFALSE || 2 || || NOF_INELIGIBLE ||
|| 08/30/2000@15:22:41 || VENTURE || 08/30/2000@15:50:15 || VENTURE ||
aList.[0].APRatingInfo || 7 || 7 || 0 || 7 || 0 || 0 ||
aList.[0].ARRatingInfo || || aList.[0].APChargeDetail || ||
aList.[0].NOFWeightByFreight || bFALSE || aList.[0].RefNumber || bFALSE
|| aList.[0].NOFStop
NonOperationalFreightList_V1 || aList || aList.[0]
FinancialsSrvc || UpdateNOF || aList

```

Load Services

The Transportation Manager load services send data to and from your system in a data structure format. Define the data structures in the input files as single line entries using the following format.

```
Data_Structure_Name || Mnemonic || Value || Value || Value
...
```

Data_Structure_Name is the type of load structure object that you are creating. It is one of the following types:

- LoadBuildPlan_V1
- SEC_V1
- StrIdList

The driver cannot read most load service data structures from an input file. Therefore, it is possible that the driver will create several data structures and put them into the output file. The following data structures are in this category:

- Charge_V1
- Load_V1
- ShipmentLeg_V1
- Stop_V1

Because these data structures are defined mainly for data retrieval, the driver will not read data for these structures within the context of an input file. However, for simplicity, these structures are still allowed within this context.

This service has the following characteristics:

- *mnemonic* is a tag
- the values that follow are specific to the data structure object you are defining
- you can omit the specification of a data field by leaving the field blank

Data Structure

A data structure can contain other data structures or lists. A load retrieval transaction creates data structures that are packaged inside a data structure list. The create plan operation will not produce a list, but will fill in the created plan ID into the LoadBuildPlan_V1 structure.

The following definition is for a data structure.

```
Data_Structure_List || Mnemonic // Value || Value || Value
```

Data_Structure_List is the type of sequence list you are creating. The supported data structure list types relevant to loads are:

- LoadList_V1
- ShipmentLegList_V1

- StopList_V1

Other data list structures may be returned as aggregates.

The `Mnemonic` field is a tag for referring to this list in later transactions. The data values for a list are the mnemonics for the data structures which indicate the list you are creating. You do not need to predefine the mnemonics.

Define the data structures before referring to them in a data transfer operation. If you try to insert a data structure that is inconsistent with its type, it will send an error to the error file.

Define all the input data elements of a load service data structure before using them in a load service data transfer operation. Otherwise, the operation will fail.

Data Transfer

You can do basic retrieval transactions, plan creation, and load event settings using the load services.

`LoadSrvc` means that you are defining a load service transaction. The next field is the operation. The appropriate data is gathered and bundled. You can use a list to transfer data in bulk between your system and the server. Otherwise, use a single value.

When you use a single value, place the actual value in the line. (For `RetrievePlanLoads`, place the plan ID string as the third parameter.) The appropriate mnemonic indicates the list. The list type for the mnemonic is checked for the given operation. If there is a mismatch, an error is sent to the error output file and the operation terminates.

Examples

Assign To Load Example

```
StrIdList || MyShipments || 100006291 || 100006730 LoadSrvc ||
AssignToLoad || MyShipments || 800426 || TRUE
```

Assign To Load Example

```
StrIdList || shpmLegIDs || 123456789 ||
LoadSrvc || AssignToNewLoad || 1101 || ShpmLegIDs
```

CreatePlan

Define the `LoadBuildPlan_V1` data structure that contains the required data values. You cannot set the value for the plan ID, since it is assigned by the API services. The error list is automatically set to the default error location. If the operation is successful, the `LoadBuildPlan_V1` structure is sent to the default output location with the data value for the plan ID assigned.

CreatePlan Example

```
LoadBuildPlan_V1 || LBPlan_1 || || Test plan || *DFT || *DFT
StrIdList || NewIDs
LoadSrvc || CreatePlan || LBPlan_1 || NewIDs
```

Disable Assign To Trip Example

```
StrIdList || IdList || 2974
LoadSrvc || DisableAssignToTrip || IdList
```

Disable Trip Continuation Example

```
StrIdList || IdList || 1143
LoadSrvc || DisableTripContinuation || IdList
```

Load Confirm Example

```
LoadConfirmData_V1 || i2_LoadConfData || 3522 || || || || || bTRUE ||
bTRUE
LoadSrvc || LoadConfirm || i2_LoadConfData
```

Load Create Example

```
UMsr_V1 || OutSHList.[0].UMsr || UMS_IMPERIAL || UMW_LB || UML_FT ||
UMD_MILES
UMsr_V1 || OutSHList.[0].TariffUMsr || UMS_IMPERIAL || UMW_LB || UML_FT
|| UMD_MILES
ShippingInfo_V1 || OutSHList.[0].ShipInfo || 70 || 50 || 50 || 12.34 ||
50 || 20 || 4 || 0 || 0
Address_V1 || OutSHList.[0].FromAddr || 8300 || MANUEL ST || || DETROIT
|| MI || USA || 48211 || FF || 0 || 0 || bTRUE || bFALSE
Address_V1 || OutSHList.[0].ToAddr || 3100 || CHERRY ST || || ATLANTA ||
GA || USA || 30344 || || 0 || 0 || bFALSE || bFALSE
Memo_V1 || OutSHList.[0].Memo || ||
ShippingInfo_V1 || OutSHList.[0].Containers.[0].ShipInfo || 70 || 50 ||
50 || 12.34 || 50 || 5 || 4 || 0 || 0
CntrOrtn_V1 || OutSHList.[0].Containers.[0].CntrOrtns.[0] || OR_LWH ||
ORALW_UNDEFINED || 0
CntrOrtn_V1 || OutSHList.[0].Containers.[0].CntrOrtns.[1] || OR_WLH ||
ORALW_UNDEFINED || 0
CntrOrtn_V1 || OutSHList.[0].Containers.[0].CntrOrtns.[2] || OR_LHW ||
ORALW_UNDEFINED || 0
CntrOrtn_V1 || OutSHList.[0].Containers.[0].CntrOrtns.[3] || OR_HLW ||
ORALW_UNDEFINED || 0
CntrOrtn_V1 || OutSHList.[0].Containers.[0].CntrOrtns.[4] || OR_WHL ||
ORALW_UNDEFINED || 0
CntrOrtn_V1 || OutSHList.[0].Containers.[0].CntrOrtns.[5] || OR_HWL ||
ORALW_UNDEFINED || 0
CntrOrtnList_V1 || OutSHList.[0].Containers.[0].CntrOrtns ||
OutSHList.[0].Containers.[0].CntrOrtns.[0] ||
OutSHList.[0].Containers.[0].CntrOrtns.[1] ||
OutSHList.[0].Containers.[0].CntrOrtns.[2] ||
OutSHList.[0].Containers.[0].CntrOrtns.[3] ||
OutSHList.[0].Containers.[0].CntrOrtns.[4] ||
OutSHList.[0].Containers.[0].CntrOrtns.[5]
WeightByFreightClass_V1 || OutSHList.[0].Containers.[0].WgtByFCs.[0] ||
50 || 50
WeightByFreightClassList_V1 || OutSHList.[0].Containers.[0].WgtByFCs ||
OutSHList.[0].Containers.[0].WgtByFCs.[0]
Container_V1 || OutSHList.[0].Containers.[0] || 101881 || PIECES || 4 ||
0 || 0 || 0 || OutSHList.[0].Containers.[0].ShipInfo || || 0 || 0 || 0 ||
DIMN_NULL || || 0 || 1 || 0 || UT_PIECES || 9999999.9999 || || || 0 ||
```

```

TRNSSRC_OTHER || SPT_NULL || || bFALSE ||
OutSHList.[0].Containers.[0].CntrOrtns || bFALSE ||
OutSHList.[0].Containers.[0].WgtByFCs || bTRUE || || bTRUE ||
ContainerList_V1 || OutSHList.[0].Containers ||
OutSHList.[0].Containers.[0]
Shipment_V2 || OutSHList.[0] || 525955 || || TV-CUST01 || *DFT || SC ||
TV-C || TV-C || || IS_MANUALLY || || bFALSE || TOM_COMP_LEVEL_DETAIL ||
|| TV-CAR01 || || || CSE_NULL || || || || FT_PRE_PAID || TV-CUST01 ||
bFALSE || bFALSE || DRY || || 0 || OutSHList.[0].Memo || SFT_LA || TV-LA
|| TOM'S LOAD-AT FOR CARRIER VIEW || OutSHList.[0].FromAddr || bFALSE ||
STT_CONSIGNEE || TV-CONS || TOM'S CONSIGNEE FOR CARRIER VIEW ||
OutSHList.[0].ToAddr || bFALSE || 10/01/2005@09:00:00 || 10/30/
2005@17:00:00 || 10/02/2005@09:00:00 || 10/31/2005@17:00:00 || bFALSE ||
GROC || *DFT || || bFALSE || 0 || VENTURE || VENTURE || 05/29/
2001@14:04:42 || 05/29/2001@14:05:11 || || || S_ASSIGNED ||
S_F_INELIGIBLE || OutSHList.[0].UMsr || OutSHList.[0].TariffUMsr ||
OutSHList.[0].ShipInfo || || INCO_BuyerSeller_NULL || || SPT_NULL ||
bFALSE || bFALSE || bFALSE || bTRUE || bTRUE || bTRUE || 999999999999.99
|| 45000 || 45000 || || bFALSE || bFALSE || || TV-CUST01 || TSM_NULL ||
bTRUE || || bFALSE || OutSHList.[0].Containers || bTRUE ||
UMsr_V1 || OutSHList.[1].UMsr || UMS_IMPERIAL || UMW_LB || UML_FT ||
UMD_MILES
UMsr_V1 || OutSHList.[1].TariffUMsr || UMS_IMPERIAL || UMW_LB || UML_FT
|| UMD_MILES
ShippingInfo_V1 || OutSHList.[1].ShipInfo || 70 || 50 || 50 || 12.34 ||
50 || 20 || 4 || 0 || 0
Address_V1 || OutSHList.[1].FromAddr || 8300 || MANUEL ST || || DETROIT
|| MI || USA || 48211 || FF || 0 || 0 || bTRUE || bFALSE
Address_V1 || OutSHList.[1].ToAddr || 3100 || CHERRY ST || || ATLANTA ||
GA || USA || 30344 || || 0 || 0 || bFALSE || bFALSE
Memo_V1 || OutSHList.[1].Memo || ||
ShippingInfo_V1 || OutSHList.[1].Containers.[0].ShipInfo || 70 || 50 ||
50 || 12.34 || 50 || 5 || 4 || 0 || 0
CntrOrtn_V1 || OutSHList.[1].Containers.[0].CntrOrtns.[0] || OR_LWH ||
ORALW_UNDEFINED || 0
CntrOrtn_V1 || OutSHList.[1].Containers.[0].CntrOrtns.[1] || OR_WLH ||
ORALW_UNDEFINED || 0
CntrOrtn_V1 || OutSHList.[1].Containers.[0].CntrOrtns.[2] || OR_LHW ||
ORALW_UNDEFINED || 0
CntrOrtn_V1 || OutSHList.[1].Containers.[0].CntrOrtns.[3] || OR_HLW ||
ORALW_UNDEFINED || 0
CntrOrtn_V1 || OutSHList.[1].Containers.[0].CntrOrtns.[4] || OR_WHL ||
ORALW_UNDEFINED || 0
CntrOrtn_V1 || OutSHList.[1].Containers.[0].CntrOrtns.[5] || OR_HWL ||
ORALW_UNDEFINED || 0
CntrOrtnList_V1 || OutSHList.[1].Containers.[0].CntrOrtns ||
OutSHList.[1].Containers.[0].CntrOrtns.[0] ||
OutSHList.[1].Containers.[0].CntrOrtns.[1] ||
OutSHList.[1].Containers.[0].CntrOrtns.[2] ||
OutSHList.[1].Containers.[0].CntrOrtns.[3] ||
OutSHList.[1].Containers.[0].CntrOrtns.[4] ||
OutSHList.[1].Containers.[0].CntrOrtns.[5]
WeightByFreightClass_V1 || OutSHList.[1].Containers.[0].WgtByFCs.[0] ||
50 || 50

```

```

WeightByFreightClassList_V1 || OutSHList.[1].Containers.[0].WgtByFCs ||
OutSHList.[1].Containers.[0].WgtByFCs.[0]
Container_V1 || OutSHList.[1].Containers.[0] || 101882 || PIECES || 4 ||
0 || 0 || 0 || OutSHList.[1].Containers.[0].ShipInfo || 0 || 0 || 0 ||
DIMN_NULL || 0 || 1 || 0 || UT_PIECES || 9999999.9999 || 0 ||
TRNSSRC_OTHER || SPT_NULL || bFALSE ||
OutSHList.[1].Containers.[0].CntrOrtns || bFALSE ||
OutSHList.[1].Containers.[0].WgtByFCs || bTRUE || bTRUE ||
ContainerList_V1 || OutSHList.[1].Containers ||
OutSHList.[1].Containers.[0]
Shipment_V2 || OutSHList.[1] || 525956 || TV-CUST01 || *DFT || SC ||
TV-C || TV-C || IS_MANUALLY || bFALSE || TOM_COMP_LEVEL_DETAIL ||
TV-CAR01 || CSE_NULL || FT_PRE_PAID || TV-CUST01 ||
bFALSE || bFALSE || DRY || 0 || OutSHList.[1].Memo || SFT_LA || TV-LA
|| TOM'S LOAD-AT FOR CARRIER VIEW || OutSHList.[1].FromAddr || bFALSE ||
STT_CONSIGNEE || TV-CONS || TOM'S CONSIGNEE FOR CARRIER VIEW ||
OutSHList.[1].ToAddr || bFALSE || 10/01/2005@09:00:00 || 10/30/
2005@17:00:00 || 10/02/2005@09:00:00 || 10/31/2005@17:00:00 || bFALSE ||
GROC || *DFT || bFALSE || 0 || VENTURE || VENTURE || 05/29/
2001@14:04:43 || 05/29/2001@14:05:11 || S_ASSIGNED ||
S_F_INELIGIBLE || OutSHList.[1].UMsr || OutSHList.[1].TariffUMsr ||
OutSHList.[1].ShipInfo || INCO_BuyerSeller_NULL || SPT_NULL ||
bFALSE || bFALSE || bFALSE || bTRUE || bTRUE || bTRUE || 9999999999999.99
|| 45000 || 45000 || bFALSE || bFALSE || TV-CUST01 || TSM_NULL ||
bTRUE || bFALSE || OutSHList.[1].Containers || bTRUE ||
ShipmentList_V2 || OutSHList || OutSHList.[0] || OutSHList.[1]
LoadCreateData_V1 || Createl || PreBuilt Load || 1699 || TV-CAR01 ||
3DAY || 48FT || 05/29/2001@12:10:00 || bFALSE
LoadSrvc || LoadBuild || OutSHList || Createl

```

Load Update Progress Example

```

SEC_V1 || LUP1.SECInfo || TRCN || 07/12/2001@13:52:00 ||
LoadUpdateProgress_V1 || LUP1 || HS || 11647 || LUP1.SECInfo
LoadUpdateProgressList_V1 || LUPList || LUP1
LoadSrvc || LoadUpdateProgress || LUPList || FALSE ||

```

Load Manifest Example

```

LoadManifestData_V1 || data || 12174 || bNULL || bFALSE
LoadManifestDataList_V1 || ds || data LoadSrvc || LoadManifest || ds

```

Payable Carrier Override Example

```

PayableCarrOvr_V1 || carr || BULK || PAY_CARR_TRNS_CARR ||
PayableCarrOvrList_V1 || carrList || carr
LoadSrvc || PayableCarrierOverride || 4764 || carrList

```

Remove Shipment Example

```

ShipmentKeyInfo_V1 || shipment1 || 100108 || 34165
ShipmentKeyInfoList_V1 || shipments || shipment1
LoadSrvc || RemoveShipment || shipments

```

Remove Shipment Leg Example

```
StrIdList || shpmLegs || 100010183
LoadSrvc || RemoveShipmentLeg || shpmLegs
```

Retrieve Condensed Load Example

```
LoadList_V1 || LoadList_2
StrIdList || LoadIdList_1 || 1841
LoadSrvc || RetrieveCondensedLoad || LoadIdList_1 || FALSE || ||
LoadList_2
```

RetrieveLoads

Place the relevant load IDs into a string ID list. Ensure the returned list (LoadList_V1) is previously defined but empty. If the operation is successful, the error list will be empty and the returned load list will contain at least one item.

The IDs that represent the loads can be reference numbers. If so, set the boolean value to True and the provide the reference type string. As with the RetrievePlanLoads routine, the calculation of charges may occur.

Retrieve Load Example

```
LoadList_V1 || LoadList_2
StrIdList || LoadIdList_1 || 1841
LoadSrvc || RetrieveLoad || LoadIdList_1 || FALSE || || FALSE ||
LoadList_2
```

Retrieve Loads By Date Examples

The following file retrieve load IDs for loads with a load start date between November 10, 1999 and November 13, 1999 inclusive.

```
StrIdList || LoadIdList_Out
LoadSrvc || RetrieveLoadsByDate || RLD_START ||
10/15/1999 || 10/20/1999 || LoadIdList_Out
```

The following file would retrieve load IDs for loads with a load start date between November 9, 1999 and the current date inclusive.

```
StrIdList || LoadIdList_Out
LoadSrvc || RetrieveLoadsByDate || RLD_START ||
9/11/1999 || || LoadIdList_Out
```

Retrieve Loads By Description Example

```
StrIdList || DescList || Test Description 1 || Test Description 2
LoadList_V1 || loads
LoadSrvc || RetrieveLoadsByDesc || DescList || loads
```

Retrieve Payable Carrier Override Example

```
PayableCarrOvrList_V1 || carrList ||
LoadSrvc || RetrievePayableCarrierOverride || 4764 || carrList
```

Retrieve Plan Loads

Place the relevant plan ID directly into this transaction command. Ensure the returned list (`LoadList_V1`) is previously defined but empty. If the operation is successful, the error list will be empty and the returned load list will contain at least one item.

This service includes a boolean value indicating whether charges should be calculated for the planned loads. The error list includes errors returned by this operation which are routed to a file.

Retrieve Plan Loads Example

```
LoadList_V1 || LoadList_1
LoadSrvc || RetrievePlanLoads || 3501 || LoadList_1 || TRUE
```

Retrieve Plan Shipments

Place the relevant plan ID directly into the transaction command for `RetrievePlanShipments`. Ensure the returned list (`ShipmentLegList_V1`) is previously defined but empty. If the operation is successful, the error list will be empty and the returned shipment leg list will contain at least one item.

Retrieve Plan Shipments Example

```
ShipmentList_V1 || ShipmentList_1
LoadSrvc || RetrievePlanShipments || 3501 || ShipmentList_1
```

Retrieve Tender Example

```
StrIdList || IDList || 12483
LoadTenderDataList_V1 || Results
LoadSrvc || RetrieveTender || IDList || Results
```

Return To Open Example

```
LoadPlan_V1 || loadData || 4029 || 1044
LoadPlanList_V1 || myLoads || logadData
LoadSrvc || ReturnToOpen || myLoads
```

The format of `LoadPlan_V1` is:

```
structname || mnemonic name || Load ID (mandatory) || Plan ID (optional)
```

If there is a Plan ID, then `ReturnToOpen` will attempt to attach the load to the plan. If it cannot, the load will not be attached to any plan.

Set All To Planned Example

```
StrIdList || MyPlans || 510181
NamedResultList || MyResults
LoadSrvc || SetAllToPlanned || MyPlans || MyResults
```

Set Load Instruction Example

11803 is the load ID and 28120 is the stop ID:

```
LoadInstruction_V1 || test || testing instructions 2 || || || ||
LoadSrvc || SetLoadInstruction || 11803 || 28120 || test
```


Set Shipment POD Example

```

POD_V1 || SSP1.POD || NormalPOD || 07/17/2001@10:00:00 || || || || ||
||
SEC_V1 || SSP1.SECInfo || POD || 07/17/2001@13:52:00 || Cambridge || ON
|| CAN || 07/27/2001@09:00:00 || 07/29/2001@09:00:00 || bFALSE || 07/30/
2001@09:00:00 || 08/01/2001@09:00:00 || bTRUE || 08/08/2001@12:00:00 ||
08/10/2001@12:00:00 || EN-CARR1 || Clinton || 15 || Landman || ABC-1 ||
bTRUE || SUSP_NOT_SUSPENDED || bFALSE
Memo_V1 || SSP1.Mmo || SetShipmentPOD-Print17-July-2001 ||
SetShipmentPOD-NonPrint17-July-2001
ShipmentUpdateProgress_V1 || SSP1 || RCCAR2 || 25638 || || || || ||
|| 100 || SSP1.Mmo || SSP1.SECInfo || SSP1.POD
ShipmentUpdateProgressList_V1 || SSPList || SSP1
LoadSrvc || SetShipmentPOD || SSPList || SI_ID || FALSE ||

```

Set Stop ETA Example

```

POD_V1 || SSE2.POD || QualityCheck || 07/13/2001@23:52:00 || 200 || 1 ||
|| || ||
POD_V1 || SSE1.POD || WinRunner || 07/13/2001@23:52:00 || 62.0100 || 0 ||
2 || || ||
StopEvent_V1 || SSE2.EventInfo || RETA || 07/13/2001@23:52:00 || 08/13/
2001@23:52:00
StopEvent_V1 || SSE1.EventInfo || RETA || 07/13/2001@23:52:00 || 07/14/
2001@23:52:00
Memo_V1 || SSE2.Mmo || PrintablememoforSSE2 || Non-printablememoforSSE2
Memo_V1 || SSE1.Mmo || PrintablememoforSSE1 || Non-printablememoforSSE1
StopUpdateProgress_V1 || SSE2 || EN-CARR1 || 11675 || || || CAR-DC1 ||
SPT_DC || || SSE2.Mmo || SSE2.EventInfo ||

```

SSE2.POD

```

StopUpdateProgress_V1 || SSE1 || HS || 11647 || || || CRMON1 ||
SPT_CONSIGNEE || || SSE1.Mmo || SSE1.EventInfo ||

```

SSE1.POD

```

StopUpdateProgressList_V1 || SSEList || SSE1 || SSE2
LoadSrvc || SetStopETA || SSEList || FALSE ||

```

Set Stop POD Example

```

POD_V1 || SLP1.POD || NormalPOD || 07/13/2001@23:52:00 || 200 || 1 || 0
|| || ||
StopEvent_V1 || SLP1.EventInfo || POD || 07/13/2001@23:52:00 || 07/14/
2001@23:52:00
Memo_V1 || SLP1.Mmo || PrintablePODmemo || Non-printablePODmemo
StopUpdateProgress_V1 || SLP1 || EN-CARR1 || 11675 || || || CAR-DC1 ||
SPT_DC || || SLP1.Mmo || SLP1.EventInfo ||

```

SLP1.POD

```

StopUpdateProgressList_V1 || SLPList || SLP1
LoadSrvc || SetStopPOD || SLPList || FALSE ||

```

Set Stop To Delivered Example

```

POD_V1 || SSTD1.POD || QualityCheck || 07/13/2001@23:52:00 || 62.0100 ||
0 || 2 || || ||

```

```

StopEvent_V1 || SSTD1.EventInfo || DLVD || 07/13/2001@23:52:00 || 07/14/
2001@23:52:00
Memo_V1 || SSTD1.Mmo || PrintablememoforSSTD1 || Non-
printablememoforSSTD1
StopUpdateProgress_V1 || SSTD1 || HS || 11647 || || || CRMON1 ||
SPT_CONSIGNEE || || SSTD1.Mmo || SSTD1.EventInfo ||

SSTD1.POD
StopUpdateProgressList_V1 || SSTDLIST || SSTD1
LoadSrvc || SetStopToDelivered || SSTDLIST || FALSE ||

```

Set To Planned Example

```

StrIdList || MyLoads || 804465
LoadSrvc || SetToPlanned || MyLoads

```

ShipmentUpdateProgress

```

SEC_V1 || SUP1.SECInfo || TRCN || 07/17/2001@13:52:00 || Cambridge || ON
|| CAN || 07/27/2001@09:00:00 || 07/29/2001@09:00:00 || bFALSE || 07/30/
2001@09:00:00 || 08/01/2001@09:00:00 || bTRUE || 08/08/2001@12:00:00 ||
08/10/2001@12:00:00 || EN-CARR1 || Clinton || 15 || Landman || ABC-1 ||
bTRUE || SUSP_NOT_SUSPENDED || bFALSE
Memo_V1 || SUP1.Mmo || Print17-July-2001 || NonPrint17-July-2001
ShipmentUpdateProgress_V1 || SUP1 || RCCAR22 || 25638 || || || || ||
|| 100 || SUP1.Mmo || SUP1.SECInfo ||
ShipmentUpdateProgressList_V1 || SUPList || SUP1
LoadSrvc || ShipmentUpdateProgress || SUPList || SI_ID || FALSE ||

```

Stop Confirm Example

```

StopConfirmData_V1 || StopConfData || 1541 || 1 || || || || || || || bNULL
LoadSrvc || StopConfirm || StopConfData ||

```

Tender Example

```

LoadTenderData_V1 || Load13004 || 13004 || || || || ||
LoadTenderDataList_V1 || Requests || Load13004
LoadTenderDataList_V1 || Results
LoadSrvc || Tender || Requests || || || Results

```

Tender Accept Example

```

TenderAcceptLoadTenderContact_V1 || Carr_Notn.SndrLdTdrCntc || LDTDRCNTC_SEN
DER || CU-BOTHP2 || zh || 444-444-4441 || 444-444-4442 || 444-444-
4443 || CCBA@I2.COM || WWW.CCBA.CA
LoadTenderContact_V1 || Carr_Notn.CarrLdTdrCntc || LDTDRCNTC_CARRIER || TCDOM1
|| en || 333-333-3331 || 333-333-3332 || 333-333-3333 || DOM3@I2.COM || WWW.DOM3.CA
NotificationData_V1 || Carr_Notn || bFALSE || 11/11/2001@11:11:11
Memo_V1 || ADMemo || PASSED - Test 130.1.1 TenderAccept NTE test - tender
accept || (private note about Tender Accept NTE)
LoadTenderData_V1 || LoadId || 11643 || NTE-CAR || || TL || || || || || Accept
User || S_TENDERED || || || ADMemo || || || || || || || || || || || || ||
LoadTenderDataList_V1 || requests || LoadId
LoadSrvc || TenderAccept || requests

```

Tender Cancel Example

```
TenderCancelLoadTenderContact_V1 | Carr_Notn.SndrLdTdrCntc | LDTDRCNTC_SEN
DER | CU-BOTHP2 | zh | 444-444-4441 | 444-444-4442 | 444-444-
4443 | CCBA@I2.COM | WWW.CCBA.CA
LoadTenderContact_V1 | Carr_Notn.CarrLdTdrCntc | LDTDRCNTC_CARRIER | TCDOM1
| en | 333-333-3331 | 333-333-3332 | 333-333-3333 | DOM3@I2.COM | WWW.DOM3.CA
NotificationData_V1 | notn | bFALSE | 11/11/2001@11:11:11
Memo_V1 | ADMemo | CANCELLED | (private note about why it was
cancelled)
LoadTenderData_V1 | LoadId | 5438 | TNDR CNC1 | | ADS1 | | | | | | |
S_TENDERED | | | ADMemo | | notn | | | | | | | | | | | | | |
LoadTenderDataList_V1 | requests | LoadId
LoadSrvc | TenderCancel | requests
```

Tender Reject Example

```
TenderRejectLoadTenderContact_V1 | Carr_Notn.SndrLdTdrCntc | LDTDRCNTC_SEN
DER | CU-BOTHP2 | zh | 444-444-4441 | 444-444-4442 | 444-444-
4443 | CCBA@I2.COM | WWW.CCBA.CA
LoadTenderContact_V1 | Carr_Notn.CarrLdTdrCntc | LDTDRCNTC_CARRIER | TCDOM1
| en | 333-333-3331 | 333-333-3332 | 333-333-3333 | DOM3@I2.COM | WWW.DOM3.CA
NotificationData_V1 | notn | bFALSE | 11/11/2001@11:11:11
Memo_V1 | ADMemo | REJECTED | (private note about why it was REJECTED)
LoadTenderData_V1 | LoadId | 5348 | TNDRREJ1 | | ADS1 | | | *DFT | | | |
S_TENDERED | | | ADMemo | | | | | | | | | | | | | | | notn
LoadTenderDataList_V1 | requests | LoadId
LoadSrvc | TenderReject | requests
```

Trigger Load Event

Place the relevant load IDs into a `StrIdList`. If the operation is successful, the error list will be empty. All load IDs must exist or the operation will fail.

Trigger Load Event Example

```
StrIdList | IdList_1 | 1841
SEC_V1 | SEC | ABCDE | 02/16/1998@11:30:00
LoadSrvc | TriggerLoadEvent | IdList_1 | LOEV_PLAN | SEC | FALSE
```

Rate Quotation Services

The rate quotation services find a list of valid lanes for shipping and obtain the charge information for each lane.

Examples**Get Possible Charges Example**

```
ChargeCriteria_V1 | i2_ChargeCriteria | ROZ | LTL | 001 | SCARBORO
| | | BRAMPTON
ChargeDetailList_V1 | i2_ChargeDetails
RateQuotationSrvc | GetPossibleCharges | i2_ChargeCriteria |
i2_ChargeDetails
```

Rate Quotation Example

```

WeightByFreightClass_V1 || w_class || 600 || 50
WeightByFreightClass_V1 || w_class2 || 400 || 55
WeightByFreightClassList_V1 || w_classes || w_class || w_class2
#ChargeDetail_V1 || c_detail || ItemId || ChargeCode || FreightClass ||
RatedAsFC || 11 || CHL_NULL || 10 || 10 || 10 || 10 || 10 || 10000 || 20
ChargeDetailList_V1 || c_details
FreightDetail_V1 || fd || w_classes || 100 || 5500 || 5000 || 0 || 3 ||
3000
RateCriteria_V1 || rate || bFALSE || SPT_LA || IYLA || || || || CAN ||
SPT_CONSIGNEE || IYCON || || || || IYCUST || || || || bFALSE || IYD
|| *DFT || 12/05/1998@10:15:00 || 12/12/1998@10:15:00 || 12/05/
1998@10:15:00 || 12/12/1998@10:15:00 || UMS_IMPERIAL || UMW_LB ||
UML_INCH || fd || || c_details || CAD || SC_COST || 1 || 100
ValidLaneList_V1 || vlanes
Details || err
RateQuotationSrvc || RateQuotationService || rate || TRUE || vlanes ||
err

```

Shipment Services

The shipment services retrieve, confirm, and trigger shipments.

Examples**Attach to Plan Example**

```

StrIdList || shpmLegIDs || 123456789 || 123456777
ShipmentSrvc || AttachToPlan || 1101 || ShpmLegIDs

```

Cancel Shipment by Tracking Number and Division Example

```

Shipment_V2 || Shpm1 || || || || || || || || MA || || || || || || || || || ||
|| || || || || MF1002-3 || || || || || || || || || || || || || || || || || ||
|| || || || || || || || || || || || || || || || || || || || || || || || || ||
ShipmentList_V2 || CanList || Shpm1
ShipmentOrderEntrySrvc || CancelShipment || CanList || SI_TRKG_NUM_DIV
||

```

Payable Carrier Override Example

```

PayableCarrOvr_V1 || carr || BULK || || || PAY_CARR_TRNS_CARR ||
PayableCarrOvrList_V1 || carrList || carr
ShipmentSrvc || PayableCarrierOverride || 100022395 || carrList

```

Responsible Customer Override Example

```

RspbCustOvr_V1 || cust || BULK || || || RSPB_CUST_ORIG_SHPG_LOC ||
RspbCustOvrList_V1 || custList || cust
ShipmentSrvc || ResponsibleCustomerOverride || 000018195 || custList

```

Retrieve Payable Carrier Override Example

```

PayableCarrOvrList_V1 || carrList ||
ShipmentSrvc || RetrievePayableCarrierOverride || 100022395 || carrList

```

Retrieve Responsible Customer Override Example

```
RspbCustOvrList_V1 || custList || cust
ShipmentSrvc || RetrieveResponsibleCustomerOverride || 000018385 ||
custList
```

Retrieve Full Shipment by ID Example

```
Shipment_V2 || Shpml || 20935 || | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
ShipmentList_V2 || RetList || Shpml
ShipmentList_V2 || OutSHList
ShipmentOrderEntrySrvc || RetrieveShipment || RetList || SI_ID ||
RES_SHIPMENT_FULL || TRUE || OutSHList
```

Retrieve Full Shipment by References and Division Example

```
RefNumber_V1 || RefNum1 || 10S || MF1002-1
RefNumberList_V1 || RefNums || RefNum1
Shipment_V2 || Shpml || | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
ShipmentList_V2 || RetList || Shpml
ShipmentList_V2 || OutSHList
ShipmentOrderEntrySrvc || RetrieveShipment || RetList || SI_RFRC_DIV ||
RES_SHIPMENT_FULL || TRUE || OutSHList
```

Retrieve Full Shipment by Tracking Number Example

```
Shipment_V2 || Shpml || | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
ShipmentList_V2 || RetList || Shpml
ShipmentList_V2 || OutSHList
ShipmentOrderEntrySrvc || RetrieveShipment || RetList || SI_TRKG_NUM ||
RES_SHIPMENT_FULL || TRUE || OutSHList
```

Retrieve Shipment ID by References Example

```
RefNumber_V1 || RefNum1 || MB || SHP-RS10.1.1
RefNumber_V1 || RefNum2 || 10S || SHP-RS10.1.1
RefNumberList_V1 || RefNums || RefNum1 || RefNum2
Shipment_V2 || Shpml || | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
ShipmentList_V2 || RetList || Shpml
StrIdList || OutSHList
ShipmentOrderEntrySrvc || RetrieveShipment || RetList || SI_RFRC ||
RES_NONE || TRUE || OutSHList
```

Retrieve Shipment Legs Example

```
StrIdList || ShipList1 || 29834
```

```
ShipmentLegList_V1 || Ship_list
ShipmentSrvc || RetrieveShipmentLegs || ShipList1 || Ship_list
```

Retrieve Shipment Legs By ID Example

```
StrIdList || mylist || 63773
ShipmentLegList_V1 || result
ShipmentSrvc || RetrieveShipmentLegsById || mylist || result
```

Return Shipment Leg From Planned Example

```
StrIdList || shpmLegId2s || 8888 || 9999 || 6666
ShipmentSrvc || ReturnShipmentLegFromPlanned || shpmLegId2s
```

Select For Optimization Example

```
SelectionCriteria_V1 || sc || 1305 || || || || || || || ||
ShipmentSrvc || SelectForOptimization || sc || || FALSE ||
```

Send To Optimzer Example

```
ShipmentSrvc || SendToOptimizer || 1305 || || FALSE
```

Set To Planned Example

```
StrIdList || MyShipments || 63605 || 63606
ShipmentSrvc || SetToPlanned || MyShipments
```

Shipment Composition Change Example

```
WeightByFreightClass_V1 || WgtByFCs_1 || 22 || 22
WeightByFreightClass_V1 || WgtByFCs_2 || 33 || 33
WeightByFreightClass_V1 || WgtByFCs_3 || 45 || 45
WeightByFreightClassList_V1 || WgtByFCs || WgtByFCs_1 || WgtByFCs_2 ||
WgtByFCs_3
Component_V1 || Comps_1 || 1 || 02/04/1999@08:08:00 || 02/04/
1999@18:45:45 || 02/06/1999@09:45:45 || 02/06/1999@18:45:45 || CHD || 33
|| 33 || 33 || 33 || 300 || 00 || 00
SKID || GROC || || || || || || || || bTRUE || WgtByFCs || bTRUE
ComponentList_V1 || Comps || Comps_1
ShipmentCompChangeData_V1 || i2_ShipmentCompChangeDataList_1 ||
000000008121
ShipmentSrvc || ShipmentCompositionChange ||
i2_ShipmentCompChangeDataList_1 || eADD
```

Shipment Confirm Example

```
ShipmentConfirmData_V1 || i2_ShipmentConfirmDataList_1 || 29374 || || ||
|| || bFALSE || bFALSE || bTRUE
ShipmentConfirmDataList_V1 || i2_ShipmentConfirmDataList ||
i2_ShipmentConfirmDataList_1
ShipmentSrvc || ShipmentConfirm || i2_ShipmentConfirmDataList
```

Trigger Shipment Leg Event – Release Example

```
StrIdList || Shipment_id || 28927
SEC_V1 || RELS || RLS || 10/09/1998@16:00:00
ShipmentSrvc || TriggerShipmentLegEvent || Shipment_id || SLEV_RELEASE
|| RELS
```

Trigger Shipment Leg Event – Unsuspend Example

```

StrIdList || Shipment_id || 100033534
SEC_V1 || UNSUSPEND || UNSUSPEND || 08/08/1999@16:00:00
SECList_V1 || SEC_list || UNSUSPEND
ShipmentSrvc || TriggerShipmentLegEvent || Shipment_id || SLEV_UNSPEND
|| UNSUSPEND ||

```

Update Shipment Legs Example

```

ShipmentLeg_V1 || result.[0] || 100028645 || || || || || || || || || ||
|| || || || || || || || || || || || || || || || || || || || ||
|| || || || || || || || || || || || || || || || || || || || ||
|| || || || || || || || || || 31 || || || || || || || || || ||

ShipmentLeg_V1 || result.[1] || 100027945 || || || || || || || || || ||
|| || || || || || || || || || || || || || || || || || || || ||
|| || || || || || || || || || 21 || || || || || || || || || ||
ShipmentLegList_V1 || result || result.[0] || result.[1]
ShipmentSrvc || UpdateShipmentLegs || result

```

Shipment Order Entry Services

Use the shipment order entry services to create, retrieve, update, and delete shipments.

Examples**Create Shipment Example**

The following example creates a shipment that has one container, two items, and no shipment reference numbers.

```

UMsr_V1 || OutSHList.[0].UMsr || UMS IMPERIAL || UMW_LB || UML_FT ||
UMD_MILES
UMsr_V1 || OutSHList.[0].TariffUMsr || UMS IMPERIAL || UMW_LB || UML_FT
|| UMD_MILES
ShippingInfo_V1 || OutSHList.[0].ShipInfo || 620 || 2000 || || || 600 ||
20 || 0 || 2 || 0
Address_V1 || OutSHList.[0].FromAddr || 5 || HAHN PL || 316 || TORONTO ||
ON || CAN || M5A4G1 ||
Address_V1 || OutSHList.[0].ToAddr || 6161 || BATHURST ST || 1202 ||
NORTH YORK || ON || CAN || M2R1Z5 ||
Memo_V1 || OutSHList.[0].Memo || ||
ShippingInfo_V1 || OutSHList.[0].Containers.[0].ShipInfo || 310 || 1000
|| || || 300 || 10 || 0 || 1 || 0
CntrOrtn_V1 || OutSHList.[0].Containers.[0].CntrOrtns.[0] || OR_LWH ||
ORALW_UNDEFINED || 0
CntrOrtn_V1 || OutSHList.[0].Containers.[0].CntrOrtns.[1] || OR_WLH ||
ORALW_UNDEFINED || 0
CntrOrtn_V1 || OutSHList.[0].Containers.[0].CntrOrtns.[2] || OR_LHW ||
ORALW_UNDEFINED || 0
CntrOrtn_V1 || OutSHList.[0].Containers.[0].CntrOrtns.[3] || OR_HLW ||
ORALW_UNDEFINED || 0

```

```

CntrOrtn_V1 || OutSHList.[0].Containers.[0].CntrOrtns.[4] || OR_WHL ||
ORALW_UNDEFINED || 0
CntrOrtn_V1 || OutSHList.[0].Containers.[0].CntrOrtns.[5] || OR_HWL ||
ORALW_UNDEFINED || 0
CntrOrtnList_V1 || OutSHList.[0].Containers.[0].CntrOrtns ||
OutSHList.[0].Containers.[0].CntrOrtns.[0] ||
OutSHList.[0].Containers.[0].CntrOrtns.[1] ||

OutSHList.[0].Containers.[0].CntrOrtns.[2] ||
OutSHList.[0].Containers.[0].CntrOrtns.[3] ||
OutSHList.[0].Containers.[0].CntrOrtns.[4] ||

OutSHList.[0].Containers.[0].CntrOrtns.[5]
ShipmentItem_V1 || OutSHList.[0].Containers.[0].ShpmItems.[0] || || ||
|| 1234567890 || 1 || 100 || 100 || 100 || 2000 || 123456789012 || || ||
IS_EXTERNAL_API ||

SLC_NOT_REQRD || S_NULL || 135686 || 71763 || IYCUST1 || 50 || || || CAN
|| 1,1,1 || bTRUE
ShipmentItemList_V1 || OutSHList.[0].Containers.[0].ShpmItems ||
OutSHList.[0].Containers.[0].ShpmItems.[0]
Container_V1 || OutSHList.[0].Containers.[0] || || SKID || 1 || 10 || 10
|| 10 || OutSHList.[0].Containers.[0].ShipInfo || Skid 1 || 0 || 0 || 0
|| DIMN_NULL || || 0 || 1 || 0 || UT_SKIDS ||

9999999 || MF-717631234567 || || 0 || TRNSSRC_API || SPT_NULL || ||
bFALSE || OutSHList.[0].Containers.[0].CntrOrtns || bTRUE || || bFALSE
||

OutSHList.[0].Containers.[0].ShpmItems || bTRUE ||
OutSHList.[0].Containers.[0].RfrncNums
ContainerList_V1 || OutSHList.[0].Containers ||
OutSHList.[0].Containers.[0]
ShipmentThruPoint_V1 || OutSHList.[0].ThruPoints.[0] || STPT_HUB ||
IYKINGSTON || IYCAR || LTL || *DFT
ShipmentThruPoint_V1 || OutSHList.[0].ThruPoints.[1] || STPT_DC ||
MILTON || || ||
ShipmentThruPointList_V1 || OutSHList.[0].ThruPoints ||
OutSHList.[0].ThruPoints.[0] || OutSHList.[0].ThruPoints.[1]
Shipment_V2 || Shpm1 || || || SI-CUST5 || SI-1 || S1 || *DFT || *DFT ||
|| IS_EXTERNAL_API || || bFALSE || TOM_ITEM_LEVEL_DETAIL || KC || || ||
|| CSE_NULL || || || || ||

FT_PRE_PAID || SI-CUST1 || bFALSE || bFALSE || DRY || || 0 ||
OutSHList.[0].Memo || SFT_LA || SI-LA1 || SHIPMENT INTERFACE - LOAD-AT 1
|| OutSHList.[0].FromAddr ||

bFALSE || STT_CONSIGNEE || SI-CN1 || SHIPMENT INTERFACE - CONSIGNEE 1 ||
OutSHList.[0].ToAddr || bFALSE || 02/02/2000@08:00:00 || 02/02/
2000@10:00:00 ||

02/02/2000@09:00:00 || 02/02/2000@10:01:00 || bFALSE || *DFT || *DFT ||
|| bFALSE || 0 || *DFT || *DFT || 08/02/2000@16:18:29 || 08/14/
2000@17:00:25 || || ||

```



```

S_PROCESSED || S_F_INELIGIBLE || OutSHList.[0].UMsr ||
OutSHList.[0].TariffUMsr || OutSHList.[0].ShipInfo || bFALSE || bFALSE
|| bTRUE || bTRUE || bTRUE || 10000 ||

45000 || 4500 || bFALSE || bFALSE || || SI-CUST5 || bTRUE || || bFALSE ||
OutSHList.[0].Containers || bTRUE ||
ShipmentList_V2 || OutSHList || Shpml
ShipmentList_V2 || OutputSHList
ShipmentOrderEntrySrvc || CreateShipment || OutSHList || FALSE || bFALSE
|| bFALSE || RES_SHIPMENT_FULL || OutputSHList

```

Delete Shipment by ID Example

```

Shipment_V2 || Shpml || 18585 || || || || || || || || || || || || || || || || ||
|| || || || || || || || || || || || || || || || || || || || || || || || || || || ||
|| || || || || || || || || || || || || || || || || || || || || || || || || || || ||
|| || || || || || || || || || || || || || || || || || || || || || || || || || || ||
Shipment_V2 || Shpm2 || 18595 || || || || || || || || || || || || || || || || ||
|| || || || || || || || || || || || || || || || || || || || || || || || || || || ||
|| || || || || || || || || || || || || || || || || || || || || || || || || || || ||
ShipmentList_V2 || DelList || Shpml || Shpm2
#StrIdList || DelList || 18585 || 18595
ShipmentOrderEntrySrvc || DeleteShipment || DelList || SI_ID ||

```

Delete Shipment by Number Example

```

Shipment_V2 || Shpml || || SH-000018615 || || || || || || || || || || || || || || || ||
|| || || || || || || || || || || || || || || || || || || || || || || || || || || ||
|| || || || || || || || || || || || || || || || || || || || || || || || || || || ||
Shipment_V2 || Shpm2 || || SH-000018616 || || || || || || || || || || || || || || || ||
|| || || || || || || || || || || || || || || || || || || || || || || || || || || ||
|| || || || || || || || || || || || || || || || || || || || || || || || || || || ||
ShipmentList_V2 || DelList || Shpml || Shpm2
ShipmentOrderEntrySrvc || DeleteShipment || DelList || SI_NUM ||

```

Delete Shipment by Reference and Division Example

```

RefNumber_V1 || RefNum1 || 10S || 10S-0816-02
RefNumberList_V1 || RefNums || RefNum1
Shipment_V2 || Shpml || || || || || || || || || || || || || || || || || || || || ||
|| || || || || || || || || || || || || || || || || || || || || || || || || || || ||
|| || || || || || bFALSE || RefNums || || || ||
ShipmentList_V2 || DelList || Shpml
ShipmentOrderEntrySrvc || DeleteShipment || DelList || SI_RFRC_DIV ||
10S

```

Delete Shipment by Reference Example

```

RefNumber_V1 || RefNum1 || 10S || 10S-0816-02
RefNumberList_V1 || RefNums || RefNum1
Shipment_V2 || Shpml || || || || || || || || *DFT || || || || || || || || || || ||
|| || || || || || || || || || || || || || || || || || || || || || || || || || || ||

```



```

ShippingInfo_V1 || OutSHList.[0].Containers.[0].ShipInfo || 310 || 1000
|| || || 300 || 10 || 0 || 1 || 0
CntrOrtn_V1 || OutSHList.[0].Containers.[0].CntrOrtns.[0] || OR_LWH ||
ORALW_UNDEFINED || 0
CntrOrtn_V1 || OutSHList.[0].Containers.[0].CntrOrtns.[1] || OR_WLH ||
ORALW_UNDEFINED || 0
CntrOrtn_V1 || OutSHList.[0].Containers.[0].CntrOrtns.[2] || OR_LHW ||
ORALW_UNDEFINED || 0
CntrOrtn_V1 || OutSHList.[0].Containers.[0].CntrOrtns.[3] || OR_HLW ||
ORALW_UNDEFINED || 0
CntrOrtn_V1 || OutSHList.[0].Containers.[0].CntrOrtns.[4] || OR_WHL ||
ORALW_UNDEFINED || 0
CntrOrtn_V1 || OutSHList.[0].Containers.[0].CntrOrtns.[5] || OR_HWL ||
ORALW_UNDEFINED || 0
CntrOrtnList_V1 || OutSHList.[0].Containers.[0].CntrOrtns ||
OutSHList.[0].Containers.[0].CntrOrtns.[0] ||
OutSHList.[0].Containers.[0].CntrOrtns.[1] ||
OutSHList.[0].Containers.[0].CntrOrtns.[2] ||
OutSHList.[0].Containers.[0].CntrOrtns.[3] ||
OutSHList.[0].Containers.[0].CntrOrtns.[4] ||
OutSHList.[0].Containers.[0].CntrOrtns.[5]
ShipmentItem_V1 || OutSHList.[0].Containers.[0].ShpmItems.[0] || || ||
|| 12345 || 1 || 100 || 100 || 100 || 2000 || 1234567890 || || ||
IS_EXTERNAL_API || SLC_NOT_REQD || S_NULL || 135686 || 71763 || IYCUST1
|| 50 || || || CAN || 1,1,1 || bTRUE
ShipmentItemList_V1 || OutSHList.[0].Containers.[0].ShpmItems ||
OutSHList.[0].Containers.[0].ShpmItems.[0]
Container_V1 || OutSHList.[0].Containers.[0] || || SKID || 1 || 10 || 10
|| 10 || OutSHList.[0].Containers.[0].ShipInfo || Skid 1 || 0 || 0 || 0
|| DIMN_NULL || || 0 || 1 || 0 || UT_SKIDS || 9999999 || MF-717631234567
|| || 0 || TRNSSRC_API || SPT_NULL || || bFALSE ||
OutSHList.[0].Containers.[0].CntrOrtns || bTRUE || || bFALSE ||
OutSHList.[0].Containers.[0].ShpmItems || bTRUE ||
OutSHList.[0].Containers.[0].RfrncNums
ContainerList_V1 || OutSHList.[0].Containers ||
OutSHList.[0].Containers.[0]
ShipmentThruPoint_V1 || OutSHList.[0].ThruPoints.[0] || STPT_HUB ||
IYKINGSTON || IYCAR || LTL || *DFT
ShipmentThruPoint_V1 || OutSHList.[0].ThruPoints.[1] || STPT_DC ||
MILTON || || ||
ShipmentThruPointList_V1 || OutSHList.[0].ThruPoints ||
OutSHList.[0].ThruPoints.[0] || OutSHList.[0].ThruPoints.[1]
Shipment_V2 || Shpm1 || 33329 || || SI-CUST5 || SI-1 || S1 || *DFT ||
*DFT || || IS_EXTERNAL_API || || bFALSE || TOM_ITEM_LEVEL_DETAIL || KC ||
|| || || CSE_NULL || || || || FT_PRE_PAID || SI-CUST1 || bFALSE ||
bFALSE || DRY || || 0 || OutSHList.[0].Memo || SFT_LA || SI-LA1 ||
SHIPMENT INTERFACE - LOAD-AT 1 || OutSHList.[0].FromAddr || bFALSE ||
STT_CONSIGNEE || SI-CN1 || SHIPMENT INTERFACE - CONSIGNEE 1 ||
OutSHList.[0].ToAddr || bFALSE || 02/02/2000@08:00:00 || 02/02/
2000@10:00:00 || 02/02/2000@09:00:00 || 02/02/2000@10:01:00 || bFALSE ||
*DFT || *DFT || || bFALSE || 0 || *DFT || *DFT || 08/02/2000@16:18:29 ||
08/14/2000@17:00:25 || || || S_PROCESSED || S_F_INELIGIBLE ||
OutSHList.[0].UMsr || OutSHList.[0].TariffUMsr || OutSHList.[0].ShipInfo
|| bFALSE || bFALSE || bTRUE || bTRUE || bTRUE || 10000 || 45000 || 4500

```

```
|| bFALSE || bFALSE || | SI-CUST5 || bTRUE || | | bFALSE ||
OutSHList.[0].Containers || bTRUE ||
ShipmentList_V2 || OutSHList || Shpml
ShipmentList_V2 || OutputSHList
ShipmentOrderEntrySrvc || UpdateShipment || OutSHList || SI_ID || | |
FALSE || FALSE || FALSE || FALSE
```

Tariff Services

Use the tariff services to create, retrieve, update, and delete Transportation Manager tariffs.

Examples

Create Tariff Example

```
StrIdList || TarIDList_Out
Memo_V1 || NULL_Memo_V1 || ""
Tariff_V1 || TFF_CD_1 || | TFF_CD_0000001 || "Tariff Create Test" ||
Default || TFFAT_GENERIC || | SCG1 || Prj_Cd || 06/23/1998 || 07/23/1998
|| USD || UMS_METRIC || UMW_TN || UML_FT || ROP_GEOGRAPHY || bNULL || | |
|| | | | | FIPM_AUTO_PAY || | | | | | | | | | | bTRUE || | | bTRUE ||
|| bTRUE || | | bTRUE || | | bTRUE || | | bTRUE
TariffList_V1 || TstListInvalidFields || TFF_CD_1
TariffSrvc || CreateTariff || TstListInvalidFields || TarIDList_Out
```

Create Tariff Entity Example

```
LaneAssc_V1 || TSTLC001 || 3185 || LTL || TRTD || CAMBR || BC14 || | | MI-
HUB || | | BC14 || bFALSE || TEMP || | | ON || CAN || BC || CAN || | | | |
|| bFALSE
LaneAsscList_V1 || LaneList || TSTLC001
TariffSrvc || CreateTariffEntity || LaneList
```

Delete Tariff Example

```
StrIdList || DelList || 16446
TariffSrvc || DeleteTariff || DelList
```

Delete Tariff Entity Example

```
TffSrvc_V1 || TffSrvcTest1 || 16056 || 4601 || | | | | | | | | | | | | | | |
|| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|| | | | | | | | | | | | | | | | | | | | | | | bTRUE || | | bTRUE
TffSrvcList_V1 || TstList || TffSrvcTest1
TariffSrvc || DeleteTariffEntity || TstList
```

Retrieve Tariff Example

```
TariffList_V1 || aList
StrIdList || IdList1 || 16404
StrIdList || IdList1 || 2818
TariffSrvc || RetrieveTariff || IdList1 || TRUE || TRUE || TRUE || TRUE
|| aList
```

Retrieve Tariff Entity Example

```

RateCd_V1 || TstRateCd11 || 16056 || RtCd11 || Rate Code 11 || || || ||
|| bTRUE
RateCdList_V1 || RtCdList || TstRateCd11
StrIdList || TffIdList1 || 16224
TariffSrvc || RetrieveTariffEntity || TffIdList1 || RtCdList

```

Update Tariff Example

```

StrIdList || TarIDList_Out
Memo_V1 || Mmo || sample mem01
TffSrvc_V1 || TffSrvcTest1 || 11638 || LTL || Updated Service2 TFF46F1 ||
|| GL01 || || || || || || || || || bFALSE || bFALSE || || || || bFALSE ||
Default || bFALSE || || CarrBOL || bFALSE || CarrMfst || CustInvSum ||
|| || || || || || || || || || || || || || || || || || || || bTRUE ||
|| bTRUE
TffSrvcList_V1 || TstSrvcList || TffSrvcTest1
Tariff_V1 || TFF_CD_46 || 11638 || || "Tariff Update2 Test" || ZOR || ||
|| SCG1 || Prj_Cd || 06/23/1998 || 11/30/1998 || || UMS_METRIC || UMW_TN
|| UML_FT || ROP_GEOGRAPHY || bNULL || || || || || || || || || || ||
|| || Mmo bFALSE || TstSrvcList || bTRUE || || bTRUE || || bTRUE || ||
bTRUE || || bTRUE
TariffList_V1 || TstUpdListInvalidFields || TFF_CD_46
TariffSrvc || UpdateTariff || TstUpdListInvalidFields || TarIDList_Out

```

Update Tariff Entity Example

```

LaneAssc_V1 || TSTLU02 || 15444 || LTL || TRTD || ALLUSA || ALLCAN || ||
|| || FINAL || bFALSE || TEMP || || ON || CAN || BC || CAN || || || ||
bFALSE
LaneAsscList_V1 || LaneList || TSTLU02
TariffSrvc || UpdateTariffEntity || LaneList

```

Tariff Entity Services – Service Equipment

You can use the create, delete, retrieve, and update tariff entity services for service equipment.

Examples**Create Service Equipment Example**

```

Rstc_V1 || 1234Rstc || || || 1234 Restriction Template || 9999 || 0 ||
99999.0000 || 41000.0000 || 0 || 9999.000 || 9999.000 || 9999.000 ||
2400.0000 || 0 || 9999.000 || 99 || 999 || 0 || 999 || 0 || 999 || 0 ||
99999999999.00 || 0 || 9 || 9999 || 9999 || 9 || 9999 || 99.00 ||
99999999999.00 || 0
TffSrvcEqmt_V1 || aList.[0].TffSrvc.[0].TffSrvcEqmt.[0] || || 1234 ||
23741 || LTL || 4 || || || || || || bTRUE || 1234Rstc
TffSrvcEqmtList_V1 || aList.[0].TffSrvc.[0].TffSrvcEqmt ||
aList.[0].TffSrvc.[0].TffSrvcEqmt.[0]
TariffSrvc || CreateTariffEntity || aList.[0].TffSrvc.[0].TffSrvcEqmt

```

Delete Service Equipment Example

```
TffSrvcEqmt_V1 || SrvcEqmt0 || || QAM || 22197 || TL || 4 || || || || || ||
||
TffSrvcEqmtList_V1 || SrvcEqmtList || SrvcEqmt0
TariffSrvc || DeleteTariffEntity || SrvcEqmtList
```

Retrieve Service Equipment Example

```
TffSrvcEqmt_V1 || SrvcEqmt0 || || QAM || 23741 || LTL || 2 || || || || || ||
|| bFALSE
TffSrvcEqmtList_V1 || SrvcEqmtList || SrvcEqmt0
StrIdList || IdList1 || 23741
TariffSrvc || RetrieveTariffEntity || IdList1 || SrvcEqmtList
```

Update Service Equipment Example

```
Rstc_V1 || 1234Rstc || || || 1234 New Restriction Template || 9999 || 0
|| 99999.0000 || 41000.0000 || 0 || 9999.000 || 9999.000 || 9999.000 ||
2400.0000 || 0 || 9999.000 || 99 || 999 || 0 || 999 || 0 || 999 || 0 ||
999999999999.00 || 0 || 9 || 9999 || 9999 || 9 || 9999 || 99.00 ||
999999999999.00 || 0
TffSrvcEqmt_V1 || aList.[0].TffSrvc.[0].TffSrvcEqmt.[0] || || 1234 ||
23741 || LTL || 5 || || || || || || bTRUE || 1234Rstc
TffSrvcEqmtList_V1 || aList.[0].TffSrvc.[0].TffSrvcEqmt ||
aList.[0].TffSrvc.[0].TffSrvcEqmt.[0]
TariffSrvc || UpdateTariffEntity || aList.[0].TffSrvc.[0].TffSrvcEqmt
```

Transport Order Services

The transport order services send data to and from your system in a data structure format. Define the data structures in the input files as single line entries using the following format.

```
Data_Structure_Name || Mnemonic || Value || Value || Value
...
```

Data_Structure_Name is the type of transport order structure object you want to create. It is one of the following:

- Component_V1
- CompRefNumber_V1
- ItineraryStep_V1
- SEC_V1
- Shipment_V1
- TransportOrder_V1
- WeightByFreightClass_V1

Data Structures

To do a transport order transaction, package each defined data structure inside a data structure list. A data structure list is defined as follows.

```
Data_Structure_List || Mnemonic // Value || Value || Value
```

`Data_Structure_List` is the type of sequence list you are creating. The supported data structure list types that are relevant to transport orders are:

- `ComponentList_V1`
- `CompRefNumber_V1`
- `ItineraryStepList_V1`
- `SECList_V1`
- `ShipmentList_V1`
- `TransportOrderList_V1`
- `WeightByFreightClassList_V1`

The transport order service has the following characteristics:

- `mnemonic` is a tag
- the values that follow are specific to the data structure object you are defining
- you can omit the specification of a data field by leaving the field blank
- a data structure can contain other data structures or lists

The data values for a list are the mnemonics for the data structures which indicate the list you are creating. You do not need to predefine the mnemonics.

Define the data structures before referring to them in a data transfer operation. Otherwise, the attempted data transfer will fail.

If you try to insert a data structure that is inconsistent with its type, it sends an error to the error file. For example, adding a transport order to a list of components generates an error.

Data Transfer

In the transport order transaction services, `TransportOrderSrvc` means that you are defining a transport order service transaction. The next field is the operation.

The appropriate information is retrieved and bundled. You use a list to transfer data in bulk between your system and the server. The list is indicated by its appropriate mnemonic. The list type for the mnemonic is checked for the given operation. If there is a mismatch between the two, an error is sent to the error output file and the operation stops.

Transport service transactions can create a list of errors from the server side. You do not need to define the error list does before using it in any of the services. This error list, and the returned transport order list in the retrieve transport service, are the only types of lists that can be undefined before using them.

You can control the destination of the retrieved data or the error list data. (See “Destination of Retrieved Data and Errors” on page 236.)

Examples

CreateTO

Define the top-level data structure along with the supporting aggregate data structures. (For additional details, see “Create” on page 237.)

Create TO Example

```
Address_V1 || FromAddr || | | | | | |
Address_V1 || ToAddr || | | | | | |
Memo_V1 || Mmo
WeightByFreightClass_V1 || WgtByFCs_1 || 10000 || 50
WeightByFreightClassList_V1 || WgtByFCs || WgtByFCs_1
CompRefNumberList_V1 || CompRef_1
Component_V1 || Comps_1 || 25 || 07/31/1998@08:00:00 || 08/31/
1998@23:00:00 || 08/15/1998@08:00:00 || 09/15/1998@23:00:00 || CHD || 0
|| 0 || 0 || 0 || 10000 || 999.00 || 999.00 || COMPONENTS || *DFT ||
|| | | | | | | | | bFALSE || WgtByFCs || bTRUE || CompRef_1
ComponentList_V1 || Comps || Comps_1
RefNumber_V1 || RfrNums_1 || CR || 5465465465465
RefNumberList_V1 || RfrNums || RfrNums_1
ItineraryStep_V1 || ItinerarySteps_1 || SPT_LA || GPLSLECAUSA0001 ||
SPT_CONSIGNEE || CUSLATNYUSA0001 || | | bNULL || | | bNULL
ItineraryStepList_V1 || ItinerarySteps || ItinerarySteps_1
TransportOrder_V1 || i2_TOList_1 || A000001ab1 || bFALSE || CLT_LA || | |
|| bFALSE || bFALSE || FT_PRE_PAID || bFALSE || bFALSE || | | MODE || 01
|| APPLE || 05 || 02 || | | | | FromAddr || ToAddr || Mmo || | | | | | | | | | | | |
|| | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|| | | | | bFALSE || Comps || bFALSE || ItinerarySteps || bFALSE ||
RfrNums
TransportOrderList_V1 || i2_TOList || i2_TOList_1
TransportOrderSrvc || CreateTO || i2_TOList || FALSE
```

DeleteTO

Place the relevant entity IDs into a `StrIdList`. All the IDs provided represent the ID for the same entity type as specified in the delete operation.

If the operation is successful, the error list will be empty. Ensure that all the entities exist or the operation will fail.

Delete TO Example

```
StrIdList || TO_id_list || KENS-4
TransportOrderSrvc || DeleteTO || TO_id_list || FALSE
```

RetrieveTO

Place the relevant transport order IDs into a `StrIdList`. Ensure that the returned `TransportOrder_V1` list is previously defined but empty.

If the operation is successful, the error list will be empty and the returned transport order list will contain at least one item, packaged in the same format in which it was created.

Retrieve TO Example

```
StrIdList || TO_id_list || KENS-4
TransportOrderList_V1 || i2_TOList
TransportOrderSrvc || RetrieveTO || TO_id_list || FALSE || || i2_TOList
```

RetrieveTOShipments

Place the relevant transport order ID directly in the command to retrieve transport order shipments. Do not place it into a `StrIdList` as with other retrieve service operations.

Ensure that the returned `ShipmentList_V1` list is previously defined but empty. If the operation is successful, the error list will be empty and the returned shipment list will contain at least one item.

Retrieve TO Shipments Example

```
StrIdList || TO_id_list || KENS-4
ShipmentList_V1 || ShipList_1
TransportOrderSrvc || RetrieveTOShipments || KENS-4 || FALSE || || FALSE
|| ShipList_1
```

SetTO

This service sets two Transport Order Service parameters. The first parameter indicates whether only one shipment can be created from a transport order. The second parameter indicates whether a container's assignment cannot be modified. For more information, refer to [“void TransportOrderSrvc::SetTOUpdate” on page 86](#).

Set TO Update Example

```
TransportOrderSrvc || SetTOUpdate || TRUE || TRUE
```

UpdateTO

Define the top-level data structure along with the supporting aggregate data structures. For additional details, see [“Create” on page 237](#).

All transport orders in the data structure list must be in the Transportation Manager database or the update will fail.

Update TO Example

```
Address_V1 || FromAddr || || || || ||
Address_V1 || ToAddr || || || || ||
Memo_V1 || Mmo
WeightByFreightClass_V1 || WgtByFCs_1 || 10000 || 50
WeightByFreightClassList_V1 || WgtByFCs || WgtByFCs_1
CompRefNumberList_V1 || CompRef_1
Component_V1 || Comps_1 || 25 || 07/31/1998@08:00:00 || 08/31/
1998@23:00:00 || 08/15/1998@08:00:00 || 09/15/1998@23:00:00 || CHD || 0
```

```

|| 0 || 0 || 0 || 0 || 10000 || 999.00 || 999.00 || COMPONENTS || *DFT ||
|| || || || || || || || bFALSE || WgtByFCs || bTRUE || CompRef_1
ComponentList_V1 // Comps || Comps_1
RefNumberList_V1 || RfrNums
ItineraryStep_V1 || ItinerarySteps_1 || SPT_LA || GPLSLECAUSA0001 ||
SPT_CONSIGNEE || CUSLATNYUSA0001 || || bNULL || || bNULL
ItineraryStepList_V1 || ItinerarySteps || ItinerarySteps_1
TransportOrder_V1 || i2_TOList_1 || A000001ab1 || || || || || || || || ||
|| || || || APPLE || 05 || 02 || || || || FromAddr || ToAddr || Mmo || || | | | | | | | | | | | | | | | |
|| || || || || || || || || || || || || || || || || || || || || ||
|| || || || || || || bFALSE || Comps || bFALSE || ItinerarySteps || bTRUE ||
RfrNums
TransportOrderList_V1 || i2_TOList || i2_TOList_1
StrIdList || i2_TOIdList || A000001000
TransportOrderSrvc || UpdateTO || i2_TOIdList || i2_TOList || FALSE || ||
FALSE

```

TriggerTOEvent

Place the relevant transport order IDs into a `StrIdList`. If the operation is successful, the error list will be empty. All transport orders must exist or the operation will fail.

Trigger TO Event Example

```

StrIdList || IdList_1 || STEVE_1
SEC_V1 || SEC || ABCDE || 02/16/1998@11:30:00
TransportOrderSrvc || TriggerTOEvent || IdList_1 || TOEV_GENMOVMENT ||
SEC || FALSE

```

Chapter 7

CORBA API

You use the Transportation Manager CORBA interface to create and access data within Transportation Manager. This chapter requires a basic knowledge of CORBA and is divided into the following topics:

- [Overview](#)
- [Venture Factory](#)
- [Venture Factory](#)
- [Venture Session](#)

Overview

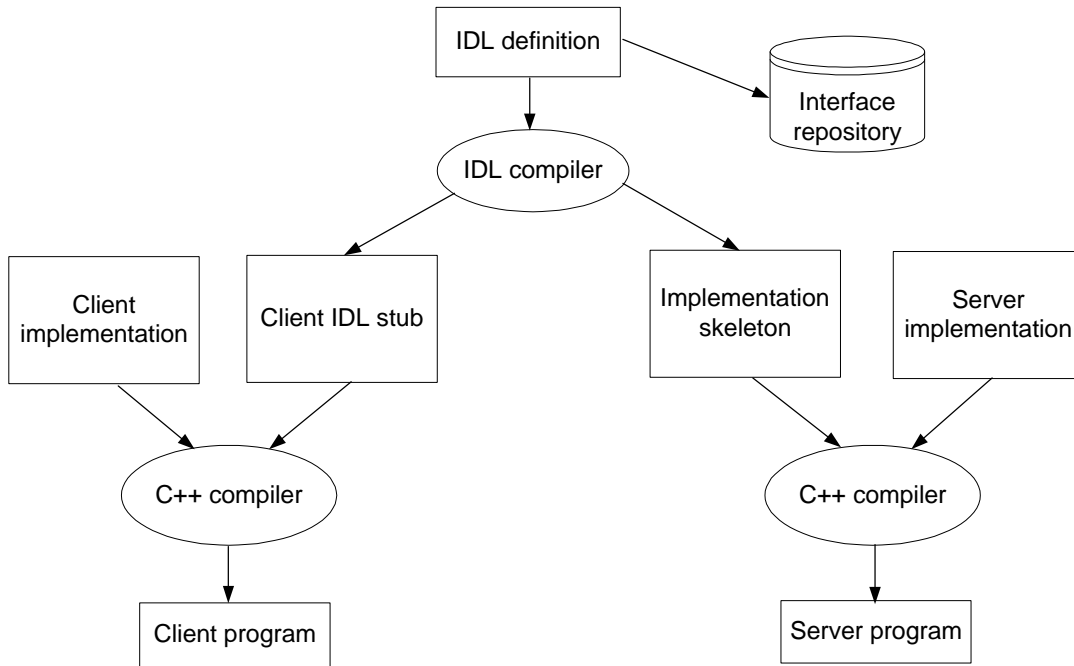
The Interface Definition Language (IDL) files distributed with each release of Transportation Manager defines the public interface. You can build your own client with any CORBA development kit that is compatible with VisiBroker.

The CORBA interface includes the services described in “API Services”. It also includes the `VentureSession` interface, which encapsulates the other services, providing a central connection point to Transportation Manager.

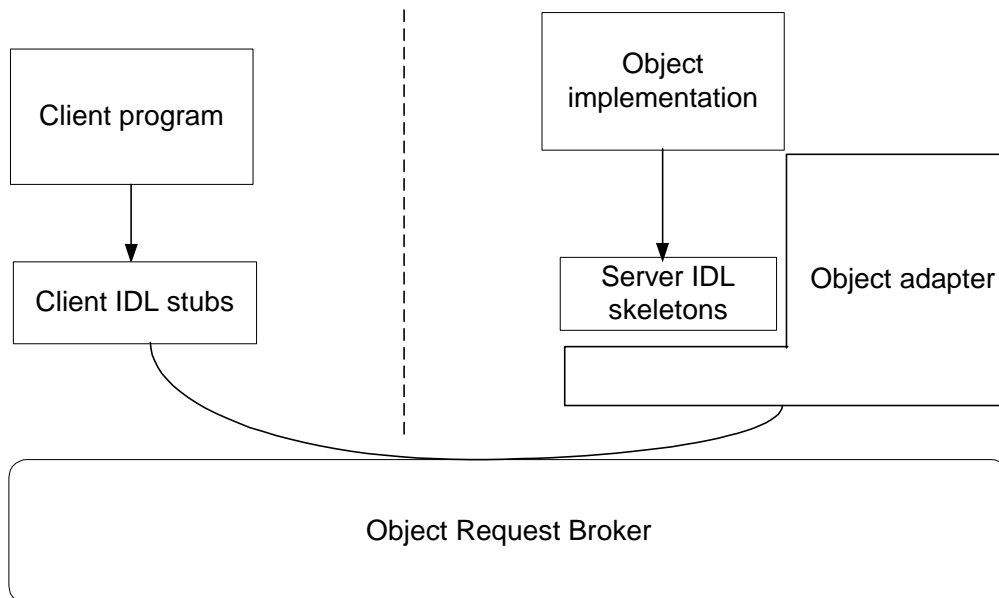
You create a logical session between the your system and Transportation Manager after obtaining a reference to a `VentureSession` object. This session maintains information about which Transportation Manager user you are connected as. It also maintains data that is cached for internal use by the other services. After establishing a connection, you can access the other services, such as Entity and Transport Order.

The following diagrams give an overview of the development and running of a CORBA program.

CORBA Program Development



Running A CORBA Program



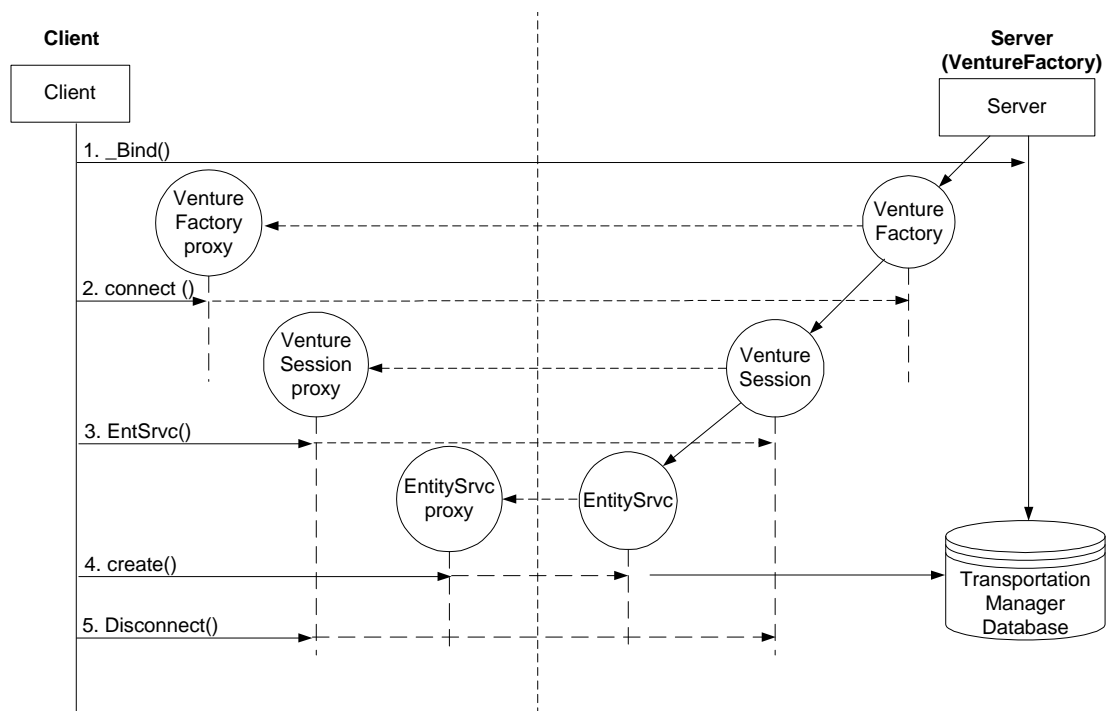
Venture Factory

The `VentureFactory` object represents the single connection point from which your system can establish a session to Transportation Manager. There is typically a single `VentureFactory` object within a system. To bind to this object, use the `_bind` method. To obtain a reference to the object, use the CORBA name service.

After you obtain a reference to the object, use the `Connect` method to create a session. All the Transportation Manager API services of that session use your default settings in the `Connect` method.

```
interface VentureFactory
{
    VentureSession Connect(in string UserName,
                          in string Password)
                          raises(VISError::Immediate);
};
```

The following diagram is a client/server scenario for various API programming entities, including `VentureFactory`. (The Client side represents your system.)



A sample client program for starting the CORBA API appears in “Sample C++ API Client” on page 333.

Venture Session

The `VentureSession` object represents a physical connection between your system and Transportation Manager. From this object, you access the Transportation Manager API services.

```
interface VentureSession
```

```
{
    enum ErrorMode
    {
        Immediate, Deferred
    };
    //Attributes

    readonly attribute EntitySrvc          EntSrvc;
    readonly attribute TransportOrderSrvc  TOSrvc;
    readonly attribute LoadSrvc            LdSrvc;
    readonly attribute FinancialsSrvc      FinSrvc;
    readonly attribute RateQuotationSrvc   RQSrvc;
    readonly attribute ShipmentSrvc        ShpmSrvc;
    readonly attribute AddrSrvc            AdrSrvc;
    readonly attribute TariffSrvc          TffSrvc;

    attribute ErrorMode                    ErrMode;
    attribute short                         LockRetries;

    //Methods

    void Disconnect()
        raises(VISError::Immediate);
};
```

Chapter 8

XML API

This chapter describes the Extensible Markup Language (XML) API. The topics included are:

- [XML Overview](#)
- [XML API Server](#)
- [XML Client](#)

XML Overview

Extensible Markup Language (XML) is a markup language for documents that have structured information. It is much more powerful than HTML, because XML allows you to create a custom markup language. Instead of using predefined tags such as <H1> for heading, you actually create and define your own tags.

XML is used with a Document Type Definition (DTD) file. The DTD defines the tags that you are using and thereby makes them valid. It defines the order, structure, and relationships of your data

For more information about XML and DTDs, see:

- <http://www.w3.org/XML>
- <http://www.xml.com>
- <http://www.xml.org>

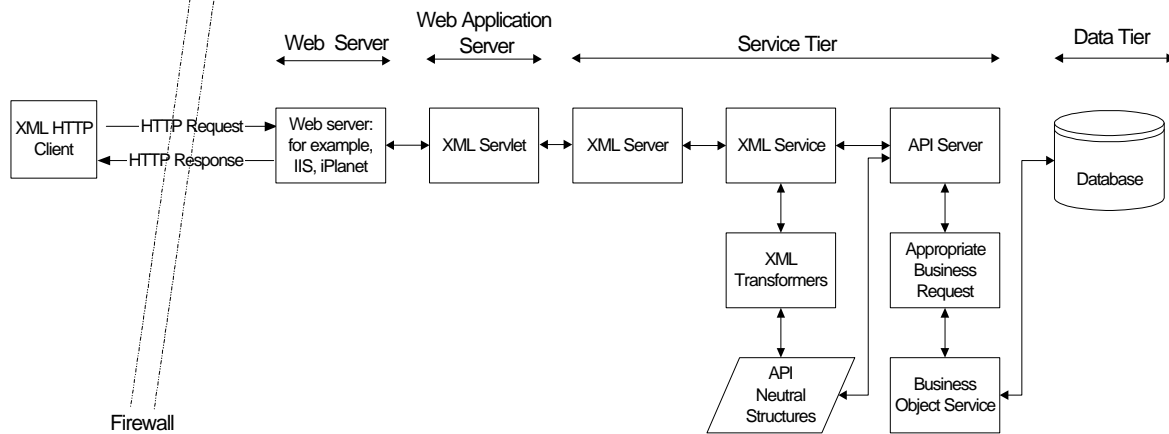
XML API Server

The XML server allows you to use API services and structures with Transportation Manager data. Unlike the existing API methods (the flat file driver and the CORBA object interface), the XML server uses XML to exchange information between the server and its clients.

Process Overview

The following list describes the XML API process:

1. The XML API HTTP client creates data and a request in XML format.
2. The request is sent to a web server that contains the XML API servlet.
3. The servlet forwards the request to an XML server that connects to an XML service.
4. The XML service does the following actions:



Naming Procedure for XML Elements

As described previously, the client should create the data and requests in XML format. An XML request consists of XML elements. After the request has been received, parsed and transformed, the correct API neutral structures should be completed.

The XML API uses a naming procedure for the various XML elements which identifies them more clearly compared to regular (non-XML) API.

The XML requests and responses are formatted using long names to reference the various XML elements in the request and response. The short names given in the mapping file (describe next) are the Transportation Manager API field names for the equivalent requests. For descriptions of these field names, refer to the Structures chapter in the *API Reference*.

A map file called XMLNames maps the XML names (or long names) and API names (or short names) to each other.

The following table gives examples of how the elements are mapped.

XML Name (Short Name)	API Name (Long Name)	Description
Shpm_Id	SystemShipmEntId	an automatically generated shipment system ID
Shpm_Num	ShipmentNumber	the shipment number
Cust_cd	CustomerCode	the customer ID code
TO_Ent_Ver_cd	ShipmentEntryVersionCode	the shipment order entry version code
TO_Ent_Typ_cd	ShipmentEntryTypeCode	the shipment order entry type code

XML Name (Short Name)	API Name (Long Name)	Description
ShipmentOrder	ShipmentOrder	a shipment order
ShipmentOrderCreate	ShipmentOrderCreate	the service that is available to create a shipment
APRate	ExecuteAPRatingFlag	indicates whether A/P rating is to be performed for the shipments
ResultContents	ResultContents	the type of output that is produced using the shipment level service
ShipmentOrderDelete	ShipmentOrderDelete	the service that is available to delete a shipment

XML Server Parameters

To properly run the XML Server, configure the database schema with a new parameter set called XML. This parameter set contains all the values that are currently in the API parameter set and an additional parameter called `DTDPATH`. This parameter should point to the Document Type Definitions (DTDs) that the XML server uses.

XML Parser

The XML API uses a third party product called Apache XML C++ Parser to parse the XML files. Parser libraries are included in the third part of the Transportation Manager folder. For information about the Apache Parser, refer to: <http://xml.apache.org/xerces-c/index.html>

Document Type Definitions (DTDs)

To improve performance, most of the user errors should be found before the user data is entered into the system. The XML file should therefore confirm to XML standards, which are:

- all XML tags (XML long names) should be valid
- only the correct XML elements should be in each composite XML element
- the data value for each XML element should be valid
 - example: the value for element `CreatedDateTime` should represent the date and time and not a numeric or string value
- fields that have a limited set of allowed values (for example, enumerated or boolean) should have only the allowed values

Most of these standards are met by using DTD files. DTDs specify the correct syntax of XML files. Each XML request should contain a reference to the proper DTD. DTDs are located on XML server's host, or are at least visible from it.

When the XML server starts, it:

- retrieves the value of `DTDPATH` parameter

- creates maps of long and short names
- transfers pointer to DTD and maps for the further processing of XML requests.

However, the client site may also include a DTD tree that the client can use to validate the XML files. This is not requested by the XML API, but it can be more convenient for the client.

For more information about DTDs, see
<http://www.w3.org/XML/1998/06/xmlspec-report>

The main DTD directory contains DTDs for all requests that are currently sent to the XML server:

- XMLNames.csv
- XML names map file
- CSVtoDTD.exe which converts the XML names map file to DTD entities
- a batch file to run CSVtoDTD.exe

DTD Directories

The DTD contains the following subdirectories:

- [BusEntities](#)
- [Common](#)
- [Ent](#)
- [Financial](#)
- [Load](#)
- [RateQuote](#)
- [ShipmentOrder](#)

BusEntities

The `BusEntities` directory contains the DTD for the common part of the business entities (`CommonBusEntity.dtd`) and DTDs for business entities.

- Auto Applied Option
- Carrier
- Consignee
- Customer Carrier Freight Audit
- Customer
- Customer Shipping Location Cross Reference
- Distribution Center
- Hub
- Load-At
- Zone

Common

The `Common` directory contains DTDs for the common API entities:

- Address
- Amount Detail
- Business Hours
- CEA Constraints
- Charge Detail
- Contact Information
- Contact Person
- Error
- Lane Availability
- Memo
- Reference Number
- Shipping Information
- Unit of Measurements
- Weight by Freight Class

Ent

The `Ent` directory contains rules for the various enumerated values and a map for short and long names, specifically:

- `BusEntEnums.ent` – the DTD rules for business entity enumerated values
- `CommonEnums.ent` – the DTD rules for common enumerated values
- `CommonTypes.ent` – the various attributes used for additional post-parsing data validation
- `FinancialEnums.ent` – the DTD rules for financial enumerated values
- `NameMap.ent` – the XML long names and the API short names map
- `ShpmOrderEnums.ent` – the DTD rules for shipment order enumerated values

Financial

The `Financial` directory contains DTDs for financial API entities:

- A/P Transaction
- A/R Transaction
- Charge Override
- G/L Transaction
- Non-Operational Freight (NOF) Stop
- Non-Operational Freight
- Rating Information

Load

The `Load` directory contains DTDs for load API entities:

- Charge
- Common Load part
- Load
- Shipment Confirmation Data
- Shipment Leg
- Stop

RateQuote

The `RateQuote` directory contains DTDs for rate quotation API entities:

- Charge Criteria
- Freight Detail
- Rate Criteria.dtd
- Schedule
- Valid Lane

ShipmentOrder

The `ShipmentOrder` directory contains DTDs for the direct shipment interface and transaction billing API entities:

XML APIs

The following XML APIs are available:

- AddressValidate
- APTransactionCommit
- APTransactionOutput
- APTransactionRetrieve
- ARTransactionCommit
- ARTransactionOutput
- ARTransactionRetrieve
- AssignToLoad
- CarrierCreate
- CarrierDelete
- CarrierRetrieve
- CarrierUpdate
- ConsigneeCreate
- ConsigneeDelete

- ConsigneeRetrieve
- ConsigneeUpdate
- CustomerCreate
- CustomerDelete
- CustomerRetrieve
- CustomerUpdate
- DCCreate
- DCDelete
- DCRetrieve
- DCUpdate
- FreightBillCancel
- FreightBillCreate
- FreightBillDetailCancel
- FreightBillDetailCreate
- FreightBillDetailRetrieve
- FreightBillDetailUpdate
- FreightBillRetrieve
- FreightBillUpdate
- GLTransactionCommit
- HubCreate
- HubDelete
- HubRetrieve
- HubUpdate
- LoadAtCreate
- LoadAtDelete
- LoadAtRetrieve
- LoadAtUpdate
- LoadBuild
- LoadConfirm
- LoadCreate
- LoadManifest
- LoadRetrieve
- LoadUpdateProgress
- NOFCreate
- NOFDelete
- NOFRetrieve

- NOFUpdate
- PlanCreate
- RateQuotationService
- SetLoadToPlanned
- SetShipmentPOD
- SetStopETA
- SetStopPOD
- SetStopToDelivered
- ShipmentConfirm
- ShipmentLegRetrieve
- ShipmentOrderCreate
- ShipmentOrderDelete
- ShipmentOrderRetrieve
- ShipmentOrderUpdate
- ShipmentPlan
- ShipmentUpdateProgress
- TenderAccept
- TenderReject
- ZoneCreate
- ZoneDelete
- ZoneRetrieve
- ZoneUpdate

XML Client

This section describes how to install the XML API client. The topics included are:

- [XML Overview](#)
- [XML API Server](#)
- [XML Client](#)

Overview of SSL

This section provides a brief introduction to Secure Socket Layer (SSL) programming. It is not intended as a comprehensive guide to SSL programming.

SSL is a communications standard that was designed to ensure that data sent through the Internet is secure. It is proven technology that has been thoroughly tested by many security experts, and is used by thousands of business in the Internet daily. Because of this, it has become the de-facto standard for secure Internet communications.

i2 uses the Sun Microsystems SSL package for its sample client program. The Java Secure Socket Extension (JSSE) 1.0.2 is a Java version implementation of the SSL and TLS. It is available for download at no cost. Documentation which provides a detailed SSL overview and a good reference of the JSSE library classes is also available for download.

You can download JSSE from the following URL.

<http://developer.java.sun.com/servlet/SessionServlet?url=http://developer.java.sun.com/developer/earlyAccess/jsse/index.html>

◆ To set up the JSSE environment for Windows NT

The following procedure is a simplified version of the one in the JSSE installation guide. For detailed information, refer to the JSSE documentation.

1. Ensure you have the Java Runtime classes in the following folder:
C:\Program Files\JavaSoft\jre\1.3\lib
2. Download and unzip JSSE to C:\jsse1.02.
3. Copy all the libraries under jsse1.02\lib to:
C:\Program Files\JavaSoft\jre\1.3\lib\ext
4. Open `java.security`, which should be located at:
C:\Program Files\JavaSoft\jre\1.3\lib\security\java.security
5. Add the following line to this file:

```
security.provider.3=com.sun.net.ssl.internal.ssl.Provider
```

If `security.provider.3` is already in use, you should choose the next available index. For example, if `security.provider.3` is the last in use, then use:

```
security.provider.4=com.sun.net.ssl.internal.ssl.Provider
```

XML API Connection Settings

The following information is required to connect to the Transportation Manager web site.

Parameter	Description
TMUser	Transportation Manager user name
TMPasswd	Transportation Manager user password

XMLClient.jar

i2 currently provides a Java client library, `XMLClient.jar`, for developing client programs to connect to the Transportation Manager XML services. This file is installed when you run the TM-Web install on the web application server. It is located in the `config\Markham_Domain\applications\tmXML\misc` folder under the WebLogic directory, for example:

D:\weblogic\config\Markham_Domain\applications\tmXML\misc\

`XMLclient.jar`

This library relieves developers from knowing the details of SSL and HTTP protocol. Note that it is not the only way to connect to the XML services. Developers can choose their own security package and programming language for their client program.

However, i2 recommends that developers use `XMLClient.jar` because it already manages cookies and some of the intricate protocol flows specific to i2. If developers still want to write their client program with their own security package, a strong knowledge of cookie handling and i2 protocol flow is required. Currently, this information is not published outside of i2.

CAUTION: Our client library (`XMLClient.jar`) currently relies on the security functions provided by JSSE. That is, the security strength of our library is dependent on the security provided by Sun Microsystems.

◆ To create a client application

This procedure describes how to create a client application. For a complete example of this program, refer to [“Sample Java Program” on page 341](#).

1. Create a file called `TestComm.java`.
2. Import the `FMXClientComm` class.


```
import FMXClientComm;
```
3. Create an instance of `FMXClientComm`.


```
FMXClientComm client = new FMXClientComm( "http://
MyWebServer:7001/tmXML/XMLAPIServlet" );
```
4. Complete the header information.


```
client.addFMXHeader ( "TM_USER", m_TMUser );
client.addFMXHeader ( "TM_PASSWORD", m_TMPasswd );
```
5. Send the XML request to and receive the response from Transportation Manager web site.


```
String reply = client.request (xmlString);
```

You can now compile and run the program.

◆ To compile and run the program

1. Include `XMLClient.jar` in the user class path:


```
set classpath=%classpath%;.\XMLClient.jar
```
2. Compile the file.


```
javac TestComm.java
```
3. Run the program.


```
java TestComm CreateShipment.xml
```


Chapter 9

API Error Messages

This chapter lists error messages in numerical order. Error codes and numbers also appear in `VisError.idl`.

No.	Error Code	Error Description	Severity
1	GEN_INTERNAL_ERROR	Unexpected internal error.	serious
2	GEN_SERVER_OUT_OF_MEMORY	Server out of memory.	fatal
3	GEN_OBJ_NOT_FOUND	Object not found.	serious
4	GEN_DUPLICATE_OBJ_ID	Duplicate object ID supplied.	serious
5	GEN_MISSING_MANDATORY_ELEMENT	Missing mandatory structure element value.	serious
6	GEN_ELEMENT_NOT_SETTABLE_DURING_CREATE	Structure element not settable during create.	serious
7	GEN_ELEMENT_NOT_SETTABLE_DURING_UPDATE	Structure element not settable during update.	serious
8	GEN_INVALID_NUMBER	Invalid number format/value.	serious
9	GEN_INVALID_DATE	Invalid date format/value.	serious
10	GEN_INVALID_TIME	Invalid time format/value.	serious
11	GEN_INVALID_TIMESTAMP	Invalid timestamp format/value.	serious
12	GEN_INVALID_DOMAIN_TABLE_VALUE	Invalid domain table value.	serious
13	GEN_RELATED_OBJ_NOT_FOUND	Could not assign the related object because it does not exist.	serious
14	GEN_RELATED_OBJ_STATUS_INVALID	The related object could not be assigned because it has an incompatible status.	serious

No.	Error Code	Error Description	Severity
15	GEN_NUMBER_OUT_OF_RANGE	The number is outside of the valid range or contains non-numeric character.	serious
16	GEN_INVALID_OBJ_ID_FMT	The object ID contains invalid characters.	serious
17	GEN_NULL_STRING_PTR_SUPPLIED	A NULL string pointer has been supplied. Ensure that you have allocated the string.	serious
18	GEN_INIT_FAILURE_DOMAIN_TABLES	Initialization error, cannot load domain tables.	fatal
19	GEN_INIT_FAILURE_LOGISTICS_SYSTEM	Initialization error, cannot load logistics system.	fatal
20	GEN_MAX_LOCK_RETRIES	Operation aborted, maximum number of lock retries exceeded.	serious
21	GEN_UNKNOWN_ERROR	An unknown error has occurred.	serious
22	GEN_DB_CONNECT_FAILURE	Cannot connect to database. Ensure that the username and password are correct.	fatal
23	GEN_NUM_KEYNOTSET	Required field is not set.	serious
24	GEN_NEED_EMPTY_FIELD	Field should be empty.	serious
25	GEN_CUST_NO_FOUND	Customer field is not defined.	serious
26	GEN_CARR_NO_FOUND	Carrier field is not defined.	serious
27	GEN_BO_ERROR	Business Logic Error.	serious
28	GEN_FAILED_TO_GET_PRINTER_INFO	Could not obtain default printer information, please ensure that a default printer has been installed.	warning
29	GEN_NOT_YET_IMPLEMENTED	Operation is not yet implemented.	serious
30	GEN_STRING_TOO_LONG	String value is too long.	serious
31	GEN_STRING_ID_OBJECT_ID_MISMATCH	String ID and Object ID do not match.	serious
32	GEN_FUTURE_TIME_INVALID	Timestamp must not be in the future.	serious
33	GEN_NEED_NOT_EMPTY_FIELD	Field must not be empty.	serious
34	GEN_STRING_TOO_SHORT	String value is too short.	serious
35	GEN_DB_ERROR	Database error.	serious
36	GEN_STRING_TRUNCATED	String value has been truncated.	warning

No.	Error Code	Error Description	Severity
37	GEN_WARN_ELEMENT_IGNORED_DURING_CREATE	Warning: Structure element not settable during create; value will be ignored.	warning
38	GEN_WARN_ELEMENT_IGNORED_DURING_UPDATE	Warning: Structure element not settable during update; value will be ignored.	warning
39	GEN_INVALID_EVENT_CODE	Event code is invalid for the current object status, list of the valid event codes follows:.	serious
40	GEN_WARN_IGNORE_FIELD_SET_TO_FALSE	Warning: Ignore field has been set to bFALSE.	warning
41	GEN_NUMBER_OUT_OF_RANGE_13	Value must be between 0 and 999999999999999.	serious
42	GEN_DIVISION_LGSTRP_MISMATCH	User's Division and/or Logistic Group ID restricted and do not match.	serious
43	GEN_ELEMENT_NOT_SETTABLE	Structure element is read only.	serious
44	GEN_STRING_INVALID	String value is invalid.	serious
45	MSTR_CONDOPT_USED_BY_TFF_CHRG	Failed to perform the operation: Master Condition/Option is used by Tariff Charge.	serious
46	MSTR_SRVC_REFERENCED	Failed to perform the operation: Master Service is referenced by another object.	serious
47	GEN_SYS_INTEGRITY_ERR	System integrity error.	fatal
48	GEN_FRHT_CLASS_STAT_INVALID	Illegal Freight class status.	serious
49	GEN_OBJECT_NOT_UNIQUE	Object is not unique.	serious
1000	COM_AUTOCORRECTED_ADDRESS	The address has been auto-corrected.	warning
1001	COM_INVALID_ADDRESS_UNIT	Invalid address (Unit).	serious
1002	COM_INVALID_ADDRESS_BLK_BLDG	Invalid address (Block).	serious
1003	COM_INVALID_ADDRESS_ST_NAME	Invalid address (Street).	serious
1004	COM_INVALID_ADDRESS_CTY_CD	Invalid address (City).	serious
1005	COM_INVALID_ADDRESS_PSTL_CD	Invalid address (Postal/Zip Code).	serious
1006	COM_INVALID_ADDRESS_STA_CD	Invalid address (State/Province).	serious
1007	COM_INVALID_ADDRESS_CTRY_CD	Invalid address (Country).	serious
1008	COM_INVALID_ADDRESS	Invalid address.	serious

No.	Error Code	Error Description	Severity
1009	COM_MISSING_ADDRESS	Internal error, missing address object.	fatal
1100	COM_NO_SPEC_AUTO_OPT_LOC	The option must be applied to at least one location.	serious
1200	COM_INVALID_TM_WND_RANGE	The 'From' time cannot be greater than the 'To' time.	serious
1201	COM_OVERLAPPED_TM_WND_RANGE	The second 'From' time must be greater than the first 'To' time.	serious
1202	COM_MISSING_BUS_HRS	Internal error, missing business hours object.	fatal
1300	COM_NON_EMPTY_FRHTCLS_FOR_PIECE	Freight Class should not be specified when using Piece.	serious
1301	COM_MISSING_FRHTCLS_CD	Freight Class is missing.	serious
1302	COM_STACK_DIFFERENT	If StackOn or StackedOn are either '0' or '1', StackOn and StackedOn have to be equal.	serious
1303	COM_MAXWGT_SMALLER_THAN_TAREWGT	Maximum Weight cannot be smaller than Tare Weight.	serious
1304	COM_IF_ANY_OF_LWH_THENALL	If any of 'Len', 'Wdth' or 'Hght' are entered, then all three must be entered.	serious
1305	COM_VOL_NOT_EQ_LWH	Volume does not equal Length x Width x Height.	serious
1306	COM_COMP_TYPE_CD_NOT_UNIQUE	Non unique Container Type code.	serious
1307	COM_COMP_TYPE_LVL_NOT_UNIQUE	Non unique container type level.	serious
1308	COM_STK_OVERFLOW	Container Stack overflow.	serious
1309	COM_INVALID_COMB_SYS_LENWGT	Invalid Combination of UOM system, length, weight and distance units.	serious
1310	COM_DUP_SHPG_LOC_XREF	Cannot cross reference shipping location more than once per customer.	serious
1311	COM_DUP_CUSTOMER_XREF	Cannot cross reference customer more than once per shipping location.	serious
1312	COM_DUP_EXTERNAL_ALIAS	This External ID already exists for this cross reference.	serious
1313	COM_DUP_PREFERRED_ALIAS	This cross reference can only have one preferred External ID.	serious
1314	COM_PREFERRED_ALIAS_NOT_SET	No preferred External ID found for this cross reference.	serious

No.	Error Code	Error Description	Severity
1315	COM_DIFF_CUST_CD	Customer code is different from the one is creating.	serious
1316	COM_DIFF_SHPG_LOC_CD	Shipping Location code is different from the one is creating or updating.	serious
1317	COM_DIFF_SHPG_LOC_TYPE	Shipping Location type is different from the one is creating or updating.	serious
1318	COM_INVALID_SHPG_LOC_XREF	Invalid shipping location cross reference.	serious
1319	COM_UNMATCH_AUTH_SHOULD_BE_NEVER	UNMATDFBAUTH_NEVER is the only value allowed since the customer is not enabled for Unmatched Audit.	serious
1320	COM_UNMATCH_AUTH_SHOULDNOT_BE_NEVER	Value should not be UNMATDFBAUTH_NEVER since the customer is enabled for Unmatched Audit.	serious
1321	COM_DUP_CUST_CARR_XREF	The Carrier/Customer relationship already exists.	serious
1322	COM_DIFF_CARR_CD	Carrier code is different from the one is creating or updating.	serious
1323	GEN_REPORT_NOT_SETTABLE	Failed to perform the operation: the report is a system provided report.	serious
1324	GEN_REPORT_INVALID_NAME	This Report Name is invalid Report Name cannot contain the following characters: / < > ? * : \.	serious
1325	GEN_REPORT_CANNOT_DELETE_ASSOC	Failed to delete the report The report is associated to other entities.	serious
1326	GEN_ILEGAL_FRHT_CLASS_ID	Freight class ID value has to be between 0 and 9999 or a string that starts with asterisk.	serious
1327	GEN_FRHT_CLASS_ID_DEC_PNT	Freight class ID value can have one digit after decimal point.	serious
1328	GEN_FRHT_CLASS_ID_DEC_PNT_ZERO	Freight class ID value cannot have a zero after decimal point.	serious
1329	GEN_FRHT_CLASS_ILEGAL_PRTY	Invalid freight class priority value.	serious
1330	GEN_FRHT_CLASS_DUP_PRTY	Freight class priority value is duplicated.	serious
1331	GEN_FRHT_CLASS_READONLY	Failed to perform the operation: freight class is reserved.	serious
1332	COM_DUP_CONTACT_ROLE	Contact with this Role is already present.	serious

No.	Error Code	Error Description	Severity
10000	VTRE_SRVC_UNKNOWN_USER	Incorrect username or password.	serious
20001	ENT_SRVC_INVALID_INS_EXP_DT	Invalid Insurance Expired date.	serious
20002	ENT_SRVC_INVALID_BOLNUM	Invalid Carrier Tracking Number.	serious
20003	ENT_SRVC_INVALID_PREF_SERV	Invalid Preferred Service.	serious
20004	ENT_SRVC_INVALID_TRANS_SRVC	Invalid transport service.	serious
20005	ENT_SRVC_INVALID_STATUS	Invalid status supplied.	serious
20006	ENT_SRVC_INVALID_STRUCTURE_TYPE	Unsupported structure type supplied to Entity Service method.	serious
20007	ENT_SRVC_INVALID_ENTITY_TYPE	Invalid entity type supplied to Entity Service method.	serious
20008	ENT_SRVC_EMPTY_COMP_LIST	At least one container should be defined for the entity.	serious
20009	ENT_SRVC_TP_ISREFERENCED	Unable to delete a trading partner that has already been referenced.	serious
20010	ENT_SRVC_INVALID_STA_CD	Invalid State code.	serious
20011	ENT_SRVC_UNKNOWN_CTRY_CD	Unknown Country code.	serious
20012	ENT_SRVC_INVALID_GEO_AREA	Invalid Geographic Area member.	serious
20013	ENT_SRVC_CITY_STATE_MISMATCH	Unknown City / State pair.	serious
20014	ENT_SRVC_CITY_PRCN_PSTL_RNG_MISMATCH	City Precise can only be true if a postal code range is specified.	serious
20015	ENT_SRVC_MISSING_RANGE_LIMITS	Missing range limits.	serious
20016	ENT_SRVC_RANGE_LIMIT_SIZE_MISMATCH	Minimum and maximum values of range must be equal in length.	serious
20017	ENT_SRVC_RANGE_LIMIT_VALUE_MISMATCH	Minimum value of range must be less than maximum value.	serious
20018	ENT_SRVC_BAD_USA_POSTAL_RANGE	Invalid USA Zip Code range.	serious
20019	ENT_SRVC_BAD_START_POSTAL_LIMIT	First characters of minimum and maximum values must match with a Postal Code range.	serious
20020	ENT_SRVC_BAD_START_USA_POSTAL_LIMIT	First 2 characters of minimum and maximum values must match with a Zip Code range.	serious
20021	ENT_SRVC_PSTL_RNG_SPANS_STATES	Postal Code range spans States.	serious
20022	ENT_SRVC_STATE_PSTL_RNG_MISMATCH	Postal Code range is not in the State specified.	serious

No.	Error Code	Error Description	Severity
20023	ENT_SRVC_UNKNOWN_PSTL_RNG	No matching range exists in the postal code table.	serious
20024	ENT_SRVC_BAD_PSTL_CD_FOR_STATE	Invalid Postal Code for the specified State.	serious
20025	ENT_SRVC_PSTL_RNG_MISSING_FROM_VALUE	Maximum Postal Code invalid without Minimum Postal Code.	serious
20026	ENT_SRVC_ZN_GEO_STA_CD_MISMATCH	Invalid State code in Geographic Area for this Zone.	serious
20027	ENT_SRVC_CITY_NAME_VALUE_MISMATCH	City cannot be specified if Postal Codes or City Precise are used.	serious
20028	ENT_SRVC_DIST_ZNGRP_MISMATCH	Distance cannot be specified if Zone is a member of a Zone Group.	serious
20029	ENT_SRVC_DIST_CITY_MISMATCH	Distance cannot be specified if City or Postal Codes have been.	serious
20030	ENT_SRVC_PSTL_CD_MIN_WILDCARD	Wildcard Postal Codes must have at least two characters defined.	serious
20031	ENT_SRVC_PSTL_CD_MASK_MISMATCH	Postal Code does not match the Postal Code Mask.	serious
20032	ENT_SRVC_ZN_DESC_MAX_LEN	Zone Description field exceeds maximum length.	serious
20033	ENT_SRVC_POSTAL_CD_MAX_LEN	Postal Code exceeds maximum length.	serious
20034	ENT_SRVC_CANNOT_BE_FIRST_PICKUP	A consignee cannot be a first pickup location.	serious
20035	ENT_SRVC_CANNOT_BE_LAST_DROP	A load at cannot be a last drop.	serious
20036	ENT_SRVC_EQMT_TYP_CD_MAX_LEN	Equipment type code exceeds maximum length.	serious
20037	ENT_SRVC_EQMT_TYP_DESC_MAX_LEN	Equipment type description exceeds maximum length.	serious
20038	ENT_SRVC_NEED_FC_FOR_RATING_UNITS	Freight Class is mandatory for the specified Rating Units.	serious
20039	ENT_SRVC_INVALID_RFRC_NUM_QLFR_ID	Invalid Reference Number Qualifier ID.	serious
20040	ENT_SRVC_DETL_CHRG_NOT_UPDATEABLE	This field is protected when VoucherGenLevelForManifest is Load (i.e. LDLEG) OR PaymentMode is Auto Pay.	serious
20041	ENT_SRVC_DELETE_ERROR	Unable to delete entity.	serious

No.	Error Code	Error Description	Severity
20042	ENT_SRVC_ZN_SHARED	Cannot delete Zone while it is a member of a Zone Group.	serious
20043	ENT_SRVC_EQMT_TYPE_INUSE	Cannot delete Equipment Type while it is still used by a Tariff Service.	serious
20044	ENT_SRVC_ZN_GEO_LIST_EMPTY	Zone must define at least one Geography.	serious
20045	ENT_SRVC_ENT_VER_ENT_TYPE_EMPTY	Both DftTOEntVer_Id and Shpm_Ent_Typ_Id are empty, Shpm_Ent_Typ_Id must be specified.	serious
20046	ENT_SRVC_ILD_INACTIVE_VLDT_ITM_NOT_NONE	If ILD is not active Vldt_Itm should be turned off or empty.	serious
20047	ENT_SRVC_VLDT_ITM_ITM_COMP_GROUP	If Validation Item is enabled, neither Item Group nor Container Type Group may be empty.	serious
20048	ENT_SRVC_CUST_SCHG_RATE_MISMATCH	Flat surcharge not supported if charge not based on carrier Resetting surcharge amount to 0.	warning
20049	ENT_SRVC_CANNOT_REPLACE_CARR	Carrier cannot be updated.	serious
20050	ENT_SRVC_NO_ADDR_NOR_SHPGLOC	Address must be supplied if there is no Shipping Location to copy it from.	serious
20051	ENT_SRVC_MISSING_INCO_SHPG_LOC_INFO	If you want to defined a Default INCO Terms Shipping Location, please provide both location code and type.	serious
20052	ENT_SRVC_SPOT_CARR_NOT_RSTD	A Spot Carrier is always a Restricted Carrier.	serious
20053	ENT_SRVC_SPOT_CARR_CTRC_TFFSRVC	A Spot Carrier may not have Tariff Services specified for a Contracted Carrier.	serious
20054	ENT_SRVC_SYSTEM_PROVIDED_GEO_CODE	You cannot specify Latitude/ Longitude and GeoCodeRecalculate_yn == bTRUE Distance will not be recalculated.	warning
20055	ENT_SRVC_BOTH_ZERO_IN_GEO_CODE	Both Latitude and Longitude cannot be set to zero because Geo-Code is mandatory for the specified Country.	serious
20056	ENT_SRVC_COUNTRY_STATE_MISMATCH	Unknown Country / State pair.	serious
20057	ENT_SRVC_UNIT_TYPE_EMPTY	If Variable Load/Unload time is specified - Load/Unload unit type cannot be empty or NULL.	serious

No.	Error Code	Error Description	Severity
20058	ENT_SRVC_LENGTH_WIDTH_HEIGHT	Length, Width and Height must be either all zeros or all non zeros.	serious
20059	ENT_SRVC_END_DATE_LESSTHAN_START	End date cannot be less than start date.	serious
20060	ENT_SRVC_CARR_ACTIVATE_INS_EXPIRED	Cannot re-activate the carrier because the insurance has expired.	serious
20061	ENT_SRVC_CARR_PUT_ON_HOLD_INS_EXPD	Carrier status set to ON HOLD because the insurance has expired.	warning
20062	ENT_SRVC_ELGB_CNT_MV_PLND_MUST_BE_TRUE	Continuous Moves Eligibility flag for Load Planned status must be TRUE when the Continuous Moves Eligibility flag is TRUE.	warning
20063	ENT_SRVC_ELGB_CNT_MV_PLND_MUST_BE_FALSE	Continuous Moves Eligibility flag for Load Planned status must be FALSE when the Continuous Moves Eligibility flag is FALSE.	warning
20064	ENT_SRVC_FOUR_CNT_MV_FLAGS_MUST_BE_EQUAL	Continuous Moves Eligibility flags for Load Tendered/Tender Accepted/Confirmed/Completed status must always be equal.	warning
20065	ENT_SRVC_FOUR_CNT_MV_FLAGS_MUST_BE_FALSE	Continuous Moves Eligibility flags for Load Tendered/Tender Accepted/Confirmed/Completed status must be FALSE when the Continuous Moves Eligibility flag is FALSE.	warning
20066	ENT_SRVC_SYSTEM_CANNOT_CALCULATE_GEO_CODE	Geo-Code cannot be re-calculated but it is mandatory for the specified Country.	serious
20067	ENT_SRVC_EXTL_AND_INTL_CODES_EMPTY	Either External or Internal code must be specified.	serious
30000	TOE_SRVC_INVALID_STRUCTURE_TYPE	Unsupported structure type supplied to the Transport Order Service method.	serious
30001	TOE_SRVC_ROUTE_AND_RATE_FAIL	Failed to route and rate.	warning
30002	TOE_SRVC_WEIGHT_EXCEEDED	The specified weight exceeds maximum weight for at least one of the Itinerary points.	serious
30003	TOE_SRVC_VOL_EXCEEDED	The specified volume causes the shipment volume limit to be exceeded.	serious

No.	Error Code	Error Description	Severity
30004	TOE_SRVC_SCLD_WGT_OVERFLOW	The specified scale weight causes the Transport Order to exceed its accuracy.	serious
30005	TOE_SRVC_VOL_OVERFLOW	The specified volume causes the Transport Order to exceed its accuracy.	serious
30006	TOE_SRVC_ODR_VAL_OVERFLOW	The specified order value causes the Transport Order to exceed its accuracy.	serious
30007	TOE_SRVC_DCLD_VAL_OVERFLOW	The specified declared value causes the Transport Order to exceed its accuracy.	serious
30008	TOE_SRVC_NMNL_WGT_OVERFLOW	The specified nominal weight causes the Transport Order to exceed its accuracy.	serious
30009	TOE_SRVC_TARE_WGT_OVERFLOW	The specified nominal weight causes the Transport Order to exceed its accuracy.	serious
30010	TOE_SRVC_LGST_GRP_INACTIVE	The specified Logistics Group is inactive.	serious
30011	TOE_SRVC_SHIP_FRM_INACTIVE	The specified shipping point is inactive.	serious
30012	TOE_SRVC_SHIP_TO_INACTIVE	The specified destination point is inactive.	serious
30013	TOE_SRVC_NO_TO_ADDR	The Shipping Address is not specified.	serious
30014	TOE_SRVC_NO_FRM_ADDR	The Destination Address is not specified.	serious
30015	TOE_SRVC_FRM_ZONES_INVALID	The Zones for the Origin Shipping Point Address could not be created.	serious
30016	TOE_SRVC_TO_ZONES_INVALID	The Zones for the Destination Shipping Point Address could not be created.	serious
30017	TOE_SRVC_INVALID_FROM_TO_DATES	The Transport Order From and To Dates are not properly specified.	serious
30018	TOE_SRVC_THGH_PNT_WGT_VOL_INVALID	The Shipping Point, Destination and/or the Through Points have an invalidly specified Weight or Volume Limit.	serious
30019	TOE_SRVC_TO_NO_EDIT	The Transport Order status does not permit editing.	serious

No.	Error Code	Error Description	Severity
30020	TOE_SRVC_SHPM_CANCEL_FAILED	Failed to cancel an empty shipment.	serious
30021	TOE_SRVC_ODR_VAL_EXCEEDS_CUST	The specified order value causes the Transport Order to exceed the limit for this customer.	serious
30022	TOE_SRVC_NO_COMP_TYPE	No Container Type Specified.	serious
30023	TOE_SRVC_NO_DCLD_VAL	No Declared Value Specified.	serious
30024	TOE_SRVC_NO_ODR_VAL	No Order Value Specified.	serious
30025	TOE_SRVC_NO_NMNL_WGHT	A Nominal Weight must be specified.	serious
30026	TOE_SRVC_NO_FC	No Freight Class has been specified.	serious
30027	TOE_SRVC_SCLD_WGT_EXCEEDS_COMP_TYPE	The specified scale weight exceeds the limit set by the Container Type for this customer.	serious
30028	TOE_SRVC_ERROR_DEL_RATE_INFO	Internal error while deleting the rating information.	fatal
30029	TOE_SRVC_ERROR_CREATING_LOADLEGS	Internal error while creating the shipment itinerary.	fatal
30030	TOE_SRVC_ERROR_SHPM_NO_EDIT	The Shipment status does not permit editing.	serious
30031	TOE_SRVC_NO_WBFC_MULTI_FC_CRT	A Multi-Freight Class Container must have at least one Freight Class weight specified.	serious
30032	TOE_SRVC_NOT_VALID_FC	The specified Freight Class is no longer valid.	serious
30033	TOE_SRVC_PIECES_MULTIFC	Containers with a Container Type rating unit of PIECE cannot be Multi-Freight Class.	serious
30034	TOE_SRVC_NO_COMPS	A Transport Order must have at least one container.	serious
30035	TOE_SRVC_NO_FROM_TO_DATES	A Container cannot be specified without Pickup and Delivery From and To Dates.	serious
30036	TOE_SRVC_TO_LESS_FROM_PKUP	The Pickup To Date cannot be less than the Pickup From Date.	serious
30037	TOE_SRVC_TO_LESS_FROM_DLVY	The Delivery To Date cannot be less than the Delivery From Date.	serious
30038	TOE_SRVC_PKUP_LESS_DLVY_FROM	The Delivery From Date cannot be less than the Pickup From Date.	serious

No.	Error Code	Error Description	Severity
30039	TOE_SRVC_PKUP_LESS_DLVY_TO	The Delivery To Date cannot be less than the Pickup To Date.	serious
30040	TOE_SRVC_LWH_MUST_ZERO_QTY_G1	Components with a quantity greater than one must have Lengths, Widths and Heights all of zero.	serious
30041	TOE_SRVC_LWH_PARTIAL_SPEC	The Length, Width and Height must all be specified if one is specified.	serious
30042	TOE_SRVC_NOT_TO_MIN_ODR_VAL	The Transport Order does not meet the required minimum order value.	serious
30043	TOE_SRVC_NO_CUST_FOR_TO	The Transport Order does not have a customer specified.	serious
30044	TOE_SRVC_NO_SHIP_FROM_PNT	The Transport Order does not have a ship from point specified.	serious
30045	TOE_SRVC_NO_SHIP_TO_PNT	The Transport Order does not have a ship to point specified.	serious
30046	TOE_SRVC_NO_LGST_GRP_FOR_TO	The Transport Order does not have a logistics group specified.	serious
30047	TOE_SRVC_NO_DIV_FOR_TO	The Transport Order does not have a division specified.	serious
30048	TOE_SRVC_DRCT_WITH_JRNY	An Itinerary Template cannot be specified for a Direct Shipment.	serious
30049	TOE_SRVC_CUST_ON_HOLD	The specified customer is currently on hold.	serious
30050	TOE_SRVC_CUST_NO_SHPM	The specified customer does not permit shipments.	serious
30051	TOE_SRVC_NOT_TO_MAX_ODR_VAL	The Transport Order exceeds the customer's maximum order value.	serious
30052	TOE_SRVC_NO_TOVER_FOR_TO	The Transport Order does not have a Transport Order Version Specified.	serious
30053	TOE_SRVC_NO_TOTYPE_FOR_TO	The Transport Order does not have a Transport Order Type Specified.	serious
30054	TOE_SRVC_UPDATE_NOT_ALLOWED	Cannot perform operation for entities that are being updated.	serious
30055	TOE_SRVC_ENTITY_NOT_EXIST	The specified entity does not exist.	serious
30056	TOE_SRVC_NUM_INVALID_SEC_HANDLE	Invalid Event Reason Code handle - probably mismatch Event Reason Code handle.	serious
30057	TOE_SRVC_NO_AR_SRVC	The Accounts Receivable Service is not provided.	serious

No.	Error Code	Error Description	Severity
30058	TOE_SRVC_JRNY_EXPIRES	The Itinerary Template Expires prior to the latest container ship-to delivery date.	serious
30059	TOE_SRVC_JRNY_NOT_VALID	The Itinerary Template is not valid for earliest container ship-from pickup date.	serious
30060	TOE_SRVC_JRNY_AND_DRCT	The Ship Direct flag cannot be set when through points are provided.	serious
30061	TOE_SRVC_NO_WBFC	At least one Weight By Freight Class must be specified.	serious
30062	TOE_SRVC_RFRC_NUM_REQ	A required reference number has not been specified.	serious
30063	TOE_SRVC_RFRC_NUM_DUP	The reference number provided is not unique.	serious
30064	TOE_SRVC_SHPG_SCLD_WGT_OVERFLOW	Overflow when the Shipment Scaled Weight is converted to Logistic System Units.	serious
30065	TOE_SRVC_SHPG_VOL_OVERFLOW	Overflow when the Shipment Volume is converted to Logistic System Units.	serious
30066	TOE_SRVC_SHPG_DCLD_VAL_OVERFLOW	Overflow when the Shipment Declared Value is converted to Logistic System Units.	serious
30067	TOE_SRVC_SHPG_ODR_VAL_OVERFLOW	Overflow when the Shipment Order Value is converted to Logistic System Units.	serious
30068	TOE_SRVC_INVALID_AR_SRVC	Invalid AR Service.	serious
30069	TOE_SRVC_INVALID_FRM_ADDR_ZONES	Invalid From Address Zones.	serious
30070	TOE_SRVC_INVALID_TO_ADDR_ZONES	Invalid To Address Zones.	serious
30071	TOE_SRVC_INVALID_PICKUP_DATE	Invalid Pick up Date.	serious
30072	TOE_SRVC_INVALID_DLVY_DATE	Invalid Delivery Date.	serious
30073	TOE_SRVC_INVALID_WGT_EXCEED_MAX_FOR_LOCATION	Weight exceeds the maximum limit for the location.	serious
30074	TOE_SRVC_INVALID_VOL_EXCEED_MAX_FOR_LOCATION	Volume exceeds the maximum limit for the location.	serious
30075	TOE_SRVC_LIST_LENGTH_NOT_MATCH	Service List length does not match.	serious
30076	TOE_SRVC_INVALID_REF_NUM_TYPE	The reference number type is not valid.	serious

No.	Error Code	Error Description	Severity
30077	TOE_SRVC_TO_ID_NOT_FOUND	No transport order could be found matching the Transport Order Identifier The transport Order ID cannot be found.	serious
30078	TOE_SRVC_MULTIPLE_TO_ID_FOUND	Multiple Transport Order IDs are found for update.	serious
30079	TOE_SRVC_INCONSISTENT_JOURNEY_STEPS	Inconsistent Shipment Itinerary Steps.	serious
30080	TOE_SRVC_INVALID_STATUS	Transport Order Status is not valid.	serious
30081	TOE_SRVC_PLAN_ID_NOT_FOUND	Plan not found.	serious
30082	TOE_SRVC_ATTACH_TO_PLAN_FAIL	Failed to attach shipments to Plan.	serious
30083	TOE_SRVC_INVALID_COMP_ITEM_VALUE	The container item value is not valid.	serious
30084	TOE_SRVC_REQ_ITN_STEP	The itinerary steps are required.	serious
30085	TOE_SRVC_UPDATE_ADDRESS_NOT_ALLOWED	New From and To addresses cannot be specified during an Update.	serious
30086	TOE_SRVC_INCORRECT_NUMBER_OF_ITN_STEP	The number of the itinerary steps are not consistent with the ones in the system.	serious
30087	TOE_SRVC_INCONSISTENT_ITN_STEP	The itinerary steps are not consistent with the ones in the system.	serious
30088	TOE_SRVC_COMP_REF_NUM_NOT_UNIQUE	Identifying component reference numbers are not unique.	serious
30089	TOE_SRVC_SINGLE_SHIPMENT_REQ_VIOLATED	Component update results in single shipment requirement being violated.	serious
30090	TOE_SRVC_MIT_SEQ_NUMBER_EMPTY	MITCC Sequence Number should not be empty.	serious
30091	TOE_SRVC_MIT_SEQ_NUMBER_NOT_EMPTY	MITCC Sequence Number should be empty.	serious
30092	TOE_SRVC_MIT_DATES_MISMATCH	Shipments with the same MITCC ID have different From/To Delivery dates range.	serious
30093	TOE_SRVC_MIT_CSLD_MISMATCH	Shipments with the same MITCC ID have different Consolidation Class.	serious
30094	TOE_SRVC_MIT_SEQ_NUM_NEG_BIG	MITCC Sequence Number is too big or negative.	serious
30095	TOE_SRVC_DIVISION_LGST_MISMATCH	User's or/and Customer's Division or/ and Logistic Group ID restricted and do not match.	serious

No.	Error Code	Error Description	Severity
30096	TOE_SRVC_NMNL_WGHT_MISMATCH	Component Nominal Weight does not match total weight of Weight by Freight classes.	warning
30097	TOE_SRVC_DFT_FREIGHT_CLASS	*DFC should not be used as Freight Class ID.	serious
30098	TOE_SRVC_FROM_LA_WEIGHT_EXCEEDED	The specified weight exceeds the limit set by the Container Type for this Load At.	serious
30099	TOE_SRVC_FROM_DC_WEIGHT_EXCEEDED	The specified weight exceeds the limit set by the Container Type for the originating Distribution Center.	serious
30100	TOE_SRVC_FROM_HUB_WEIGHT_EXCEEDED	The specified weight exceeds the limit set by the Container Type for the originating Hub.	serious
30101	TOE_SRVC_TO_CONSIGNEE_WEIGHT_EXCEEDED	The specified weight exceeds the limit set by the Container Type for this Consignee.	serious
30102	TOE_SRVC_TO_DC_WEIGHT_EXCEEDED	The specified weight exceeds the limit set by the Container Type for the destination Distribution Center.	serious
30103	TOE_SRVC_TO_HUB_WEIGHT_EXCEEDED	The specified weight exceeds the limit set by the Container Type for the destination Hub.	serious
30104	TOE_SRVC_THRU_DC_WEIGHT_EXCEEDED	The specified weight exceeds the limit set by the Container Type for a through-point Distribution Center.	serious
30105	TOE_SRVC_THRU_HUB_WEIGHT_EXCEEDED	The specified weight exceeds the limit set by the Container Type for a through-point Hub.	serious
30106	TOE_SRVC_GLOBAL_TRK_LIMIT_WEIGHT_EXCEEDED	The specified weight exceeds the limit set by the global default: maximum truck weight.	serious
30107	TOE_SRVC_FROM_LA_VOL_EXCEEDED	The specified volume exceeds the limit set by the Container Type for this Load At.	serious
30108	TOE_SRVC_FROM_DC_VOL_EXCEEDED	The specified volume exceeds the limit set by the Container Type for the originating Distribution Center.	serious
30109	TOE_SRVC_FROM_HUB_VOL_EXCEEDED	The specified volume exceeds the limit set by the Container Type for the originating Hub.	serious

No.	Error Code	Error Description	Severity
30110	TOE_SRVC_TO_CONSIGNEE_VOL_EXCEEDED	The specified volume exceeds the limit set by the Container Type for this Consignee.	serious
30111	TOE_SRVC_TO_DC_VOL_EXCEEDED	The specified volume exceeds the limit set by the Container Type for the destination Distribution Center.	serious
30112	TOE_SRVC_TO_HUB_VOL_EXCEEDED	The specified volume exceeds the limit set by the Container Type for the destination Hub.	serious
30113	TOE_SRVC_THRU_DC_VOL_EXCEEDED	The specified volume exceeds the limit set by the Container Type for a through-point Distribution Center.	serious
30114	TOE_SRVC_THRU_HUB_VOL_EXCEEDED	The specified volume exceeds the limit set by the Container Type for a through-point Hub.	serious
30115	TOE_SRVC_GLOBAL_TRK_LIMIT_VOL_EXCEEDED	The specified volume exceeds the limit set by the global default: maximum truck volume.	serious
30116	TOE_SRVC_COMP_TYPE_INACTIVE	The specified Component type is inactive.	serious
30117	TOE_SRVC_CUST_DIVISION_MISMATCH	Customer's Division is restricted and does not match TO Division.	serious
30118	TOE_SRVC_USER_DIVISION_MISMATCH	User's Division is restricted and does not match TO Division.	serious
30119	TOE_SRVC_CUST_LGST_GRP_MISMATCH	Customer's Logistic Group is restricted and does not match TO Logistic Group.	serious
30120	TOE_SRVC_USER_LGST_GRP_MISMATCH	User's Logistic Group is restricted and does not match TO Logistic Group.	serious
30121	TOE_SRVC_FT_COLLECT_SCHRG_CUSTOMER	Freight Term cannot be Collect for a Surcharge Based Customer.	serious
30122	TOE_SRVC_UNKNOWN_EQMT_TYPE	Unknown Equipment Type.	serious
30123	TOE_SRVC_NO_AR_SRVC_FOR_SCHG_BASED_CU ST	Preferred AR Service ID cannot be provided for Surcharge based Customer.	serious
30124	TOE_SRVC_JRNY_INVALID_FOR_CUSTOMER	Journey template is not valid for the customer.	serious
30125	TOE_SRVC_EQMT_TYPE_MISMATCH	Specified Equipment Type does not match Carrier/Service/Customer combination.	serious

No.	Error Code	Error Description	Severity
30126	TOE_SRVC_AR_SRVC_NO_CUST	The Service is not defined for the customer.	serious
30127	TOE_SRVC_CANNOT_CLOSE	The Transport Order cannot be closed - it has unassigned containers.	serious
30128	TOE_SRVC_CNTR_ORTN_PRFN_INVALID	Container Orientation Preference value must be between 0 and 9.	serious
30129	TOE_SRVC_CUST_AUTH_FOR_TRANS_DISABLE	The specified customer is not authorized for Transportation.	serious
30130	TOE_SRVC_CUST_COMP_TYPE_UMSR_MISMATCH	Customer's and Component Type Group Units of measurement don't match.	serious
30131	TOE_SRVC_CANNOT_DELETE_CONTAINER	This is the only container assigned to a shipment Deleting this will result in an empty shipment Shipments with a status of Assigned or greater cannot be empty.	serious
30132	TOE_SRVC_INVALID_ADDRESS_ACCEPTED	Address is invalid but accepted.	warning
30133	TOE_SRVC_CARR_SRVC_INCOMPATIBLE	The service is not defined for the specified carrier.	serious
30134	TOE_SRVC_BATCH_NUM_BIG	Batch Number is too big.	serious
40000	LOAD_SRVC_INVALID_STRUCTURE_TYPE	Unsupported structure type supplied to Load Service method.	serious
40001	LOAD_SRVC_GEN_PLAN_ID_FAILED	Plan ID generation failed.	serious
40002	LOAD_SRVC_INVALID_REF_NUM_TYPE	Invalid Reference Number Type.	serious
40003	LOAD_SRVC_LOAD_ID_NOT_FOUND	No Load found for the given Reference Number.	serious
40004	LOAD_SRVC_MULTIPLE_LOAD_ID_FOUND	Multiple Loads found for the given Reference Number.	serious
40005	LOAD_SRVC_INVALID_EVENT	Invalid trigger event for Load.	serious
40006	LOAD_SRVC_STATUS_CHANGE_FAIL	Failed to Advance the Load status.	serious
40007	LOAD_SRVC_SHIPMENT_NOT_FOUND	No Shipments found for the given Plan.	serious
40008	LOAD_SRVC_LOAD_IN_CONFIRMING_ALREADY	Load is already in confirming status.	warning
40009	LOAD_SRVC_LOAD_IN_CONFIRMED_ALREADY	Load is already in confirmed status.	warning
40010	LOAD_SRVC_LOAD_LLD_NOT_CONFIRMED	At least one shipment leg cannot be confirmed.	serious
40011	LOAD_SRVC_LOAD_RR_FAILED	Routing and rating of the Load failed.	serious

No.	Error Code	Error Description	Severity
40012	LOAD_SRVC_LOAD_NO_TARIFF	Error retrieving the corresponding tariff information.	serious
40013	LOAD_SRVC_LOAD_INVALID_RFRC_NUM_TYP	Load confirming failed; Missing Reference number type "MB".	serious
40014	LOAD_SRVC_LOAD_AR_RATE_FAIL	AR rating for shipment failed, confirmation continue.	warning
40015	LOAD_SRVC_ONE_LOAD_TO_CONFIRM_ONLY	Only one load is allowed at a time for load confirmation.	serious
40016	LOAD_SRVC_WEIGHT_EXCEED_VARIANCE	Total weight exceeded weight variance limitation at load confirmation.	serious
40017	LOAD_SRVC_INVALID_MBOL_NUM	The format of the specified MBOL number is invalid.	serious
40018	LOAD_SRVC_LOAD_NOT_ON_PLAN	The specified Load is not attached to a Plan.	serious
40019	LOAD_SRVC_BAD_PLAN	Load is attached to an invalid Plan.	serious
40020	LOAD_SRVC_ADV_TO_PLANNED_BAD_STATUS	Could not set Load to Planned because of current status.	serious
40021	LOAD_SRVC_ADV_TO_PLANNED_NOT_ROUTED	Cannot set Load to Planned until it has been routed.	serious
40022	LOAD_SRVC_CANNOT_REMOVE_FROM_PLAN	Unable to remove Load from Plan.	serious
40023	LOAD_SRVC_NOT_MANIFEST_LOAD	The specified load is not of type manifest load.	serious
40024	LOAD_SRVC_INVALID_MANIFEST_NUM	Invalid manifest number.	serious
40025	LOAD_SRVC_MANIFEST_NUM_NOT_UNIQUE	The manifest number is not unique.	serious
40026	LOAD_SRVC_MANIFEST_CONSOLID_SHPM_FAIL	Shipment consolidation failed during load manifesting.	serious
40027	LOAD_SRVC_MANI_REFNUM_FAIL	Reference number generation failed during load manifesting.	serious
40028	LOAD_SRVC_SHPM_LEG_HOLD	Shipment Leg is on hold and cannot be removed.	serious
40029	LOAD_SRVC_SHPM_LEG_NOT_UNASSIGNED	Shipment Leg cannot be unassigned from load.	serious
40030	LOAD_SRVC_SHPM_LEG_INCORRECT_STATUS	Shipment Leg status does not allow it to be reversed to Null Load Leg.	serious
40031	LOAD_SRVC_BAD_RATING	Rating of the Load is not valid.	serious
40032	LOAD_SRVC_CARR_SRVC_ON_TRIP	Cannot specify Carrier or Service for a Load Leg associated with a Trip.	serious

No.	Error Code	Error Description	Severity
40033	LOAD_SRVC_CARR_SRVC_COMMIT_MISMATCH	Cannot commit an unspecified Carrier or Service.	serious
40034	LOAD_SRVC_UNKNOWN_CARR_SRVC	Unknown Service ID for the specified Carrier.	serious
40035	LOAD_SRVC_PLAN_NOT_OPEN	Plan is not in OPEN status.	serious
40036	LOAD_SRVC_TENDER_NO_DETL	Cannot Tender a Load with no Shipments.	serious
40037	LOAD_SRVC_ATTACH_TO_PLAN_FAIL	Failed to attach Load to a Plan.	serious
40038	LOAD_SRVC_LOAD_SHPM_PLAN_MISMATCH	Load and Shipment Leg are not on same Plan.	serious
40039	LOAD_SRVC_ALREADY_ASSIGNED	Shipment Leg is already assigned to a Load.	serious
40040	LOAD_SRVC_BAD_ASSIGN_HELD_SHPM_LEG	Shipment Leg is on hold and cannot be assigned to the Load.	serious
40041	LOAD_SRVC_BAD_ASSIGN_TO_LOAD	Cannot assign the Shipment Leg to the Load.	serious
40042	LOAD_SRVC_SHPM_APPT_CONFLICT	Conflicting Appointments exist between the Shipment Leg and its associated Pick/Drop Stops.	serious
40043	LOAD_SRVC_ADV_TO_ASSIGNED_BAD_STATUS	Cannot assign Shipment Leg to Load due to current status.	serious
40044	LOAD_SRVC_NOT_OPEN	Load is not in OPEN status.	serious
40045	LOAD_SRVC_CANNOT_BUILD_STOPS	Failed to build stops for the Load.	serious
40046	LOAD_SRVC_CANNOT_SET_CDTY_CD	Failed to set a Commodity Code for the Load.	serious
40047	LOAD_SRVC_INVALID_RANGE_OR_DUPLICATE	Load ID is not in valid range or duplicate.	serious
40048	LOAD_SRVC_AUTO_TDR_ACTIVE	Auto-Tendering is already active for the Load.	serious
40049	LOAD_SRVC_AUTO_TDR_NOT_ENABLED	Auto-tendering is not enabled Load was not tendered.	serious
40050	LOAD_SRVC_AUTO_TDR_NO_MAX_NUM	Maximum number of eligible Carriers for auto-tendering not defined Load was not tendered.	serious
40051	LOAD_SRVC_TENDER_BAD_AUTO_ACCEPT	Unable to auto-accept Load Tender.	serious
40052	LOAD_SRVC_BAD_AUTO_TDR_STAT_UPDATE	Unable to update AutoTender status Load was not Tendered.	serious

No.	Error Code	Error Description	Severity
40053	LOAD_SRVC_AUTO_TDR_SUSP_LOAD	Cannot Auto-Tender a suspended Load.	serious
40054	LOAD_SRVC_CONFIRMING_LOCKED	Load is locked by another user for confirmation.	serious
40055	LOAD_SRVC_AUTO_TDR_NOT_ACTIVE_FOR_LOAD	Load has no active Auto Tender process.	serious
40056	LOAD_SRVC_NO_AUTO_TDR_INFO_FOR_LOAD	Failed to auto tender Load: no associated Auto Tender info.	serious
40057	LOAD_SRVC_RET_TO_PLANNED_LL_NOT_BUILT	Cannot return Load to PLANNED: Load Leg is not a built Load.	serious
40058	LOAD_SRVC_BAD_TFF_SRVC	Tariff Service not found.	serious
40059	LOAD_SRVC_BAD_TFF_SRVC_FOR_CARR	Tariff Service invalid for Carrier.	serious
40060	LOAD_SRVC_LOAD_NOT_PLAN_NOT_OPEN	Cannot remove shipment from a load that is not Open and not Attached to Plan.	serious
40061	LOAD_SRVC_AUTO_TDR_RR_FAILED	Routing and Rating of the Load during auto-tender failed Load was not tendered.	serious
40062	LOAD_SRVC_AUTO_TDR_RR_NO_NOTIF	No Carrier/Tariff/Service from rateshop list has auto notification capability Load was not tendered.	serious
40063	LOAD_SRVC_AUTO_TDR_STATUS_CHANGE_FAIL	Failed to advance the Load status to Tendered.	serious
40064	LOAD_SRVC_ADV_TO_OPEN_BAD_STATUS	Could not set Load to OPEN because of current status.	serious
40065	LOAD_SRVC_RET_TO_OPEN_LL_NOT_BUILT	Cannot return Load to OPEN: Load Leg is not a built Load.	serious
40066	LOAD_SRVC_RET_TO_OPEN_TDR_ACTIVE	Cannot return load to OPEN: current tender must be canceled.	serious
40067	LOAD_SRVC_SET_PLANNED_NO_SHIPMENTS	Cannot Set a Load to Planned with no Shipments.	serious
40068	LOAD_SRVC_AUTO_TDR_CARRS_EXHAUSTED	No more carriers for auto tendering Load was not tendered.	warning
40069	LOAD_SRVC_NOT_FIRST_LOAD_OF_TRIP	Carrier override is allowed for first load of trip only.	serious
40070	LOAD_SRVC_CARR_NOT_ENABLE_FOR_CM	Override carrier has not been enabled for continuous moves processing.	serious
40071	LOAD_SRVC_NEED_CARR_FOR_TENDER_ID	Carrier code must be supplied to identify a Tender.	serious

No.	Error Code	Error Description	Severity
40072	LOAD_SRVC_NEED_OPTLSTAT_FOR_TENDER_ID	Current operational status must be supplied to identify a Tender.	serious
40073	LOAD_SRVC_STOP_NOT_ON_LOAD	Stop does not belong to Load.	serious
40074	LOAD_SRVC_NO_LDNG_INST_ON_DROP_STOP	Loading Instruction cannot be set for a Drop Stop.	serious
40075	LOAD_SRVC_INVALID_LOAD_DATE_TYPE	Invalid date type for RetrieveLoads-ByDate.	serious
40076	LOAD_SRVC_DESCRIPTION_TOO_LONG	Load Description is too long and will be truncated.	warning
40077	LOAD_SRVC_UNKNOWN_DATE_TYPE	Unrecognized date type.	serious
40078	LOAD_SRVC_SHIPMENT_HAS_MEMO	There is memo associated with the shipment.	serious
40079	LOAD_SRVC_ACCEPT_ACTL_COST_EXCEEDED	Cannot Accept Tender because actual cost exceeded the load's cost limit.	serious
40080	LOAD_SRVC_NOT_ALLOWED_RATING	Cannot change rating results because financial processing has been started for the load.	serious
40081	LOAD_SRVC_SPOT_RATE_CURR_MISMATCH	Cannot apply Spot Rate because of currency mismatch.	serious
40082	LOAD_SRVC_SPOT_RATE_EMPTY	Cannot apply Spot Rate because the supplied value is empty.	serious
40083	LOAD_SRVC_MULTILEG_SHIPMENT	Shipments must be single-legged only.	serious
40084	LOAD_SRVC_NOT_THE_SAME_PLAN	Shipments should have the same Plan ID.	serious
40085	LOAD_SRVC_OPEN_STATE	Shipments Entry version should specify that created shipments must be in Processed state.	serious
40086	LOAD_SRVC_CUSTOMER_DIV_LGST	All customers must have the same Division and Logistic group or they should not be restricted to Division and Logistic group.	serious
40087	LOAD_SRVC_ADV_TO_PLANNED_FAILED	Could not set Load to Planned.	serious
40088	LOAD_SRVC_LOAD_NOT_IDENTIFIED	Either Load ID or Load Tracking Number or Master BOL Number or Reference number and Type should be defined to identify the load (only one of them).	serious

No.	Error Code	Error Description	Severity
40089	LOAD_SRVC_LOAD_IDENTIFIER_NOT_UNIQUE	Load cannot be uniquely identified by given values.	serious
40090	LOAD_SRVC_STOP_NOT_IDENTIFIED	Either Shipping Location and Type or Stop Sequence Number should be defined to identify the stop (only one of them).	serious
40091	LOAD_SRVC_STOP_LOCATION_NOT_UNIQUE	Shipping Location and Type is not enough to identify the stop.	serious
40092	LOAD_SRVC_CARRIER_ON_HOLD	Carrier is on hold and cannot be specified.	serious
40093	LOAD_SRVC_CUST_SRVC_CARR_EQMT_NOT_RATED	Load with this combination of Customer/Service/Carrier/Equipment/Scheduled date cannot be rated.	serious
40094	LOAD_SRVC_SHPM_LEG_NOT_IDENTIFIED	Either Shipment Leg ID or Sequence Number or BOL Number should be defined to identify the shipment leg (only one of them).	serious
40095	LOAD_SRVC_REF_FLAG_NOT_TRUE	To identify shipment by reference parameter idsAreReferences must be set to bTRUE.	serious
40096	LOAD_SRVC_NO_LOADS_IN_PLAN	No loads found in plan Please ensure that a specified plan ID exists.	warning
40097	LOAD_SRVC_NO_TDR_INFO_FOR_LOAD	Load is not in proper operational status for Retrieve Tender operation.	warning
40098	LOAD_SRVC_WRONG_LOAD_CARRIER	Wrong Carrier for the Load.	serious
40099	LOAD_SRVC_INCORRECT_OPTLSTAT	Incorrect Load current operational status.	serious
40100	LOAD_SRVC_NOT_DROP_STOP	Stop is not a drop stop.	serious
40101	LOAD_SRVC_NOT_BUILD_LOAD	Cannot perform the operation: Load Leg is not a build Load.	serious
40102	LOAD_SRVC_LOAD_IN_TRIP	Unable to delete the load Load is part of a trip.	serious
40103	LOAD_SRVC_LOAD_NOT_DELETED	Failed to delete load.	serious
40104	LOAD_SRVC_NO_PLAN_ID	Failed to build load Plan not found.	serious
40105	LOAD_SRVC_LOAD_IN_OPTM_REQ	Unable to delete the load Load is included in optimization request.	serious
40106	LOAD_SRVC_CONFIRMSTOP_NOT_IDENTIFIED	Either Stop ID or Shipping Location and Type or Stop Sequence Number should be defined to identify the stop (only one of them).	serious

No.	Error Code	Error Description	Severity
40107	LOAD_SRVC_NO_STOPS_ON_LOAD	Load doesn't have stops.	serious
40108	LOAD_SRVC_INVALID_CREATE_STATUS	Status for a new load can be set to either Planned (default) or to Open.	serious
40109	LOAD_SRVC_COMPLETE_NO_START_END	Load Start and/or End Dates can be specified only if Load's Schedule-Completed flag is false.	serious
40110	LOAD_SRVC_DIFFERENT_DIVISIONS	Shipments must have the same Divisions to be put on one Load.	serious
40111	LOAD_SRVC_DIVISION_NOT_ALLOWED_FOR_USER	Division is not allowed for the User.	serious
40112	LOAD_SRVC_STAUS_AND_ENABLE_NOTIF_FLAG	EnableLoadLevelNotification flag should be specified only if Load status is Tender Accepted.	warning
40113	LOAD_SRVC_WARN_ELEMENT_NOT_EMPTY_IGNORED	Warning: Only SEC code, trailer, driver and seal number can be specified in SEC structure for Confirmation Everything else is ignored.	warning
50000	FNCL_SRVC_INVALID_STRUCTURE_TYPE	Unsupported structure type supplied to Financial Service method.	serious
50001	FNCL_SRVC_COMMIT_FAIL	Failed to Commit the transaction.	serious
50002	FNCL_SRVC_NOF_NOT_CANCELED	NON Operational Freight entity cannot be canceled.	serious
50003	FNCL_SRVC_NOF_CANNOT_BE_DELETED	NON Operational Freight entity cannot be deleted.	serious
50004	FNCL_SRVC_NOF_NOT_DELETED	NON Operational Freight entity not deleted.	serious
50005	FNCL_SRVC_AP_VCHR_NOT_DELETED	NON Operational Freight AP Voucher not deleted.	serious
50006	FNCL_SRVC_NOF_REFERENCE_IGNORED	Reference number of NOF type ignored - will be generated by system.	warning
50007	FNCL_SRVC_TYPE_LOC_ID_ADDRESS_EMPTY	Either Shipping Location type and ID or Address must be specified.	serious
50008	FNCL_SRVC_FRHT_BILL_NUM_NOT_UNIQUE	Freight Bill cannot be uniquely identified by Freight Bill Number.	serious
50009	FNCL_SRVC_FRHT_BILL_NUM_CARR_NOT_UNIQUE	Freight Bill cannot be uniquely identified by Freight Bill Number and Carrier.	serious
50010	FNCL_SRVC_FRHT_BILL_NUM_CARR_NOT_EXIST	Freight Bill cannot be identified by Freight Bill Number and Carrier.	serious

No.	Error Code	Error Description	Severity
50011	FNCL_SRVC_NO_AUDIT_NO_MATCH_FOR_CREATE	Freight Bill Detail cannot be created in Audit-No-Match mode - create NOF first.	serious
50012	FNCL_SRVC_CREATE_DETAIL_WHILE_UPDATE	An attempt is made to create a new Freight Bill Detail while updating - use Create mode.	serious
50013	FNCL_SRVC_NO_DATA_FOR_DETAIL	Either Sequential Number or User Reference Number should be specified to identify Freight Bill Detail.	serious
50014	FNCL_SRVC_FB_ID_NUM_EMPTY	Either Freight Bill ID or Freight Bill Number should be specified to identify parent Freight Bill.	serious
50015	FNCL_SRVC_FB_ID_NUM_CONTRADICT	Freight Bill ID and Freight Bill Number refer to different Freight Bills.	serious
50016	FNCL_SRVC_FBD_ID_NUM_EMPTY	Either Freight Bill Detail ID or Sequential Number should be specified to identify Freight Bill Detail.	serious
50017	FNCL_SRVC_FBD_ID_NUM_CONTRADICT	Freight Bill Detail ID and Sequential Number do not refer to the same Freight Bill Detail.	serious
50018	FNCL_SRVC_DUP_FRHT_BILL_NUM	Freight Bill Number already exists. AP Transaction may fail to be generated for this Freight Bill, the Freight Bill will be invalidated.	warning
50019	FNCL_SRVC_CUST_NO_AUDIT_ENABLED	No vouchers were found which are in eligible status. The Freight Bill Detail will be saved in an un-matched status.	warning
50020	FNCL_SRVC_NO_MATCHES_FOUND	Audit Failed when auditing against Voucher. The Freight Bill Detail amounts are NOT within the defined tolerance limits.	warning
55000	RQ_SRVC_INVALID_STRUCTURE_TYPE	Unsupported structure type supplied to Rate Quotation Service method.	serious
55001	RQ_SRVC_INVALID_VALUE	Invalid number format/value.	serious
55002	RQ_SERVER_RETRUN_CODE	Routing and Rating Server return code (0 means successful).	warning
55003	RQ_SERVER_MESSAGE	Routing and Rating Server return message.	warning
55004	RQ_SERVER_DIAGNOSTIC_MESSAGES	Routing and Rating Server diagnostic messages.	warning

No.	Error Code	Error Description	Severity
55005	RQ_SERVER_CONNECTION_FAILURE	Routing and Rating Server could not be contacted.	serious
55006	RQ_SRVC_CUST_CDTYCODE_NOTMATCH	Commodity does not belong to the customer.	serious
55007	RQ_SRVC_CUST_SRVCODE_NOTMATCH	Service does not exist on the customer.	serious
55008	RQ_SRVC_SUPPORT_SORT_BY_COST_ONLY	Only SC_COST is supported in the current release.	serious
60000	SHPM_SRVC_INVALID_STRUCTURE_TYPE	Unsupported structure type supplied to Shipment Service method.	serious
60001	SHPM_SRVC_INVALID_VALUE	Unrecognized value supplied to the Shipment Service method.	serious
60002	SHPM_SRVC_LOAD_AR_RATE_FAIL	AR rating for shipment failed, confirmation continue.	warning
60003	SHPM_SRVC_LEG_IN_CONFIRMED_ALREADY	Shipment leg is already in confirmed status.	warning
60004	SHPM_SRVC_STATUS_CHANGE_FAIL	Failed to Advance the Shipment Leg status.	serious
60005	SHPM_SRVC_LEG_IN_SUSPEND	Shipment Leg is in suspend status, confirmation is not allowed.	serious
60006	SHPM_SRVC_LEG_RR_FAILED	Routing and rating of the Shipment Leg failed.	serious
60007	SHPM_SRVC_LOAD_NO_TARIFF	Error retrieving the corresponding tariff information.	serious
60008	SHPM_SRVC_INVALID_BOL_NUM	The format of the specified BOL number is invalid.	serious
60009	SHPM_SRVC_LOAD_INVALID_RFRC_NUM_TYP	Load confirming failed; Missing Reference number type "MB".	serious
60010	SHPM_SRVC_CREATE_MANIFEST_LD_FAILED	The Creation of Manifest load failed.	serious
60011	SHPM_SRVC_TO_HAS_MULTIPLE_SHIPMENTS	The Transport Order has more than one shipment.	serious
60012	SHPM_SRVC_BAD_PLAN	Shipment Leg is attached to an invalid Plan.	serious
60013	SHPM_SRVC_ADV_TO_PLANNED_BAD_STATUS	Could not set Shipment Leg to Planned because of current status.	serious
60014	SHPM_SRVC_CANNOT_REMOVE_FROM_PLAN	Unable to remove Shipment Leg from Plan.	serious
60015	SHPM_SRVC_BAD_SHIPMENTLEG	Invalid Shipment Leg ID.	serious

No.	Error Code	Error Description	Severity
60016	SHPM_SRVC_DIVISION_CONFLICT	Shipment Leg's division or logistic group conflict with current plan.	serious
60017	SHPM_SRVC_PLAN_ATTACHED	Shipment Leg already attached to another plan.	serious
60018	SHPM_SRVC_CANCEL_SHPM_INCORRECT_STATUS	Shipment status does not allow to cancel it.	serious
60019	SHPM_SRVC_RETURN_SHPM_INCORRECT_STATUS	Shipment status does not allow to return it to Processed status.	serious
60020	SHPM_SRVC_NONLLD	Only unassigned shipment legs can be selected for optimization.	serious
60021	SHPM_SRVC_WRONG_STATUS	Invalid object status for this operation.	serious
60022	SHPM_SRVC_HELD	Shipment Legs is in "Held" status: operation rejected.	serious
60023	SHPM_SRVC_NO_PLAN	Plan not found.	serious
60024	SHPM_SRVC_PICKUP_NOT_MATCH	Pickup from and to Date does not match.	serious
60025	SHPM_SRVC_DELIVER_NOT_MATCH	Deliver from and to Date does not match.	serious
60026	SHPM_SRVC_CDTY_NOT_MATCH	Commodity Code does not match.	serious
60027	SHPM_SRVC_ASSN_TO_LOAD	Shipment leg cannot be confirmed because its Assign to Load flag is set to TRUE.	serious
60028	SHPM_SRVC_ASSN_TO_LOAD_PLANNED	Shipment leg cannot be set to PLANNED - Tariff Service's Assign to Load flag is TRUE.	serious
60029	SHPM_SRVC_EMPTY_SHPM_IS_INVALID	Shipments with a status of Assigned or greater cannot be empty.	serious
60030	SHPM_SRVC_UNSUSPEND_ASSIGNED_LEG	Shipment Leg already assigned to a load - cannot be Unsuspended.	serious
60031	SHPM_SRVC_INVALID_EVENT	Invalid trigger event for Shipment Leg.	serious
60032	SHPM_SRVC_NO_SHPMLEG_FOR_OPTIMIZED	No shipment legs are selected for optimization, change the loads/trips inclusion flag to true for the request.	serious
60033	SHPM_SRVC_NO_SHIPMENT	Specified Transport Order has no Shipments.	serious
60034	SHPM_SRVC_DIVISION_CONFLICT_PLAN	Shipment Leg's division conflicts with current plan.	serious

No.	Error Code	Error Description	Severity
60035	SHPM_SRVC_LGST_GROUP_CONFLICT_PLAN	Shipment Leg's logistic group conflicts with current plan.	serious
60036	SHPM_SRVC_HEADER_CONTAINERS_IGNORE	Both IgnoreHeader and IgnoreContainers flags to UpdateShipment cannot be TRUE.	serious
60037	SHPM_SRVC_NO_REFERENCE_IN_INPUT	Shipment should be identified by reference but no references specified in input.	serious
60038	SHPM_SRVC_MULTIPLE_REFERENCES_SAME_TYPE	More than one reference number of the same type specified in input.	serious
60039	SHPM_SRVC_SHPM_NOT_FOUND_BY_REFERENCE	No Shipment found for the given Reference Number.	serious
60040	SHPM_SRVC_OUTPUT_NOT_MATCH_RESULT_CONTENTS	Shipment Output List structure does not match specified Result Contents.	serious
60041	SHPM_SRVC_SEQUENCE_EMPTY	List does not contain any entry - at least one must be provided.	serious
60042	SHPM_SRVC_SHPM_DATA_INVALID	Shipments cannot be created with provided data.	serious
60043	SHPM_SRVC_SEQUENCE_NOT_EMPTY	List should not contain any entry.	serious
60044	SHPM_SRVC_ODR_VAL_NOT_EMPTY	Order Value Shipping Info field cannot be specified for Item Level Detail.	serious
60045	SHPM_SRVC_DCLD_VAL_NOT_EMPTY	Declared Value Shipping Info field cannot be specified for Item Level Detail.	serious
60046	SHPM_SRVC_FC_ID_EMPTY_IN_LIST	Freight Class ID cannot be left empty if there is more than one Weight by Freight Class.	serious
60047	SHPM_SRVC_FC_ID_EMPTY_IN_CUSTOMER	No Freight Class ID specified in the Customer profile.	serious
60048	SHPM_SRVC_SHPM_NOT_FOUND_BY_TRKG_NUM	Shipment not found by given Shipment Tracking Number.	serious
60049	SHPM_SRVC_NO_DIVISION_FOR_IDENT	No division specified for non-restricted user to identify Shipment.	serious
60050	SHPM_SRVC_DIVISION_NOT_MATCH_RESTRICTED	Specified division does not match restricted user's division.	serious
60051	SHPM_SRVC_SHPM_NOT_FOUND_BY_RFRC_AND_DIV	Shipment not found by Reference number and Division.	serious
60052	SHPM_SRVC_SHPM_NOT_FOUND_BY_TRKG_NUM_AND_DIV	Shipment not found by Shipment Tracking Number and Division.	serious

No.	Error Code	Error Description	Severity
60053	SHPM_SRVC_SHPM_NOT_UNIQUE_BY_REFERENCE	Shipment cannot be uniquely identified by Reference Number.	serious
60054	SHPM_SRVC_SHPM_NOT_UNIQ_BY_RFRC_AND_DIV	Shipment cannot be uniquely identified by Reference Number and Division.	serious
60055	SHPM_SRVC_SHPM_NOT_UNIQ_BY_TRKG_NUM_AND_DIV	Shipment cannot be uniquely identified by Shipment Tracking Number and Division.	serious
60056	SHPM_SRVC_SHPM_NOT_UNIQUE_BY_TRKG_NUM	Shipment cannot be uniquely identified by given Shipment Tracking Number.	serious
60057	SHPM_SRVC_STATUS_FIELD_NOT_UPDATED	Field's value cannot be updated - Shipment has reached Confirming or greater status.	serious
60058	SHPM_SRVC_CONT_NOT_UNIQUE_BY_ID_OR_REFERENCE	Container cannot be uniquely identified by Element ID or Reference Number.	serious
60059	SHPM_SRVC_SYSTEM_SHIPMENT_NUMBER	Shipment number's prefix SH- is reserved for system use Another number will be generated.	warning
60060	SHPM_SRVC_ONLY_REFERENCES_UPDATED	The shipment's status does not allow any updates except container reference numbers.	warning
60061	SHPM_SRVC_AUTOGEN_REFERENCE_NO_OVRD	BOL number or container tracking numbers - User input not allowed for confirmation of this shipment leg.	warning
60062	SHPM_SRVC_NO_CNTR_RFRC_TRKG	Cannot find the container reference number required to identify the container.	serious
65001	TFF_SRVC_TFFEFTDT_GT_EXPDDT	Tariff effective date is later than expiratory date.	serious
65002	TFF_SRVC_TFFDT_RNG_OVERLAP_OTH_VER	Tariff effective date range overlaps with another tariff version date.	serious
65003	TFF_SRVC_CUST_SURCHRG_CARR	Customer is a 'Surcharge Based on carrier' type. Tariff for this customer is not allowed.	serious
65004	TFF_SRVC_CARR_MANDATORY	Carrier Code field Mandatory.	serious
65005	TFF_SRVC_OBJ_CANNOT_BE_DELETED	The Specified Tariff Entity cannot be deleted.	serious
65006	TFF_SRVC_OPER_NOT_ALLOW_ON_REFTFF	The Specified operation is not allowed on a referential tariff.	serious

No.	Error Code	Error Description	Severity
65007	TFF_SRVC_ISREF_CANNOT_BE_DELETED	The Specified Tariff Service/Charge is being referenced, deletion is not allowed.	serious
65008	TFF_SRVC_SRVC_NOT_IN_MASTER	The Specified Service / Charge is not defined in the master tariff.	serious
65009	TFF_SRVC_INVALID_MSTR_TFF	The Specified Master tariff is invalid.	serious
65010	TFF_SRVC_FAILED_COPY_MASTER_TARIFF	Error occurred while copying service charges from master tariff.	serious
65011	TFF_SRVC_CNFT_ASSN2LD_COMPTRKG	Only one of 'Container Tracking' or 'Assign to Load' Could be True.	serious
65012	TFF_SRVC_COMBO_PRNT_FMT	Print label flag and Shipment label format setting is not consistent.	serious
65013	TFF_SRVC_COMBO_BOL_FMT	Print BOL flag and BOL Format setting is not consistent.	serious
65014	TFF_SRVC_INVALID_MAXMIN_CHRG	Minimum charge dollar is greater than maximum charge dollar.	serious
65015	TFF_SRVC_INVALID_MINMAX_RSTC_PCES	Restriction: Minimum pieces is greater than Maximum pieces.	serious
65016	TFF_SRVC_INVALID_MINMAX_RSTC_WGT	Restriction: Minimum weight is greater than Maximum weight.	serious
65017	TFF_SRVC_INVALID_MINMAX_RSTC_VOL	Restriction: Minimum volume is greater than Maximum volume.	serious
65018	TFF_SRVC_INVALID_MINMAX_RSTC_SKID	Restriction: Minimum pallet is greater than Maximum pallet.	serious
65019	TFF_SRVC_INVALID_MINMAX_RSTC_SHPM	Restriction: Minimum shipments is greater than Maximum shipments.	serious
65020	TFF_SRVC_INVALID_MINMAX_RSTC_LOAD	Restriction: Minimum loads is greater than Maximum loads.	serious
65021	TFF_SRVC_INVALID_MINMAX_RSTC_SHPM_VAL	Restriction: Minimum shipment value is greater than Maximum shipment value.	serious
65022	TFF_SRVC_INVALID_MINMAX_RSTC_ODR_VAL	Restriction: Minimum ordered value is greater than Maximum ordered value.	serious
65023	TFF_SRVC_REFTFF_CHRG_CANNOT_BE_DELETED	Referential tariff condition cannot be deleted.	serious
65024	TFF_SRVC_CHRG_NOT_MULTI_FC	The Condition/Option does not support multi Freight Class.	serious
65025	TFF_SRVC_CHARGE_NOT_TAX	Master condition must not be a tax condition.	serious

No.	Error Code	Error Description	Severity
65026	TFF_SRVC_CUST_CHRGBACK_NOT_APPL	Charge back customer is not application for this tariff type.	serious
65028	TFF_SRVC_INVALID_MLTICOMP_LVL	Invalid multi component level value or value in conflict with data accumulation level setting or multi freight class support.	serious
65029	TFF_SRVC_INVALID_DIM_FIELDS	Either both Dimensional fields must contain a value or neither of them.	serious
65030	TFF_SRVC_INVALID_OVERSIZE_FIELDS	Either both Oversize fields or none has to be non null.	serious
65031	TFF_SRVC_INVALID_MIN_LGST_FREE	Minimum Logistic Free field is not zero when Logistic free is not enabled.	serious
65032	TFF_SRVC_MISSING_NEFT_FIELD	At least one NET effect field must be specified.	serious
65033	TFF_SRVC_INVALID_CHRG_BY_RNG_SET	Condition/Option charge is not by range but range code is specified.	serious
65034	TFF_SRVC_DUPLICATE_CHRG_PRTY	Condition/Option priority setting conflicts with other charges.	serious
65035	TFF_SRVC_INVALID_SCHRG_PRTY	Supercede Condition/Option has to have greater priority.	serious
65036	TFF_SRVC_CHRG_NOT_BY_RANGE	Associated Condition/Option is not rated by range, Range rate cannot be specified.	serious
65037	TFF_SRVC_MINMAX_CHRG_NOT_BYRANGE	Minimum/Maximum charge dollar cannot be changed if it is not charged by range.	serious
65038	TFF_SRVC_BASERATE_NO_CHNG_NONCLIP	Base rate cannot be changed if the rate is not clipped.	serious
65039	TFF_SRVC_INVALID_CHRG_CODE	Invalid condition/option code.	serious
65040	TFF_SRVC_CRSRFRC_FC_FND	Cross references cannot be nested Freight class already is used as a 'Rated As' freight class.	serious
65041	TFF_SRVC_MISSING_LANE_ORIG	Missing origin point for the lane.	serious
65042	TFF_SRVC_MISSING_LANE_DEST	Missing destination point for the lane.	serious
65043	TFF_SRVC_BOTH_LANE_ORIG_SPEC	Both origin zone and hub are provided, please specify only one.	serious
65044	TFF_SRVC_BOTH_LANE_DEST_SPEC	Both destination zone and hub are provided, please specify only one.	serious

No.	Error Code	Error Description	Severity
65045	TFF_SCHG_SRVC_IN_CARR_TFF	Surcharge service cannot be specified for carrier tariff.	serious
65046	TFF_SCHG_CHRG_IN_CARR_TFF	Surcharge charge cannot be specified for carrier tariff.	serious
65047	TFF_SCHG_RATE_IN_CARR_TFF	Surcharge rate cannot be specified for carrier tariff.	serious
65048	TFF_NO_UPDATE_FIPYMT_GENERIC_TFF	Freight invoice payment mode cannot be updated for Generic tariff.	serious
65049	TFF_UPDATE_FOR_ACTIVE_TARIFF	Attempt to update Weight or Length units for Active tariff.	serious
65050	TFF_SRVC_NEW_SRVC_IN_REFF	New Services cannot be specified in Referential Tariff.	serious
65051	TFF_SRVC_COMP_TRKG_MISMATCH	Tracking Level and Container Tracking Indicator do not match.	serious
65052	TFF_SRVC_ASSN2LD_ALWSTOP_CONFLICT	Assn_To_Ld_yn flag must be TRUE if Alw_Stop_yn is TRUE.	serious
65053	TFF_SRVC_CACF_CCF_MC_GENERIC	For Generic Tariffs CarrCslD_Ctl_enu, CustCslD_Ctl_enu, and CslD_PrcnLvl_enu must be NULL.	serious
65054	TFF_SRVC_CACF_NONE_MC_NOT_NULL	For Carrier and CSC Tariffs if CarrCslD_Ctl_enu is CACF_NONE, CslD_PrcnLvl_enu must be NULL.	serious
65055	TFF_SRVC_CCF_NONE_MC_NOT_NULL	For Customer Tariffs if CustCslD_Ctl_enu is CCF_NONE, CslD_PrcnLvl_enu must be NULL.	serious
65056	TFF_SRVC_ALW_STOP_RSTC_MAX_STOP	Allow Stops flag cannot be false if the maximum number of stops on the Restriction is greater than 0.	serious
65057	TFF_SRVC_DLVY_SCHD_BE_EMPTY	Field should be empty according to specified Scheduling Method.	serious
65058	TFF_SRVC_EQMT_OBJ_CANNOT_BE_DELETED	Tariff Service Equipment entity cannot be deleted.	serious
65059	TFF_SRVC_EQMT_TYPE_NOT_UNIQUE	Service Equipment Type is not unique per Tariff/Service.	serious
65060	TFF_SRVC_EQMT_PRTY_NOT_UNIQUE	Service Equipment Priority is not unique per Tariff/Service.	serious
65061	TFF_SRVC_EQMT_LIST_EMPTY	For Carrier Tariffs, at least one Service Equipment Type must be specified.	serious

No.	Error Code	Error Description	Severity
65062	TFF_SRVC_CACF_NOT_NONE_MC_NULL	For Carrier and CSC Tariffs if CarrCslD_Ctl_enu is not CACF_NONE, CslD_PrcnLvl_enu cannot be NULL.	serious
65063	TFF_SRVC_CCF_NOT_NONE_MC_NULL	For Customer Tariffs if CustCslD_Ctl_enu is not CCF_NONE, CslD_PrcnLvl_enu cannot be NULL.	serious
65064	TFF_NO_UPDATE_CARR_CSCT_TFF	Freight invoice payment mode cannot be updated for Carrier or CSCT Tariff when Freight Audit at Charge Level is TRUE for the Carrier owning this Tariff.	serious
65065	TFF_ENABLE_ONLY_FOR_CUSTOMER_TFF	Responsible customer fields are only enabled for Customer Tariff.	serious
65066	TFF_INVALID_ENUM_VAL_FOR_PIECE_SUMMARIZED	Invalid enumerated value since the setting of the Data Accumulation Level on the Master Charge is SUMMARIZED or PIECE.	serious
65067	TFF_INVALID_ENUM_VAL_FOR_STOP	Invalid enumerated value since the setting of the Data Accumulation Level on the Master Charge is STOP.	serious
65068	TFF_CANNOT_REFERENCE_SAME_CUST_CD	Responsible Customer ID must not reference the same Customer ID as the one associated with the Tariff to which the Tariff Charge is attached.	serious
65069	TFF_CANNOT_REFERENCE_SAME_CARR_CD	Payable Carrier ID must not reference the same Carrier ID as the one associated with the Tariff to which the Tariff Charge is attached.	serious
65070	TFF_ENABLE_ONLY_FOR_CARR_CSCT_TFF	Payable Carrier fields only enabled for Carrier or CSCT tariff.	serious
65071	TFF_SRVC_INVALID_RATE_FORSHPM_COMB	Shipment rating is allowed only for cheaper rating method.	serious
65072	TFF_NON_EDITABLE_FOR_REFERENTIAL_TFF	Field not editable for referential tariff.	serious
65073	TFF_NON_CUSTOMER_TFF_FALSE_ONLY	Cannot set this field to true since this is a non-customer tariff.	serious
65074	TFF_RATE_ALREADY_DEFINED_FALSE_ONLY	Cannot set this field to true since rate has been already defined in either master or referential tariff.	serious
65075	TFF_NON_CUST_REF_TFF_NOT_EDITABLE	Cannot edit this field since there are existed non-customer referential tariff.	serious

No.	Error Code	Error Description	Severity
65076	TFF_TRIP_ONLY_SUPPORTED_FOR_CONDITION	Trip level rating can only be applied to conditions.	serious
65077	TFF_TRIP_NOT_VALID_CUST_TFF	Trip level rating cannot be applied to customer tariffs.	serious
65078	TFF_TRIP_SUPERCEDE_NOT_ALLOWED	Trip level ratable charges cannot supercede other charges.	serious
65079	TFF_TRIP_SUPERCEDE_NOT_VALID	A superceded charge cannot be trip ratable.	serious
65080	TFF_TRIP_NO_LOAD_RATEABLE_CONDITIONS	A tariff/service combination must have at least one non-trip level ratable condition.	serious
65081	TFF_SPOT_CARR_CTRC_SRVC	Contracted Tariff Service cannot belong to a Spot Carrier.	serious
65082	TFF_OUT_OF_RANGE_FOUR_DIGIT	Restriction: Value must be between 0 and 9999.	serious
65083	TFF_MINMAX_RSTC_PCES_OUTOFRANGE	Restriction: Minimum/Maximum pieces cannot exceed 9999999.	serious
65084	TFF_MINMAX_RSTC_SKID_OUTOFRANGE	Restriction: Minimum/Maximum pallets cannot exceed 9999999.	serious
65085	TFF_MINMAX_RSTC_SHPM_OUTOFRANGE	Restriction: Minimum/Maximum shipments cannot exceed 9999.	serious
65086	TFF_MINMAX_RSTC_LOAD_OUTOFRANGE	Restriction: Minimum/Maximum loads cannot exceed 9999.	serious
65087	TFF_MINMAX_RSTC_STOP_OUTOFRANGE	Restriction: Maximum stops cannot exceed 9999.	serious
65088	TFF_SRVC_DUPLICATE_EQMT_PRTY	Equipment Type priority setting conflicts with other Equipment Types of the same Service.	serious
65089	TFF_SRVC_LANE_ALREADY_EXIST	Lane already existed.	serious
70000	ADRV_SRVC_INVALID_STRUCTURE_TYPE	Unsupported structure type supplied to Address Validation Service method.	serious
70001	ADRV_SRVC_INVALID_ADDRESS	Invalid address.	serious
70002	ADRV_SRVC_EMPTY_ADDRESS	Empty address.	serious
70100	DS_SRVC_ITNR_POINTS_LESS_THAN_2	Number of Itinerary Points cannot be less than 2.	serious
70101	DS_SRVC_INVALID_ARRIVAL_DATE	Invalid Arrival Date.	serious
70102	DS_SRVC_INVALID_DEPARTURE_DATE	Invalid Departure Date.	serious

No.	Error Code	Error Description	Severity
70103	DS_SRVC_ARRIVAL_DATE_NOT_EMPTY	Arrival Date and Time must be empty.	serious
70104	DS_SRVC_DEPARTURE_DATE_NOT_EMPTY	Departure Date and Time must be empty.	serious
70105	DS_SRVC_ARRIVAL_DATE_EMPTY	Arrival Date and Time cannot be empty.	serious
70106	DS_SRVC_DEPARTURE_DATE_EMPTY	Departure Date and Time cannot be empty.	serious
70107	DS_SRVC_ARRIVAL_BIGGER_DEPARTURE	Departure Date and time must be later or the same as Arrival.	warning
70108	DS_SRVC_ARVL_FROM_BIGGER_TO_DEPARTURE	Arrival From time is later than Arrival To or Departure time.	serious
70109	DS_SRVC_ARVL_TO_BIGGER_DEPARTURE	Arrival To time is later than Departure time.	serious
70110	DS_SRVC_DPTR_FROM_BIGGER_TO	Departure From time is later than Departure To time.	serious
70111	DS_SRVC_INVALID_STRUCTURE_TYPE	Unsupported structure type supplied to Delivery Schedule Service method.	serious
70112	DS_SRVC_ISREF_CANNOT_BE_DELETED	The Specified Delivery Schedule Service object is being referenced, deletion is not allowed.	serious
70113	DS_SRVC_OBJ_CANNOT_BE_DELETED	The Specified Delivery Schedule Service object cannot be deleted.	serious
70114	DS_SRVC_BOTH_ZONE_LOC_SPEC	Both Zone and Location are provided in Itinerary Point, please specify only one.	serious
70115	DS_SRVC_ITNR_POINT_NUMBER_TOO_BIG	Itinerary Point number is too big.	serious
70116	DS_SRVC_ITNR_POINT_NUMBER_TOO_SMALL	Itinerary Point number is too small.	serious
70117	DS_SRVC_ITNR_POINT_NUMBER_INCORRECT	Itinerary Point sequential number incorrect.	serious
70118	DS_SRVC_FIRST_ITNR_POINT_INCOMPATIBLE	First Itinerary Point cannot be Consignee.	serious
70119	DS_SRVC_LAST_ITNR_POINT_INCOMPATIBLE	Last Itinerary Point cannot be LoadAt.	serious
70120	DS_SRVC_ITNR_POINTS_SAME	Itinerary Points cannot be repeated.	serious
70121	DS_SRVC_DLVS_SCHD_MISMATCH	Itnr_TmTbl_DS_Cd in Itinerary_V1 does not match Delivery Schedule.	serious
70122	DS_SRVC_ITNR_POINTS_MISMATCH	Number of Itinerary Points in Itinerary and Timetable does not match.	serious

No.	Error Code	Error Description	Severity
70123	DS_SRVC_NO_ITNR_POINTS	No Itinerary Points for Lane Performance.	serious
70124	DS_SRVC_BUSINESS_DAYS_INVALID	Business Days can be specified only for Elapsed Days Delivery Schedule.	serious
70125	DS_SRVC_INCOMP_TIMETABLE_ENTRY	TimeTable entry date basis does not match Delivery Schedule date basis.	serious
70126	DS_SRVC_NO_TIMETABLE_ENTRY	Either FixedPnt or DayWeek or ElapsedPnt elements must be specified in Timetable Entry.	serious
70127	DS_SRVC_TIMETABLE_ENTRY_LESS_TWO	Number of Timetable Entries cannot be less than two.	serious
70128	DS_SRVC_TIMETABLE_ATTACHED	Timetable cannot be updated because it is attached to Itinerary.	serious
70129	DS_SRVC_ARRIVAL_WEEK_NOT_EMPTY	Arrival week must be empty.	serious
70130	DS_SRVC_ARRIVAL_WEEK_INCORRECT	Arrival week incorrect.	serious
70131	DS_SRVC_BUSINESS_DAYS_MANDATORY	Business Days must be specified for Elapsed Days Delivery Schedule.	serious
70132	DS_SRVC_TMTBL_ENTRY_MISMATCH	Timetable entry cannot have Timetable Entry Points of another type.	serious
70133	DS_SRVC_TMTBL_ITNR_DETACHED	Itinerary has been detached from Timetable during Timetable Delete operation.	warning
70134	DS_SRVC_ITNR_ACTIVE_NOT_DELETED	Itinerary cannot be deleted because of its Active status.	serious
70135	DS_SRVC_REFER_SRVC	Delivery Schedule cannot be deleted because it is referenced in at least one Tariff Service.	serious
70136	DS_SRVC_ARRIVAL_DAY_EMPTY	Arrival day should not be empty.	serious
70137	DS_SRVC_ARRIVAL_DAY_NOT_EMPTY	Arrival day should be empty.	serious
70138	DS_SRVC_DEPARTURE_DAY_EMPTY	Departure day should not be empty.	serious
70139	DS_SRVC_DEPARTURE_DAY_NOT_EMPTY	Departure day should be empty.	serious
70140	DS_SRVC_ELAPSED_DAYS_NOT_ZERO	Number of elapsed days must be zero.	serious
80000	XML_PARSING_ERROR	XML parsing error.	serious

Appendix A

Sample C++ API Client

The following sample client program connects to the CORBA API Server and demonstrates a `RetrieveZone` operation. For this sample client, the project must also include the IDL-generated source files:

- `VIS_c.cpp`
 - `VISCommon_c.cpp`
 - `VISEntity_c.cpp`
 - `VisError_c.cpp`
- `VISTypes_c.cpp`

```
#include <iostream>
#include <fstream>
#include "VIS_c.hh"
#include "VISEntity_c.hh"

int main( int argc, char *argv[] )
{
    CORBA::ORB_var orb;

    try
    {
        VentureFactory_var factory;

        // First, obtain an object reference to the VentureFactory.
        try
        {
            orb = CORBA::ORB_init( argc, argv );
            if( CORBA::is_nil(orb) )
            {
                cout << "ORB pointer is NULL." << endl;
                return 2;
            }

            CORBA::Object_var objref;
            {
```

```

        // Assuming the IOR string is in a file called
        // "VentureFactory.ior" in the current directory.
        // Replace as appropriate.
        ifstream iorFile( "VentureFactory.ior" );
        char iorBuffer[102400];
        iorFile >> iorBuffer;
        iorFile.close();

        objref = orb->string_to_object( iorBuffer );
    }

    if( CORBA::is_nil(objref.in()) )
    {
        cout << "Stringified IOR resulted in NULL object
reference."
            << endl;
        return 2;
    }

    // Narrow the CORBA::Object reference to the stub object.
    factory = VentureFactory::_narrow( objref.in() );
    if( CORBA::is_nil(factory) )
    {
        cout << "Stringified IOR does not narrow to a
VentureFactory"
            << endl;
    }
}
catch( CORBA::SystemException& err )
{
    cout << "Caught a CORBA::SystemException building the obj ref"
        << endl;
    cout << err << endl;
    return 2;
}
catch( ... )
{
    cout << "Caught an unknown exception building the obj ref" <<
endl;
    return 2;
}

// The VentureFactory is now available for use. Login as a
// Transportation Manager user, and retrieve a list of Zones.
try
{
    // Login returns a VentureSession object reference,
    // if successful.
    VentureSession_var session = factory->Connect( "*DFT", "*DFT"
);

    EntitySrvc_var entityService = session->EntSrvc();
    ZoneList_V1 outZones;
    outZones.length(0);
    CORBA::Any outAny;

```

```

// Pre-load the Any with an empty ZoneList_V1 sequence,
// so the server knows which entity we want.
outAny <<= outZones;
VIS::StrIdList ids;
ids.length(2);
ids[0] = CORBA::string_dup( "ALLUSA" );
ids[1] = CORBA::string_dup( "ALLCAN" );
VISError::Details_var errors;
entityService->Retrieve( ids, outAny, errors );
outAny >>= outZones;
size_t zoneCount = outZones.length();
for( size_t i = 0; i < zoneCount; ++i )
{
    Zone_V1& nextZone = outZones[i];
    cout << i << ": " << nextZone.Zn_cd
        << ", " << nextZone.Zn_Desc
        << endl;
}

// Disconnect from the server.
session->Disconnect();
}
catch( VISError::Immediate& err )
{
    cout << "Caught an Immediate error during session" << endl;
    cout << err << endl;
    return 2;
}
catch( CORBA::SystemException& err )
{
    cout << "Caught a CORBA::SystemException during session" <<
endl;

    cout << err << endl;
    return 2;
}
catch( ... )
{
    cout << "Caught an unknown exception during session" << endl;
    return 2;
}
} // End the VentureFactory_var scope.
catch( CORBA::SystemException& err )
{
    cout << "Caught a CORBA::SystemException in VentureFactory's
scope"
        << endl;
    cout << err << endl;
    return 2;
}
catch( ... )
{
    cout << "Caught an unknown exception in VentureFactory's scope"
        << endl;
    return 2;
}

```

```
    }  
    return 0;  
}
```


Appendix B

XML API File Samples

This chapter includes the following sample files:

- [Create Customer XML File](#)
- [Retrieve Customer XML File](#)
- [Create Shipment XML File](#)
- [Sample Java Program](#)

Create Customer XML File

```
<?xml version="1.0"?>
<!DOCTYPE BusinessEntityCreate SYSTEM "urn:/BusEntityCreate.dtd" >
<BusinessEntityCreate>
  <Customer>
    <CustomerCode>XMLTEST</CustomerCode>
    <CustomerDescription>WinRunner</CustomerDescription>
    <LanguageCode>en</LanguageCode>
    <CorporateCustomerFlag Value="bFALSE"/>
    <DefaultProfitCenterCode>WR</DefaultProfitCenterCode>
    <CustomerServiceRepresentativeCode>WR</
CustomerServiceRepresentativeCode>
    <CurrencyCode>USD</CurrencyCode>
    <UnitOfMeasure>
      <SystemUnitOfMeasureEnumVal Value="UMS_IMPERIAL"/>
      <WeightUnitOfMeasureEnumVal Value="UMW_LB"/>
      <LengthUnitOfMeasureEnumVal Value="UML_FT"/>
      <DistanceUnitOfMeasureEnumVal Value="UMD_MILES"/>
    </UnitOfMeasure>
    <AllowShipmentsFlag Value="bTRUE"/>
    <ARCCreditTermsCode>01</ARCCreditTermsCode>
    <BarcodeTypeCode>39</BarcodeTypeCode>
    <RoutingPrecisionEnumVal Value="ROP_GEOGRAPHY"/>
    <StatusVerificationEnumVal
Value="SV_POD_OR_DELVY_NOTIFICATION_NOT_REQUIRED"/>
    <SurchargeBasedVoucherCurrencyEnumVal
Value="VC_TFF_PAYMENT_CNCY"/>
  </Customer>
</BusinessEntityCreate>
```

```

    <TariffSelectionControlCode Value="TS_SPEC_FIRST"/>
    <RateShopTariffSelectionControlEnumVal
Value="TCRS_AR_TARIFF_HIEARCHY"/>
    <CarrierPaymentRequiredFlag Value="bFALSE"/>
    <MaximumVariancePercentPositive>50</
MaximumVariancePercentPositive>
    <MaximumVarianceAmountPositive>500</
MaximumVarianceAmountPositive>
    <MaximumVariancePercentNegative>50</
MaximumVariancePercentNegative>
    <MaximumVarianceAmountNegative>500</
MaximumVarianceAmountNegative>
    <InvoiceGroupLevel Value="IGL_ONE_INVC_PER_VCHR"/>
    <InvoicePaymentModeEnumVal Value="FIPM_AUTO_PAY"/>
    <RespondToTenderHours>0</RespondToTenderHours>
    <EnabledForTransporationFlag Value="bTRUE"/>
    <CarrierPaymentResponsibilityEnumVal Value="FT_PRE_PAID"/>
    <ItemValidationTypeEnumVal Value="VI_NULL"/>
    <ItemLevelDetailEnumVal Value="ILD_NOT_ACTIVE"/>
    <ComponentTypeGroupCode>WR</ComponentTypeGroupCode>
    <DefaultBillToCustomerCode>WRBILLTO</DefaultBillToCustomerCode>
    <SalesPersonCode>WR</SalesPersonCode>
    <DefaultShipmentEntryVersionCode>WR</
DefaultShipmentEntryVersionCode>
    <DefaultCommodityCode>*DFT</DefaultCommodityCode>
    <FreightClassCode> 50 </FreightClassCode>
    <LogisticsGroupCode>WR</LogisticsGroupCode>
    <DivisionCode>WR</DivisionCode>
    <DefaultShipmentEntryTypeCode>WR</DefaultShipmentEntryTypeCode>
    <CustomerStatusEnumVal Value="S_ACTIVE"/>
    <CreatedByUserCode>VENTURE</CreatedByUserCode>
    <UpdatedByUserCode>WR</UpdatedByUserCode>
    <Address>
      <Block>80</Block>
      <Street>WHITEHALL DR</Street>
      <City>MARKHAM</City>
      <State>ON</State>
      <CountryCode>CAN</CountryCode>
      <PostalCode>L3R0P3</PostalCode>
    </Address>
    <Contact>
      <PrimaryTelephoneNumber>905-944-      8088</
PrimaryTelephoneNumber>
      <SecondaryTelephoneNumber>905-944-8088</
SecondaryTelephoneNumber>
      <FaxNumber>944-0364</FaxNumber>
      <EmailAddress>qouser1@i2.com</EmailAddress>
      <URL>www.i2.com</URL>
    </Contact>
    <IgnoreContactsFlag Value="bFALSE"/>
    <ContactPerson>
      <ContactPersonRoleCode>WR</ContactPersonRoleCode>
      <ContactPersonName>WinRunner</ContactPersonName>
      <LanguageCode>en</LanguageCode>

```

```

        <PrimaryTelephoneNumber>905-944-8088</
PrimaryTelephoneNumber>
        <SecondaryTelephoneNumber>905-944-8088</
SecondaryTelephoneNumber>
        <FaxNumber>944-0364</FaxNumber>
        <EmailAddress>qauser1@i2.com</EmailAddress>
        <URL>www.i2.com</URL>
    </ContactPerson>
</Customer>
</BusinessEntityCreate>

```

Retrieve Customer XML File

```

<?xml version="1.0"?>
<!DOCTYPE BusinessEntityRetrieve SYSTEM "urn:/BusEntityRetrieve.dtd" >
<BusinessEntityRetrieve>
    <Customer>
        <CustomerCode>XMLTEST</CustomerCode>
    </Customer>
</BusinessEntityRetrieve>

```

Create Shipment XML File

```

<?xml version = "1.0"?>
<!DOCTYPE ShipmentOrderCreate SYSTEM "urn:/ShipmentOrderCreate.dtd">
<ShipmentOrderCreate>
    <Shipment>
        <SystemShipmentId>26755</SystemShipmentId>
        <CustomerCode>IYCUST</CustomerCode>
        <ShipmentEntryVersionCode>IYOEV</ShipmentEntryVersionCode>
        <ShipmentEntryTypeCode>IY</ShipmentEntryTypeCode>
        <LogisticsGroupCode>IYLG</LogisticsGroupCode>
        <DivisionCode>IYD</DivisionCode>
        <ShipmentEntryModeEnumVal Value =
" TOM_COMP_LEVEL_DETAIL" />
        <ConsigneeGroupCode>SCG1</ConsigneeGroupCode>
        <FreightTermsEnumVal Value = "FT_PRE_PAID" />
        <BillToCustomerCode>IYCUST</BillToCustomerCode>
    <ShipmentConsolidationClassCode> CHD </ShipmentConsolidationClassCode>
        <ShipFromLocationTypeEnumVal Value = "SFT_LA" />
        <ShipFromLocationCode>IYLA</ShipFromLocationCode>
        <ShipFromDescription>IY's Load-At</ShipFromDescription>
        <ShipFromAddress>
            <Block>5</Block>
            <Street>HAHN PL</Street>
            <Unit>316</Unit>
            <City>TORONTO</City>
            <State>ON</State>
            <CountryCode>CAN</CountryCode>
            <PostalCode>M5A4G1</PostalCode>
        </ShipFromAddress>
        <ShipToLocationTypeEnumVal Value = "STT_CONSIGNEE" />
        <ShipToLocationCode>IYCON</ShipToLocationCode>
    </Shipment>
</ShipmentOrderCreate>

```

```

<ShipToDescription>IY's Consignee</ShipToDescription>
<ShipToAddress>
  <Block>6161</Block>
  <Street>BATHURST ST</Street>
  <Unit>1202</Unit>
  <City>NORTH YORK</City>
  <State>ON</State>
  <CountryCode>CAN</CountryCode>
  <PostalCode>M2R1Z5</PostalCode>
</ShipToAddress>
<PickupFromDateTime>01/02/2001@10:00:00</PickupFromDateTime>
<PickupToDateTime>01/02/2001@11:00:00</PickupToDateTime>
<DeliveryFromDateTime>01/04/2001@10:00:00 </DeliveryFromDateTime>
<DeliveryToDateTime>01/04/2001@11:00:00</DeliveryToDateTime>
<CommodityCode>GROC</CommodityCode>
<CurrentShipmentOperationalStatusEnumVal Value = "S_PROCESSED"/>
<CurrentShipmentFinancialStatusEnumVal Value = "S_F_INELIGIBLE"/>
<UnitOfMeasure>
  <SystemUnitOfMeasureEnumVal Value = "UMS_IMPERIAL"/>
  <WeightUnitOfMeasureEnumVal Value = "UMW_LB"/>
  <LengthUnitOfMeasureEnumVal Value = "UML_FT"/>
  <DistanceUnitOfMeasureEnumVal Value = "UMD_MILES"/>
</UnitOfMeasure>
<TariffUnitOfMeasure>
  <SystemUnitOfMeasureEnumVal Value = "UMS_IMPERIAL"/>
  <WeightUnitOfMeasureEnumVal Value = "UMW_LB"/>
  <LengthUnitOfMeasureEnumVal Value = "UML_FT"/>
  <DistanceUnitOfMeasureEnumVal Value = "UMD_MILES"/>
</TariffUnitOfMeasure>
<ShippingInformation>
  <ScaledWeight>100</ScaledWeight>
  <NominalWeight>100</NominalWeight>
  <Skids>1</Skids>
</ShippingInformation>
<MaximumWeight>45000</MaximumWeight>
<MaximumVolume>40000</MaximumVolume>
<ItemGroupCode>IYCUST</ItemGroupCode>
<ComponentTypeGroupCode>IYCUST</ComponentTypeGroupCode>
<IgnoreContainersFlag Value = "bFALSE"/>
<Container>
  <SystemContainerId>49481</SystemContainerId>
  <ContainerTypeCode>SKID</ContainerTypeCode>
  <Quantity>1</Quantity>
  <ContainerShippingInformation>
    <ScaledWeight>100</ScaledWeight>
    <NominalWeight>100</NominalWeight>
    <Skids>1</Skids>
  </ContainerShippingInformation>
  <RatingUnitEnumVal Value = "UT_SKIDS"/>
  <IgnoreContainerOrientationsFlag Value = "bFALSE"/>
  <IgnoreWeightByFreightClassFlag Value = "bFALSE"/>
  <WeightByFreightClass>
<FreightClassNominalWeight> 100 </FreightClassNominalWeight>
  <FreightClassCode> 50 </FreightClassCode>

```

```

        </WeightByFreightClass>
    </Container>
</Shipment>
<IgnoreAllShipmentReferenceNumbersFlag Value = "FALSE"/>
<ResultContents Value = "RES_SHIPMENT_FULL"/>
</ShipmentOrderCreate>
Retrieve Shipment XML File
<?xml version = "1.0"?>
<!DOCTYPE ShipmentOrderRetrieve SYSTEM "urn:/ShipmentOrderRetrieve.dtd"
>
<ShipmentOrderRetrieve>
    <Shipment>
        <SystemShipmentId>21155</SystemShipmentId>
    </Shipment>
    <ShipmentIdType Value="SI_ID"/>
    <ResultContents Value="RES_SHIPMENT_FULL"/>
</ShipmentOrderRetrieve>

```

Sample Java Program

```

/
* A HTTP client to interface i2 XML API interface
*
* This program demonstrates how to use the FMXClientComm class to
* connect to the i2 Transportation Manager web server. It sends a XML
* request and receives the corresponding XML response.
*
* However, this program will not demonstrate how to generate/parse the
* XML data, instead it reads the XML request from a text file constructed
* according to the i2's XMLAPI DTD.
*
* To construct a XML request, please refer to the DTD section of this
* documentation. For a complete XML file, you can refer to the XML sample
* file included in this section.
*
* Copyright (c) 2001 by i2 Technologies Inc. All Rights Reserved.

```

```

import java.util.*;
import java.net.*;
import java.io.*;
import FMXClientComm;

public class TestComm
{
    /** Default constructor. */
    public TestComm()
    {
    }

    public static void main(String[] args)
    {
        try

```

```
{
    if (args.length < 1 || args.length > 2)
    {
        System.out.println("Usage: java TestComm [inputFile]");
        System.out.println("Example:java TestComm
CreateShipment.xml");
    }
    else
    {
        m_XMLfile = args[0];

        if ( m_ServiceURL == null || m_ServiceURL.length() == 0 ||
            m_TMUser == null || m_TMUser.length() == 0 ||
            m_TMPasswd == null || m_TMPasswd.length() ==0 )
        {
            System.out.println("Failed : Missing manadatory value");
            return;
        }

        FMXClientComm client = new FMXClientComm( m_ServiceURL);
        client.addFMXHeader ("TM_USER", m_TMUser);
        client.addFMXHeader ("TM_PASSWORD", m_TMPasswd);
        BufferedReader requestStream =
            new BufferedReader(
                new InputStreamReader(
                    new FileInputStream( m_XMLfile )));

        String s1="";
        String line = requestStream.readLine();
        while (line != null)
        {
            s1 = s1 + line;
            System.out.println("Sending ..." + line);
            line = requestStream.readLine();
        }

        String reply = client.request (s1);

        System.out.println("\n\nReceiving ... \n\n" + reply);

    }
}
catch (FileNotFoundException fnfe)
{
    System.out.println("TestComm - file not found exception");
    fnfe.printStackTrace();
}
catch (Exception e)
{
    System.out.println("TestComm - generic exception");
    e.printStackTrace();
}
}
```

```
    private static String m_ServiceURL="http://MyWebServer:7001/tmXML/  
XMLAPIServlet";  
    private static String m_TMUser="*DFT";  
    private static String m_TMPasswd="DFT";  
    private static String m_XMLfile;  
}
```


Appendix C

API Changes for VisiBroker Support

API Server Port from Orbix to VisiBroker

The port of the API server from Orbix 3.0.1 to VisiBroker 4.1 requires several changes to its installation and setup.

The most visible change is the connection method. Under legacy Orbix, the API server was registered with the Orbix daemon process. A client could use the Orbix-proprietary `_bind()` call, specifying only the host that the server was running on. Although the API server would be assigned a different port each time it was run, the daemon always ran on a fixed port. The client could always connect to the daemon through its known port to resolve a connection to the API server.

Under VisiBroker, clients must use the CORBA-compliant method of connecting through an Interoperable Object Reference (IOR). Accordingly, the API Server has new entries in its parameter set to make its IOR available to clients, and the Flat File Driver has new command-line switches to locate the API Server's IOR.

API Server Parameter Set Entries

To replace the Orbix-proprietary `_bind()` call, several API Server Parameter Set entries are available to configure the connection information.

The `_bind()` call opened a connection with the API server, and returned an object reference for the `VentureFactory`. It took advantage of the fact that it could first communicate with the Orbix daemon, which is available on the named host at a known port.

Starting with release 5.2, the client must obtain an Interoperable Object Reference, or IOR, which contains enough information to connect a client built with any CORBA-compliant ORB to the API Server.

The following parameters have been added to support Visibroker:

- `IORFilePath` (updated)
- `IDInNamingService`
- `ListenerPort`

- NamingServiceHost
- NamingServicePort
- MinThreadsInPool
- MaxThreadsInPool

For details of these parameters, refer to [“API Parameter Set” on page 24](#).

Starting the VisiBroker Name Server

To register the API Server’s IOR with the CORBA naming service, you must start the VisiBroker name server, as configured in the API Server’s Parameter Set. In particular, the name server must be run on the machine specified in NamingServiceHost, at the port specified in NamingServicePort. For example, to start the name server on port 14090, issue the command:

```
nameserv -J-Dvbroker.se.iiop_tp.scm.iiop_tp.listener.port=14090
```

It should be possible to write this argument to a file, and simply name the file. For example, suppose we have a file called `ns.properties` with the contents:

```
-Dvbroker.se.iiop_tp.scm.iiop_tp.listener.port=14090
```

Then we should be able to start the name server with the command:

```
nameserv -config ns.properties
```

Although the `-config` switch worked in VisiBroker 4.0, it does appear to affect the name server’s listener port in VisiBroker 4.1.

Flat File Driver Command Line Parameters

The following flat file driver command line switches have been removed:

Parameter	Description
B	specified a marker for the <code>Orbix _bind()</code> call
DB	specified the database connect string for collocated mode collocated mode is no longer supported.
h	needed for the old <code>Orbix _bind()</code> call <code>_bind()</code> can no longer be used to connect a client to the API Server

In their place, the following list of new parameters is used to specify connection parameters:

- ior
- nsname
- nshost
- nsport

For details of these parameters, refer to [“Flat File Driver Command Line Options” on page 222](#).

The order of precedence of these parameters is:

1. nsname, nshost, and nsport must all be used together to retrieve an IOR from the naming service
2. Either ior or the set of nsname, nshost, and nsport can be specified, but not both. That is, the client must get the IOR to the API Server either from an IOR string file, or from the naming service.
3. If neither of these IOR sources has been specified, a default IOR string filename of “VentureFactory.ior” will be searched for in the current directory.

The flat file driver’s output and error reporting has also been updated. Previously, a few operations always wrote all regular and error output, a few others wrote regular output only if there were no serious or fatal errors (that is, if there was no error output, or if there were only warnings), while the rest of the operations wrote no regular output if there was any error output at all – warnings, or serious or fatal errors.

All operations now write all output – regular and error output – returned by the API Server. Therefore, the flat file driver users will see the same output that the custom CORBA clients will see.

Flat File Driver Daemon Parameters

The changes to the command-line parameters for the Flat File Driver have also been reflected in the Flat File Driver Daemon. Specifically, the following switches have been removed:

Parameter	Description
h	needed for the old Orbix _bind() call. _bind() can no longer be used to connect a client to the API server
DB	specified the database connect string for collocated mode collocated mode is no longer supported

Likewise, the following switches have been added:

- ior
- nsname
- nshost
- nsport

For details of these parameters, refer to [“Flat File Driver Daemon Command Line Options” on page 227](#).

Older versions of the daemon had to invoke the flat file driver in collocated mode. More recent versions supported either running in collocated mode, or as a CORBA client to an external API Server.

As of release 5.2, collocated mode is no longer supported by the flat file driver, so it must now act as a CORBA client to the API Server. As outlined by the command line switches above, both methods of obtaining an IOR to the server (IOR string file or naming service) are supported by the daemon.

Changes to Custom CORBA Clients

Replacing the `_bind()` call in an Orbix 3.0.1 client

Typically, Orbix 3.0.1 (and earlier) clients connected to the API Server with the Orbix-proprietary `_bind()` call:

```
VentureFactory_var factory = VentureFactory::_bind( "", hostname );
```

If this is true for your custom CORBA client, you will have to update it get its initial object reference from a more CORBA-compliant source. The API Server will publish its Interoperable Object Reference (IOR) either in string format to a file, or to a naming service.

To connect using an IOR string file, the `_bind()` call above line can be replaced by something like the following:

```
int dummyArgc = 0;
CORBA::ORB_var orb = CORBA::ORB_init( dummyArgc, NULL, "Orbix" );
// The IOR filename we want to use is stored in an STL string
// named iorFilename
if( iorFilename == "" )
{
    cout << "No connection info supplied." << endl;
    return false;
}

char factoryIORstring[10240];
factoryIORstring[0] = '\0';
{
    FILE* iorFile = fopen(iorFilename.c_str(), "r" );
    if( 0 == iorFile )
    {
        cout << "Unable to open input file for IOR string: ["
            << iorFilename << "]" << endl;
        return false;
    }

    fscanf( iorFile, "%s\n", factoryIORstring );
    fclose( iorFile );
}

if( 0 == factoryIORstring )
{
    cout << "No IOR String found in file: " << iorFilename << endl;
    return false;
}

CORBA::Object_var objref = orb->string_to_object( factoryIORstring
);

if( CORBA::is_nil( objref.in() ) )
{
    cout << "Stringified IOR resulted in NULL object reference." <<
endl;
```

```

        return false;
    }

    // Narrow the CORBA::Object reference to the stub object.
    factory = VentureFactory::_narrow( objref.in() );

    if( CORBA::is_nil(factory) )
    {
        cout << "Stringified IOR does not narrow to a VentureFactory
object"
            << endl;
        return false;
    }

    return true;

```

The key steps here are:

1. Use `CORBA::ORB_init()` to obtain an ORB reference.
2. Read the IOR string from the file.
3. Convert the string to an object reference with `CORBA::ORB::string_to_object()`.
4. Narrow the object reference to a `VentureFactory` reference with `VentureFactory::_narrow()`.

Note: Before exiting your client, remember to call `VentureSession::Disconnect`.

Porting an Orbix 3.0.1 client to VisiBroker 4.1 for C++

To move Orbix 3.0.1 clients to VisiBroker 4.1, there are a few more issues to be aware of.

New IDL compiler command-line parameters.

Where the Orbix IDL compiler could be run with the command:

```
IDL -A VIS.idl
```

The VisiBroker IDL compiler is run as follows:

```
idl2cpp -type_code_info -src_suffix cpp VIS.idl
```

The `-type_code_info` switch causes the IDL compiler to generate Any support.

The `-src_suffix cpp` switch simply causes the compiler to use `.cpp` as the source file suffix, instead of `.cc`. This is useful for MSDEV, which does not recognize `.cc` files as C++ source, by default.

CPP Files

For the file `VIS.idl`, for example, the Orbix IDL compiler generated the following C++ files: `VIS.hh`, `VISC.cpp`, and `VISS.cpp`. The VisiBroker IDL compiler generates the following C++ files, instead: `VIS_c.hh`, `VIS_c.cpp` (assuming the `-src_suffix cpp` switch was used), `VIS_s.hh`, and `VIS_s.cpp`.

Any source or header file in your project which used to `#include` one of the Orbix IDL generated `*.hh` files must be updated to `#include` a `*_c.hh` file now, instead. A client project that used to include `VIS*C.cpp` source files must replace them with the equivalent `VIS*_c.cpp` files. (A server project must include both `VIS*_s.cpp` and `VIS*_c.cpp` files.)

Compiler Command Line Macros

Orbix-specific macros must be removed from the compiler command-line arguments, including: `EXCEPTIONS`, `ORBIX_DLL`, and `USE_ORBXPROT`. These can be found in the MSDEV GUI by selecting Project Settings > C/C++ (Category: Preprocessor) > Preprocessor definitions.

For a project's Win32 – Release configuration, the VisiBroker headers handle the linking with the correct VisiBroker libraries. For the Win32 – Debug configuration, the headers try to link with the VisiBroker debug libraries, but these are not shipped with the distribution. To correct this problem, undefine the `_DEBUG` macro by selecting Project Settings > C/C++ (Category: Preprocessor). In the Preprocessor definitions box, remove the macro `_DEBUG`. Also, in the Undefined symbols box, add the macro `_DEBUG` (simply removing it from the list of definitions may not work).

The VisiBroker documentation suggests defining the `_VIS_NOLIB` macro, instead of changing the `_DEBUG` one, and then explicitly adding the VisiBroker libraries to the link list. This worked for the basic VisiBroker libraries, but not for the naming service library.

itmi.lib

Remove `itmi.lib` from link list, but leave `wsock32.lib` in the list.

CORBA::Any packing/unpacking

For a given user-defined IDL structure, for example, `MyStruct`, CORBA supports the following Any inserters and extractors:

```
void operator<<=(CORBA::Any&, const MyStruct&);
void operator<<=(CORBA::Any&, MyStruct*);
CORBA::Boolean operator>>=(const CORBA::Any&, MyStruct*&);
```

(according to OMG document ptc/98-09-03, adopted January 1999, "C++/C 1.4 RTF Results").

That is, `MyStruct` may be packed into an Any either by copying it, or by giving control of `MyStruct` to the Any (by pointer). `MyStruct` can be extracted from the Any only by pointer.

In contrast, VisiBroker 4.1 generates only the following inserter and extractor:

```
void operator<<=(CORBA::Any&, const MyStruct&);
CORBA::Boolean operator>>=(const CORBA::Any&, MyStruct&);
```

That is, `MyStruct` may only be packed into an Any or extracted from it by copying. There are several possible scenarios for updating existing Orbix 3.0.1 code to VisiBroker 4.1 code, but in general:

- `MyStruct` must be fully populated before copying it to an Any

- `MyStruct` can be retrieved from an `Any` by copying to an allocated `MyStruct` instance: a pointer that has not been allocated can no longer be used

The following code is a simple example of the original Orbix 3.0.1 code:

```
MyStruct* inStruct = new MyStruct;
// Code to populate inStruct has been omitted.
CORBA::Any inAny;
inAny <<= inStruct;
```

This becomes for VisiBroker 4.1:

```
MyStruct inStruct;
// Code to populate inStruct has been omitted.
CORBA::Any inAny;
inAny <<= inStruct;
```

The major difference is that in the Orbix example, `inStruct` is allocated on the heap. The `Any` takes ownership of the pointer on insertion. In the VisiBroker example, `inStruct` is allocated on the stack. The `Any` takes a complete copy of the struct on insertion. The code changes are very straightforward.

Now consider the following more complicated example. This first section of code could be used in an Orbix 3.0.1 client:

```
MyStruct* inStruct = new MyStruct;
CORBA::Any inAny;
inAny <<= inStruct;
// Code to populate inStruct follows here.
```

This alternative section of code works the same way as the previous one, in Orbix 3.0.1:

```
MyStruct* inStruct = new MyStruct;
CORBA::Any* inAny = new CORBA::Any(_tc_MyStruct,
(void*)inStruct, 1);
// Code to populate inStruct follows here.
```

Since `inAny` holds a pointer to `inStruct`, some client applications have been packing the struct into an `Any` first, and later populating the struct. The `Any` holds a pointer to the struct that is being updated, and so with Orbix, the updates are reflected in the `Any`.

VisiBroker 4.1 requires that the `Any` be packed by copying the struct, so the struct must be fully populated before being packed into the `Any`:

```
MyStruct inStruct;
// Code to populate inStruct must be here.
// After fully populating inStruct:
CORBA::Any inAny;
inAny <<= inStruct;
```

Note: Before exiting your client, remember to call `VentureSession::Disconnect`.

Connecting from a VisiBroker client using the IOR String File

The code sample under the heading, replacing the `_bind()` call in an Orbix 3.0.1 client can be used for VisiBroker clients as well, with the one following change:

```
CORBA::ORB_var orb = CORBA::ORB_init( dummyArgc, NULL, "Orbix"
);
```

becomes

```
CORBA::ORB_var orb = CORBA::ORB_init( dummyArgc, NULL );
```

Connecting from a VisiBroker client using the Naming Service

Obtaining the API Server's IOR from the VisiBroker naming service involves two extra steps. First, prepare an argc/argv pair that inform the ORB of where to find the naming service, and supply this argc/argv pair to the ORB_init call:

```
// Build an argc/argv pair to pass to ORB_init()
dummyArgc = 3;
dummyArgv = new char*[dummyArgc];
// cxxStringDup is like strdup, but uses operator new.
dummyArgv[0] = this->cxxStringDup( "MyClientName" );

// The next two arguments allow resolve_initial_references() to find
// the name service, without using the OSAgent.
dummyArgv[2] = this->cxxStringDup( "-ORBInitRef" );

// Assume we receive an ANSI string argument called nsHost that holds the
// hostname running the NameServer, and an integer argument called nsPort
// that holds the TCP/IP port that the NameServer is listening on.
std::string argBuf = "NameService=iioploc://" + nsHost + ":";
char numBuf[20];
sprintf( numBuf, "%d", nsPort );
argBuf += numBuf;
argBuf += "/NameService";
dummyArgv[3] = this->cxxStringDup( argBuf.c_str() );

// Call ORB_init with our argc/argv pair.
CORBA::ORB_var orb = CORBA::ORB_init( dummyArgc, dummyArgv );
Second, replace the code that reads the IOR string from a file and
generates a VentureFactory object reference from it with the following:
// Get a reference to the Naming Service root_context
CORBA::Object_var nsObjRef =
orb->resolve_initial_references( "NameService" );
CosNaming::NamingContext_var nsRoot = 0;
if( !(CORBA::is_nil( nsObjRef.in() ) )
    {
nsRoot = CosNaming::NamingContext::_narrow( nsObjRef );
    }
else
    {
    cout << "Cannot resolve NameService" reference." << endl;
    return false;
    }

// Look up the object reference for the API Server.

// Assume we receive the ID used to register the API Server
```



```

// in an ANSI string argument called idInNamingService.
CosNaming::Name name;
name.length(1);
name[0].id = CORBA::string_dup( idInNamingService.c_str() );
name[0].kind = CORBA::string_dup( "VentureFactory" );

CORBA::Object_var objref = nsRoot->resolve( name );

if( CORBA::is_nil( objref.in() ) )
{
    cout << "Unable to find VentureFactory object reference in
NameService"
        << endl;
}

// Narrow the CORBA::Object reference to the stub object.
VentureFactory_var factory = VentureFactory::_narrow( objref.in() );

if( CORBA::is_nil(factory) )
{
    cout << "Corrupted VentureFactory object reference in
NameService"
        << endl;
}

```

Porting an OrbixWeb 3.2 client to VisiBroker 4.1 for Java

As with a C++ CORBA client, the OrbixWeb proprietary `bind()` call must be replaced with code to retrieve the API Server's Interoperable Object Reference (IOR).

Connecting from a VisiBroker for Java client using the IOR String File

To obtain the API Server's IOR from the IOR string file, you can replace the `bind()` call with code such as:

```

ORB orb=org.omg.CORBA.ORB.init( (String [])null, null );
// Assume the IOR filename is stored in the variable sIORfilename/
BufferedReader breader = new BufferedReader(new
FileReader(sIORfilename));
String sIOR = new String();
sIOR = breader.readLine();
breader.close();

org.omg.CORBA.Object obj = orb.string_to_object(sIOR);
VentureFactory factory = VentureFactoryHelper.narrow(obj);

```

The key steps are:

1. Obtain an ORB reference from `org.omg.CORBA.ORB.init()`.
2. Read the IOR string from the file.
3. Convert the string to an object reference with `orb.string_to_object()`.

4. Narrow the object reference to a VentureFactory reference with `VentureFactoryHelper.narrow()`.

Note: Before exiting your client, remember to call `VentureSession::Disconnect`.

Connecting from a VisiBroker for Java client using the Naming Service

As with the C++ client, obtaining the API Server's IOR from the VisiBroker Naming Service involves two extra steps. First, prepare a string array of arguments that inform the ORB of where to find the naming service, and supply this array to the `ORB.init()` call:

```
// Allocate the String array to hold the arguments.
String[] dummyArgv = new String[2];

// Assume hostname is available in the variable sNShost,
// and port is available in uNSport.
dummyArgv[0] = new String("-ORBInitRef");
dummyArgv[1] = new String("NameService=iioploc://" + sNShost +
":" + uNSport + "/NameService");

ORB orb=org.omg.CORBA.ORB.init(dummyArgv, null);
```

Second, replace the code that reads the IOR string from a file with the following:

```
// Get a reference to the Naming Service root_context
org.omg.CORBA.Object nsObj =
orb.resolve_initial_references("NameService");
NamingContext nsRoot = NamingContextHelper.narrow(nsObj);

// Locate an account manager through the Naming Service.
// Assume "id" is available in the variable sNSid.
NameComponent[] name = new NameComponent[1];
name[0] = new NameComponent();
name[0].id = new String(sNSid);
name[0].kind = new String("VentureFactory");
org.omg.CORBA.Object factoryObj = nsRoot.resolve(name);

// Narrow the CORBA::Object reference to the stub object,
// checking the type along the way using _is_a.
VentureFactory factory = VentureFactoryHelper.narrow(factoryObj);
```

Note: Before exiting your client, remember to call `VentureSession::Disconnect`.

Changes to the Java API Client Library

The Java API Client library itself handles most of the porting from Orbix to VisiBroker automatically. The following changes are visible to the users of this library:

The static `ServiceManager.getServiceManager()` method used to take a single string argument, with the Transportation Manager username, password, and API server hostname all coded into it. This method now takes the following input arguments:

- **String username** - the Transportation Manager username
- **String password** - the password for the supplied username
- **String iorFilename** - the path to a file containing the API Server's IOR string
- **String idInNamingService** - the name with which the API Server will identify itself in the naming service; this corresponds to the API Server parameter set entry `IDInNamingService`
- **String namingServiceHost** - the host running the naming service
- **int namingServicePort** - the TCP/IP port on which the naming service is listening

The arguments `username` and `password` are always mandatory. You must provide `iorFilename` or `idInNamingService`, `namingServiceHost`, and `namingServicePort`.

Note: The method `ServiceManager.disconnect()` method is also provided. Ensure that you call this method before your client exits.

Index

A

- A/P transactions
 - commit [38, 252](#)
 - retrieve [40, 255](#)
 - retrieve quantity [40, 255](#)
 - structure [141](#)
- A/R transactions
 - commit [38, 252](#)
 - retrieve [41, 255](#)
 - retrieve quantity [41, 255](#)
 - structure [142](#)
- accept, tenders [66, 266](#)
- accruals [147](#)
- addresses
 - overriding [207](#)
 - service [30, 229](#)
 - structure [90](#)
 - validate [30](#)
- API
 - installing, CORBA [285](#)
 - overview [13](#)
 - parameter set [24](#)
 - server [221](#)
- architecture [18](#)
- assign to
 - load [46, 259](#)
 - new load [47, 259](#)
 - trip, disable [260](#)
- associations, lane [191](#)
- attach to plan [69, 268](#)
- attributes, updating [21](#)
- auto-applied option [91](#)

B

- Bill of Lading (BOL) [112](#)
- bill-to [122](#)
- block operations [21](#)
- BOL (Bill of Lading) [112](#)
- business days [105](#)

- business hours [93](#)

C

- cancel
 - freight bill details [38](#)
 - freight bills [37](#)
 - shipments [69, 75, 268](#)
 - tenders [67, 267](#)
- carriers
 - create [237](#)
 - cross-reference [139](#)
 - delete [246](#)
 - equipment availability [93](#)
 - retrieve [248](#)
 - set status [249](#)
 - structure [111](#)
 - update [250](#)
- CEA constraints [93](#)
- charges
 - criteria [172](#)
 - detail [173](#)
 - overrides [94](#)
 - possible [68](#)
 - structure [94](#)
 - tariff [197, 199](#)
- charges, freight bill [146](#)
- CIS entity structures [139](#)
- client application, creating [296](#)
- client/server scenarios [285](#)
- clips [194](#)
- codes, error [22](#)
- commit
 - A/P transactions [38, 252](#)
 - A/R transactions [38, 252](#)
 - G/L transactions [38, 252](#)
- Common [291](#)
- common
 - services, flat file [226](#)
 - structures [90](#)
- component type groups

- create 238
- delete 246
- retrieve 248
- update 250
- component types
 - cross-reference 139
 - remove 247
- components
 - reference number 211
 - structure 209
 - tracking number 178
 - types 95
- condensed loads, retrieve 54, 263
- conditions 81
- confirm
 - loads 48, 260
 - shipments 74, 270
 - stops 63
- consignees
 - create 239
 - delete 246
 - retrieve 248
 - set status 249
 - structure 115
 - update 250
- consolidating
 - components 207
 - in hubs 130
 - shipments 207
- constraints, equipment 93
- contacts
 - information 97
 - structure 96
- contacts, load tender 157
- containers
 - orientation 208
 - plan structure 183
 - structure 181
 - updating 88
- continuation flag 225
- CORBA
 - defined 13
 - implementing 285
 - installing API 285
 - objects 221
 - program development 284
 - program execution 284
 - services 283
 - uninstalling 285
- create
 - carriers 237
 - component type groups 238
 - consignees 239
 - customers 240
 - delivery schedule entities 31, 232
 - delivery schedules 30, 230
 - distribution centers 242
 - entities 33
 - equipment types 243
 - fields 90
 - freight bill details 39
 - freight bills 39
 - harmonized tariffs 243
 - hubs 243
 - item group lists 244
 - load plans 259
 - load-ats 244
 - loads 48, 260, 262
 - NMFCs 246
 - NOFs 39, 252
 - objects 19
 - plans 47
 - prebuilt loads 50
 - service equipment 277
 - services 237
 - shipments 76, 271
 - tariff entities 82, 276
 - tariffs 82, 276
 - transport orders 84, 280
- create customer sample 337
- create shipment sample 339
- creating a client application 296
- cross-references
 - carrier 139
 - component type 139
 - customer 139
 - customer location 139
 - equipment type 140
 - location 98
 - reference number type 140
 - service 140
- customer locations, cross-reference 139
- customers
 - carrier and freight audit 97
 - create 240
 - cross-reference 139
 - delete 246
 - retrieve 248
 - set status 249
 - shipping location cross-reference 98

- structure 118
 - update 250
- D**
- daemons
 - flat file 227
 - run log 228
 - data structures
 - contents 235
 - loads 258
 - transport order services 279
 - data transfers
 - entity service 236
 - load services 259
 - transport order 279
 - days of the week
 - timetable point 106
 - deadhead 196
 - deconsolidation 130
 - defaulting values 19
 - deferred mode 22
 - delete
 - carriers 246
 - component type groups 246
 - consignees 246
 - customers 246
 - delivery schedule entities 31, 232
 - delivery schedules 31, 232
 - distribution centers 247
 - entities 34, 246
 - equipment types 247
 - harmonized tariffs 247
 - hubs 247
 - INCO terms 34
 - item groups 247
 - load-ats 247
 - NMFCs 247
 - NOFs 40, 255
 - prebuilt loads 52
 - service equipment 278
 - shipments 77
 - shipments by ID 273
 - shipments by number 273
 - shipments by reference 273
 - shipments by reference and division 273
 - shipments by tracking number 274
 - shipments by tracking number and division 274
 - tariff entities 82, 276
 - tariffs 82, 276
 - transport orders 85, 280
 - delivery schedule entities
 - create 31, 232
 - delete 31, 232
 - retrieve 32, 232
 - update 33, 234
 - delivery schedules 177
 - create 30, 230
 - delete 31, 232
 - detach itineraries 32
 - itinerary 108
 - retrieve 32, 232
 - services 30, 229
 - structures 105, 109
 - update 33, 232
 - dependencies 285
 - destinations, specifying 206
 - detach
 - all itineraries 232
 - itineraries, delivery schedule 32
 - direct 212
 - disable
 - assign to trip 47, 260
 - trip continuation 48, 260
 - distribution centers
 - create 242
 - delete 247
 - retrieve 248
 - set status 249
 - structures 124
 - update 251
 - document type definitions (DTD) 287, 289, 290
 - domain table types 23
 - domiciles
 - equipment 99
 - structure 127
- E**
- elapsed business days, timetable point 106
 - elements 17
 - elementTag 22
 - enabling methods, flat file 226
 - Ent 291
 - entities
 - create 33
 - delete 34
 - structures 111

- structures, CIS 139
- entity IDs, retrieved 140
- entity services 33, 235
 - data transfer 236
- enumerated types 17
- equipment availability 93
- equipment tariff services 205
- equipment types 128
 - create 243
 - cross-reference 140
 - delete 247
 - retrieve 248
 - update 251
- errors
 - code 22
 - fatal 22
 - handling 21
 - message 297
- event reason codes 103
- external aliases 99

F

- FAK 174, 206
- fatal errors 22
- file parameters 225
- Financial 291
- financial
 - services 37, 252
 - structures 141
- fixed point timetables 105
- flat file driver
 - common services 226
 - daemon 227
 - enabling methods 226
 - functions and facilities 226
 - operating as client 226
 - operating as server 226
 - system interfaces 222
- forward-declaring 235
- freight audit, customer and carrier 97
- freight bill details
 - cancel 38
 - charge 146
 - create 39
 - retrieve 42
 - structure 145
 - update 45
- freight bills 43, 44
 - cancel 37

- create 39
- retrieve 41
- structure 143
- update 45
- freight classes
 - cross reference 191
 - entry 206
 - weight by 219
- freight details, structure 174
- functions and facilities, flat file 226

G

- G/L transactions
 - commit 38, 252
 - retrieve 43, 255
 - retrieve quantity 43, 255
 - structure 147
- geocodes 91
- geographic areas 98
- get possible charges 68, 267

H

- harmonized tariffs
 - create 243
 - delete 247
 - retrieve 248
 - update 251
- headers
 - shipment 189
 - shipment leg 163
- holidays 100
- hubs
 - create 243
 - delete 247
 - retrieve 248
 - set status 249
 - structure 130
 - update 251

I

- IDL files 90
- immediate mode 21
- implementing, CORBA 285
- inbound 91
- INCO terms 133
 - delete 34
 - retrieve 35
- installing, CORBA API 285
- instructions, load 155

- integration architecture 18
- item group lists
 - create 244
 - retrieve 248
- item groups, delete 247
- item masters
 - remove 247
 - structure 134
 - update 251
- items 22
 - remove 35
 - shipment 214
- itineraries 112
 - lane performance data 110
 - lane performances 110
 - points 107
 - points, performance update 234
 - step 211
 - timetable 108

J

- java program sample 341
- journey steps 211

K

- key info, shipments 160

L

- lanes
 - association 191
 - availability 100
 - defined 67
 - valid 177
- legs, shipment. *see* shipment legs
- Less Than Truckload (LTL) 19
- Load 292
- load selection criteria 156
- load services 46
- load tenders
 - defined 64
 - notification 159
 - service 65
 - structure 157
- load-ats 135
 - create 244
 - delete 247
 - retrieve 249
 - set status 249
 - update 252

loads

- assign to 46
- build plan 153
- confirm 260
- confirm, services 48
- confirm, structure 154
- create 48, 260, 262
- create data 154
- defined 46
- event, trigger 267
- instruction 155
- manifests, flat file 262
- manifests, service 49
- manifests, structure 155
- plan, create 259
- plan, structure 153, 156
- retrieve 263
- retrieve, plan 56, 264
- retrieve, service 54
- select 57
- services 258
- services, data structures 258
- services, data transfer 259
- set all to planned 57
- set shipment POD 58
- set stop ETA 59
- set stop POD 59
- set stop to delivered 60
- set to planned 60
- shipment update progress 61
- spot rate 49, 156
- structure 149
- tender contact 157
- trigger event 63
- update progress 49, 159, 262
- logistics groups 123
- LTL (Less Than Truckload) 19

M

- manifest loads 49, 262
- match pay 97, 112
- memos 101
- mnemonic 258
- msg 22
- multi-thread processing 30

N

- named result 101
- naming convention 288

NMFC 184
NMFCs
 create 246
 delete 247, 255
 retrieve 249
NOFs
 create 39, 252
 delete 40
 retrieve 44, 255
 stops 171
 structure 166
 update 40
 weight by freight class 171
non-operational freight, *see* NOFs
notification data 159
numbers, component tracking 178

O

objects
 create 19
 retrieving 20
 updating 20
operating
 as client, flat file 226
 as server, flat file 226
optimization, selecting 270
optimizer, send to 73, 270
options, defined 81
origin points, specifying 206
outbound 91
overviews
 process 287
 XML 287

P

parameter set 24
payable carrier override
 example 262, 268
 load service 50
 retrieve 56, 70, 263, 268
 shipment service 69
person 96
planning request structure 191
planning requests 184
plans
 create 47
 load 153, 156
 shipments 78
 structure, shipment 191

PODs 102
possible charges 267
prebuilt loads
 create 50
 delete 52
process overview 287
processing, multi-thread 30

R

ranges, rate 194
rate quotations
 get charges 68
 services 67, 267
 structures 172
RateQuote 292
rates
 code 193
 criteria 174
 range 194
 structure 192
rating 66, 91
 information 170
read only values 20
reason codes 103
recoverable errors 22
reference numbers
 component 211
 structure 102
 types, cross-reference 140
registry keys 285
reject tenders 67, 267
relationships 20
remove
 component types 247
 item masters 247
 items 35
 shipment legs 53, 263
 shipments 53, 262
Req'd field 89
responsible customer override
 example 255, 268
 financials service 40
 retrieve 44, 71, 255, 269
 shipment service 70
 structure 102
restrictions, tariff 195
retrieve
 A/P transactions 40, 255
 A/P transactions, quantity 40, 255

- A/R transactions [41, 255](#)
 - A/R transactions, quantity [41, 255](#)
 - carriers [248](#)
 - component type groups [248](#)
 - condensed loads [54, 263](#)
 - consignees [248](#)
 - customers [248](#)
 - data [236](#)
 - delivery schedule entities [32, 232](#)
 - delivery schedules [32, 232](#)
 - distribution centers [248](#)
 - entities [34, 247](#)
 - equipment types [248](#)
 - errors [236](#)
 - freight bill details [42](#)
 - freight bills [41](#)
 - G/L transactions [43, 255](#)
 - G/L transactions, quantity [43, 255](#)
 - harmonized tariffs [248](#)
 - hubs [248](#)
 - INCO terms [35](#)
 - item group lists [248](#)
 - load-ats [249](#)
 - loads [54, 263](#)
 - loads by date [55, 263](#)
 - loads by description [55, 263](#)
 - NMFCs [249](#)
 - NOFs [44, 255](#)
 - objects [20](#)
 - payable carrier override [56, 70, 263, 268](#)
 - plan loads [56](#)
 - plan loads, flat file [264](#)
 - plan shipments [56, 264](#)
 - responsible customer override [44, 71, 255, 269](#)
 - service equipment [278](#)
 - shipment legs [71, 269](#)
 - shipment legs by ID [71, 270](#)
 - shipments [78, 86, 269](#)
 - tariff entities [83, 277](#)
 - tariffs [83](#)
 - tariffs, flat file [276](#)
 - tenders [65, 264](#)
 - transport order shipments [86, 281](#)
 - transport orders [85, 280](#)
 - retrieve customer sample [339](#)
 - retrieve shipment sample [341](#)
 - retrieved entity IDs [140](#)
 - retrieveTM entity IDs [36](#)
 - return shipment legs from planned [72, 270](#)
 - return to open [264](#)
 - routing [91](#)
- S**
- samples
 - create customer [337](#)
 - create shipment [339](#)
 - java program [341](#)
 - retrieve customer [339](#)
 - retrieve shipment [341](#)
 - SCC (Shipment Consolidation Class) [207](#)
 - schedules
 - delivery [177](#)
 - stop [177](#)
 - SEC [103](#)
 - select
 - for optimization [270](#)
 - loads [57](#)
 - shipment leg for optimization [72](#)
 - selection criteria [178](#)
 - send to optimizer [73, 270](#)
 - serious errors [22](#)
 - server parameters [289](#)
 - service equipment
 - create [277](#)
 - delete [278](#)
 - retrieve [278](#)
 - update [278](#)
 - service exception codes [103](#)
 - services
 - address [30, 229](#)
 - common [226](#)
 - create [237](#)
 - cross-reference [140](#)
 - delivery schedule [229](#)
 - entity [33, 235](#)
 - equipment [277](#)
 - financial [37, 252](#)
 - load [46, 258](#)
 - rate quotation [67, 267](#)
 - shipment [69, 268](#)
 - shipment order entry [271](#)
 - starting and stopping [285](#)
 - tariff [203, 276](#)
 - tariff equipment [205](#)
 - transport order [84, 278](#)
 - Transportation Manager [18](#)
 - set all to planned [264](#)
 - set load instruction [264](#)

- set shipment POD 265
- set status
 - carriers 249
 - consignees 249
 - customers 249
 - distribution centers 249
 - entities 35
 - flat file 249
 - hubs 249
 - load-ats 249
- set stop POD 265
- set stop to delivered 265
- set to planned 266
 - flat file 270
 - shipment legs 73
- set transport order update 86
- shipment
 - services 268
 - structure 212
- Shipment Consolidation Class (SCC) 207
- shipment legs
 - confirmed 180
 - defined 46
 - header 163
 - remove 53, 263
 - retrieve 71, 269
 - retrieve by ID 71
 - return from planned 72
 - select for optimization 72
 - set to planned 73
 - structure 160
 - trigger event 74, 270, 271
 - update 61, 75, 271
- shipment order entry
 - services 75
 - structure 181
- shipment order entry services 271
- shipments
 - cancel 69, 268
 - cancelling 75
 - composition 179
 - composition, change 73, 270
 - confirm 74, 270
 - create 271
 - creating 76
 - delete by ID 273
 - delete by number 273
 - delete by reference 273
 - delete by reference and division 273
 - delete by tracking number 274
 - delete by tracking number and division 274
 - deleting 77
 - entity structure 185
 - header 189
 - item 214
 - itinerary 112
 - key info 160
 - plan structure 191
 - planning 78
 - planning requests 184
 - remove 53, 262
 - retrieve 86, 269
 - retrieve plan 56, 264
 - retrieving 78
 - service 69
 - structure 212
 - structures 178
 - through point 191
 - through points 191
 - update progress 164, 266
 - updating 80
- shipping information 104
- shipping location cross-references 98
- spot rate 49, 156
- SSL 294
- standard data types 18
- starting and stopping API 285
- status, setting 249
- steps 211
- stop confirm
 - data structure 165
 - example 266
- stop events 104
- stops
 - confirm 63
 - defined 46
 - schedule 177
 - structure 164
 - update progress 166
- structTag 22
- structures
 - common 90
 - entity 111
 - financial 141
 - load 149
 - NOF 166
 - overview 89
 - rate quotation 172
 - shipment 178

- tariff 191
 - transport order 206, 216
 - surcharges
 - rate 194
 - service 195
 - structure 194
 - sysMsg 22
 - system
 - interfaces, flat file 222
 - overview 17
- T**
- tariff entities
 - create 82, 276
 - delete 82, 276
 - retrieve 83, 277
 - update 84, 277
 - tariff services
 - flat file 276
 - overview 81
 - structure 203
 - tariffs
 - charge 199
 - charges 197
 - create 82, 276
 - delete 82, 276
 - overview 81
 - restriction 195
 - retrieve 83, 276
 - service equipment 205
 - structures 191, 196
 - update 84, 277
 - tenders
 - accept 66, 266
 - cancel 67, 267
 - flat file 266
 - reject 67, 267
 - retrieve 65, 264
 - service 65
 - threads, multi processing 30
 - through points
 - structure 191
 - through points, specifying 206
 - timetable entries 107
 - timetable point, elapsed business days 106
 - tracking numbers, component 178
 - transactions
 - A/P 141
 - A/R 142
 - G/L 147
 - transport order services, data structures 279
 - transport orders
 - create 84, 280
 - data transfer 279
 - defined 84
 - delete 85, 280
 - retrieve 85, 280
 - retrieve shipments 281
 - services 84, 278
 - set update 86
 - structures 206, 216
 - trigger 87
 - trigger event 282
 - update 87, 281
 - Transportation Manager
 - architecture 18
 - services 18
 - trigger
 - load events 63, 267
 - shipment leg events 74, 270, 271
 - transport orders 87
 - transport orders, event 282
 - trip continuation, disable 260
- U**
- uninstalling, CORBA 285
 - units of measurement structure 104
 - update
 - attributes 21
 - carriers 250
 - component type groups 250
 - consignees 250
 - customers 250
 - delivery schedule entities 33, 234
 - delivery schedules 33, 232
 - distribution centers 251
 - entities 36, 249
 - equipment types 251
 - fields 90
 - freight bill details 45
 - freight bills 45
 - harmonized tariffs 251
 - hubs 251
 - item masters 251
 - itinerary points 234
 - load-ats 252
 - NOFs 40, 45
 - objects 20

- performance [234](#)
- service equipment [278](#)
- shipment legs [75](#), [271](#)
- shipments [80](#)
- tariff entities [84](#), [277](#)
- tariffs [84](#), [277](#)
- transport orders [87](#), [281](#)

V

- valid lanes [177](#)
- values
 - defaulting [19](#)
 - read only [20](#)
- Venture
 - factory [285](#)
 - session [285](#)
- VentureFactory IOR [26](#)
- vffder.exe
 - file parameters [225](#)
 - overview [222](#)
 - services [225](#)
- VIS
 - date [18](#)
 - num [19](#)
 - time [19](#)
 - timestamp [19](#)
 - vbool [18](#)
- VIS_V1 t [19](#)
- VisiBroker [345](#)

W

- warning errors [22](#)
- weights by freight class [219](#)
 - NOFs [171](#)

X

- XML
 - overview [287](#)
 - server parameters [289](#)

Z

- zones
 - in rate quotation service [67](#)
 - structure [138](#)