



ADMINISTRATOR'S GUIDE

Descartes® Route Planner™

17.05

May 2017

© 2017 The Descartes Systems Group Inc. All rights reserved.

Information in this document is subject to change without notice. The Descartes Systems Group Inc. and its affiliates and subsidiaries ("Descartes") make no warranties, express or implied, in respect of this document or its contents.

Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, no part of this document may be reproduced, stored in or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of The Descartes Systems Group Inc..

Descartes may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in any written license agreement from Descartes, the furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property.

Descartes Route Planner, Descartes wGLN, Descartes Mobile, Descartes Global Logistics Network (GLN), Descartes, the phrase "Helping Customers Deliver," and the Descartes triangle logo are either registered trademarks or trademarks of Descartes and/or its affiliates or subsidiaries.

Microsoft, Windows, Windows Server, Active Directory and Internet Explorer are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

Google Maps mapping service is a registered trademark of Google Inc.

HERE maps is a registered trademark of HERE North America, LLC.

The names of other actual companies and products mentioned herein may be the trademarks of their respective owners.

Descartes® Route Planner™

17.05

The Descartes Systems Group Inc.

120 Randall Drive

Waterloo, Ontario Canada, N2V 1C6

Phone: 519-746-8110

Internet: <http://www.descartes.com>

Customer Support

In North America: 1-877-786-9339

Outside North America: +800 -7866-3390

e-mail: servicedesk@descartes.com

Table of Contents

Welcome to Descartes Route Planner	9
Features.....	9
Learning to Use Descartes Route Planner	10
About This Guide	10
Document Conventions	10
Descartes Route Planner Workflow	12
Data Model Introduction	13
Logical Data Areas	13
Schedule Data	15
Supporting Data.....	16
Order Management (Supporting System)	18
Resource Management (Supporting System).....	20
Database Structure	21
Setting up Descartes Route Planner	24
Managing Commodity Constraints	24
Commodity Constraints	24
Managing Routing Parameters.....	26
Routing Parameters	26
Managing Templates.....	30
Schedule Templates.....	30
Route Templates.....	34
Stop Templates.....	44
Managing Bucket Templates	45
Managing Message Templates	46
Resource Template Sets	47
Managing Territories	50
Creating Territories.....	50
Editing a Territory.....	53
Activating and Deactivating a Territory.....	55
Assigning Territories to Schedules and Schedules to Territories.....	55
Deleting a Territory.....	57
Importing and Exporting Territories	57
Managing Territory Swap Sets.....	58
Swap Definition.....	59
No Swap Definition	60

Using Swaps and No Swaps Together	60
Major and Minor Territories.....	61
Zone File Path.....	61
Importing and Exporting Territory Swap Files (TXT or XML).....	61
Managing Datasets.....	63
Data Slice	63
Nestings	66
Values	67
Configuring Background Optimizer (BGO) Properties	68
Configuring the Application	70
Published Settings	71
wGLN Settings	72
AutoPublish Settings	73
Call Out/Notify Settings.....	74
Managing Drivers.....	74
Managing Messages.....	76
Managing Key Wizards.....	77
Customizing Icons for Routes and Stops	79
API Tracking for Resources, Schedule and Orders	80
Setting TreeControl Layout to Org Default	80
Managing Callouts and Notifications.....	82
Creating and Managing Party Users	85
Glossary	88
Appendix A: Database Field Definitions.....	90
General Definitions.....	90
Generic Attributes.....	90
Generic Window Attributes.....	91
Table Definitions	94
FWLocation.....	94
FWLocationWindow	106
FWLocContInvTrack.....	107
FWOrder.....	108
FWOrderLine.....	113
FWOLineDetail	118
FWOrderInfo.....	122
FWOrderLineInfo	123

FWStopTemplate	123
FWActivity	127
FWActivityWindow	135
FWStop.....	137
FWRouteTemplate	151
FWResource	170
FWRoute.....	191
FWRouteHist.....	199
FWSchedule.....	201
FWScheduleTemplate	229
FWBucket.....	254
FWBucketTemplate	256
FWAlert	259
FWAlertCode.....	260
FWDataSetGroup	263
FWDataSetNesting	265
FWDataSetValue.....	267
FWDataSetGroupSetSchedule	270
FWDriver	271
FWCrewAssignment.....	274
FWCostProfile	274
FWCostDefinition	274
FWTrailer.....	276
FWTruck.....	278
FWOrderLeg	282
FWProductType	282
FWRequirementSet.....	284
FWRequirement.....	284
FWAVLParm.....	285
FWCommodityConstraint.....	288
FWGPSStatus.....	288
FWAutoPublish.....	290
FWExtAcntConfig	291
FWCalloutNotifyProfile	292
FWAppConfig.....	294
FWFieldData	294
FWScheduleGroup	294

FWResTemplateSet.....	296
Appendix B: System Value Definitions.....	297
Appendix C: Advise Component Settings	307
Depots (currently needed only if re-charging is allowed)	307
Resources	308
New Job for which to find best time slots	314
Time window buckets.....	317
Commodity Restrictions.....	317
Settings from Current Scheduler Advise (defaults in [Settings] section of adapi initialization file) ...	318
Settings from RMPI (defaults in [Settings] section of adapi initialization file)	318
New Advise Settings (defaults in [Settings] section of adapi initialization file)	318
Debug settings (defaults in [Settings] section of adapi initialization file)	322
Path engine (papi) settings (defaults in [Papi] section of adapi initialization file).....	323
Appendix D: Housekeeping Information	326
Monitoring Descartes Route Planner Components.....	326
Manually Running the RestartCom.exe	326
Automatically Running the RestartCom.exe	328
Configuring RestartCom.exe to Monitor the Memory of Services and Check IIS	330
Setting up Archiving	334
Archiving	334
Archiving Orphan Locations.....	335
Cleaning Up the MessageLog	336
Extracting Data from the LNOSFW Database	337
Silent Installation Using LNOSSetup.exe	339
How It Works	339
Appendix E: Adding Generic Reports to LNOS Integrated Reporting Services/Route Planner.....	341
Preconditions	341
Reports Import	341
RPS Installation.....	343
Activating Customer Organization.....	348
Restoring RPS Functions	349
Generate Additional RPS Database Tables	350
Appendix F: Federated SSO with LNOS and SAML 2	352
General.....	352
Key Benefits.....	352

How to Support F.SSO	352
Pros/Cons of Solution	355
Setup	355
Non-code related	355
Code Related	355
ASP LNOS UI change	355
Proof of concept	355
Requirements	355
Steps and Downloads	355
Step 1: Upgrade and Re-configure LNOS Servers	358
Configure LNOS UI (ASP versions)	359
Verification	360
Step 2: Setup of a Core UI.NET	361
Install Core UI.NET	361
Setup VirDir Instance	361
Quick First Time Test	362
Reconfigure .NET UI to use the DCFSession	365
Setup the FedSSO Application into LNOS URI	366
Step 3: Enable Claims Authentication	366
Required software	366
General Concept	366
Enable Application with WIF	366
Setup LNOS Specific Claims list	373
Setup STS Trust with Application	374
Step 4: Add Claims Mapping Rules	380
Claims Mapping Rules	380
LNOS Claims	380
Claim setup	381
Complete Round Trip Test	381
Appendix G: Support and Handling of Miss Cases	383
API: UnassignRescheduleOrder	383
API: UpdateStatus	383
Appendix H: Mutual Client Authentication between Descartes Route Planner and Descartes wGLN	389
Descartes wGLN Authentication Extensions	389
Mutual SSL Authentication to Descartes wGLN	389

API Key Implementation Descartes wGLN Outbound..... 389

Descartes Route Planner Authentication Extensions 390

Appendix I: Retrieving LNOS Pathing Service Version Information.....392

Welcome to Descartes Route Planner

Descartes® Route Planner™ (formerly known as LNOS Fleetwise™) is a premier enterprise-class Route Planner system. The broad set of Descartes Route Planner features and capabilities are specifically designed to enhance the management and optimization of pick-up and delivery order routing and scheduling. Descartes Route Planner offers:

- a rich user interface
- sophisticated optimization algorithms
- wireless interfaces that help ensure the execution of your plans

Descartes Route Planner is designed to help:

- improve profits and service
- increase Route efficiency
- add flexibility to fleet operations

Features

Descartes Route Planner provides the following key features for an easy-to-use and effective fleet Route Planning system:

- **Easy-to-use Business Document Interfaces**—self contained XML documents with process codes for simplified interfacing
- **Extensive Administrative Functions**—system configuration, settings and data management
- **Pickup and Delivery Reservations Module**—quickly find and reserve available times on your fleet for appointments
- **Route Planning with background optimization**—sophisticated planning and optimization functions that work continuously behind the scenes to improve your Routes
- **Dispatch Management**—view and manage (add, remove, or reassign) work to Routes in progress
- **Wireless Dispatch Interfaces**—easily communicate with drivers in the field to assign new work, get statuses, and monitor Route progress
- **Descartes' Logistics Network Operating System™ (LNOS) Technology**—based on Microsoft®'s latest .NET architecture, LNOS enhances performance, scalability, and interoperability with other Descartes LNOS applications

Learning to Use Descartes Route Planner

For more information about Descartes Route Planner, please refer to the following sources:

Descartes Route Planner Getting Started Guide

This guide introduces novice and intermediate Descartes Route Planner users to the Descartes Route Planner interface, explains basic tasks within the application, and describes how to customize the Descartes Route Planner interface.

Descartes Route Planner User's Guide

This guide builds on the material in the *Getting Started Guide* and focuses on the dispatch and routing functionality of Descartes Route Planner. This guide covers Route planning activities and dispatch activities, including all activities associated with planning and managing Routes being executed by drivers.

Descartes Route Planner Online Help

Descartes Route Planner includes an online help system. It contains explanations of many features and functions within the Descartes Route Planner application, as well as procedures to guide users through the application's basic functionality.

About This Guide

This guide provides information on the administrative functions within Descartes Route Planner, including:

- viewing, filtering, and managing data
- managing templates, codes, and routing parameters
- entering orders

This guide also provides initial field reference information in an appendix.

Document Conventions

This document uses the following conventions:

- Names of windows, frames, dialogs, menus, list boxes, and lists begin with uppercase and are bolded. (**Tool** menu, **Save** button)
- Key combinations appear in mixed case. If the keys are joined by a plus sign (+), press and hold the first key simultaneously with the remaining keys (for example, CTRL+ALT+DEL).
- User-typed text appears in Courier New font. (Enter `USERID` in the login field.)
- Cross-references to other documents, or to sections within the current document, appear in underlined italics. (See *Saving a File* for details.)
- *Italics* are used for emphasis throughout this document.

➡ **Note**— Information important to a particular task or function is introduced with the note format and icon.

❗ **Tip**— Information that may make completing a task easier, but isn't essential to the task, is introduced with the tip format and icon.

⚠ **Warning**— This warning format indicates information that you need to pay particular attention to. Ignoring information presented as a warning could lead to damage to your software or data. Disregarding information presented as a warning may result in damage to your software or data.

Descartes Route Planner Workflow

1 Setting Up Descartes Route Planner

- The administrator sets up the different templates.
- The administrator sets up the Schedules (planning, reservation, dispatch).
- The administrator sets up resources which will also create the Route link between Schedule and resource entities.
- The administrator sets up the buckets.
- The administrator sets up the Depot Locations.

2 Getting Data Into Descartes Route Planner

- The administrator ensures that the necessary data for resources, locations, orders, and buckets (if Reservations is being used) is available to Descartes Route Planner.
- LNOS Dataflow™, an intermediate data transfer application; can be used to support the data transfer into Descartes Route Planner.
- The administrator ensures that new incoming data (orders, resources, reservation windows, etc.) enters Descartes Route Planner using these methods.

3 Assigning Data to a Schedule

- The administrator sets up groups - also called Schedules - to organize the incoming data into logical problem sets.
- Descartes Route Planner translates orders into activities and Stops. It creates empty Routes and Unassigned Stops when resources and orders are imported into Descartes Route Planner.

4 Creating Routes

- The Route planner uses Descartes Route Planner to create routing plans using the empty Routes and Unassigned Stops.
- The Route planner can use several methods to optimize the Routes and schedule Stops according to your organization's established criteria and order requirements.
- The Route planner reviews the Schedules and Routes and verifies that Stops have been assigned to Routes optimally and that resources are being used efficiently.

5 Executing Routes

- Planned Routes and associated data are moved to the dispatch Schedule for execution by the dispatcher.
- The dispatcher can assign new orders and optimize Routes as new orders arrive.
- The dispatcher can communicate with drivers using wireless technology to monitor the progress of Routes and communicate Route changes.

Data Model Introduction

This section addresses the Descartes Route Planner data model with a focus on the following areas:

- high-speed access/updates
- availability
- scalability
- stateless processing
- future functionality and deployment flexibility
- simplicity in usability and manageability
- integration with existing Descartes Route Planner implementations



Note— This section is focused on the data model and not the Descartes Route Planner functional design. In some cases, it includes functional and API references that are needed to explain the proposed data structures.

Logical Data Areas

Examining the scheduling (optimization) problem at a high level means asking the question, “How can the problem be divided into logical data areas?”

The illustration below divides the problem into several logical areas.

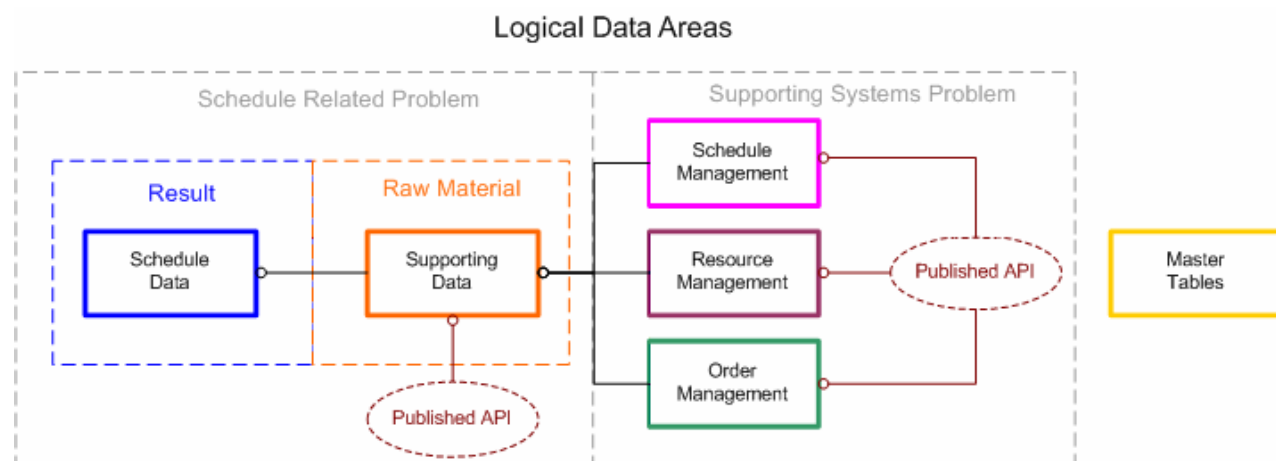


Figure 1: Logical Data Areas

It first separates the Schedule-related problem data from the supporting external system data identified by the large dashed boxes below.

Within the Schedule-related problem, the data are further separated into two additional areas called the result (Schedule data), and the raw material (supporting data).

The supporting system data are also divided into three additional areas: **Schedule**, **Resource**, and **Order Management**. The following sections look at each data area in more detail.

Schedule Data

The Schedule area represents a group of data (tables) that holds all of the Schedule-specific data attributes. The key aspect here is that multiple Schedules are supported for different Schedule types, from reservations to real-time dispatch Schedules. The flexibility exists to have only one Schedule (as with previous Descartes Route Planner (LNOS Fleetwise) versions), or to have as many different Schedules as each implementation requires.

A Schedule data set represents the result for a given optimization problem. Going forward, the Schedule data set tables will be the focal point for all optimization information (Schedule and all associated Bucket, Route, and Stop information needed to import and export all Schedule details).

Supporting Data

The supporting data area represents a group of data tables providing the data needed to select and solve a specific scheduling problem. Think of the supporting data as the raw material of what needs to be done. It supplies the how, the who, and the what, that needs to be done. Different supporting data combinations represent different possible scheduling problems.

At a minimum, the Schedule data and supporting data areas provide enough data points to create and customize different Schedule possibilities. Think of the Schedule data as the result for a possible solution to a specific scheduling problem, just as a cake is the result of a specific recipe. Think of the supporting data as the raw material needed to create the cake: flour, eggs, oven, and a baker to bake the cake. Now think of the baking activity in virtual space where several baking scenarios can be performed using the same raw materials in different amounts and combinations to produce different cakes. The same can be done with different supporting data to produce different Schedules.

Q. Why separate and divide the two?

Q. Why not just feed the data from the external or supporting systems straight into a specific Schedule?

The reason is that a separate supporting data area provides a buffer that can be incorporated with new data coming from the external systems. This provides the flexibility to integrate with most generic order or resource management systems without having them provide all specific data attributes needed by the scheduling technology. This also allows Descartes Route Planner to remove and summarize data attributes at the Schedule level so the optimizer does not need to deal with unnecessary data attributes and table joins.

Example: the current order line details (such as individual measure amounts) can be summarized into one amount at the supporting data level. In addition, the supporting data area provides the capability for any Schedule to be reset to its default state.

➡ **Note—** The raw material (supporting data) can be supplied via external sources such as a published API, Business Documents, or managed from within one of the Descartes Route Planner application components. The supporting systems represent some of the other Descartes Route Planner application components needed to round out the whole scheduling solution.

The current solution includes the following supporting systems:

- Order Element Management
- Resource Element Management
- Schedule Element Management
- Master Tables

Schedule Data

The Schedule data area breaks down into the following detail data tables:

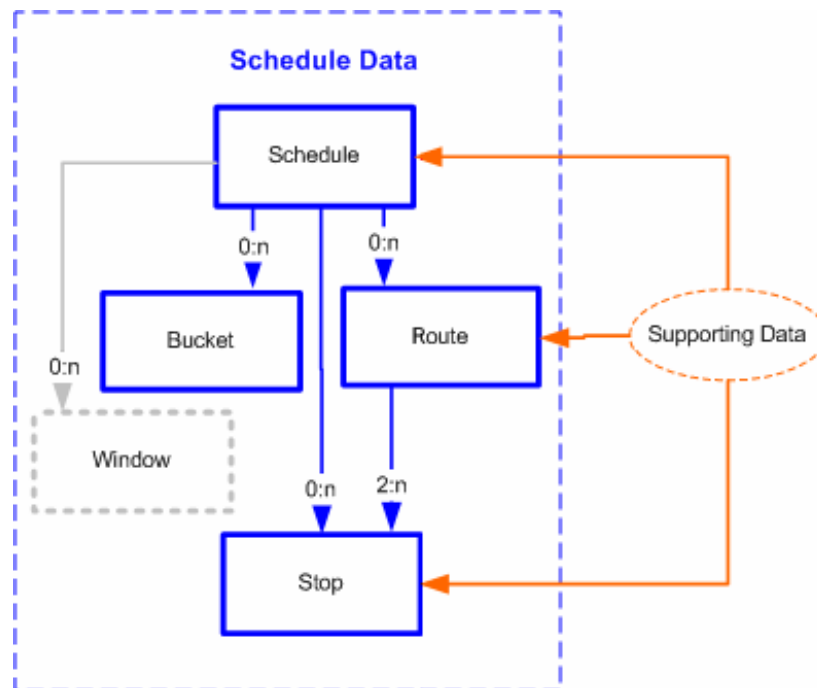


Figure 2: Schedule Data

Schedule— Data that represent specific Schedule rules such as Schedule settings, optimization parameters, business rules, routing parameters, bucket rules, etc.

The Schedule data attributes typically represent the business perspective in a Schedule problem.

- A Route cannot exist without a Schedule.
- A Stop cannot exist without a Schedule.
- A bucket cannot exist without a Schedule.
- A Schedule can exist without a Route, Stop, bucket, or window.
- The Schedule receives its initial attributes from a corresponding (Schedule) supporting data element.
- The Schedule can only reference one corresponding (Schedule) supporting data element.

Route— Data that represent the equipment, personnel, availability, and rules needed to perform the work. The Route data attributes represent the physical perspective in a typical Schedule problem. A Route record is a link record between a specific Schedule and specific corresponding (resource) supporting data element.

- A Route can only belong to one Schedule.
- A Route can have many Stops associated with it.
- The Route can only reference one corresponding (resource) supporting data element.
- A Route will have different time and capacity constraints from its corresponding (resource) supporting data element.

Stop— Data that represent the work needed to be done at a specific location. The Stop data attributes represent the customer order perspective in a typical Schedule problem. A Stop record is a link record between a specific Schedule and specific corresponding supporting data element (activity).

- A Stop can only belong to one Schedule.
- The Stop can only reference one corresponding (activity) supporting data element.
- A Stop will have different time quantity constraints from its corresponding (activity) supporting data element.

Bucket— Data that represent the time-related business rules associated with the availability and user presentation of when the work can be performed.

- The bucket data are optional and only used when the Schedule is used to perform reservation type functionality.
- A bucket must belong to only one Schedule.

Supporting Data

The supporting data are defined in the next section but identified here as a group to indicate the source (raw material) for the different Schedule tables (Schedule, Route, and Stop). Internal rules will be enforced so that the corresponding supporting data

(raw material) is available. The supporting data provides the link back to the corresponding orders, resources, drivers, and customers.

It is possible to override the settings on the different Schedule tables once they have been created. You might want to override settings to fine-tune the way the Schedule is built. Additional functionality to manage the Schedule data will be available through the different Descartes Route Planner applications.

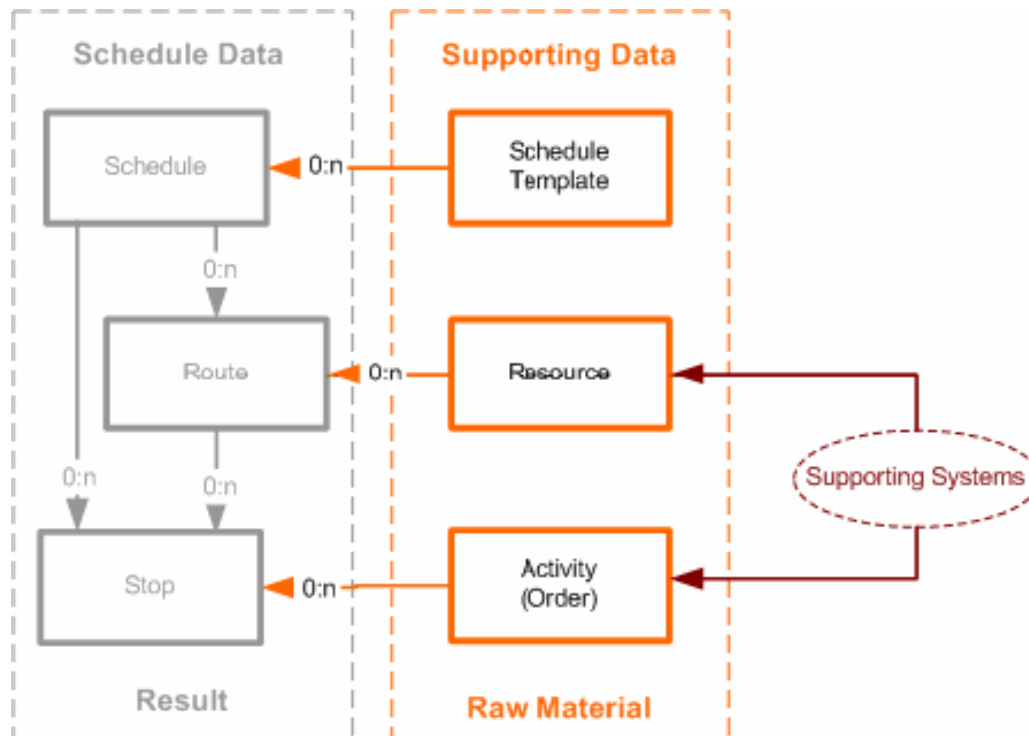


Figure 3: Supporting Data

The supporting data breaks down into the following data structure:

Activity (Orders) — The data representing the order view or a specific shipment leg view that needs to be Scheduled. An activity represents one leg of the order that has a specific geographic location and point in time. As we build out this data model we will see that an external order can have more than one target activity. Typically, an order will have an origin and destination activity, but that doesn't mean that we will always have two activities for each order. We can have one, two, or more. The activity table bridges the gap between external order information and internal scheduling settings. Both external and internal data are combined on each activity record. This is important because the existing customer order records do not need to be cluttered with additional scheduling-specific data details.

- Activity records are created whenever an order comes into Descartes Route Planner. An order can be populated via an API, a Business Document, a third party order entry system, or a Descartes Route Planner order management system. Typically all Descartes Route Planner orders will be populated via Business Documents.
- All order line items can be consolidated into one activity record.
- Additional scheduling attributes can be introduced at the activity level.
- One Activity can be related to another Activity through the same order key relationship.
- This relationship would typically be used to relate a pick-up with a drop-off leg of a double-ended type order.
- In the future, this relationship can be used to model order splitting where different order lines can be split across different activities.
- Separating the activity from the order allows us to potentially support:
 - Blanket orders
 - Consignment orders
 - Multi-task orders

Resource— The data that represents a consolidated physical view of the thing (Resource) that actually performs the work. Typically, this would include the physical vehicle, driver, some time restrictions, and scheduling attributes. This is a little different from the previous resource representations, particularly with respect to the date aspect. The new resource can be used as the default state of a Route that will be associated with it.

➡ **Note**— The resource can only be linked to a Schedule through a Route record.

- The resource records can be populated via an API, Business Documents, a third party resource management systems, generated via Route Templates (using the Descartes Route Planner UI).
- When a resource is created and a Route Template specified, many of the scheduling attributes will be initialized to the Route template values.

Schedule Template— A table where different Schedule templates with different scheduling attributes can be created. Once created and referenced, new Schedules can be created based on the Schedule template settings. Many of the Schedule attributes can be predefined, minimizing the effort needed to create Schedules from scratch. For the most part, the Schedule and Schedule Template attributes are the same and used as defaults during the Schedule creation process.

Order Management (Supporting System)

Descartes Route Planner preserves certain Order-related tables internally. These tables are needed for internal processing and display purposes for other Descartes

Route Planner applications, such as Dispatch. The current Descartes Route Planner order management data set breaks down into the following data structure:

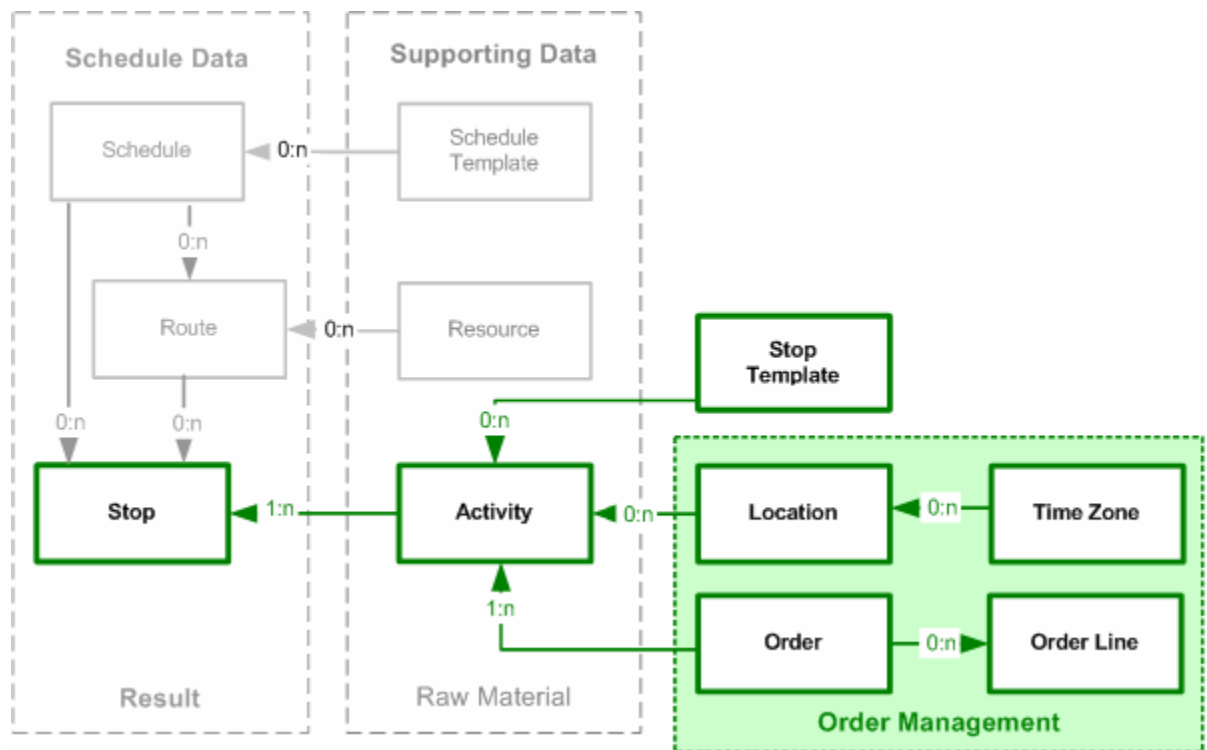


Figure 4: Order Management

Order— This data table holds the customer order header type information.

An order within Descartes Route Planner represents a common unit of work that needs to be Scheduled on one resource. We can have many activities but all the activities must be handled as a unit by the same resource. Each activity can have its own Schedule settings and location details but they must ship on the same physical resource.

Order Line— This data table holds the order line information (quantity ordered and specific item details).

Descartes Route Planner does not provide a table to preserve a master copy of the item (product) list. The reason for this is that the item master table is maintained externally and the required attributes have been included with every Order Line record.

Location— This is a master type table that holds the customer location-specific information.

- The Location table is required and associated with the Activity directly, not with the Order. The reason for this is to support the complex order type of multiple tasks (activities). For example, an order typically supports two locations, which are represented by the activities, but a complex order may have three or more activities associated with one order. This data structure is well positioned to support order splitting in the future.

Stop Template— This data table holds the default scheduling attributes to be used during any scheduling function. Whenever an activity is created it will utilize the Stop template values as needed.

Activity— Please see the previous *Supporting Data* section for details.

Resource Management (Supporting System)

Currently, Descartes Route Planner has a basic resource management system with its own data elements that are needed to support the other Descartes Route Planner application components. The resource management system consists of the following data tables: Route Template, Location, and Time Zone.

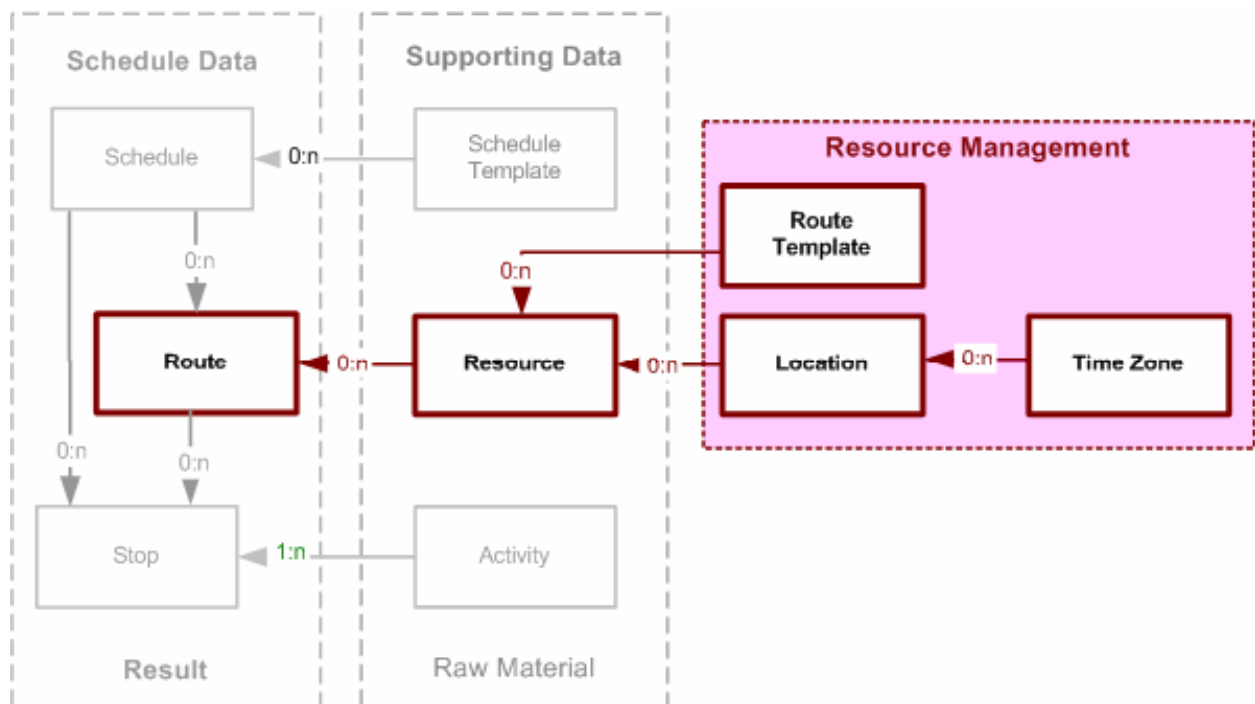


Figure 5: Resource Management

Route Template— This data table holds the default scheduling attributes to be used during any scheduling function. Whenever a resource is created and a Route template is specified, it will utilize the Route template values as needed.

Location— This is a master type table that holds the depot location-specific information.

Time Zone— This is a master type table that holds the time zone references. Please see the *Order Management* section for details.

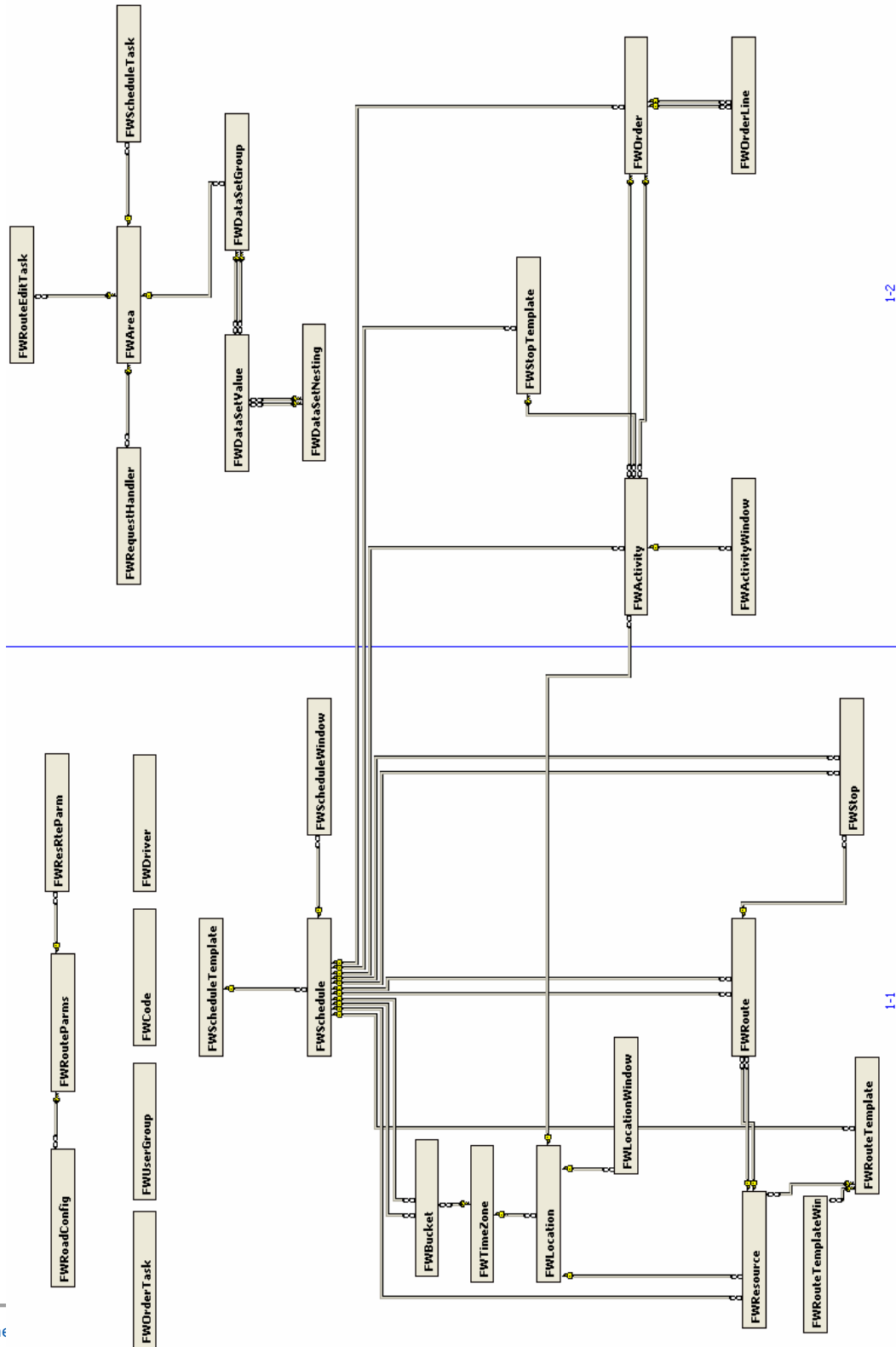
Resource— Please see the previous *Supporting Data* section for details.

Route— Please see the previous *Schedule Data* section for details.

➡ **Note**— The Route records are used to maintain the link between a specific Schedule and resource records.

Database Structure

The following diagram depicts the relationships between the Descartes Route Planner Database tables.



The main relationships that must be taken into consideration are:

Order Creation

The affected tables are:

FWOrder

FWOrderLine (1..n)

FWLocation (1..n)

FWLocationWindow (1..n)

FWActivityWindow (1..n)

Every Order created, will generate the following records:

FWActivity

One record for every leg of the order (i.e. double-ended jobs will have two activity records, one for the Pick-up and one for the Delivery).

FWStop

One record for every FWActivity created.



Note— FWStop.ScheduleKey field will be null if assigned to a Route.

Resource creation

The affected Tables are:

FWResource

FWLocation (1..n. Initial and Final stations)

Every resource will generate the following records:

FWRoute

FWStop (1..n. One record for the initial and one for the final station)

Setting up Descartes Route Planner

The Descartes Route Planner administrator uses the Administrator options to set up Descartes Route Planner. Administrator options are used during the initial setup of Descartes Route Planner and to add or modify background data.

Managing Commodity Constraints

Commodity Constraints

Commodity rules allow the user to specify that certain Orders should not be on a Resource at the same time as other Orders because of a particular characteristic. For instance, lawn chemicals and bananas should not be carried on the same truck, or perhaps frozen and ambient items cannot be on the same vehicle at the same time. Additionally, certain Depots may not be prepared to handle certain commodities.

Elements and Attributes

The use of commodity rules requires that commodities be identified on individual Orders and Depots. The rules regarding which commodities conflict with others are specified in this section.

Behavior

Because of the restrictions placed on the use of Resources by commodity rules, there will often be a need to have a Resource visit a Depot during its Route to unload Orders with one kind of commodity so that it can subsequently serve Orders with a conflicting commodity. This is handled by recharging. Only Orders with the commodities supported by the Depot can be dropped off or picked up at the Depot.

More formally, recharging works as follows:

In order for a Resource to serve a dropoff-only Order with a particular commodity, the Resource must have previously visited a Depot that supports the Order's commodity. This could either be the Depot of the Resource's InitialStation or a Depot at which a recharge occurred. Of course, if it serves a dropoff-only Order with a conflicting commodity between that Depot visit and the new Order, then it must recharge (thus dropping off the prior Order). Similarly, after picking up a pickup-only Order, a Resource must recharge at a Depot with the Order's commodity before serving an Order with a conflicting commodity.

For multi-task Orders a recharge is not required between Orders with conflicting commodities.

Please note that a Depot *can* support conflicting commodities.

It is important to remember that capabilities and requirements on Depot and Resources will limit the recharge options.

Limitations

There are two known limitations to the recharging behavior associated with commodity rules. The first is that a Resource will not make an additional recharge at the end of the Route to unload any Orders with commodities that cannot be served by its FinalStation. Thus, while there will not be commodity conflicts between the Orders on the segment between its second to last Depot visit and the end of the Route, it will make a special detour to unload pickup-only Orders at a Depot that has the commodities on those Orders.

To access Commodity Rules:

- 1 From the main menu, select: **Data > Commodities**.

The list of Commodity Rules appears:

Data / Commodities				
Commodities				Find
Key	Description	ConflictKey	Schedule Key:	Active
DRY		FROZEN	ZSV_13_2_1_2	True
FROZEN		DRY		True
Water	water	frozen	a	True
888	water	frozen	a	True
Water	water	frozen	a_copy	True
888	water	frozen	a_copy	True

If no rules are specified, an empty list will appear with a **New** button in the center of the page.

- 2 Right-click on a selected commodity rule record and select one of the management options: **New**, **Edit**, **Clone** or **Delete**. It is also possible to double-click a specific commodity rule record to directly edit that record.

When double-clicked, the selected Commodity Rule page appears:

Commodity Constraint

Key

Description

ConflictKey

Active

☐

CLEANING TIME AT RECHARGE DEPOT

Cleaning Time:

Minutes

Sequence-dependent:

☐

SCHEDULES

ScheduleKey
RechargeDemo
DispatchingSchedule

ScheduleKey

NEW

DELETE

- 3 Enter the appropriate values on this page:

Key: Specify the Commodity Key.

Description: Specify a Description for this rule.

Conflict Key: Specify the Commodity that conflicts with the Specified above.

Active: Activate or Deactivate this rule

Cleaning Time: Specify the cleaning time required at the recharge depot. Select the checkbox if this is sequence-dependent.

Schedule Key: Select a schedule from the ScheduleKey drop-down menu, and click **New** for each schedule beyond the first you would like to add. To delete a schedule, select the desired schedule from the list and click **Delete**. Leaving the **Schedules** list blank upon creation will apply the constraint to all schedules.

- 4 Click **Save and Close** to save this rule, or **Close** to exit this screen without saving the changes.

Managing Routing Parameters

Routing Parameters

Routing Parameters are used to manage the speeds and costs that apply to a Resource when the optimizer is solving a routing problem. Routing Parameters also

manage the Schedule as to when particular routing rules apply to a particular resource.

As the name indicates, users can define parameters that apply to routing (geography, cost, and traffic conditions). For example, Routing Parameters can be used to slow down drivers' speeds during rush hour, or make left turns more expensive than right turns.

A resource record can have one or many Routing parameters that function at different times during the day.

To access Routing Parameters:

- 1 From the **Home** menu, select: **Data > Routing Parameters**.

The list of Routing Parameters appears.

Fleetwise / Routing Parameters			
Routing Parameters			Find
RouteParmsKey	Comment	Description	Active
DEFAULTS	Descartes Default Settings	Descartes Default Settings	True
RSDemo	Reservations Demo	Reservations Demo	True
RCDemo	Recharge Demo	Recharge Demo	True
FCDemo	Fleet Consolidate Demo	Fleet Consolidate Demo	True
BGDDemo	Bulk Gas Demo	Bulk Gas Demo	True
CMDemo	Commodities Demo	Commodities Demo	True
HSDemo	Hours of Service Demo	Hours of Service Demo	True
ALDDemo	AVL Demo	AVL Demo	True
LTDemo	LTL Demo	LTL Demo	True
TTDDemo	Territory Demo	Territory Demo	True

By default, Descartes Route Planner already has a Routing Parameter set called "Defaults." This is a reserved name that the optimizer uses whenever a Routing Parameter is not specified during a certain time of the day.

- 2 Right-click on a selected Routing Parameter record and select one of the management options:
 - **New:** Create a new parameter.
 - **Edit:** Edit the selected parameter.
 - **Clone:** Displays the Clone Routing Parameter page, which allows the user to copy the selected parameter and make any necessary edits.
 - **Delete:** Delete the selected parameter.
 - **Update Road and Overall Hr Cost:** Displays the **Update Road and Overall Hr Cost** window.

Data / Routing Parameters / Update Road and Overall Hr Cost

Update Road and Overall Hr Cost

SAVE AND CLOSE CANCEL

Route Parm's

RouteparmsKey: DEFAULTS

OverallHourCost:

Road Cost:

Use the associated fields to edit the Overall Hour Cost and Road Cost and click **Save** or **Save and Close** to save the data and/or return to the **Routing Parameters** list page.

- 3 It is also possible to double-click a specific Routing Parameter record to directly edit it. When double-clicked, the selected **Routing Parameter** page appears:

Data / Routing Parameters / New Routing Parameters

New Routing Parameters

SAVE CANCEL

RouteparmsKey

Comment

Active ☒

Road Speeds

#	Type Of Road	RuralSpeed	SuburbanSpeed	SemiUrbanSpeed	UrbanSpeed
1	HIGHWAY	96.559	88.51	80.46	72.421
2	PRIMARY	80.46	64.368	48.276	38.621
3	SECONDARY	64.368	48.276	35.399	25.751
4	SERVICE	64.368	48.276	35.399	25.751
5	LOCAL	40.23	32.191	28.969	20.92
6	RAMP	32.191	32.191	32.191	32.191
7	FERRY	16.088	16.088	16.088	16.088
8	JEEP	16.088	16.088	16.088	16.088
9	ALL FRY	16.088	16.088	16.088	12.87

RuralSpeed km/hr

SuburbanSpeed km/hr

SemiUrbanSpeed km/hr

UrbanSpeed km/hr

DenseSpeed km/hr

RuralCost /km

SuburbanCost /km

SemiUrbanCost /km

UrbanCost /km

DenseCost /km

RoadType

OverallHourCost

DegradationTimeFactor

UturnCost

LeftTurnCost

SharpLeftTurnCost

RightTurnCost

SharpRightTurnCost

BearLeftCost

BearRightCost

UturnTime

LeftTurnTime

SharpLeftTurnTime

RightTurnTime

SharpRightTurnTime

BearLeftTime

BearRightTime

Description

External Pather Speed Category

Allow U-Turns: ☒

Allow Toll Roads: ☒

Allow Ferries: ☒

Apply Traffic Pattern: ☐

Dist Cost: /km

Hour Cost: /hour

Allow

☒ NonCommercialRoadsOnly ☒ BusRoadsOnly

☒ MilitaryRoadsOnly ☒ AllowPrivateRoads

☒ AllowVehicle ☒ AllowWalkways

☒ Allow Ferries

Speeds

SpeedCat #	Speed (km/hr)
1	96.561
2	88.514
3	80.467
4	56.327
5	56.327
6	56.327
7	56.327
8	32.187

Enter the appropriate values on this page. Select the **Apply Traffic Pattern** setting on the **New/Edit Routing Parameters** page to enable the use of traffic patterns in RMPI.

RMPI's use of Traffic Patterns and Premium Truck Data in the external pather is regulated by the user's license terms. Descartes Route Planner will check the user's license terms to determine whether or not the use of Traffic Patterns or Premium Truck Data is permitted.

- 4 Click **Save** to save this rule, or **Close** to exit this screen without saving the changes.

Managing Templates

Schedule Templates

Schedule templates lets users create a series of templates that can serve as the basis for creating new Schedules. Users can set up templates representing commonly used Schedules, so users can create Schedules more quickly and accurately.

Managing Schedule Templates

To access a Schedule template:

- 1 From the main menu, select: **Data > Templates > Schedule**.

The list of Schedule templates appears:

Schedule Templates					Find
ID	Template	Name	Type	Status	
1	DefaultScheduleTemplate	Default Schedule Template			

- 2 Right-click on a selected Schedule template record and select one of the management options: **New**, **Edit**, or **Delete**. It is also possible to double-click a specific Schedule template record to directly edit that record.

When double-clicked, the selected **Schedule Template** page appears:

Description

Template:

Name:

Type:

Status:

Comment:

ScheduleGroup:

Adjustment

Profit:

Ignore

EmptyRoutes: ☐

Territories: ☐

Enable

StopConsolidation: ☐

NewStop:

SecureParham:

Limits

MaxWindowWait: seconds

Time Window Warning Threshold: seconds

MaxBetweenIntras:

MaxUnservedOrders:

#ofMeasures:

Min InterStop Travel Time: seconds

Restrict

ViableOrders: ☐

ZonableOrders: ☐

SwaptToNeighbors: ☐

SwaptToPoints: ☐

SwapWithinTerritory: ☐

Seed

DesiredArea: ☐

Route: ☐

ZonedOrders: ☐

Assignment

Auto Assign Close Orders: ☐

Assign Inassigned Orders: ☐

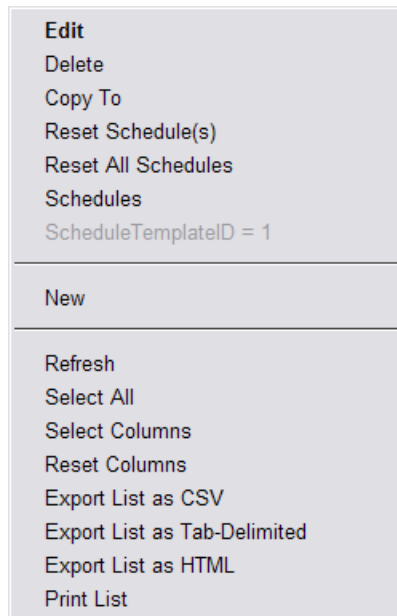
3 Enter data in the appropriate fields.

4 When finished entering the data, click **Save**, and then click **Close**

This template can now be used as the basis for adding new Schedules.

Applying Schedule Template Settings to Multiple Schedules

Users can apply settings from a schedule template to all created schedules or a particular subset by using the right-click menu in the **Schedule Templates** list page.



- 1 Right-click on a schedule template in the list and select **Reset Schedule(s)** to apply the settings to one or more schedules.
- 2 Click **OK** in the confirmation dialog. The **Select Schedule** dialog box appears.

Fleetwise / Schedule Templates / Select Schedule							
Select Schedule							
ID	Schedule	Template:	Name	Type	State	Status	Start
7	ReservationsDemo	DefaultScheduleTemplate	Reservations Demo		1		1- JAN- 2006 00:01
8	RechargeDemo	DefaultScheduleTemplate	Recharge Demo		1		1- JAN- 2006 00:01
9	FleetConsolidateDemo	DefaultScheduleTemplate	Fleet Consolidate Demo		1		1- JAN- 2006 00:01
10	BulkGasDemo	DefaultScheduleTemplate	Bulk Gas Demo		1		1- JAN- 2006 00:01
11	CommodityDemo	DefaultScheduleTemplate	Commodity Demo		1		1- JAN- 2006 00:01

Rows 1 - 11 of 11 Rows per Page: 10 / 20 / 50 Page 1

- 3 Select one or more schedules from the list and right-click to display the right-click menu. Click **Select** to apply the template settings.
- 4 To apply the template settings to all schedules in the list, select **Reset All Schedules** and click **OK** in the confirmation dialog to complete.

Creating a Schedule Template from an Existing Schedule

Users can create a schedule template from a schedule by selecting the **Create Template** right-click menu option for a schedule on the **Schedules** list page.

To create a schedule template from a schedule:

- 1 Right-click on a schedule on the **Schedules** list page and select **Create Template** from the right-click menu. The **Create Template** window appears.

- 2 Enter a Schedule Template Key and click **Create**.
- 3 The **Schedules** list page refreshes. The new schedule template will be available on the **Schedule Templates** list page.

Route Templates

Route templates provide default routing settings to associate with a resource when it is created.

Tip—Resource settings override settings taken from the Route template.

Managing Route Templates

To access a Route template:

- 1 From the main menu, select: **Data > Templates > Route**.

The list of Route templates appears:

Fleetwise / Route Templates										
Route Templates										Find
ID	Template:	Schedule	Type	Status	Driver	Territory	Amount	Parameters	Earliest Start Time	Latest End Time
11	DefaultRouteTemplate	DefaultSchedule	DEFAULT	DEFAULT			0		07:00:00	19:00:00

- 2 Right-click on a selected Route template record and select one of the management options: **New**, **Edit**, **Create Resources**, or **Delete**.

The selected **Route** template page appears:

Data / Route Templates / Edit Route Template

SAVE SAVE AND CLOSE DELETE CLOSE

Description

Template: MUR_Full_RouteTemp
Schedule: MUR_TestFullSchedule
Name: RouteTemplate Name Full
Resource Key Generation: DefaultRscKeyWizard
Class:

Equipment

Driver: Dummy_driver1
Crews: ASSIGN A DRIVER AND CREW
Truck:
Trailer:
Tractor:
Parameters:

Location

Ignore Asset Locations:
Initial Location: Res_Depot1
Final Location: Res_Depot3
Territory:

Capacity

FullThreshold: 2
Full Bonus: 2
PartialPenalty: 2
Ignore Asset Measures:
Measure1: 2000 Factor 1 RechargeRate 0.5
Measure2: 5000 Factor 1 RechargeRate 0.8
Measure3: 2000 Factor 1 RechargeRate 0.4
Measure4: 1000 Factor 1 RechargeRate 0.6
Measure5: 1000 Factor 1 RechargeRate 1
Measure6: 1000 Factor RechargeRate 0.6
Measure7: 1000 Factor RechargeRate 0.9
Measure8: 1000 Factor 1 RechargeRate 0.8
Measure9: 1000 Factor 1 RechargeRate 0.9
WorkUnits: 2
Empty Weight: 0.91 kg
Height: 0.61 m
Length: 0.61 m
Width: 0.61 m

Constraints

Requirements: Requirement_key test111
Requirement Set:
Hazardous Materials: 1
FreezeInd:
Commercial Truck: StraightTruck
RoutePositionInd: ☒

Recharge Setting

Allow Recharges: ☒
Opportunistic Recharge Threshold:
Recharge to Capacity: ☒
End Route with Unloaded or Loaded Vehicle: ☒
Allow Empty Start: ☒
End Vehicle Measures Required: ☒ Measure1 ☐ Measure2
☒ Measure3 ☐ Measure4
☐ Measure5 ☐ Measure6
☐ Measure7 ☐ Measure8
☐ Measure9
Commodity Transition Factor: 2

Availability

EarliestStartDay: 0 LatestStartDay:
Earliest Start Time: 06:00 Latest Start Time:
EarliestEndDay: LatestEndDay: 0
Earliest End Time: Latest End Time: 11:00
EarliestServiceDay: LatestServiceDay:
EarliestServiceTime: LatestServiceTime:
Prefix: Suffix:

Frequency

☒ Sunday ☒ Monday
☒ Tuesday ☒ Wednesday
☒ Thursday ☒ Friday
☒ Saturday

Resource Generation

Days From Today:
Number Of Days:
Number Of Resources:
NEW DELETE

Routing Parameters

StartTime	EndTime	RouteparmsKey
StartTime	EndTime	RouteparmsKey
EndTime	RouteparmsKey	

NEW DELETE

Strings

UDFString1	UDFString2	UDFString3	UDFString4	UDFString5	UDFString6	UDFString7	UDFString8	UDFString9
2	2	2	2	2	2	2	2	2

Integers

UDFInteger1	UDFInteger2	UDFInteger3
2	2	2

Numbers

UDFNumber1	UDFNumber2	UDFNumber3
2	2	2

date Time

FWResourceUDFDateTime1: 2013-11-14T01:47:00
FWResourceUDFDateTime2: 2013-11-12T01:47:00
FWResourceUDFDateTime3: 2013-11-01T01:47:00
Comment: test

Scheduled Stops

LocationKey	StartTime	EndTime	Duration	Profit

Location Key:
StartTime:
EndTime:
Duration:
Profit:
NEW DELETE

Membership

Party: Select Party...

- 3 Enter data in the appropriate fields.
- 4 When finished entering data, click **Save**.

The new template can now be used as the basis for adding new resources.

To edit multiple route templates:

- 1 Select the desired resources from the **Route Templates** list page, right-click and select **Edit Selected Templates**.

The **Edit Multiple Route Template** page appears. It is blank so that the user can enter new values for the selected resources.

- 2 Enter or select the appropriate information in the appropriate fields.
- 3 Click **Save** to save the new information for the selected route templates.

The new information is applied to the selected route templates.

Route template windows can also be created to set specific time windows that define a resource's availability. Route template settings are associated directly with the Route template and are used to validate resource availability.

To create New Resources from Route Templates:

- 1 From the list of Route Templates, right-click a Route Template and select: **Create Resources**.

The following screen appears:

Fleetwise / Route Templates / Create Resources

Create Resources

CREATE CLOSE

Date Range

From Date: To Date:

Number of Resources: Master Route: ☐

Key Wizard:

Resources Created:

- 2 Enter the date range and the number of Resources to create for each day.
- 3 When finished entering data, click **Create**. This will create the total number of resources for the provided date range. When finished, the **Note** area will display the number of resources created. Click **Close** to exit this window.

Capping Generated Window Open and Close Values

The Earliest Hr Gen OpenWindow and Latest Hr Gen CloseWindow fields allow users to put a cap on how early a generated time window can open and how late a generated time window can close. If the generated time windows resulting from any of the following operations exceed the cap values set in the Earliest Hr Gen OpenWindow and Latest Hr Gen CloseWindow fields, the cap value is used instead.

- **Lock Generated Time Windows** operation
- Route moved to another schedule
- All routes moved to another schedule
- Status updated from first status call

Availability			
EarliestStartDay		LatestStartDay	
EarliestStartTime		LatestStartTime	
EarliestEndDay		LatestEndDay	
EarliestEndTime		LatestEndTime	
EarliestServiceDay		LatestServiceDay	
EarliestServiceTime		Latest Service Time	
Prefix		Suffix	
Earliest Hr Gen OpenWindow			
Latest Hr Gen CloseWindow:			

For example, if a user selects the **Lock Generated Time Window** option for a route in the **Routes** quadrant and the generated open time window is earlier than the cap value entered in the Earliest Hr Gen OpenWindow field, the cap value is used instead.

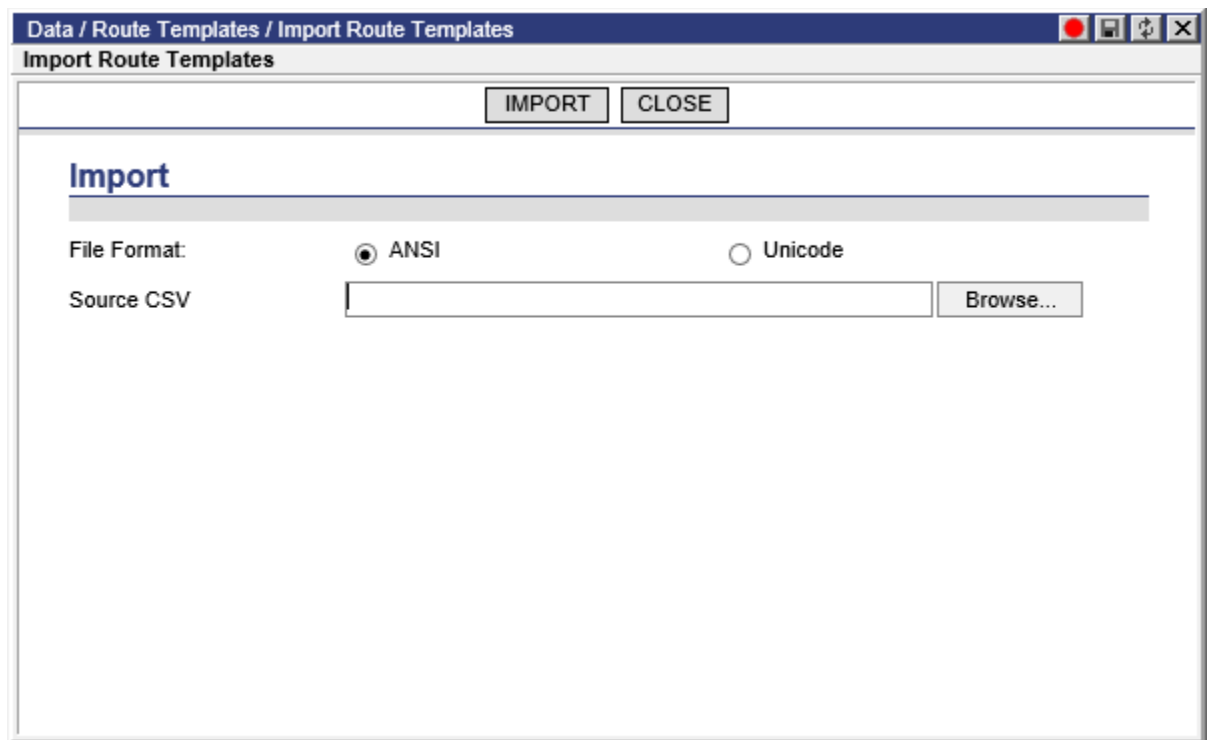
Please note the following the following exceptions regarding this feature:

- If the Projected Arrive Time is earlier than the Earliest Hr Gen OpenWindow value, the Projected Arrival Time is used instead.
- If the Projected Departure Time is later than the Latest Hr Gen CloseWindow value, the Projected Departure Time is used instead.
- **Note**— In these cases, users should fix the route starting and ending times since the resource settings conflict with the cap values provided.
- If no value is entered in these fields, no cap is enforced.

Import Route Templates Via CSV File

Users can create route templates by populating and importing a CSV template. To create route templates via import:

- 1 Navigate to **Data > Templates > Routes**.
- 2 Right-click on the page and select the **Export the CSV Template** option from the right-click menu. From the browser prompt, save the file to your machine.
- 3 Open the CSV in a spreadsheet application and enter the necessary data.
- 4 Right-click on the **Route Templates** list page and select the **Import CSV Data** right-click menu option. The **Import Route Templates** window appears.



- 5 Select the File Format.
- 6 Click **Browse** to and select the CSV template using the dialog.
- 7 Click **Import**.

Review Key Parameters Before Resource Creation

To assist with the creation of resources from route templates extemporaneously, select the **Edit and Create Resources** right-click option. With this option, users can review and edit the key parameters of a route template before creating resources from it.

- 1 From the **Route Template** list page, right-click on the desired route template and select the **Edit and Create Resources** right-click menu option. The **Review Key Parameters Before Creating Resources** page appears.

No	Route Template	Driver	Truck	Trailer	Crew	EarliestStart	EarliestEnd	EarliestService	LatestStart	LatestEnd	LatestService	Schedule	Ignore
1	Create_resource_aqa_2	Dumy_driver1	ZSV_Tr	Trailer_z	Crew	19-MAY-2014 10:10	19-MAY-2014 10:10	19-MAY-2014 10:10	19-MAY-2014 10:11	19-MAY-2014 10:11	19-MAY-2014 10:11	FF	

- 2 Select a Resource Key Wizard and an Order Key Wizard from the appropriate drop-down menus.
- 3 Specify a Driver, Truck and Trailer using the appropriate drop-down menus. Users can choose the **Select Driver**, **Select Truck** or **Select Trailer** option from the **Driver**, **Truck** and **Trailer** drop-down menus respectively to display a window with a list of the selected assets.

No	Route Template	Driver	Truck	Trailer	Crew	EarliestStart	EarliestEnd	EarliestService	LatestStart	LatestEnd	LatestService	Schedule	Ignore
1	DefaultRouteTemplate	Select 1	Select 1	Select 1	Crew							LTLDemo	

The list of assets can be filtered using the Find search box. To assign an asset to the resource, select one from the list, right-click and choose the **Select This [Asset]** option. For drivers, users can select multiple rows and apply this option. The drivers will be displayed in the associated **Crew** page.

Data / Route Templates / Review Key Parameters Before Creating Resources / Drivers

Drivers

Driver Key	Driver Name	Email	Phone Number 1:
ZSV_party_driver	ZSV_party_driver		
ZSV_Driver_2	ZSV_Driver_2		
ZSV_Driver_13_2_1_2	ZSV_Driver_13_2_1_2		
ZSV_Driver_1_X_active	ZSV_Driver_1_X_active_dummy_2		
xyz	xyz		
testDriver	ZSV_driver		
test_driver_7	test_driver_7		
test_driver_6	test_driver_6		
test_driver_5	test_driver_5		
test_driver_2	test_driver_2		
TEST_CSV_DRIVER_party	TEST_CSV_DRIVER		
TEST_CSV_DRIVER	TEST_CSV_DRIVER		
TA_API_CreateDriver148265_Test_1	TA_API_CreateDriver148265_Test_1_Name		
TA_2014_010_Driver_2	TA_2014_010_Driver_2_Name		
TA_2014_010_Driver_1	TA_2014_010_Driver_1_Name		
party_party_driver	party_party_driver		

Rows 1 - 50 of 78 Rows per Page: 10 / 20 / 50 Page 1 2 << < > >>

Click the Crew field to display the **Crew** page, for finer control of the driver assignment process.

Data / Route Templates / Review Key Parameters Before Creating Resources / Crew

Crew

OK CANCEL

Filter

Find where: no party SEARCH FILTER

Drivers and Crew

AVAILABLE DRIVER | Row 1 - 15 of 17 | << < > >>

* Driver Key	Driver Name	Comment	Role
Dummy_driver1	Dummy_driver1		
Dummy_driver2	Dummy_driver2		
test_driver_5	test_driver_5		dummy
Driver_11132013_75852117	Driver_cost	testovaci driver	driver
ZSV_Driver_1_X_active	ZSV_Driver_1_X_active_dummy_2		
Driver_dummy_ZSV_5854	Driver_dummy_ZSV_5854		
MUR_TestDriverAPI	Driver API MUR		
TA_API_CreateDriver148265_Test_1	TA_API_CreateDriver148265_Test_1_Name		
Driver_testik	Tester		
AVL1	AVL1		
Driver_4302014_93433129	adsfdfsdfsfdf		
Driver_ZSV_no_party	Driver_ZSV_no_party	Driver_ZSV no party	
TEST_CSV_DRIVER	TEST_CSV_DRIVER		
bc_5987234	test driver 2		
bc_1D99	test driver 3		

< >

SELECTED DRIVER | Row 1 - 2 of 2 | << < > >>

↑ Driver Key	Driver Name	Comment	Role	Lead
Dummy_driver1	Dummy_driver1			☒
test_driver_5	test_driver_5		dummy	☐

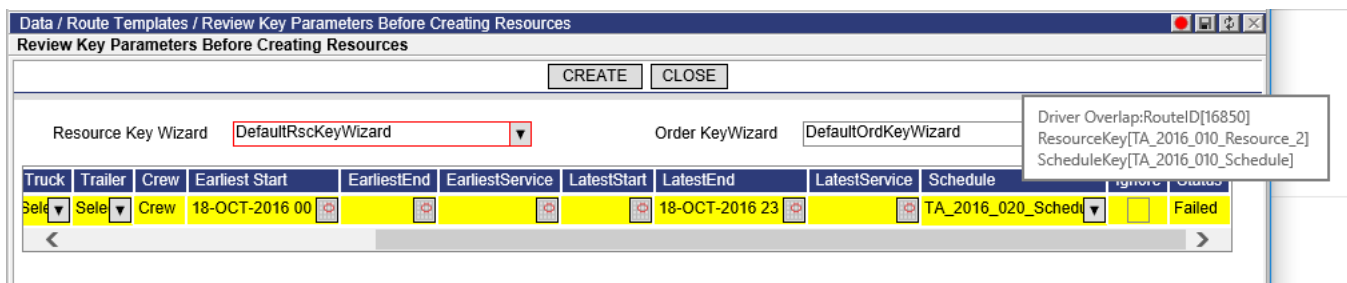
>> ADDALL >>
> ADD >
< REMOVE <
<< REMOVEALL <<

- 4 Select relevant date information using the calendar controls and select a schedule from the Schedule drop-down menu.

➡ **Note**— The **Resource Key Wizard** menu, Earliest Start and Latest End fields are required.

- 5 Click **Create** when finished. The system returns a success dialog in response. Click **OK** to return to the page.

When users attempt to create resources from the **Review Key Parameters Before Creating Resources** page and the system returns a "Failed" status, the system will display the reason for the conflict and the related ResourceKey and ScheduleKey in a tooltip if users hover the mouse cursor over the Route Template or Status columns.



Setting Route Template Settings

The **Settings** tab is used to set up the following information for the Route template:

- service duration
- bonus or discount information
- any penalties to be applied to the route
- any limits to be applied to the route
- speed factors that may effect the route
- any working hour rules (i.e., driver/transportation rules)

Data / Route Templates / Settings

Edit Route Template | **Settings** | BreaksRests

SAVE DELETE CLOSE

Description		Limits		Working Hour Rules	
Template:	MUR_Full_RouteTemp	MaxStops	200	Last Date Worked	1-JAN-1901
Name	RouteTemplate Name Full	MaxDistance	32186.88 km	Work Week Length	2
		MinPaidOrders	2	Work Week Limit	2
		MinTravelDistance	2	Weekly Rest Length	2
		MaxDwell	20000	Work Week Reset Time	2
		MaxElapsed	20000	Work Week History1	2
		DynamicZoningDistance	32.19 km	Work Week History2	2
		Max Distance From Depot:	80.47 km	Work Week History3	2
		MaxDrivingTime	20000	Work Week History4	2
		Max Redispatches:	2	Work Week History5	2
		Redispatch Rest Time:	8800	Work Week History6	2
		Max Redispatch Time:	3600	Work Week History7	2
				Work Week History8	2
				Work Week History9	2
				Work Week History10	2
				Work Week History11	2
				Work Week History12	2
				Work Week History13	2
				Work Week History14	2

Duration		Speed	
Service	180	SpeedFactor	2
PreService	240		
ServiceFactor	2		
Recharge	2		
Startup:	2		
Windup:	2		
SkipInitialLoadTime::	0		

Bonus	
SameStopBonus	2
DiscountInitialLeg	2
DiscountFinalLeg	2

Penalty	
PerMeter	1
PerOrder	2
Service	2
MissedWindow	2
LatestEnd	1000
WaitTime	3600

#	Work	Duration
1	2	120
2		0
3		0
4		0
5		0
6		0
7		0

#	Speed	Threshold
1	2	2
2		
3		
4		
5		
6		
7		

Speed	2
Threshold	2
Adapti Driving Time Factor:	2

#	Work	Duration
1	2	120
2		0
3		0
4		0
5		0
6		0
7		0

Penalty	2
Duration	120

To set the route settings:

- 1 From the **Route Templates** list page, right-click the selected Route template and select **Edit**. The **Edit Route Templates** page appears.
- 2 Click **Settings**.
- 3 Enter data in the appropriate fields.
- 4 When finished entering data, click **Save**.

Setting up Breaks and/or Rests

The BreaksRests tab is used to set up resources' breaks and/or rests.

Fleetwise / Route Templates / BreaksRests

Edit Route Template Settings **BreaksRests**

SAVE DELETE CLOSE

Description

Template: DefaultRouteTemplate

Name:

Rests

Duration: 0

Max Elapsed Time: 0

Max Driving Time:

First Stop Threshold: 0

Allow Early: ☐

Early Time Threshold: 0

Early Wait Threshold: 0

Use Full Wait: ☒

Opportunistic Early Rest: ☒

Breaks

Duration: 0

Max Elapsed Time: 0

Max Driving Time:

Max Breaks: 10

On Duty Ind: ☒

To set up breaks and/or rests:

- 1 From the **Route Templates** list page, right-click the selected Route template and select **Edit**. The **Edit Route Templates** page appears.
- 2 Click **BreaksRests**.
- 3 Enter data in the appropriate fields.
- 4 When finished entering data, click **Save**.

Stop Templates

Stop templates establish the routing settings associated with Stops, and provide many of the default settings needed during scheduling.

Managing Stop Templates

To access a Stop template:

- 1 From the main menu, select: **Data > Templates > Stop**.

The list of Stop templates appears:

Fleetwise / Stop Templates

Stop Templates Find

Template:	Schedule	Type	Status	Territory	On Load	Service	Profit
DefaultStopTemplate	DefaultSchedule	DEFAULT	Default		0	0	0

- 2 Right-click on a selected Stop template record and select one of the management options: **New**, **Edit**, or **Delete**. It is also possible to double-click a specific Stop template record to directly edit that record.

The selected **Stop** template page appears:

Description

Template:

Schedule:

Constraints

Territory:

Requirements:

Commodities:

Duration

Service:

PreService:

Route Settings

PreferredInd: ☒

Interspersable: ☒

OnLoad:

RoutePosition: ☒

SpecialHandling:

AllowConsolidation: ☒

Rating

Profit:

DepreciationHour:

Comment:

- 3 Enter data in the appropriate fields.
- 4 When finished entering data, click **Save**.

This template can now be used as the basis for adding new Stops.

Managing Bucket Templates

To access a Bucket template:

- 1 From the main menu, select: **Data > Templates > Buckets**.

The list of Bucket templates appears:

ID	Template	Name	Schedule ID	Schedule Key	Description	Multi-	End Day	StartTime	EndTime
1	MDayTest		6	DefaultSchedule		True	2	08:00:00	22:00:00
2	OneDayTest	One Day Bucket	6	DefaultSchedule		False	0	08:00:00	08:00:00
3	test_bucket_1	test_bucket_2	922	buckets		True	0	08:00:00	10:00:00
4	copy_test_bucket_2	copy_test_bucket_2	922	buckets		False	0	08:00:00	10:00:00

- Right-click on a selected bucket template record and select one of the management options: **New**, **Edit**, or **Delete**. It is also possible to double-click a specific bucket template record to directly edit that record.

The selected **Bucket** template page appears:

Data / Bucket Templates / Edit Bucket Template

Edit Bucket Template

SAVE SAVE AND CLOSE DELETE CLOSE

Description

Template:

Name:

Type:

Comment:

Buckets

Schedule:

Territory:

* Multiple territories must be delimited by comma(,).

Multi-Day: ☐

Start Time:

Bucket Size: Hours

Number of Buckets per Day:

Frequency

Use Frequency: ☐

☐ Sunday
 ☐ Monday

☐ Tuesday
 ☐ Wednesday

☐ Thursday
 ☐ Friday

☐ Saturday

- Enter data in the appropriate fields.
- When finished entering data, click **Save**.

This template can now be used as the basis for creating new buckets.

Managing Message Templates

To access a Message template:

- From the main menu, select: **Data > Templates > Message**.

The list of Message templates appears:

Data / Message Templates

Message Templates

MessageKey	Schedule Key	Subject	Body
Hello	FleetConsolidateDemo	test	test test test test
aqa	AQA	test	aqa test
test_template_message		test_template_message_subject	test_template_message_body
test_template_reassign	Reassign	subject	body body body body body body body body

- Right-click on a selected message template record and select one of the management options: **New**, **Edit**, or **Delete**. It is also possible to double-click a specific message template record to directly edit that record.

The selected **Message** template page appears:

- Enter data in the appropriate fields.
- When finished entering data, click **Save**.

This template can now be used as the basis for adding new messages.

Resource Template Sets

Users can use Resource Template Sets to represent and organize given routes on a given work day.

Creating Resource Template Sets

To create a Resource Template Set:

- From the main menu, select **Data > Templates > Resource Template Sets**. A list of Resource Template Sets appears.

ResourceTemplateSet ID	ResourceTemplateSet Key	Name	Description	Name	Comment
1	Test1	T1		T1	

- Right click in the **Resource Templates Sets** page and select **New** from the right-click menu. The **New Resource Template Set** page appears.

[illegible]

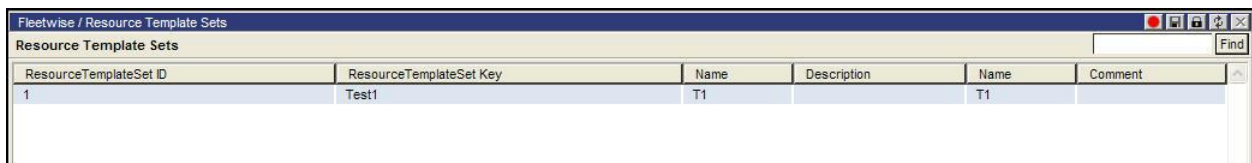
- 3 Fill in the Resource Template Set field with an ID unique to the database.
- 4 Enter a name for the Template Set in the Name field.
- 5 Enter a Description and any Comments.
- 6 Move saved route templates from the **Available Route Template** table to the **Selected Route Template** table to add to the resource template set using the following functions:
 - **ADDALL**— Moves all route templates from the **Available Route Template** table to the **Selected Route Template** table, adding them to the set.
 - **ADD**— Moves selected route templates from the **Available Route Template** table to the **Selected Route Template** table, adding them to the set.

- **REMOVE**— Moves selected route templates from the **Selected Route Template** table to the **Available Route Template** table, removing them from the set.
 - **REMOVEALL**— Moves all route templates from the **Selected Route Template** table to the **Available Route Template** table, removing them from the set.
- 7 Click **Save** or **Save and Close** to save the new resource template set. Clicking **Close** will close the page without saving the set.

Editing Resource Template Sets

To edit a resource template set:

- 1 From the main menu, select **Data > Templates > Resource Template Sets**. A list of resource template sets appears:



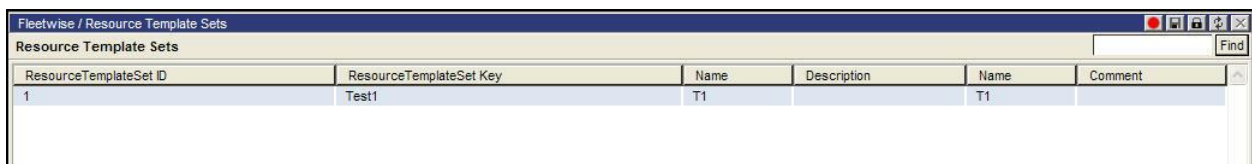
ResourceTemplateSet ID	ResourceTemplateSet Key	Name	Description	Name	Comment
1	Test1	T1		T1	

- 2 Right-click on a resource template set and select **Edit** from the right-click menu. The **Edit Resource Template Set** page appears.
- 3 Edit data as needed in the appropriate fields.
- 4 Use the **Route Template** tables to add or remove route templates from the resource template set.
- 5 Click **Save** or **Save and Close** to save the new resource template set, **Close** to close the page without saving the changes and **Delete** to remove the set entirely.

Generating Resources Using Resource Template Sets

To generate resources from resource template sets:

- 1 From the main menu, select **Data > Templates > Resource Template Sets**. A list of resource template sets appears:



ResourceTemplateSet ID	ResourceTemplateSet Key	Name	Description	Name	Comment
1	Test1	T1		T1	

- 2 Select one or more resource template sets, right-click in the page and select **Generate Resources** from the right-click menu. The **Create Resources** dialog box appears:

- 3 Enter data into the appropriate fields, using the calendar button (📅) to select dates for the FromDate and ToDate fields.
- 4 Click **Create** to generate the specified resources or **Close** to cancel.

Managing Territories

Territories can be used to create geographic boundaries that logistically speaking create a constraint on Vehicles that travel within each of them. Vehicles can either stay in one territory or swap territories when it makes sense. This function allows users to have the application keep the territory definition in its database rather than external systems.

Creating Territories

To create territories on a map:

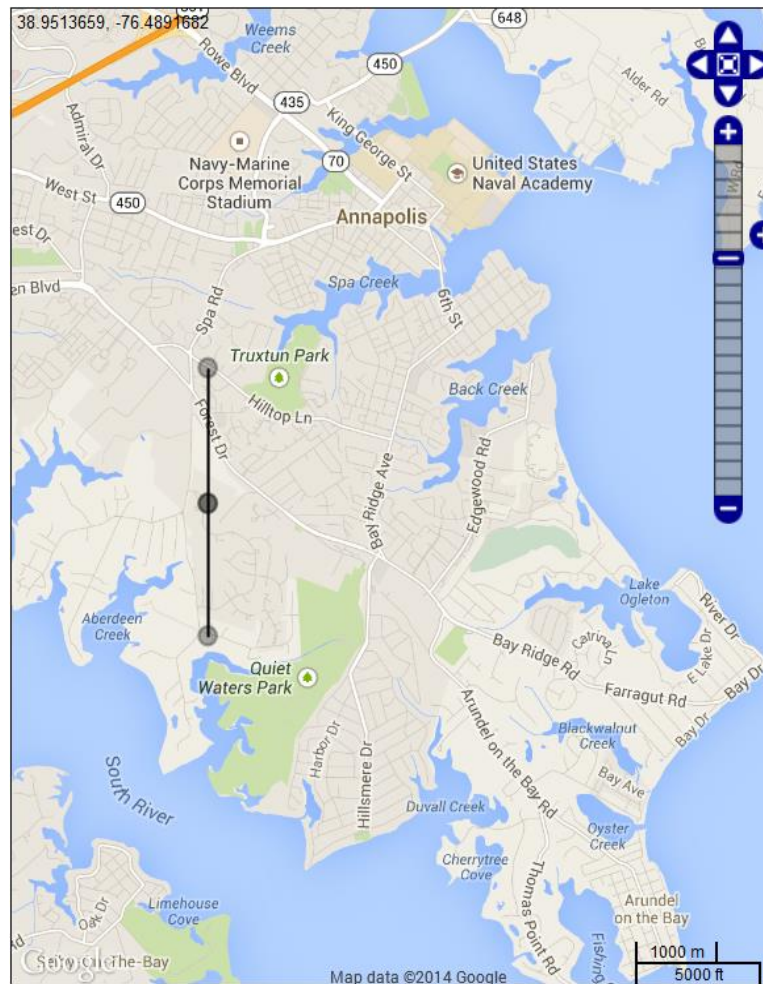
- 1 Select **Data > Territories > Definition** from the main menu. The **Territories** page appears.

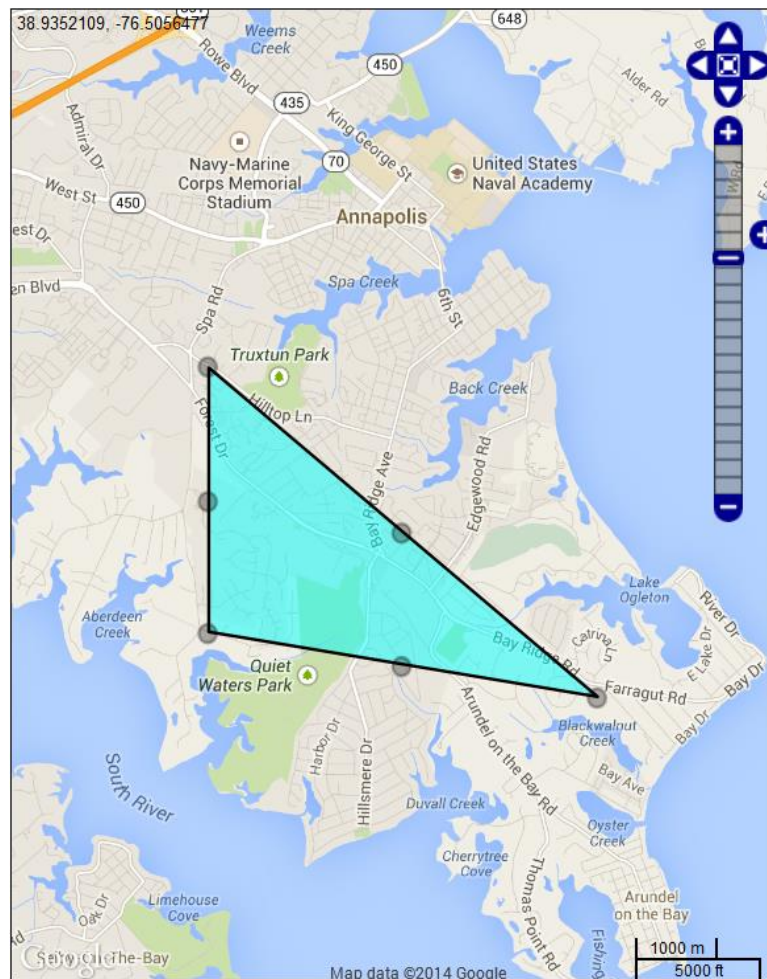
Name	TerritoryKey	ScheduleKey	Active	Description
T1	T1	15MD	True	
t2	t2	15MD	True	
T_4-OCT-2006 17:22:57:841	TerXav1	PlanningSchedule	True	
T_4-OCT-2006 17:27:43:850	TerXav2	PlanningSchedule	True	

- 2 If there are no territories created, click **New**. If territories exist, right-click in the **Territories** page and select **New**. The **New Territory** page displays as shown below.

Map data © 2015 Google

-  **Note**— You can use the zoom in and zoom out tools to adjust the map view.
- To set the map center, enter all relevant information and click **Set Map Center**
- Specify the Territory Generation allowance percentage.
- Enter a name or use the default name.
- Enter the territory key.
- Select a Requirement Set and enter Requirements. The values entered in these fields will be applied to the order to which the territory is assigned.
- Select a color for the territory.
- Double-click on a starting point for the territory and move the mouse pointer to the next location to draw a line and left-click. Click the nodes on the line and drag to expand the territory's scope.





Map data © 2015 Google

10 Right-click and select **Create Territory** to complete the territory area.

➡ **Note**— The latitude and longitude for the territory vertexes are displayed under Territory Vertexes.

11 To create multiple territories, under **Territory Name**, click **New**.

12 Repeat steps 5-10.

13 Click **Save**.

➡ **Note**— Use the Clear Depots, Clear Stops and Clear Territories buttons to remove all specified information from the map.

Editing a Territory

To edit a territory:

- 1 Select **Data > Territories** from the main menu. The **Territories** page appears.

Name	TerritoryKey	ScheduleKey	Active	Description
T1	T1	15MD	True	
t2	t2	15MD	True	
T_4-OCT-2006 17:22:57:841	TerXav1	PlanningSchedule	True	
T_4-OCT-2006 17:27:43:850	TerXav2	PlanningSchedule	True	

- 2 Right-click on the desired territory and select **Edit**. The **Territory** page displays as shown below.

Data / Territories / Territory

Territory

SAVE SAVE AND CLOSE CLOSE

Territory Design

SET CENTER

City: State:

Country: Zip Code:

SET MAP CENTER

TERRITORY GENERATION

Allowance: %

Territory Name

Name	Key	Color	Description	GlobalInd
TC1	TC1_13_1_3_4	Cyan		False

ID: 14 Color Code: Cyan

Name: TC1 Territory Key: TC1_13_1_3_4

Description:

Requirement Set: Requirements:

Global Ind: ☐

TERRITORY ASSIGNMENT RULE

Table: Location

Criteria: UDFString1 = TC1

DELETE TERRITORY

Map data © 2015 Google

- 3 Edit the Name field if desired.

- 4 Edit the Territory Key field.

➡ **Note**— The territory vertexes cannot be redrawn without entering in a new Territory Key.

- 5 Double-click on a starting point to redraw the territory and move the mouse pointer to the next location to draw a line and left-click. Continue drawing the lines for your territory until you are on the next to the last leg.
- 6 Right-click and select **End** to complete the territory area.
- 7 Click **Save** to save the edits to the territory.

Activating and Deactivating a Territory

A territory can be activated or deactivated depending on the setting you select. Activated means that a route can be assigned to it and the territory will have a True status in the Active column on the **Territories** page. Deactivated means that it will not be available for use in routing and the territory will have a False status in the Active column on the **Territories** page.

To activate a route:

- 1 Select **Data > Territories** from the main menu. The **Territories** page appears.
- 2 Right-click a territory with a False status in the Active Column and select **Activate/Deactivate**. The status in the Active column will change to True.

To deactivate a route:

- 1 Select **Data > Territories** from the main menu. The **Territories** page appears.
- 2 Right-click a territory with a True status in the Active Column and select **Activate/Deactivate**. The status in the Active column will change to False.

Assigning Territories to Schedules and Schedules to Territories

Users can view territories by schedule from the **Schedules** tab on the **Territories** list page and then **Assign Schedules** on the **Territories** list page or **Assign Territories** on the associated **Schedules** list page. These options allow users to see which territories are associated with which schedules and then assign and unassign new schedules or territories to the other.

Data / Schedules							
Territories Schedules							
ID	Schedule	Template	Name	Type	State	Status	Start
1868	TA_2014_010_Schedule	TA_2014_010_ScheduleTemplate	TA_2014_010_Schedule_Name	2	1		23-NOV-2014 07:00
1827	ZSV_14_2_1_8	DefaultScheduleTemplate	Default Schedule Template	2	1		18-NOV-2014 09:34
1828	BG_14_2_1_8	DefaultScheduleTemplate	Default Schedule Template	2	1		18-NOV-2014 09:34
1829	BG_copy	DefaultScheduleTemplate	Default Schedule Template	2	1		18-NOV-2014 09:34
1844	BG_copy2	DefaultScheduleTemplate	Default Schedule Template	2	1		18-NOV-2014 09:34
1845	BG_copy3	DefaultScheduleTemplate	Default Schedule Template	2	1		18-NOV-2014 09:34
1857	BG_14_2_1_8_optimizeSelectedScenario	DefaultScheduleTemplate	Default Schedule Template	2	1		18-NOV-2014 09:34
1825	Redispatch	DefaultScheduleTemplate	Default Schedule Template		1		13-NOV-2014 12:27

To view, assign territories to or unassign territories from schedules:

- 1 From the **Schedules** tab on the **Territories** list page, right-click on one or more schedules in the list and select the **Assign Territories** right-click option. The **Assign Territories** page appears.

Assign Territories

SAVE CLOSE

Filter

Find SEARCH

Assign Territories For Selected Schedules

SELECTED TERRITORY | Row 1 - 4 of 4

TerritoryKey
TC1_13_1_3_4
Res_territory_ReservationsDemo
TC3_ReservationsDemo
TC1_ReservationsDemo

<< ADDALL << < ADD < > REMOVE > >> REMOVEALL >>

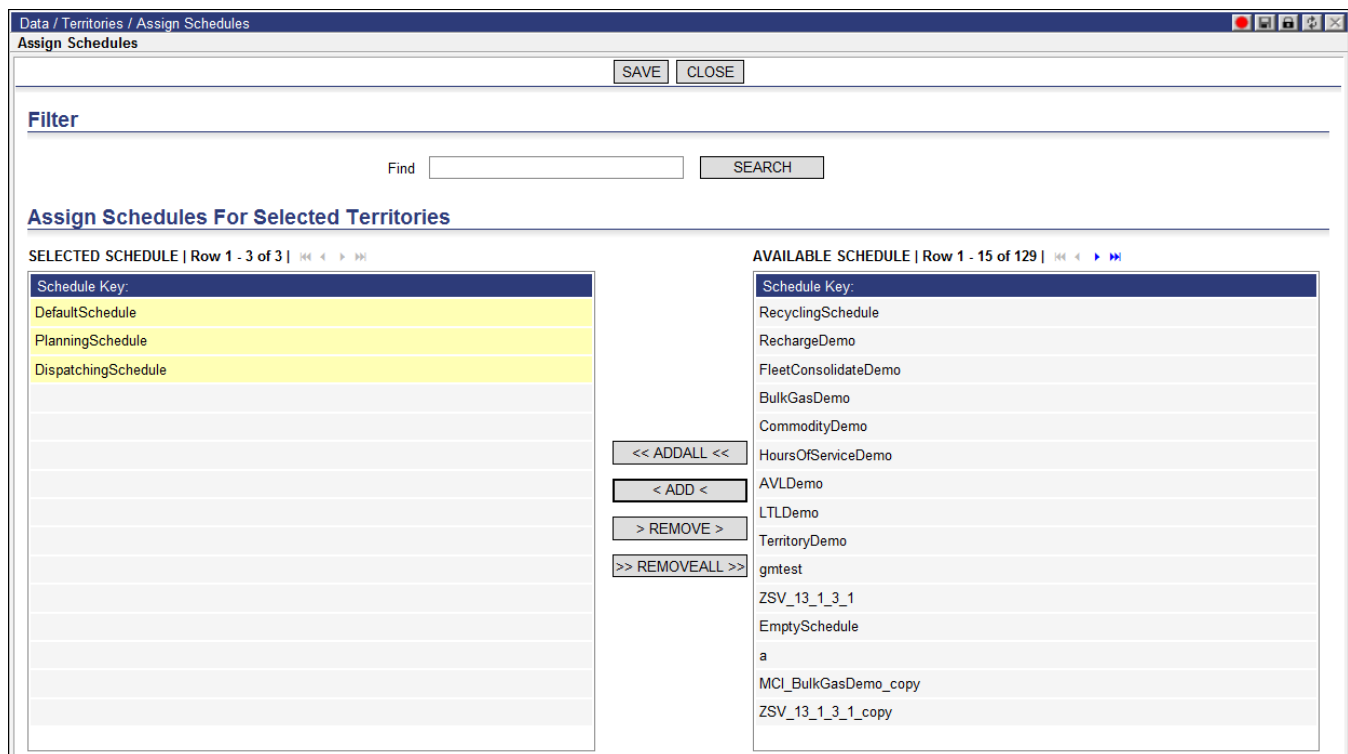
AVAILABLE TERRITORY | Row 1 - 15 of 65

TerritoryKey
Res_territory_Reservation_ZSV
TC3_Reservation_ZSV
ddd
1111
test9999
asdfasdfsdf
avl
ca
CA2
AVL_13_2_1_1
CA3
CA4
Fleet
MUR_terr_full
Minneapolis

- 2 Use the Filter to search for specific territories. To add territories, select one or more rows from the Available Territory section and click the **ADDALL** or **ADD** buttons. The selected items will display in the Selected Territory section. To remove territories, select one or more rows from the Selected Territory section and click the **REMOVE** or **REMOVEALL** buttons.
- 3 Click **Save** to complete the assignment process.

To view, assign schedules to or unassign schedules from territories:

- 1 From the **Territories** list page, right-click on one or more territories in the list and select the **Assign Schedules** right-click option. The **Assign Schedules** page appears.



- 2 Use the Filter to search for specific schedules. To add schedules, select one or more rows from the Available Schedule section and click the **ADDALL** or **ADD** buttons. The selected items will display in the Selected Schedule section. To remove schedules, select one or more rows from the Selected Schedule section and click the **REMOVE** or **REMOVEALL** buttons.
- 3 Click **Save** to complete the assignment process.

Deleting a Territory

To delete a territory:

- 1 Select **Data > Territories** from the main menu. The **Territories** page appears.
- 2 Right-click on the desired territory and select **Delete**.

Importing and Exporting Territories

Users can import and export territories through the UI.

To import territories:

- 1 On the **Territories** list page, right-click in the list select **Import** from the right-click menu. The **Import Territory** window appears.

- 2 Use the Browse button to select the Source XML.
- 3 Choose a schedule from the **Schedule** drop-down menu
- 4 Click **Load** to import.

To export territories:

- 1 Select **Export** from the right-click menu. The **Export Territory** window appears.

- 2 Enter the Export File Path and click **Save**.

Managing Territory Swap Sets

In a Territory Swap Set, users can define any of the following:

- Pairs of Territories that the Optimizer can swap Orders between. (type Swap)
- Pairs of Territories that the Optimizer cannot swap Orders between. (type No Swap)
- Major / Minor Territories. (type Major/Minor)

Users can create new Territory Swap Sets from the Data > Territories > Swaps list page. Only one Territory Swap Set can be linked to a schedule at a time. In the Use section of the **Schedule Settings** page, enable Territory Swap Set and Zone File Path options (see Zone File section below).

To create a new territory swamp set:

- 1 From the **Swaps** list page, right-click and select **New** from the right-click menu. The **New Territory Swap Set** page appears.

- 2 Enter a unique name for the new swap set.
- 3 In the Territory Swap Definition section, add the desired records. Each record defines either a Swap, NoSwap or Major/Minor territories and has a unique name entered in the Territory Swap Key table.
- 4 Either type a Territory name into the Territory Key box or use the browse button to select a Territory previously setup with FW Territory Planner.
- 5 For Swaps and NoSwaps definitions type a Territory name in the Definition field or use the browse button.
- 6 For Major/Minor Territories definitions a comma separated list of Territory Names can be typed into the Definition field.
- 7 Use the drop down list to select the type of definition for this record.

Swap Definition

If two Territories are set to swap, the Optimizer can create a route that crosses between these two Territories. A swap definition that is one Territory name and an

asterisk (*) indicates that the Territory can be swapped with any other Territory. Two asterisks instead of Territory names indicates that any territory can be swapped with any other territory.

Territory Swap Definition

TerritorySwapKey	TerritoryKey	TerritorySwapDef	TerritorySwapType
Territory1-swap	Territory1	Territory2	1
TerritoryA-swap	TerritoryA	*	1
All swap	*	*	1

If the Optimizer exchanges Orders between two resources in different Territories, the resulting routes will serve two different Territories. It is therefore important to only define pairs of Territories which can be served by one resource. For example, if the service area is divided into FarWest, West, East, and FarEast, then it could be desirable to allow swaps between FarWest and West, but probably not between FarWest and FarEast.

No Swap Definition

The No Swaps record type defines pairs of Territories that the Optimizer cannot swap Orders between. If two Territories are paired in a No Swaps record, then the Optimizer cannot create a route that crosses between these two Territories. A No Swap definition that is one Territory name and an asterisk (*) indicates that a territory cannot be swapped with any other territory:

TerritorySwapKey	TerritoryKey	TerritorySwapDef	TerritorySwapType
Nothing Swapping	*	*	2

An asterisk instead of both Territory Names indicates that no Territories can be swapped.

Using Swaps and No Swaps Together

Swaps and NoSwaps can be used in conjunction. The default swapping is determined by the **Allow Swaps For All** checkbox. If **Allow Swaps For All** is not selected, then pairs cannot be swapped unless they are specified by a Swaps type record in the Territory Set Definition. If **Allow Swaps For All** is selected, then pairs can be swapped unless they are specified by a NoSwaps type record in the Territory Set Definition.

For example, given four Territories named TerritoryA, TerritoryB, TerritoryC, and TerritoryD, the following records allow swaps between TerritoryA and TerritoryC, and between TerritoryA and TerritoryD. All other swaps are prohibited.

Major and Minor Territories

Descartes Route Planner can divide Territories into major and minor Territories. Territories setup using Territory Planner or already defined on an order by the customer can be used as they are or grouped together into a Major Territory by a Territory Swap Set definition record.

- **Major Territory:** A major territory is any territory that is not contained by another territory. A major territory only appears once as the Territory Key in all the records in the Territory Swap Set.
- **Minor Territory:** A minor territory is any territory that is contained by another territory.

In the following example of Major/Minor type records in a Territory Swap Set, the only major Territories are WestBoroughs and EastBoroughs. The remaining Territories are all minor Territories.

Territory Swap Definition

TerritorySwapKey	TerritoryKey	TerritorySwapDef	TerritorySwapType
ManhattanSet	Manhattan	Manhattan1, Manhattan2	0
BronxSet	Bronx	Bronx1, Bronx2	0
WestBoroughsSet	WestBoroughs	Manhattan, Bronx	0
BrooklynSet	Brooklyn	Brooklyn1, Brooklyn2, Brooklyn3	0
QueensSet	Queens	Queens1, Queens2	0
EastBoroughsSet	EastBoroughs	Brooklyn, Queens, StatenIsland	0

The system is only concerned with major Territories. In the example shown above, a resource with a territory of Queens1 is considered within the EastBoroughs territory, and is eligible to serve all Orders in Brooklyn, Queens, or Staten Island. A resource with a territory of Bronx2 is considered within the WestBoroughs territory, and is eligible to serve all Orders in Manhattan and the Bronx.

Zone File Path

This schedule property defines a temporary file that the system can use when generating a territory swap file for the optimizer. The path must exist but the file will be created or overwritten as necessary. The file name must have an lza extension. Since the file must be unique across schedules it is suggested that the schedule key be part of the name. So for schedule Branch21 the Zone File Path might be:

C:\LNOS\Fleetwise\Branch21.lza

Importing and Exporting Territory Swap Files (TXT or XML)

Users can import or export territory swap files (.txt or XML) from the **Data > Territories > Swaps** list page. These files can replace existing swap sets.

To import:

- 1 Right-click on the list page and select **Import** from the right-click menu. The **Import Territory Swap** window appears.

The screenshot shows the 'Import Territory Swap' window. At the top, there are two tabs: 'Import Territory Swap' and 'Export Territory Swap'. Below the tabs are two buttons: 'LOAD' and 'CLOSE'. The main content area is titled 'Import Territory Swap' and contains a 'Source XML' text input field followed by a 'Browse...' button. Below this is a section titled 'Note' with a scrollable text area.

- 2 Select the Source XML and schedule and click **Load** to import.

To export:

- 1 Select the **Export Territory Swap** tab. The **Export Territory Swap** window appears.

The screenshot shows the 'Export Territory Swap' window. At the top, there are two tabs: 'Import Territory Swap' and 'Export Territory Swap'. The 'Export Territory Swap' tab is selected. Below the tabs is a 'SAVE' button. The main content area is titled 'Export Territory Swap Set' and contains an 'Export File Path' text input field. Below this is a section titled 'Note' with a scrollable text area.

- 2 Enter the Export File Path and click **Save**.

The following associated business document is available for use:

Listener: <http://<servername>/stad/listener/DocFWConfigListener.asp>

INPUT:

```

<DocFWImport>
<Header senderID="LNOSFWAnalyzer" ReceiverID="TOE"
SendDateTime="12/9/2005 9:35:19 AM"
extDocControlID="LNOSFWAnalyzer" CompanyName="FW"
LoginName="FW-Admin" Password="cs" />
<Request>
<DocTerritorySwapSet MessagePurpose="" ProcessCode=""
TerritorySwapSetKey="" AllowSwapForAll="">
<FWTerritorySwap Description="" TerritorySwapKey="" TerritoryKey=""
TerritorySwapType="" TerritorySwapDef="" >
</FWTerritorySwap>
</DocTerritorySwapSet>
</Request>
</DocFWImport>

```

Managing Datasets

➡ **Note**— For more information on the configuration of BGO properties, please see the *Descartes Route Planner BGO User's Guide*.

Data Slice

Data Slices identify a problem set used to differentiate between different Schedule problem subsets. These Data Slices are typically used to define a problem for background optimization (BGO), planning, or dispatch. Data Slices are predefined and used to query different Schedule problem subsets within the dispatch application.

Managing Data Slices

To access a data slices:

- 1 From the main menu, select: **App Setup > Dataset > Data Slice**. The list of data slices appears.
- 2 Right-click on a selected data slice record and select one of the management options:
 - **Edit:** Displays the **Edit Data Slice** page
 - **Clone:** Allows users to clone a data slice by entering a new Data Set Group Key value.
 - **Assign Data Slice Set:** Displays the **Assign Data Slice Set window**, allowing users to select a data slice set to assign to the data slice from a dropdown menu.

- **Delete:** Removes the selected data slice
- **History:** Displays the Data Slice Group History list page for the selected data slice
- **Activate:** Activates the selected data slice(s)
- **Deactivate:** Deactivates the selected data slice(s)
- **Values:** Displays the Data Set Values list page
- **New:** Displays the new Data Slice page, allowing users to create a new data slice

It is also possible to double-click a specific data slice record to directly edit that record.

When creating or editing a data slice, the selected **Data Slice** page appears:

Data Slice

Data Slice Name: DefaultBGO

AreaKey: -NoAreas- AREAS

Type: Background Optimizer

Active: ☐ (This must be on for the Data Slice to be optimized.)

Use Dispatch-BGO: ☒ (This flag must be on if selecting routes in execution.)

MasterRoute: ☐

Status: Free

ServiceName: -None-

LastEvaluated: 19-OCT-2016 01:46

Data Slice Set: ZVE

NOTE: Active check box.

The Data Slice will not be assigned to a BGO and optimized unless this is on. Applies to all types of Data Slice.

Leave this off while setting up the Data Slice initially or making changes to it.

If this is turned off while a BGO is optimizing the Data Slice then it will stop on the next refresh.

The refresh interval is determined by optimizer properties RefreshData.

NOTE: XML Logging Mins.

This setting should only be changed when asked to do so by Descartes Support for troubleshooting.

Setting this to X minutes makes the BGO write out all the xml sent back and forth between the BGO and Route Planner when optimizing this Data Slice. It will automatically turn off after X mins and this setting reset to 0.

XML Logging Mins: 10

- 3 Enter data in the appropriate fields. The **Active** setting is disabled by default in order to allow users to configure the data slice first and then activate it when configuration is complete. In cases where the data slice is active and being optimized and this setting is turned off, the system will behave as follows:
 - The data slice will not be assigned to a BGO instance.
 - Or, if a BGO instance is currently optimizing the data slice, the next time that the BGO requests a refresh of the data slice, the data slice will not be returned, causing BGO instance to move on to another data slice. The refresh interval is controlled by the RefreshData optimizer property:
com.descartes.escheduler.optimizer.refresh_inosfw_data
- 4 Click the **Areas** button to return a list of configured areas in the **Areas** window. Double-click an area to populate the AreaKey field.

Data Slice

Data Slice Name:

AreaKey

App Setup / Data Slice / New Data Slice / Areas

Areas

AreaID	AreaKey
1	-NoAreas-
2	area1
3	nonactive
4	no
5	
6	
7	noactive
8	-NoAreas-nonactive
9	-NoAreas-nonactive
10	-NoAreas-no
11	XYZ

Rows 1 - 11 of 11 Rows per Page: 10 / 20 / 50 Page 1

NOTE: XML Logging Mins.

- 5 Enter a value in the XML Logging Minutes to enable logging of all XMLs between the BGO and Descartes Route Planner from the user interface for a certain time period (in minutes). After the set number of minutes expires, logging will stop and the setting will revert to a value of "0" minutes. This enhancement eliminates

the need for the user to remember to disable logging after it has been enabled. When enabled for long periods of time, XML logging could use a significant amount of disk space.

6 When finished entering data, click **Save**.

This data slice can now be used as the basis for defining the Schedule problem set. The data slice is the first of three table definitions needed to complete the definition. In addition to the data slice setting, the data nesting and data value table need to be defined.

Nestings

Nestings specify settings that narrow down a problem set based on nested criteria. Users can define up to six different nested attributes for a given table.

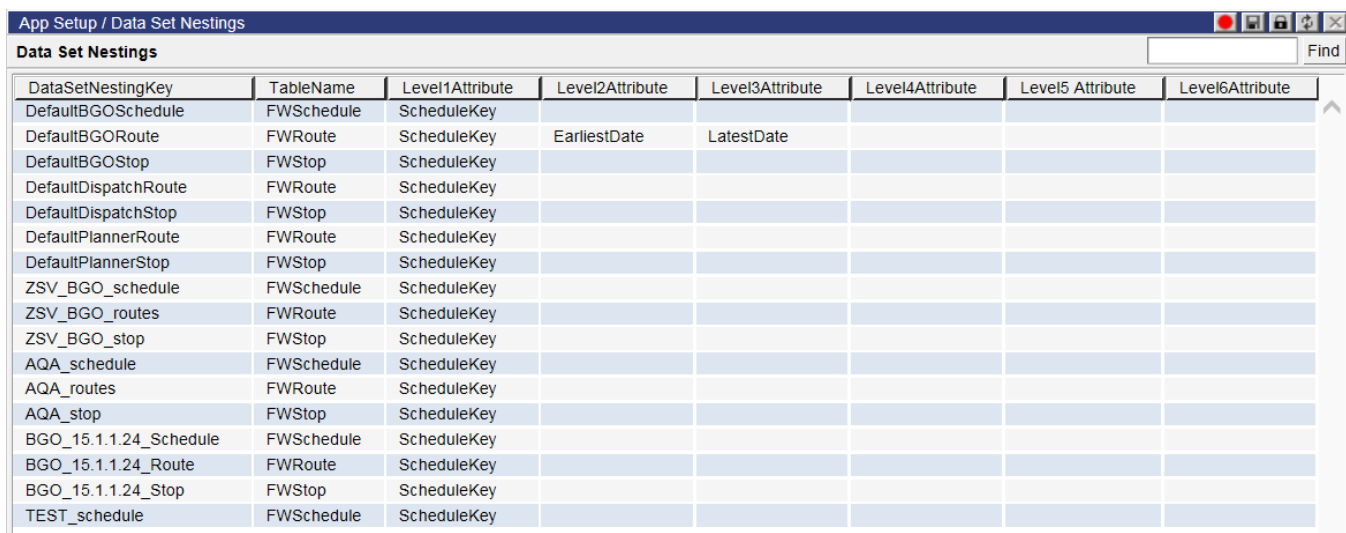
***Example:** If specifying a territory at the highest level, and a driver at the next level, the problem is restricted to that driver within that territory, instead of that driver within all available territories.*

This example represents two nested attributes.

Managing Data Nestings

To access a data nesting record:

- 1 From the main menu, select: **App Setup > Dataset > Nestings**. The list of nestings appears:



DataSetNestingKey	TableName	Level1Attribute	Level2Attribute	Level3Attribute	Level4Attribute	Level5 Attribute	Level6Attribute
DefaultBGOSchedule	FWSchedule	ScheduleKey					
DefaultBGORoute	FWRoute	ScheduleKey	EarliestDate	LatestDate			
DefaultBGOSTop	FWStop	ScheduleKey					
DefaultDispatchRoute	FWRoute	ScheduleKey					
DefaultDispatchStop	FWStop	ScheduleKey					
DefaultPlannerRoute	FWRoute	ScheduleKey					
DefaultPlannerStop	FWStop	ScheduleKey					
ZSV_BGO_schedule	FWSchedule	ScheduleKey					
ZSV_BGO_routes	FWRoute	ScheduleKey					
ZSV_BGO_stop	FWStop	ScheduleKey					
AQA_schedule	FWSchedule	ScheduleKey					
AQA_routes	FWRoute	ScheduleKey					
AQA_stop	FWStop	ScheduleKey					
BGO_15.1.1.24_Schedule	FWSchedule	ScheduleKey					
BGO_15.1.1.24_Route	FWRoute	ScheduleKey					
BGO_15.1.1.24_Stop	FWStop	ScheduleKey					
TEST_schedule	FWSchedule	ScheduleKey					

- 2 Right-click on a selected data nesting record and select one of the management options: **New**, **Edit**, or **Delete**. It is also possible to double-click a specific data nesting record to directly edit that record.

The selected **Data Nesting** page appears:

- 3 Enter data in the appropriate fields.
- 4 When finished entering data, click **Save**.

Values

Data values link data groups and data nestings, and identify the actual data to be used. Data values provide the link between a many-to-many relationship with data groups and data nestings, and provide the ability to specify the actual data values needed to perform specific problem set extractions.

Managing Data Values

To access a data value record:

- 1 From the main menu, select: **App Setup > Dataset > Values**.

The list of data values appears:

DataSetGroupKey	DataSetNestingKey	EndLevel	Level1Value	Level2Value	Level3Value	Level4Value	Level5Value	Level6Value
DefaultBGO	DefaultBGOSchedule	1	"					
DefaultBGO	DefaultBGORoute	1	"					
DefaultBGO	DefaultBGOSStop	1	"					
BGO	ZSV_BGO_schedule	1	'BGO'					
BGO	ZSV_BGO_routes	1	'BGO'					
BGO	ZSV_BGO_stop	1	'BGO'					
TEST	TEST_schedule	1	'ZVE1'					
BGO_party	BGO_15.1.1.24_Schedule	1	'BGO_party'					
BGO_party	BGO_15.1.1.24_Route	1	'BGO_party'					
BGO_party	BGO_15.1.1.24_Stop	1	'BGO_party'					

- 2 Right-click on a selected data value record and select one of the management options: **New**, **Edit**, or **Delete**. It is also possible to double-click a specific data value record to directly edit that record.

The selected **Data Value** page appears:

DataSet Value

DataSetGroupKey: DefaultBGO

DataSetNestingKey: DefaultBGOSchedule

EndLevel: 1

Level	Attribute	Operator	Value
Level 1	ScheduleKey	=	
Level 2			
Level 3			
Level 4			
5:			
6:			

LevelXValue: The following values are acceptable:
 String -> Must be within single quotes: 'TC1'
 Number -> Free form: 12.4
 Date -> It must be within quotes and in the format that sql server has been setup to. Example: 'mm/dd/yyyy hh:mm:ss'

LevelXOperator: Accepts any SQL Server operator and can be enter in a free form:
 Examples: =, IN, >=, <=, LIKE, <>, DATE>=, DATE<=, DATE=, <, >, =WITHNULL, INWITHNULL

- 3 Enter data in the appropriate fields.
- 4 When finished entering data, click **Save**.

The specified data values, data nesting, and data groups can now be used by the other Descartes Route Planner applications to extract smaller pieces of the Schedule for processing.

Configuring Background Optimizer (BGO) Properties

The BGO properties contained in the optimizer.properties file can be setup in the database from the Descartes Route Planner user interface. From the **App Setup > Dataset > Properties** page, users can create or import new sets of Data Set Properties to link to Data Slice Sets (**App Setup > Dataset > Data Slice Sets**). Data Set Properties and Data Slice Sets can then be linked to Data Slice Set Schedules (**App Setup > Dataset > Data Slice Set Schedules**).

➡ **Note**— For more information on the configuration of BGO properties, please see the [Descartes Route Planner BGO User's Guide](#).

App Setup / Data Set Properties	
Data Set Properties	
Set Name	Description
OptimizeAll	Run opt all
Default-ReadOnly	Properties
Base	base set

Multiple sets of optimizer properties can be generated that perform different optimizations. For instance, a set of properties that only runs Intra/Resequencing and Assignment could be setup called "IntraAssignOnly" and another set that runs all the BGO incremental optimizations appropriate for reservations could be called "ReservationsOpts".

App Setup / Data Set Properties / New Data Set Properties	
New Data Set Properties	
SAVE CANCEL	
Option Set Name:	
Description:	
DATA SLICE OPTIMIZATION TIME / SETTINGS	
DataSliceOptimizationTime	60
RouteOptimizationCount	2
AssignCount	2
MinimumCycleTime	60
OptimizeChangedRoutesFirst	☒
RefreshData	5
LNOSTimeout	10
HandleExceptionsCount	0
ExceptionSleep	5
SliceCompleteCount	0
PostStatistics	☐
CONTROL WHICH OPTIMIZATION PHASES TO RUN	
AssignUnassignedOrders	1 ▼
Intra	1 ▼
BatchAssign	1 ▼
Inter	1 ▼
CheckRouteLoc	0 ▼
CheckOrderDateViolation	0 ▼

Data slices can then be linked to sets of optimizer properties and can be setup to use a set of optimizer properties at a certain time of day.

So all reservation data slices that select today+1 could use the ReservationsOpts properties from 00:01 to 18:00 (the cutoff time). Outside of those times the data slices will not be optimized.

If a data slice is linked to optimizer properties in the database, then those are used by the BGO instead of the ones in the actual optimizer.properties file. If a data slice is not configured to use the properties from the database, the BGO will use the actual optimizer.properties file when optimizing this slice. This allows for backward compatibility so that when upgrading to Descartes Route Planner 15.2, the BGO properties do not have to be setup in the database immediately, the BGO will continue to work as before. Also the BGO can cope with cases where some data slices are configured to use optimizer properties from the database and some are not. The BGO can switch between these cases.

Configuring the Application

This functionality specifies the communication channel to use when publishing data to other systems.

To access this functionality, from the main menu, select: **App Setup > Application Configuration**.

The following screen appears:

The screenshot shows the 'Application Configuration / Application Configuration' window. It has tabs for 'Application Configuration', 'Key Wizard', and 'Time Zone'. The 'Application Configuration' tab is active. At the top are 'SAVE' and 'SAVE AND CLOSE' buttons. The window is divided into four main sections:

- Publishing Settings:**
 - AccountName: test
 - ConnectionType: Local Path (dropdown)
 - Local Publish Path: C:\inetpub\wwwroot\LNOS FW UI\GeneratedFiles\PublishedFile
 - SenderID: 12
 - ReceiverID: 23
 - Publish Stop: ☒
 - Order Status Caption Text: (empty)
 - TEST CONNECTION button
- wGLN Settings:**
 - Use wGLN: ☒
 - wGLN URL: http://wirelessuat.gln.com
 - wGLN Handler URL: https://wirelessuat.gln.com/wGlnEnterprise/MRMInterfaceService
 - wGLN Company Name: 855060828
 - wGLN User Name: adistribuidora
 - wGLN User Password: (masked with dots)
 - wGLN Connection Verified: ☐
 - TEST CONNECTION button
- AutoPublish Settings:**
 - UseAutoPublish: ☒
 - MaxTimeBeforeDepart: 1 Minutes
 - MaxTimeDiffProjDepartedDate: 1 Minutes
 - MaxTimeDiffProjArrivedDate: 1 Minutes
 - MaxDistDiffProjectedDistance: 0.0999 km
 - MaxDiffPotentialAvailCapacity1: 100
 - MaxDiffPotentialAvailCapacity2: 100
 - MaxDiffPotentialAvailCapacity3: 100
 - MaxDiffPotentialAvailCapacity4: 100
 - MaxDiffPotentialAvailCapacity5: 100
 - MaxDiffPotentialAvailCapacity6: 100
 - MaxDiffPotentialAvailCapacity7: 100
 - MaxDiffPotentialAvailCapacity8: 100
 - MaxDiffPotentialAvailCapacity9: 100
 - MaxDiffNumberOfStops: 3
- Call Out/Notify Settings:**
 - Use Call Out/Notification: ☒
 - Call Out/Notification Profile: (dropdown menu)

From this page, the user can use the fields to adjust Publishing Settings, wGLN Settings, AutoPublish Settings and Call Out/Notify Settings.

Published Settings

The 'Published Settings' dialog box contains the following fields:

- AccountName: EA3006
- ConnectionType: Local Path (dropdown)
- Local Publish Path: C:\Test
- SenderID: EA3006
- ReceiverID: EA3006
- Publish Stop: ☒
- Order Status Caption Text: (empty)
- TEST CONNECTION button

- In the AccountName, ConnectionType, Local Publish Path, SenderID and ReceiverID fields, enter the appropriate information and click **Save** or **Test Connection**. The following communication channels are supported:
 - Ftp
 - http
 - local path
- Select the **Publish Stop** checkbox to enable the individual stop publishing and tracking. A new column, Published, becomes available at the stop level that tracks whether or not the stop or associated route has been published. A new right-click option, **Publish**, becomes at the stop level of the **Route Detail** quadrants.
- In the OrderKey Status Caption Text field, enter any customized text for use in the Order Search Status tracking page. For more information on Order Tracking, see the *Tracking Order Status* section in the *Descartes Route Planner User's Guide*.

wGLN Settings

Users that use wGLN as their mobile communications gateway can configure their wGLN settings within Route Planner.

wGLN Settings

Use wGLN: ☒

wGLN URL:

wGLN Handler URL:

Network Repository URL:

wGLN Company Name:

wGLN Authentication:

wGLN User Name:

wGLN User Password:

To configure wGLN, click the Use wGLN checkbox and fill in the appropriate fields. Click **Save** or **Test Connection** when finished.

For the Descartes wGLN Asset Grouping feature to work properly, one of the following URLs must be configured in the wGLN Settings section of the Application Configuration page for the Network Repository URL. For more information on this feature, please see the *Descartes Route Planner User's Guide*.

- <https://wirelessuat.gln.com/rest/v1/wglnconfiguration> - test

- <https://wireless.gln.com/rest/v1/wglconfiguration> – production

AutoPublish Settings

AutoPublish allows the Route Planner user to automatically generate routes to wireless devices based on the predefined, configurable criteria on the **Application Configuration** page.

AutoPublish Settings

UseAutoPublish

☒

MaxTimeBeforeDepart:

MaxTimeDiffProjDepartedDate:

MaxTimeDiffProjArrivedDate:

MaxDistDiffProjectedDistance:

MaxDiffPotentialAvailabilityCapacity1:

MaxDiffPotentialAvailabilityCapacity2:

MaxDiffPotentialAvailabilityCapacity3:

MaxDiffPotentialAvailabilityCapacity4:

MaxDiffPotentialAvailabilityCapacity5:

MaxDiffPotentialAvailabilityCapacity6:

MaxDiffPotentialAvailabilityCapacity7:

MaxDiffPotentialAvailabilityCapacity8:

MaxDiffPotentialAvailabilityCapacity9:

MaxDiffNumberOfStops:

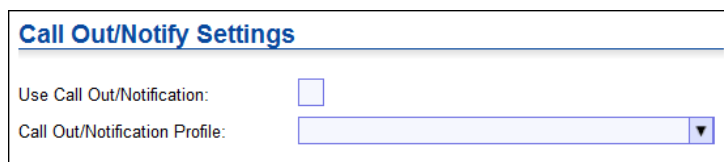
To turn on and configure AutoPublish Settings:

- 1 Click the **Use AutoPublish** checkbox.
- 2 Fill in each field with the preferred criteria for AutoPublish.
 - **MaxTimeBeforeDepart**— Auto publish 'x' minutes before planned departure.

- **MaxTimeDiffProjDepartedDate**— Redispatch when start time is changed by 'x' minutes.
 - **MaxTimeDiffProjArrivedDate**— Redispatch when end time is changed by 'x' minutes.
 - **MaxDistDiffProjectedDistance**— Redispatch when route distance is changed by 'x' miles/kilometers.
 - **MaxDiffPotentialAvailCapacity1...9**— Redispatch when route volume is changed by 'x' measures.
 - **MaxDiffNumberOfStops**— Redispatch when number of stops are changed.
- 3 When finished, click **Save** to save your changes and continue editing, or **Save and Close** to exit the **Configure Application** page.

Call Out/Notify Settings

To enable the callout/notification functionality globally, select **App Setup > Application Configuration**.



Call Out/Notify Settings

Use Call Out/Notification: ☐

Call Out/Notification Profile: ▼

Select the **Use Call Out/Notification** checkbox and enter the relevant profile number to enable the functionality. For more information on callouts and notifications, see the [Managing Callouts and Notifications](#) section in this document.

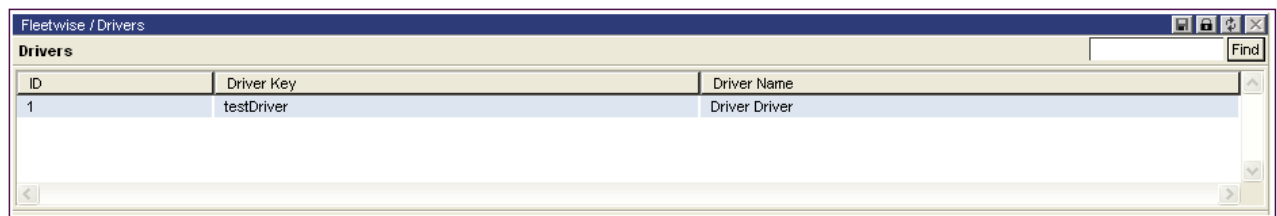
Managing Drivers

Driver Management allows users to create Drivers that can be later assigned to a Route. Typically, Drivers are assigned to a Route when the dispatching function is used.

To access Driver Management:

- 1 From the main menu, select: **Data > Assets > Drivers**.

The following screen appears.



ID	Driver Key	Driver Name
1	testDriver	Driver Driver

- 2 Right-click on a selected Driver record and select one of the management options: **New**, **Edit**, or **Delete**. It is also possible to double-click a specific data value record to directly edit that record.

The selected **Driver** page appears:

Driver

Driver Key: Driver_7212014_102731330

Driver Name:

Segment:

Role:

Location:

Email:

Phone Number 1:

Phone Number 2:

Certificates:

License Number:

License Issue Date:

License Exp Date:

Earliest Start Time:

Latest Start Time:

Earliest End Time:

Latest End Time:

Service Factor:

CostProfile: Select Cost Profile...

Party: Select Party...

Dummy:

Comment:

wGLN

wGLN SyncInd:

wGLN Response:

WGLN Group:

3 Enter information in the fields as needed.

Driver Key: This field is a unique identifier for the Driver. It is important to notice that this Driver Key will be used by the wireless functionality of Descartes Route Planner, if implemented, to identify the incoming wireless messages and to update the appropriate Route in the Schedule.

Driver Name: A unique name for this record.

Note— For more information on the other fields, see Creating a New Driver in the Descartes Route Planner User's Guide. For information on making edits to

an existing driver, see Editing a Driver in the Descartes Route Planner User's Guide.

Managing Messages

The **Messages** page allows users to view, update, delete or publish preconfigured messages that can be relayed to a Driver.

To access this functionality:

- 1 From the main menu, select: **Data > Messages**.

The following screen appears:

Fleetwise / Messages							
Messages							
From Key	To Key	Schedule Key	Template:	Subject	Send Date	Sent	Message Key
1	1		False			False	2
2	2		True			False	2
1	1		False			False	1

- 2 Right-click on a selected Message record and select one of the management options: **View**, **Status Setting**, **Delete** or **Publish**. It is also possible to double-click a specific data value record to directly edit that record.

- 3 Click **View** to display the **View Messages** page:

Data / Messages / View Messages

View Messages

CLOSE

Message

Route ID

8709

Schedule Key:

Recharge_copy_without_assets

Subject

yyy

Body

yyy

Status

Read

From

fw-admin

Driver Key

test_driver_4

TruckKey

ZSV_Truck_11

Trailer Key

Resource Key

Resource-1508-2

Direction

Sent

Send Date

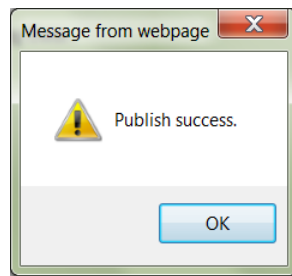
25-OCT-2013 17:39

Acknowledged Date:

Read Date:

25-OCT-2013 17:39

- 4 To update the status of a message, select **Status Setting**. The Status field becomes a drop-down menu, allowing the user to select a new status for the message.
- 5 Click **Publish** to publish the message. Users will receive the following dialog if the message is successfully published.

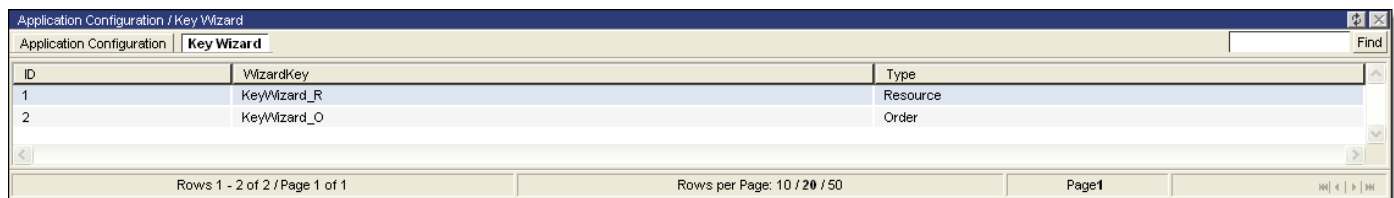


Managing Key Wizards

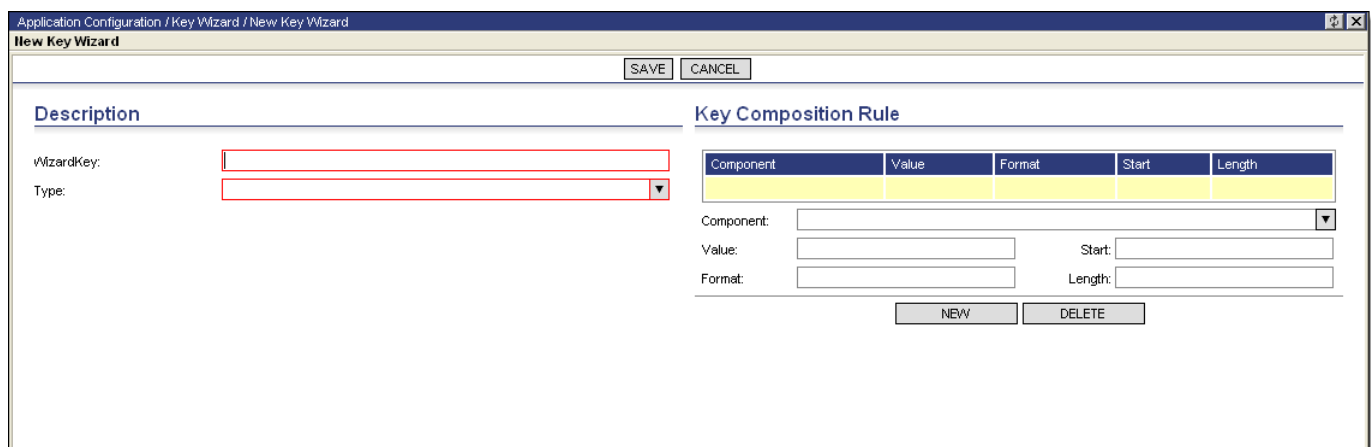
This functionality allows users to create Key Wizards that can be used later in Routing.

To access Key Wizards:

- 1 From the main menu, select: **App Setup > Key Wizard**.



- 2 Right-click on a selected WizardKey and select one of the management options: **New**, **Edit**, **Copy To**, or **Delete**. It is also possible to double-click a specific data value record to directly edit that record.



- 3 Enter the WizardKey name and select the type.
- 4 Under **Key Composition Rule**, select a component and enter the valid fields for it. The following components are available:
 - **Counter** – no additional fields are valid
 - **Date** – Value and Format fields are valid. Users can input the following formatting values for date components:
 - **%a** Abbreviated weekday name
 - **%A** Full weekday name
 - **%b** Abbreviated month name
 - **%B** Full month name
 - **%c** Date and time representation appropriate for locale
 - **%d** Day of month as decimal number (01 ? 31)
 - **%H** Hour in 24-hour format (00 ? 23)
 - **%I** Hour in 12-hour format (01 ? 12)
 - **%j** Day of year as decimal number (001 ? 366)
 - **%m** Month as decimal number (01 ? 12)
 - **%M** Minute as decimal number (00 ? 59)
 - **%p** Current locale's A.M./P.M. indicator for 12-hour clock
 - **%S** Second as decimal number (00 ? 59)
 - **%U** Week of year as decimal number, with Sunday as first day of week (00 ? 53)
 - **%w** Weekday as decimal number (0 ? 6; Sunday is 0)
 - **%W** Week of year as decimal number, with Monday as first day of week (00 ? 53)
 - **%x** Date representation for current locale
 - **%X** Time representation for current locale
 - **%y** Year without century, as decimal number (00 ? 99)
 - **%Y** Year with century, as decimal number
 - **%Z** Either the time-zone name or time zone abbreviation, depending on registry settings; no characters if time zone is unknown
 - **%%** Percent sign
 - **Delimiter** – Value field is valid
 - **FinalLocationKey** – Start and Length fields are valid
 - **InitialLocationKey** – Start and Length fields are valid
 - **Literal** – Value field is valid
 - **MasterKey** – Start and Length fields are valid
 - **Prefix** – Start and Length fields are valid
 - **Random** – Format field is valid
 - **ScheduleKey** – Start and Length fields are valid

- **SeqNo** – Value field is valid
- **Suffix** – Start and Length fields are valid
- **SysDate** – Value and Format fields are valid
- **Territory** – Start and Length fields are valid

To add additional components, click **New**.

5 Click **Save**.

Customizing Icons for Routes and Stops

Users can configure quadrant panes to display custom icons and colors depending on user-defined criteria for routes and stops. The following UI elements are customizable:

- Line color for a route on map
- Icon for a stop on map
- Column 'CustomIcon' on Route tree
- Column 'CustomIcon' on Unassign Stops tree
- Column 'CustomIcon' on route/stop levels of Route Detail tree

To configure these settings:

1 Select **App Setup > Icon Customization > Mappings**.

2 Right-click in the list and select **New**. The **New Mapping Step** page appears.

3 Enter the **MappingKey** and then specify the **Domain** and **Content** using the drop-down menus. When selecting Tree for the domain, choose a node from the drop-down menu that appears.

4 Click the **Next Step** button. The **New Mapping Step 2** page appears.

- 5 Specify details in the Customization section. Select the **Is Default** checkbox to make the selected icon or color the default.
- 6 Select a custom icon image or color from the drop-down menu. Click **New** to add another icon or color to the table.

Once configuration is saved to database, add the new column 'CustomIcon' to the associated trees in order to see the icons.

Custom icons need to be placed in the folder /Inos fw ui/AppIcon/CustomIcons. A set of basic shape icons (diamond, circle, square, star, pentagon, triangle) in various colors is included in the package. Users can add their own version of custom icons in this same folder for use in the application.

Items with custom icons can be manipulated by using the following right-click options on the **Dashboard**:

- **Show by Custom Icon:** Filters and displays content based on a specified custom icon profile.
- **Hide by Custom Icon:** Disables the Show by Custom Icon functionality.

API Tracking for Resources, Schedule and Orders

Users can track Schedule APIs, Resource APIs, Published Route APIs and Order APIs through the UI.

To access these API lists, select **Data > API Tracking > Schedules, Resources, Publish Routes** or **Orders**. Use the respective tabs to navigate between types of APIs.

Setting TreeControl Layout to Org Default

This functionality allows users to set an organization-level layout default. Users can reset to that default instead the default.

There are two ways to set layout to org level.

Select **User Profile > Copy Quadrant ProfileItems To Org**. The system then copies all of the existing layouts to an org default.

Home	Data	Summary Reports	App Setup	Data Filters	User Profile	UIF Tools	Setup	Help																																																			
Home / Dashboard					Change Password																																																						
Dashboard					Color Scheme																																																						
Routes					Reset All Columns																																																						
Customize					Preferences																																																						
Reset					Change User Group																																																						
Advanced Find					Profile Items																																																						
Show All					Copy Quadrant ProfileItems To Org																																																						
Auto H					Ext Config Account																																																						
Routes Row 1 - 18 of 18																																																											
<table><tr><th>Route ID</th><th>Resource Key</th><th>Name</th></tr><tr><td>3792</td><td>LTL D1 V1 23</td><td>Truck</td></tr><tr><td>3793</td><td>LTL D1 V2 23</td><td>Truck</td></tr><tr><td>3794</td><td>LTL D1 V3 23</td><td>Truck</td></tr><tr><td>3795</td><td>LTL D1 V4 23</td><td>Truck</td></tr><tr><td>3796</td><td>LTL D1 V5 23</td><td>Truck</td></tr><tr><td>3797</td><td>LTL D1 V6 23</td><td>Truck</td></tr><tr><td>3798</td><td>LTL D1 V7 23</td><td>Truck</td></tr><tr><td>3799</td><td>LTL D1 V8 23</td><td>Truck</td></tr></table>					Route ID	Resource Key	Name	3792	LTL D1 V1 23	Truck	3793	LTL D1 V2 23	Truck	3794	LTL D1 V3 23	Truck	3795	LTL D1 V4 23	Truck	3796	LTL D1 V5 23	Truck	3797	LTL D1 V6 23	Truck	3798	LTL D1 V7 23	Truck	3799	LTL D1 V8 23	Truck	<table><tr><td>23-JAN-2007 07:00</td><td>23-JAN-2007 07:00</td><td>23-JAN-2007 07:00</td></tr><tr><td>23-JAN-2007 07:00</td><td>23-JAN-2007 07:00</td><td>23-JAN-2007 07:00</td></tr><tr><td>23-JAN-2007 07:00</td><td>23-JAN-2007 07:00</td><td>23-JAN-2007 07:00</td></tr><tr><td>23-JAN-2007 07:00</td><td>23-JAN-2007 07:00</td><td>23-JAN-2007 07:00</td></tr><tr><td>23-JAN-2007 07:00</td><td>23-JAN-2007 07:00</td><td>23-JAN-2007 07:00</td></tr><tr><td>23-JAN-2007 07:00</td><td>23-JAN-2007 07:00</td><td>23-JAN-2007 07:00</td></tr><tr><td>23-JAN-2007 07:00</td><td>23-JAN-2007 07:00</td><td>23-JAN-2007 07:00</td></tr><tr><td>23-JAN-2007 07:00</td><td>23-JAN-2007 07:00</td><td>23-JAN-2007 07:00</td></tr></table>				23-JAN-2007 07:00	23-JAN-2007 07:00	23-JAN-2007 07:00	23-JAN-2007 07:00	23-JAN-2007 07:00	23-JAN-2007 07:00	23-JAN-2007 07:00	23-JAN-2007 07:00	23-JAN-2007 07:00	23-JAN-2007 07:00	23-JAN-2007 07:00	23-JAN-2007 07:00	23-JAN-2007 07:00	23-JAN-2007 07:00	23-JAN-2007 07:00	23-JAN-2007 07:00	23-JAN-2007 07:00	23-JAN-2007 07:00	23-JAN-2007 07:00	23-JAN-2007 07:00	23-JAN-2007 07:00	23-JAN-2007 07:00	23-JAN-2007 07:00	23-JAN-2007 07:00
Route ID	Resource Key	Name																																																									
3792	LTL D1 V1 23	Truck																																																									
3793	LTL D1 V2 23	Truck																																																									
3794	LTL D1 V3 23	Truck																																																									
3795	LTL D1 V4 23	Truck																																																									
3796	LTL D1 V5 23	Truck																																																									
3797	LTL D1 V6 23	Truck																																																									
3798	LTL D1 V7 23	Truck																																																									
3799	LTL D1 V8 23	Truck																																																									
23-JAN-2007 07:00	23-JAN-2007 07:00	23-JAN-2007 07:00																																																									
23-JAN-2007 07:00	23-JAN-2007 07:00	23-JAN-2007 07:00																																																									
23-JAN-2007 07:00	23-JAN-2007 07:00	23-JAN-2007 07:00																																																									
23-JAN-2007 07:00	23-JAN-2007 07:00	23-JAN-2007 07:00																																																									
23-JAN-2007 07:00	23-JAN-2007 07:00	23-JAN-2007 07:00																																																									
23-JAN-2007 07:00	23-JAN-2007 07:00	23-JAN-2007 07:00																																																									
23-JAN-2007 07:00	23-JAN-2007 07:00	23-JAN-2007 07:00																																																									
23-JAN-2007 07:00	23-JAN-2007 07:00	23-JAN-2007 07:00																																																									

OR,

- 1 Select **User Profile > Profile Items**. The **Profile Items** list page appears.

Profile Item Management / Profile Items	
Profile Items	
Key	Value(s)
FW.OrgSettings	Org:MapSource=Google;MapServerURL=http://maps.google.com/maps/api/js?sensor^1fa
Psw - Minimum Length	
Psw - Blank Allowed	
Psw - Num Char Types Required	
Psw - Change Every x Days	
Psw - Num Prior Passwords Kept	
Psw - FailedLoginsBeforeLockout	
TC_routesTree_LAYOUT	Org:PE4gSz0iQExCum91dGVzlj48QyBXPSI4MCI@QExCum91dGVJRDwvQz48QyBXPSI5
TC_alertsTree_LAYOUT	
TC_unassignedStopsTree_LAYOUT	
TC_favoriteSchedulesTree_LAYOUT	
TC_routeDetail1Tree_LAYOUT	
TC_routeDetail2Tree_LAYOUT	
TC_routeDetail1GSTree_LAYOUT	
TC_routeDetail2GSTree_LAYOUT	
TC_alertManagementTree_LAYOUT	
TC_filtersTree_LAYOUT	
TC_locationListTree_LAYOUT	
TC_resourceListTree_LAYOUT	

- 2 Right-click on the 'TC_XXX_LAYOUT' profile item and select **Add Org Item** or **Edit Org Item** from the right-click menu. The system then copies all of the existing layouts to an org default.

App Setup / Call Out Notification Profile / Edit Call Out Notification Profile

Edit Call Out Notification Profile

SAVE SAVE AND CLOSE CLOSE

Call Out Notification Profile

Key: ZSV_notif_profile

Name: ZSV_notif_profile

Notify Events:

☒ En Route	☐ Arrived
☒ Completed	☐ Delayed
☐ Skipped	☐ No Freight
☐ Missed	☐ Returned
☐ Hooked	☐ Spotted

Email Notifications

Use Notification Ind: ☒

Notify Minutes Before Arrive: 10 Minutes

Notify Minutes After Complete: 888 Minutes

Other Email to Notify (seperated with ";"): zs@d.com;a@gmail.com

Callouts

Use Call Out Ind: ☒

Call Minutes Before Arrive: 5 Minutes

Call Minutes After Complete: 5 Minutes

Preferred Language: Polish

Use Location Phone field: FWLocation.MobilePhoneNumber

SMS Notifications

Use SMS Notification Ind: ☒

SMS Notify Minutes Before Arrive: 8 Minutes

SMS Notify Minutes After Complete: 8 Minutes

Use SMS Location Field: Phone

Configure the following fields after specifying Key, Name and status events for the profile.

- Callouts:
 - Select **Use Call Out Ind** to enable callouts
 - Call X minutes before arrive
 - Call X minutes after completion
 - Preferred language
 - Select the field containing the phone number you wish to use for callouts from the **Use Location Phone** drop-down menu
- Email Notifications
 - Select **Use Notification Ind** to enable email notifications
 - Send email X minutes before arrive
 - Send email X minutes after completion
 - Enter any additional email addresses in the Other Email to Notify field
- SMS Notifications:
 - Select **Use SMS Notification Ind** to enable text message notifications

- Send email X minutes before arrive
- Send email X minutes after completion
- Select the field containing the phone number you wish to use for SMS notifications from the **SMS Location Field** drop-down menu

Profile definitions are sent at the route/stop level in the DocRoute Detail. Routes contain organization or schedule callout/notification profiles. Stops contain the location callout/notification profile. wGLN reads and applies the notification or callout profile settings.

To enable this functionality globally, select **App Setup > Application Configuration**.

Call Out/Notify Settings

Use Call Out/Notification: ☐

Call Out/Notification Profile: ▼

Select the **Use Call Out/Notification** checkbox and enter the relevant profile number to enable the functionality.

Callout/notification settings can be configured granularly by location or schedule as well. From the **Edit Location** or **Edit Schedule** page, users can set a Callout/Notification Profile and set whether or not the particular location or schedule uses global settings.

UDFDateTime3: ⚙️

Times

Use Call Out/Notification: ▼

Call Out/Notification Profile: ▼

Use Call Out/Notification: ▼

Call Out/Notification Profile: ▼

In the **Use Call Out/Notification** drop-down menu, select one of three options for the location or schedule:

- **Use global settings:** Applies the global settings established on the **Application Configuration** page
- **Use location/schedule settings:** Applies the specific profile entered on the **Edit Location** or **Edit Schedule** page
- **Do not call:** Excludes the location or schedule from callouts and notifications

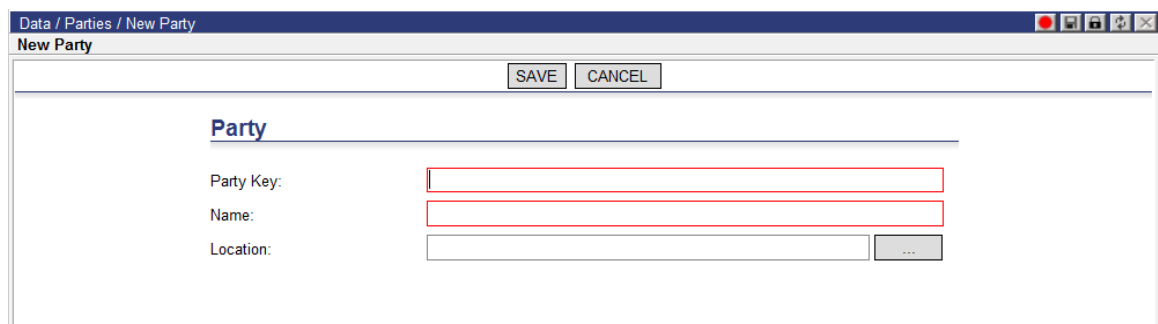
Callout/notification information is displayed in columns on the **Locations** and **Schedules** list pages:

- Callout/Notify Profile Key
- Callout/Notify Profile Name
- Use Callout/Notify Profile (yes/no)

Creating and Managing Party Users

Descartes Route Planner supports party users. The ability to create party users allows administrators to limit access to certain routes and orders within a schedule for particular users.

To create a party, select **Data > Parties** from the main menu. Right-click in the **Parties** list page and select **New Party** from the right-click menu.



Enter the Party Key and Name of the party and then select a Location using the ellipse button. Click **Save** when finished.

To assign users to the party, from the **Parties** list, right-click on the desired party and select **Assign Users**. The **Assign Users** page appears, with functionality similar to the **Crews** page.

[illegible]

Use the **ADDALL**, **ADD**, **REMOVE** and **REMOVEALL** buttons to create a list of assigned users in the Selected User table from the users in the Available User table. The tables can be filtered for specific criteria using the Find field. Click **Save** when finished.

The following items can be assigned to party users:

- Resources
- Orders
- Assets
 - Trailer
 - Trucks
 - Tractors
 - Drivers
- Route Templates

Users can assign and unassign parties in two ways:

- From the edit form (**Edit Resource**, **Edit Driver**, etc.), select a party from the **Party** drop-down menu and save. To unassign a party, select the null option, **Select User...**
- From dashboard quadrants (**Routes**, **Unassigned Stops**, etc.), right-click on a row in the quadrant and select the **Assign Party** option from the right-click menu. The **Select Party** window appears. Double-click the desired party to add it to the route or stop. The assigned party is listed in the Party column in the quadrant.

To unassign a party from a quadrant row, right-click on a row in the quadrant and select the **Unassign Party** option.

➡ **Note**— Orders are assigned to a party on two conditions: (1) When they are assigned to a Resource's party or (2) when an admin user assigns the order to a party.

➡ **Note**— All optimization calls will be filtered using the PartyKey that a user belongs to

Routes can be generated from Route Templates that have a PartyKey or have an assigned Driver/Trailer or Truck assigned to a Party.

Glossary

Administrator – A role definition that has access to all application functionality. The administrator manages the setup and flow of data within Descartes Route Planner.

Bucket – A service time window interval used to offer reservation slots to customers. Buckets are identified by date/time boundaries.

Dispatcher – A role definition that allows execution functionality in the system. Wireless and Status updates are included in this role's functionality.

LNOS Dataflow – Descartes' core functionality for Publish/Subscribe mechanisms. It allows integration of documents across other LNOS-based Descartes applications as well as external applications.

LNOS – Logistics Network Operating System, acronym used for Descartes' standard development architecture.

Order – A movement request of goods from one geographic location to another. An order in Descartes Route Planner can be either a transportation order or a bill of lading, and supports the information included in either type of document.

Planner – A role definition used for planning purposes. Some data editing capabilities are enabled for this role as well as all optimization commands.

Resource – A data definition of a physical unit that can actually perform the work. It is used in the planning and execution of orders. A resource has characteristics such as capacity, costs, availability and restrictions, etc. For example, a truck or a truck/trailer/driver are both resources.

Rmpi – The Descartes Route Planner optimization engine.

Route – Routes are used to link a specific Schedule with a specific resource. All the Resource settings are copied to the Route record and can be modified within a Route so that different settings can be used to build different Routes from the same resource and different Schedules.

➡ **Note**— Routes are created automatically by Descartes Route Planner, based on the Schedule setting provided with the resource users create. Routes cannot exist without Schedules and resources.

Schedule – A group of entities that represent the different aspects of the Schedule settings, Routes, and their Stops. A Schedule is used to communicate the results of an optimization function for export or execution. A Schedule is also a collection of logistics tables that can be combined to create a logistics plan. These tables include Schedules, Routes, Stops, and buckets. Different Schedules let users group these logistics tables, so that users can combine or split routing problems to suit user's business model.

Stop – A physical geographic point that represents a portion or entirety of an order. It holds all the planning information of an order.

Template - Users can set up templates that represent commonly used Schedules, Routes, and Stops in a user's organization, so that users can create Schedules, Routes, and Stops quickly and accurately.

Transportation Order - A movement request of goods from one geographic location to another, specifying date and time, pickup and delivery locations, and quantities.

Appendix A: Database Field Definitions

General Definitions

Generic Attributes

- ➡ **Note**— All duration values are specified in seconds (integer) and negative values are not allowed unless otherwise specified.
- ➡ **Note**— All float values are positive where negative values are not allowed unless otherwise specified.
- ➡ **Note**— All distance values are specified in meters unless otherwise specified.

Attribute Name	Type	Field Size	Description	Required
UniqueID	int	4	This attribute is assigned to a unique numeric value by the system when it is created. Once created it cannot be modified by the user or the system. All uniqueIDs are internal and needed for the LNOS server framework and must be on every table.	Required – system assigned
UniqueBusinessKey	varchar	50	This attribute is assigned to a unique string value by the user when it is created. Once created, it cannot be modified by the user or the system. The Unique Key values are typically provided by the user and are more meaningful to the user. Not all LNOS tables will have a unique Key value because certain child table records are always associated with their parent table record and the parent will always have a unique Key. Every child table record must have a unique ID file regardless.	Required – can be user assigned or system defaulted
ForeignID	int		This attribute is assigned to the TableID value that corresponds to the TableBusinessKey value on the FWTableName. It must be assigned whenever the TableBusinessKey value has been modified. It must be validated against the foreign table for existence. Used to link two tables together.	Required or Optional – system assigned

ForeignBusinessKey	varchar	50	This attribute is a foreign lookup key assigned by the user. It must exist on the foreign FWTableName. Typically a pop-up window is provided for the user to select a valid ForeignBusinessKey from a displayed list of valid keys. As with the foreign ID, the foreign Key is used to link two tables together.	Required or Optional – user assigned
Company	varchar	50	It is used to specify a specific company in a multi-company environment. Currently it is ignored and can be used for display purposes.	Required when multiple companies exist
Comment	varchar	255	It is used to provide additional information about a specific record.	
CreatedBy	int		This is an integer field that is set to the user id of the user who created this record. It is not user modifiable. Use same userid as modifiedby.	Required - system assigned
CreatedDate	datetime	8	This is a date field that should be saved in Universal Metric Time (UMT). It is updated only when the record was first created. It is not user modifiable.	Required – system assigned
ModifiedBy	int		This is an integer field that is set to the user id of the user who created or updated this record. It is not user modifiable.	Required - system assigned
ModifyDate	datetime		This is a date field that should be saved in the local time zone of the server with the following precision: yyyy:mm:ddThh:mm:ss:sss. It is updated every time this record is updated and created. It is not user modifiable.	Required - system assigned

Generic Window Attributes

Attribute Name	Type	Field Size	Description	Required
Type	varchar	50	This attribute is assigned a specific string value by the user. Valid values must be validated against the FWCode table type that is equal to	Required – user assigned

			WindowType. This value should be selectable from a list of valid values (FWCode table). Valid Values: - ARRIVE – Vehicle must arrive at this location within this time window - DEPART – Vehicle must complete servicing this stop within this time window.	
EarliestDate	datetime	8	This attribute is assigned a specific date value by the user. It only holds the date format (yyyy:mm:dd). The time is not used in this attribute. The reason for this is that it represents the starting date range for when this time window will start. If not set, it can start on any date.	Required – user assigned
EarliestTime	datetime	8	It only holds the time format (hh:mm:ss). The date is not used in this attribute. The reason for this is that it represents the starting time range for a specific date specified by the EarliestDate attribute.	Required – user assigned
LatestDate	datetime	8	This attribute is assigned a specific date value by the user. It only holds the date format (yyyy:mm:dd). Other date formats will also need to be supported and handled by the client applications. The time is not used in this attribute. The reason for this is that it represents the ending date range for when this time window will end. Both LatestTime and LatestDate cannot be empty. If the LatestDate is not specified, then the time windows will repeat indefinitely. When specified, it must be greater than or equal to the EarliestDate.	Required – when the LatestTime Attribute is not empty Required – when the frequency field is set to 0
LatestTime	datetime	8	This attribute is required if the LatestDate attribute was entered. It only holds the time format (hh:mm:ss). The date is not used in this attribute. The reason for this is	Required – when the frequency field is set to 0

			that it represents the ending time range for a specific date. This attribute must be greater than the EarliestTime and EarliestDate combination. It is possible to have the time be less than the EarliestTime, but it must be for a future date value.	
Frequency	varchar	50	This attribute is assigned a string value that represents different days of the week. This will allow the user to enter specific days of the week when this window is active. More than one day can be entered but they will need to be delimited by a comma or space. Valid values: • " " or null = Everyday • 1 = Monday • 2 = Tuesday • 3 = Wednesday • 4 = Thursday • 5 = Friday • 6 = Saturday • 7 = Sunday • A blank or null Frequency value means that all the days of the week are valid including Saturday and Sunday. It is the same as specifying "1, 2, 3, 4, 5, 6, 7". • Mixed multiple values represent different days of the week. They should be specified from lowest to highest. Duplicate values are not allowed. Default value is 'empty'.	
ProfitFactor	float		Portion of the order/activity's profit that is gained by serving it within the time window. One or blank gains all the profit. Default value is 0.	
VariationDuration	int		This attribute works in conjunction with the VariationProfit attribute. It is assigned a specific number that represents a duration in seconds. It is	

			used to offset the window "open" and "close" times by this amount without causing violations. Default value is 0.	
VariationProfit	float		Portion of the order/activity's profit that is gained by serving it outside the time window but within the VariationDuration. Range 0.0 to 1.	

Table Definitions

FWLocation

This table is used to specify specific location related information. Typically used to represent a DEPOT or CUSTOMER type location.

Attribute Name	Type	Field Size	Description	Required
LocationID	int	4	Unique internal DB Identifier of the Location record.	Required
LocationKey	nvarchar	50	Business key to identify the activity this Location refers to.	Required
TimeZoneKey	nvarchar	50	Select from a list of valid codes on the FWTimeZone.	
ModifiedBy	int	4	See <u>ModifiedBy</u> under <u>Generic Attributes</u> .	
ModifyDate	datetime	8	See <u>ModifiedDate</u> under <u>Generic Attributes</u> .	
CreatedBy	int	4	See <u>CreatedBy</u> under <u>Generic Attributes</u> .	
CreatedDate	datetime	8	See <u>CreatedDate</u> under <u>Generic Attributes</u> .	
Comment	nvarchar	500	See <u>Comment</u> under <u>Generic Attributes</u> .	
Company	nvarchar	50	This field is used to specify the company name for the location.	
Type	nvarchar	50	This attribute is assigned a specific value by the user when it is created.	Required

			Valid values must be validated against the FWCode table of type LocationType. Valid values: (default = Customer) - CUSTOMER = both customer pickup and dropoff - DEPOT = both depot pickup and dropoff - VENDOR = currently not supported - similar to customer but may require different processing - GPS = currently not supported - but can be used for tracking the execution of the route between stops	
Status	nvarchar	50	This attribute is assigned a specific value by the user when it is created. Valid values must be validated against the FWCode table of type LocationStatus. Valid values: (default = Temp) - DELETE = a previous master is now tagged for deletion - MASTER = can persist by itself - TEMPORARY = can only exist if there is at least one activity or resource associated with it - PENDING = not fully validated - typically used to identified locations that have not been geocoded, but where code has other missing validations	Required
ContactFirstName	nvarchar	255	This is a string value assigned by the user.	
ContactLastName	nvarchar	255	This is a string value assigned by the user.	
Name	nvarchar	255	Typically used for customer name or depot name. It is required because the User Interface (UI) application references the attribute throughout.	Required
ContactPhone	nvarchar	50	This is a string value assigned by the user. This value should be validated against valid phone formats, both domestic and International.	

ContactMobilePhone	nvarchar	50	Contains the mobile number for a location contact used in the Use Location Phone and Use SMS Location drop-down menus on the Call Out Notification Profile	
Territory	nvarchar	50	It should be validated against the FWCode table of type Territory.	
ShipCity	nvarchar	255	City of shipping address.	
ShipCountry	nvarchar	50	It will default to U.S. but can be modified to any country value during implementation.	Required
ShipCrossStreet	nvarchar	255	It is typically used in conjunction with the ShipStreetName(1 and 2). It should be empty if the ShipStreetName is empty.	
ShipState	nvarchar	255	State of shipping address. Outside the U.S., province information is stored here; therefore, we cannot restrict this field to U.S. states only.	Required
ShipStreetName	nvarchar	255	The Street name information of the shipping address.	
ShipStreetName2	nvarchar	255	The additional Street name information of the shipping address. This field must be empty if ShipStreetName is empty.	
ShipDistrict	nvarchar	255	Outside the U.S., District information is stored here so we cannot restrict this field to the U.S. states only.	
ShipStreetNumber	nvarchar	255	Street number of shipping address.	
ShipZipcode	nvarchar	50	Short zip code only. This value should be validated against valid zip and postal code formats domestic and International.	
Latitude	float	8	This is a numeric value provided by an external system or function. It is used as the original external latitude value. It does not mean that the address has been officially geocoded within the LNOS environment. The GeocodeLatitude attribute represents	Required – if the Longitude attribute is specified.

			the actual LNOS geocoded value. Negative values are allowed. Note— If the GeocodeLatitude is blank and this value changes the Activity status needs to be changed to "modified."	
Longitude	float	8	This is a numeric value provided by an external system or function. It is used as the longitude original value. In certain applications the user is able to modify the value. Negative values are allowed. Note— If the GeocodeLongitude is blank, and this value is changed, then the Activity status needs to be changed to "modified."	Required – if the Latitude attribute is specified.
GeocodeLatitude	float	8	This is a numeric value provided by an external system or function. It is used for the calculated (LNOS) value and cannot be modified by the user. Note— If this value changes, the Activity status needs to be changed to "modified."	Required – if GeocodeLongitude is specified.
GeocodeLongitude	float	8	This is a numeric value provided by an external system or function. It is used for the calculated (LNOS) value and cannot be modified by the user. Note— If this value changes, the Activity status needs to be changed to "modified."	Required – if GeocodeLatitude is specified.
GeocodeOneCell	int	4	This value is an internal data element that will not be modified or displayed to the user. It is a future requirement and used to help determine the side of street.	Required if RelativePosition is specified.
GeocodeRelativePosition	float	8	This value is an internal data element that will not be modified or displayed to the user. It is a future requirement and used to help determine the side of street.	Required if OneCell is specified
GeocodeQuality	smallint	2	This is a hex value provided by an external system or function. It is not user modifiable. Whenever this value	

is presented to the user it will need to be transformed as follows:

- 0x0001 = WRONG_PARITY -The containing address range has the wrong parity.
- 0x0002 = RANGE_ROUNDED -A range with numeric and/or character suffixes had to have its low street number rounded down or its high street number rounded up in order to contain the given street number.
- 0x0004 = WRONG_STREET_NUM_HYPHENATION -Had to insert or delete a dash from the street number to find a containing range.
- 0x0008 = WRONG_CITY - City misspelled.
- 0x0010 = WRONG_XSTREET - Bad suffix or direction in cross street name, base name okay unless WRONG_XSTREET_BASE also set.
- 0x0020 = WRONG_STREET - Bad suffix or direction in main street name, base name okay unless WRONG_STREET_BASE is also set.
- 0x0040 = NO_CONTAINING_RANGE - Street number not contained in range, closest range used.
- 0x0080 = WRONG_XSTREET_BASE - Base name of cross street misspelled.
- 0x0100 = WRONG_STREET_BASE - Base name of main street misspelled.
- 0x0200 = WRONG_ZIP - Improper Zip/postal code
- 0x0400 = WRONG_STATE - State misspelled.

Note— The left column provides the bit mask, represented as a hexadecimal value.
The GeocodeFlags field contains the text strings that match the code in GeocodeQuality.

GeocodeScore	smallint	2	This is a string value provided by an external system or function. It is not user modifiable. A higher value indicates a better resolution. It is used in several UI screens and as a potential input value to other external functions.
GeocodeSource	nvarchar	255	This is a string value provided by an external system or function. It is not user modifiable. Valid values: • GeoStan (an external map provider) • *.lmb = the name of a map database file • *.lzb = the name of a zip code geocoding database file.
GeocodeType	nvarchar	50	This is a numeric value provided by an external system or function. It is not user modifiable. Valid values: • 0 = Not Resolved • 1 = Resolved By City Name • 2 = Resolved By Zip Code • 3 = Resolved By Zip Plus2 • 4 = Resolved By Zip Plus4 • 5 = Resolved As Point • 6 = Resolved Regularly
GeocodeFilter	smallint	2	This is a string value used to interface with RMPI. It is provided by the user and is modifiable. It is not used by the new Geocode service but is used simply to control the behavior of the RMPI geocode functionality. Valid values: (default = 0) • 0 = No filtering. Any resolution is acceptable. • 1 = Resolutions with a wrong zip code and one other wrong item are rejected. If the zip code is valid, or if everything but the zip code is valid, the resolution is accepted.

			• 2 = Resolutions with a wrong zip code are rejected. If the zip code is valid, the resolution is accepted. Note— Settings other than zero can be useful when geocoding on multiple maps. See FileX setting in the [Map] sections of your configuration files.	
GeocodePolicy	tinyint	1	This is a string value used to interface with RMPI. It is provided by the user and is modifiable. It is not used by the new Geocode service but is simply used to control the behavior of the RMPI geocode functionality. Valid values: • -1 = Never use Latitude/Longitude. • 0 = Use the Latitude/Longitude unless the address is resolved by the geocoder (not zip/city). • 1 = Use the Latitude/Longitude unless the address resolved perfectly. • 2 = Use the Latitude/Longitude always. • 3 = Use the GeocodeLatitude/GeocodeLongitude fields always.	
GeocodeFlags	nvarchar	255	A text string containing the strings that match the code in GeocodeQuality. The strings are separated by a space. The possible strings are: • HYPHENATION CHANGE - Had to insert or delete a dash from the street number to find a containing range. • NO CONTAINING RANGE - Street number not contained in any range, closest range used. • RANGE ROUNDED - A range with numeric and/or character suffixes had to have its low street number rounded down or its high street number rounded up in order to contain the given street number. • WRONG CITY - City misspelled.	

			• WRONG PARITY - The containing address range has the wrong parity. • WRONG STATE - State misspelled. • WRONG STREET - Bad suffix or direction in main street name, base name OK unless public static final integer RFLAG_WRONGSTREETBASE also set. • WRONG STREET BASE - Base name of main street misspelled. • WRONG XSTREET - Bad suffix or direction in cross street name, base name OK unless public static final integer RFLAG_WRONGXSTREETBASE also set. • WRONG XSTREET BASE - Base name of cross street misspelled. • WRONG ZIP - The resolved zip code has changed.	
Requirements	nvarchar	255	This is a string value used in conjunction with the requirements attribute of the <i>FWResource/FWRoute</i>. When specified, the stop associated with this location will only be scheduled on a route that also has the same requirements specified. If no route has this requirement specified then this stop will not be scheduled. Multiple values can be specified (delimited by a space). Each separate value must be validated against the FWCode table of type Requirements. These values are linked together with other requirements on the corresponding activity record. If the location is a Depot and Recharge is turned on then a resource must have all the requirements of a Depot in order to recharge there.	
Commodities	nvarchar	255	This is a string value used in conjunction with other Commodity values. For a Depot these are the Commodities that it supports. For Customers these are the values of the	

			commodities for this location and will be combined with an Order's commodities to generate the actual set for an Order. An empty value implies that all commodities are possible. Multiple values can be specified (delimited by a space). Each separate value must be validated against the FWCode table of type Commodities.	
Profit	float	8	This is a numeric value that represents a base profit for this location (customer). It is added to the other profit values to represent a total profit on the activity record. It is different in that it is cumulative instead of linked together. Defaults to 0.	
ServiceDuration	int	4	This is an integer value that represents the average service time in seconds applied at this location. It is added to the ServiceDuration of an order/activity. It is only applied once for multiple orders at the same location. Defaults to 0.	
PreServiceDuration	int	4	This is an integer value that represents the average pre-service time such as parking - in seconds that is applied at this location. It can be taken before the time window of an order/activity starts. Defaults to 0.	
RechargeDuration	int	4	This is an integer value that represents the average recharge service time in seconds that is applied at this location. It is typically cumulative. Defaults to 0.	
Recharge	bit	1	For Depots only. If this value is set to true, then the Depot can recharge the Measures that are set to 1. If no measures are set to 1, then all measures can be recharged by this Depot. Defaults to 0.	

RechargeRate	float	8	Specifies how quickly a Measure can be recharged. If this is set to 10, then 10 units of a measure can be transferred per second.	
EarliestRecharge	datetime	8	For Depots specifies the earliest time of the day that a Recharge can take place.	
LatestRecharge	datetime	8	For Depots specifies the latest time of the day that a Recharge can take place.	
Measures1-9	float	8	For Depot locations the Measure fields are used to represent which measure amount (1 thru 9) is supplied by the depot. This field is set to 0 (off) or 1 (on). If none are set, then all measures can be supplied. Defaults to empty.	
Measure1-9RechargeRate	float	8	It is used to represent which measure amounts (1 thru 9) can be recharged and at what rate. This is typically used for depot type locations. The occurrence must match the occurrence specified by the MeasuresSupplied values. If Measure1RechargeRate=10, then 10 units of Measure1 can be transferred per second. Default is 0.	
UDFString1-9	nvarchar	500	User Defined Fields for Strings 1...9	
UDFInteger1-3	int	4	User Defined Fields for integers 1..3	
UDFNumber1-3	float	8	User Defined fields for numbers 1...3	
ServiceFactor	float	8	Factor used to multiply the ServiceDuration of a Measure x. Values greater than one (1) specify a slower Service time. Values less than one (1) specify faster service times.	
RequirementSetID	int	4	The Requirement Set ID taken from the FWRequirementSet table.	
LockLatLong	bit	1	This field is used to lock a longitude and latitude of a location. Values are: 0 = Off 1 = On Default is 0.	Required – unless it is specified explicitly

AutoCorrectGeocodeInd	bit	4	This field will be set to 1 when the Autocorrection on Geocoding information process overwrote the original longitude/latitude of this record. Default value: <null>	
RechargeSetupPenalty	float	8	One time cost that a resource incurs whenever it reloads or unloads using this location.	
RechargePenaltyMeasure1...9			Penalty per unit incurred at this location when a resource loads Measure "x".	
AllowEmptyStart	bit	1	Enables Resources to start empty and allows them to load at their home or remote location. To activate this functionality on a schedule, the AllowEmptyStart property must be set to "1".	
StartupTime	nvarchar		Allows RMPI to add a startup time at the beginning of a route. This time can be specified by resource and/or by location. If both exist, RMPI adds them together.	
WindupTime	nvarchar		Allows RMPI to add a shut down time at the end of a route. This time can be specified by resource and/or by location. If both exist, RMPI adds them together.	
AutoCorrectLatitude	float		- If the reported latitude in a status message (see AVLParms) is to be considered for updating a customer, update FWLocation.GeocodeLatitude and the AutoCorrectLatitude with last reported status GPS - If FWLocation.GeocodeScore <= MaxScoreToAutoCorrect and the latitude is not locked, then update FWLocation.GeocodeLatitude and	

			the AutoCorrectLatitude with last reported status GPS Otherwise just save latitude reported in status GPS in the AutoCorrectLatitude	
AutoCorrectLongitude	float		- If the reported longitude in a status message (see AVLParms) is to be considered for updating a customer, then update GeocodeLongitude and the AutoCorrectLongitude with last reported status GPS - If GeocodeScore <= MaxScoreToAutoCorrect and the longitude is not locked, then update GeocodeLongitude and the AutoCorrectLongitude with last reported status GPS - Otherwise save reported longitude GPS in AutoCorrectLongitude	
UDFDateTime1-3	nvarchar		User-defined date/time fields	
UseCallOutNotifyInd	int		This field will activate or deactivate Callouts/Notifications for this org 0: use the higher level setting 1: use the local setting 2: do not call	
CallOutNotifyProfileId	int		This field will identify the primary key of the Callout/Notification Profile to use	
MaxDockDoors			Specifies the maximum number of dock doors to display in the Dock Door View quadrant.	
NrOfLoadingBatches			Specifies the number of loading batches to display in the Dock Door View quadrant.	
RemoveChildWhenEmpty	int		0: The system will not remove time windows from existing locations when none are specified on the import file 1: The system will remove time windows from existing locations when none are specified on the import file	

FWLocationWindow

 **Note**— See Generic Window Attributes for details.

This table is used to specify specific time windows for a given location. It is optional in that it does not need to exist when an FWLocation table is created.

Create: Must be associated with a location record

Delete: Must be deleted whenever the associated location record is deleted.

Update: individual record only

Attribute Name	Type	Field Size	Description	Required
LocationWindowID	int	4	Unique internal DB Identifier of the Location window record.	Required
LocationID	int	4	Unique internal DB Identifier of the Location record. Must exist on the FWLocation table.	Required
CreatedBy	int	4	See <i>CreatedBy</i> under <i>Generic Attributes</i> .	
CreatedDate	datetime	8	See <i>CreatedDate</i> under <i>Generic Attributes</i> .	
Type	nvarchar	50	See <i>Type</i> under <i>Generic Window Attributes</i> .	Required
EarliestTime	datetime	8	Beginning time for time window	
LatestTime	datetime	8	Ending time for time window	
EarliestDate	datetime	8	Beginning date for time window X. Values of "ScheduleDate + X and 'Today + X' can be used to create relative dates.	
LatestDate	datetime	8	Ending date for time window X. Values of 'ScheduleDate + X' and 'Today + X' can be used to create relative dates.	
Frequency	nvarchar	255	Designates which weekdays or dates the item is available to be scheduled. For weekdays, you can use the standard abbreviations, full names, or numbers 1 through 7 corresponding to days Monday through Sunday. Dates may be	

			written as M/D/Y, M-D-Y, MMDDYY, or MMDDYYYY. For example, a setting of "135" schedules the item for Monday, Wednesday and Friday. A setting of "Mon Wed Fri 6/30/1999 7/1/1999" schedules the item for Monday, Wednesday, and Friday and for 6/30/1999 and 7/1/1999.	
ProfitFactor	float	8	Portion of the job or task's profit gained by serving it within the time window.	
VariationDuration	int	4	A "fudge factor" for time windows. Extends every time window's open and close times by the specified duration.	
VariationProfit	float	8	Portion of the task's profit gained when it is served outside the defined time window, but within the allowed variation. Encourages Optimizer to assign jobs inside the variation if the time window cannot be achieved. For example, a value of 0.75 gives a job served within the variation 75% of its normal profit.	
ModifiedBy	int	4	See <i>ModifiedBy</i> under <i>Generic Attributes</i> .	
ModifyDate	datetime	8	See <i>ModifiedDate</i> under <i>Generic Attributes</i> .	
Comment	nvarchar	500	See <i>Comment</i> under <i>Generic Attributes</i> .	
OpenDateTime	datetime	8	Opening Date/Time of the Time Window	
CloseDateTime	datetime	8	Closing Date/Time of the time window	

FWLocContInvTrack

The FWLocContInvTrack table contains fields specifying the relationship between the location and the ProductTypeKey inventory.

Field	Type	Field Size	Description
LocContInvTrackID	int	4	Defines the primary key for this table

LocationID	int	4	Defines the Location ID for the current LocContInvTrack record
ProductTypeID	int	4	Defines the ProductType ID for the current LocContInvTrack record
LastReportedInventory	float		The Last Reported and current inventory for this product type key at this location. This value may be a fraction.
LastReportedDate	datetime	8	This field reports that Last Reported Date of the Ending Inventory

FWOrder

This table is used to identify order header type information that describes the order as a whole.

Attribute Name	Type	Field Size	Description	Required
OrderID	int	4	Unique internal DB Identifier of the Order record.	Required
OrderKey	nvarchar	50	Business key to identify the activity this Order refers to.	Required
StopTemplateKey	nvarchar	50	Business key to identify the Stop template this Order refers to. Must exist on the FWStop table.	
ScheduleKey	nvarchar	50	Business key to identify the Schedule this Order refers to. If a value is specified, it must exist on the FWSchedule table. An order with a bad schedule key is rejected. Default is DefaultSchedule.	
CustomerOrder	nvarchar	50	This is a string value representing an external customer order number. The reason it is optional is because most implementations will use the OrderKey value as the unique external customer order number. The reason for another customer order attribute is to support temporary orders or order splitting in the future. This field will provides flexibility and a point of reference back to the original	

			customer order. It can also be used if for some reason the original customer order is not unique and we need to track it as reference.	
ModifiedBy	int	4	See <u>ModifiedBy</u> under <u>Generic Attributes</u> .	
ModifyDate	datetime	8	See <u>ModifiedDate</u> under <u>Generic Attributes</u> .	
CreatedBy	int	4	See <u>CreatedBy</u> under <u>Generic Attributes</u> .	
CreatedDate	datetime	8	See <u>CreatedDate</u> under <u>Generic Attributes</u> .	
Comment	nvarchar	500	See <u>Comment</u> under <u>Generic Attributes</u> .	
Company	nvarchar	50	This field is used to specify a specific company name in a multi-company environment.	
Name	nvarchar	255	Used for the customer name. It is required because the UI application references this attribute throughout. Typically it is the same as the Location Name, but it can be different if the order was placed by one company and shipped to another.	
Type	nvarchar	50	This field is used to identify a different order type that may require a different process. For example: we can have pickup, drop-off, or double-ended orders which may require different FWActivity records created. Valid Values: • DROPOFF – no origin FWActivity record needs to be created • PICKUP – no destination activity needs to be created • PICKUP/DROPOFF – both a pickup and destination FWActivity records will need to be created • RECHARGE – Reload type job • COMPLEX – multi-leg order	Required

PreferredResourceKey	nvarchar	50	This value is used to specify that a specific resource be used whenever possible. This attribute is used in conjunction with the PreferredResourceInd attribute, which is used to force this job on the specific resource.	
PreferredResourceInd	bit	1	This value is used to specify when the optimizer must consider the Preferred Resource during the assignment process.	
Profit	float	8	This attribute holds the additional profit at the order level that is accumulated with the FWOrderLines, Stop Template and Location table profits whenever the FWActivity record gets created.	
Requirements	nvarchar	255	This attribute holds the additional requirements at the order level that is linked with the FWOrderLines requirements, Stop Template, and Location tables whenever the FWActivity record gets created.	
Commodities	nvarchar	255	This attribute holds the additional requirements at the order line level that is linked with the commodities from the Order lines, Stop Template, and Location tables whenever an activity is created.	
ServiceDuration	int	4	This is an integer value that represents the average service time in seconds for this order.	
PreServiceDuration	int	4	This attribute is used mostly to model parking time at each stop, but users can include other activity times spent before performing customer's jobs or using a depot. When it's used to model parking time, it differentiates between situations with easy parking and difficult parking. For example, parking at a downtown office building usually takes longer than parking in an office park.	
PUServiceDuration	int	4	Used in Double ended orders. This field indicates the ServiceDuration at	

			the Pickup location. This number will be added to the total ServiceDuration of the FWActivity ServiceDuration field.	
PUPreServiceDuration	int	4	Used in Double ended orders. This field indicates the PreServiceDuration at the Pickup location. This number will be added to the total PreServiceDuration of the FWActivity PreServiceDuration field.	
DeliveryServiceDuration	float	8	Used in Double ended orders. This field indicates the ServiceDuration at the Delivery location. This number will be added to the total ServiceDuration of the FWActivity ServiceDuration field.	
DeliveryPreServiceDuration	float	8	Used in Double ended orders. This field indicates the PreServiceDuration at the Delivery location. This number will be added to the total PreServiceDuration of the FWActivity PreServiceDuration field.	
DeliveryPostServiceDuration	float	8	Used in Double ended orders. This field indicates the PostServiceDuration at the Delivery location. This number will be added to the total PostServiceDuration of the FWActivity PostServiceDuration field.	
Status	nvarchar	255	Execution Status	
RestrictPreferredResource	bit	1	Designates whether or not this order should be restricted to a PreferredResource Key value when optimizing or planning.	
MasterRoute	bit	1	Designates if this Order is a Master Route Order or not.	
OrginalOrderKey	nvarchar	50	Designates the original order key that was used to generate this Order record (when splitting or creating itinearys or rescheduling this order)	
ScheduleID	int	4	Designates the Schedule ID to which this order belongs to.	

RequirementSetID	int	4	The Requirement Set ID from FWRequirementSet table.	
UDFString1-9	nvarchar	500	User Defined Fields for Strings 1...9	
UDFInteger1-3	int	4	User Defined Fields for integers 1..3	
UDFNumber1-3	float	8	User Defined fields for numbers 1...3	
ItineraryOrderType	int	4	Designates if this order is an itinerary Order type	
RescheduledInd	bit	1	Designates if this order has been rescheduled	
UnplannedInd	bit	4	Index to identify Unplanned Orders	
UDFDateTime1-3	nvarchar		User-defined date/time fields	
UDFURL1-3	nvarchar	500	User-defined URL fields	
ExtDocControlID	nvarchar	50	User's business document ID	
OrderState	Int	4	1 for order from order split, 2 from cloned and 3 from rescheduled	
OrigPreferredResourceKey	nvarchar	50	Stores the original value of preferredResourceKey. In the unlocking of a route event, this value is used to reset PreferredResourceKey	
OrigPreferredResourceInd	bit	1	Stores the original value of PreferredResourceInd. In the unlocking of a route event, this value is used to reset PreferredResourceInd	
OrigRestrictPreferredResource	bit	1	Stores the original value of RestrictPreferredResource. In the unlocking of a route event, this value is used to reset RestrictPreferredResource	
Version	nvarchar	64	Order version number. Route Planner will not update an order if the version is the same or older by string comparison	
BOLNo	nvarchar	128	BOL Number	
TMLoadNo	nvarchar	64	Load number from Descartes Transpotarion management Module. It is used as indicator that the order is tendered from TM	

TMBestContractName	nvarchar	128	The contract name of best rate	
TMBestCarrierCode	nvarchar	16	The carrier code of best rate	
TMBestTotalCost	float	8	The total cost of best contract	
TMSelContractName	nvarchar	128	The contract name of selected rate	
TMSelCarrierCode	nvarchar	16	The carrier code of selected rate	
TMSelTotalCost	float	8	The total cost of selected contract	
FreightAllowance			Represents the least expensive value between carrier and vendor costs.	

FWOrderLine

This table is used to represent the different product related information. It is a child of the order table which cannot exist on its own.

Attribute Name	Type	Field Size	Description	Required
OrderLineID	int	4	Unique internal DB Identifier of the Order Line record.	Required
OrderID	int	4	Unique internal DB Identifier of the Order record. Must exist in the FWOrder table.	Required
ProductKey	nvarchar	50	This value is used to reference the Product business key from the external Order Entry TAD system.	Required
ModifiedBy	int	4	See <i>ModifiedBy</i> under <i>Generic Attributes</i> .	
ModifyDate	datetime	8	See <i>ModifiedDate</i> under <i>Generic Attributes</i> .	
CreatedBy	int	4	See <i>CreatedBy</i> under <i>Generic Attributes</i> .	
CreatedDate	datetime	8	See <i>CreatedDate</i> under <i>Generic Attributes</i> .	
Comment	nvarchar	500	See <i>Comment</i> under <i>Generic Attributes</i> .	
ProductName	nvarchar	255	This value is used to describe the product being moved. It is	

			needed for online display information.	
VendorName	nvarchar	255	This value is used to describe the product vendor being moved. It is needed for online display information.	
Quantity	float	8	This value cannot be equal to or less than zero.	Required
ServiceDuration	int	4	This numeric value in seconds is added to the other service values from the FWProduct, FWOrder, FWStopTemplate and FWLocation tables. See the StopTemplate table for more details.	
PreServiceDuration	int	4	This numeric value in seconds is added to the other pre-service values from the FWOrder and FWStopTemplate and FWLocation. See the StopTemplate for more details.	
Profit	float	8	This numeric value is added to the other profit values from the FWOrder and FWStopTemplate and FWLocation.	
Requirements	nvarchar	255	This is a string value that is linked with the requirements for the FWOrder, FWStopTemplate, and FWLocation. Each value must exist on the FWCode table for type Requirements. A selection list should be provided where one or more requirements can be selected. See the StopTemplate for more details.	
Commodities	nvarchar	255	This attribute holds the additional commodities at the order line level that is linked with the commodities from the Order, Stop Template, and Location tables whenever an activity is created. Each value must exist on the FWCode table for type Commodities. A	

			selection list should be provided where one or more commodities can be selected. See the StopTemplate for more details.	
Measure1-9	float	8	This numeric value is added to the other Measure1-9 amount values supplied with the activity input function.	Measure1 is Required
Measure1-9 Factor	float	8	This numeric value is used to adjust the actual corresponding measure amount. It is applied when the FWActivity Measures are created, but only for the OrderLine Measure portion.	
ServiceRate	float	8	Rate used to calculate the service duration to service Measure x units. Specified in units/second.	
ActivityID	int	4	Unique Identifier of the FWActivity record.	
ActivityKey	nvarchar	50	Business key to identify the activity this Order Line refers to. Must exist in the FWActivity table.	
UDFString1-9	nvarchar	500	User Defined Fields for Strings 1...9	
UDFInteger1-3	int	4	User Defined Fields for integers 1..3	
UDFNumber1-3	float	8	User Defined fields for numbers 1...3	
OriginalMeasure1-9	float	8	Original reading reported for Measure 1...9	
ApplyToMeasure	tinyint	1	Specifies the Measure number to which the specified ServiceRate should be applied.	
ProductTypeId	int	4	Unique Identifier of the FWProductType record.	
ProductTypeKey	nvarchar	50	Business key to identify the product type this Order Line	

			refers to. Must exist in the FWProductType table.	
PUServiceDuration	int	4	Used in Double ended orders. This field indicates the ServiceDuration at the Pickup location. This number will be added to the total ServiceDuration of the FWActivity ServiceDuration field.	
PUPreServiceDuration	int	4	Used in Double ended orders. This field indicates the PreServiceDuration at the Pickup location. This number will be added to the total PreServiceDuration of the FWActivity PreServiceDuration field.	
DeliveryServiceDuration	float	8	Used in Double ended orders. This field indicates the ServiceDuration at the Delivery location. This number will be added to the total ServiceDuration of the FWActivity ServiceDuration field.	
DeliveryPreServiceDuration	float	8	Used in Double ended orders. This field indicates the PreServiceDuration at the Delivery location. This number will be added to the total PreServiceDuration of the FWActivity PreServiceDuration field.	
RequirementSetID	int	4	The Requirement Set ID taken from the FWRequirementSet table.	
ActualMeasure1-9	float	8	Actual reading reported for Measure 1...9	
ProductTypeKey	nvarchar		ProductTypeKey for this order line. If not present, then ActualMeasures must be provided or will be assumed 0.	

ActualQuantity	float		Required if ProductTypeKey is used.	
UDFString1...9, UDFInteger1...9, UDFNumber1...9	Int, nvarchar, float		User defined fields at the Order Line level.	
ActualMeasure1-9	float		Captures the actual measures for the Order Line.	
Status			Captures UpdateStatus message when posted. Value can be any string.	
UDFDateTime1-3			User-defined date/time fields	
Category	nvarchar	100	Captures category for order lines	
ProdCategory	nvarchar	100	Captures product category for order lines	

FWOLineDetail

When a user wishes to use crates, FWOLineDetail table represents current OrderLines. Then the crates related information is represented by FWOrderLine table.

Attribute Name	Type	Field Size	Description	Required
OLineDetailID	int	4	Unique internal DB Identifier of the Crate Line record.	Required
OLineDetailKey	nvarchar	50	Business key to identify OLineDetail.	Required
OrderLineId	int	4	Unique internal DB Identifier of the OrderLine record. Must exist in the FWOrderLine table.	Required
ProductKey	nvarchar	50	This value is used to reference the Product business key from the external Order Entry TAD system.	
ProductName	nvarchar	255	This value is used to describe the product being moved. It is needed for online display information.	
ProductTypeId	int	4	Unique Identifier of the FWProductType record.	
ProductTypeKey	nvarchar	50	Business key to identify the product type this Order Line refers to. Must exist in the FWProductType table.	
ModifyBy	int	4	See <i>ModifiedBy</i> under <i>Generic Attributes</i> .	
ModifyDate	datetime	8	See <i>ModifiedDate</i> under <i>Generic Attributes</i> .	
CreatedBy	int	4	See <i>CreatedBy</i> under <i>Generic Attributes</i> .	

CreatedDate	datetime	8	See <i>CreatedDate</i> under <i>Generic Attributes</i> .	
Comment	nvarchar	500	See <i>Comment</i> under <i>Generic Attributes</i> .	
Quantity	float	8	This value cannot be equal to or less than zero.	Required
Measure1-9	float	8	This numeric value is added to the other Measure1-9 amount values supplied with the activity input function.	Measure1 is Required
Measure1-9Factor	float	8	This numeric value is used to adjust the actual corresponding measure amount. It is applied when the FWActivity Measures are created, but only for the OrderLine Measure portion.	
OriginalMeasure1-9	float	8	Original reading reported for Measure 1...9	
ApplyToMeasure	tinyint	1	Specifies the Measure number to which the specified ServiceRate should be applied.	
ActualMeasure1-9	float	8	Actual reading reported for Measure 1...9	
ActualQuantity	float		Required if ProductTypeKey is used.	
ServiceRate	float	8	Rate used to calculate the service duration to service Measure x units. Specified in units/second.	

ServiceDuration	int	4	This numeric value in seconds is added to the other service values from the FWProduct, FWOrder, FWOrderLine , FWStopTemplate and FWLocation tables. See the StopTemplate table for more details.	
PreServiceDuration	int	4	This numeric value in seconds is added to the other pre-service values from the FWOrder, FWOrderLine , FWStopTemplate and FWLocation. See the StopTemplate for more details.	
Profit	float	8	This numeric value is added to the other profit values from the FWOrder, FWOrderLine , FWStopTemplate and FWLocation.	
Requirements	nvarchar	255	This is a string value that is linked with the requirements for the FWOrder, FWOrderLine , FWStopTemplate, and FWLocation. Each value must exist on the FWCode table for type Requirements. A selection list should be provided where one or more requirements can be selected. See the StopTemplate for more details.	

Commodities	nvarchar	255	This attribute holds the additional commodities at the order line level that is linked with the commodities from the Order, FWOrderLine , Stop Template, and Location tables whenever an activity is created. Each value must exist on the FWCode table for type Commodities. A selection list should be provided where one or more commodities can be selected.	
PUServiceDuration	int	4	Used in Double ended orders. This field indicates the ServiceDuration at the Pickup location. This number will be added to the total PreServiceDuration of the FWActivity ServiceDuration field.	
PUPreServiceDuration	int	4	Used in Double ended orders. This field indicates the PreServiceDuration at the Pickup location. This number will be added to the total PreServiceDuration of the FWActivity PreServiceDuration field.	
DeliveryServiceDuration	float	4	Used in Double ended orders. This field indicates the ServiceDuration at the Delivery location. This number will be added to the total ServiceDuration of the FWActivity ServiceDuration field.	

DeliveryPreServiceDuration	float	8	Used in Double ended orders. This field indicates the PreServiceDuration at the Delivery location. This number will be added to the total PreServiceDuration of the FWActivity PreServiceDuration field.	
RequirementSetID	int	4	The Requirement Set ID taken from the FWRequirementSet table.	
Status			Captures UpdateStatus message when posted. Value can be any string.	

FWOrderInfo

The FWOrderInfo and FWOrderLineInfo stores order and order line information collected by drivers (payment information, for example).

Attribute Name	Type	Field Size	Description	Required
OrderInfoID	int	4	Unique internal DB Identifier of the OrderInfo record.	Required
OrderID	Int	4	Unique internal DB Identifier of the Order record from FWOrder.	Required
Name	nvarchar	250	Name of the property.	Required
Value	nvarchar	250	Value of the property.	Required
Category	nvarchar	50	Category of the property (e.g. Payment)	Required

FWOrderLineInfo

Attribute Name	Type	Field Size	Description	Required
OrderInfoLineID	int	4	Unique internal DB Identifier of the OrderLineInfo record.	Required
OrderLineID	Int	4	Unique internal DB Identifier of the Order Line record from FWOrderLine.	Required
Name	nvarchar	250	Name of the property.	Required
Value	nvarchar	250	Value of the property.	Required
Category	nvarchar	50	Category of the property (e.g. Payment)	Required

FWStopTemplate

The FWStopTemplate table is used to introduce the different stop record routing settings. When a stop template is specified, it provides many of the default settings needed by the scheduling engine. The order import function may have certain attributed already set which will take precedence over the stop template value. This provides a level of flexibility to override certain setting for a specific order without having to create a different stop template record for every possible setting combination. Please keep in mind that certain attributes are cumulative as apposed to overrides values.

Attribute Name	Type	Field Size	Description	Required
StopTemplateID	int	4	Unique internal DB Identifier of the Stop Template record.	Required
StopTemplateKey	nvarchar	50	Business key to identify the activity this Stop template refers to. Defaults to the StopTemplateKey of the FWOrder. If this value is ever changed, the reset function needs to be performed so that the new template values are reflected on this activity.	Required
ScheduleKey	nvarchar	50	Business key to identify the Schedule this Stop template refers to.	

			If entered, it must exist on the FWSchedule table. A selection list should be provided.	
ModifiedBy	int	4	See <u>ModifiedBy</u> under <u>Generic Attributes</u> .	
ModifyDate	datetime	8	See <u>ModifyDate</u> under <u>Generic Attributes</u> .	
Comment	nvarchar	500	See <u>Comment</u> under <u>Generic Attributes</u> .	
CreatedBy	int	4	See <u>CreatedBy</u> under <u>Generic Attributes</u> .	
CreatedDate	datetime	8	See <u>CreatedDate</u> under <u>Generic Attributes</u> .	
DeliveryRatio1-9	float	8	This attribute is used to reclaim capacity. A value of 1.0 will reclaim the full measure amount. A value of 0.5 will reclaim 50% of the amount, and a value of 0.0 will not reclaim any of the amounts. This setting is only applicable if it is a DROPOFF or PICKUP type stop.	
InterspersableInd	bit	1	This Boolean attribute is used to set the Interspersable indicator for the related stop records that came from a DROPOFF/PICKUP order type record. It indicates if other stops can be scheduled between the PICKUP and DROPOFF type stops for that order. It is not applicable to single leg order types (PICKUP or DROPOFF).	
OnLoadPosition	tinyint	1	Use to explicitly specify first or last position on a route. Valid values: 0 = ignored 1 = must be first on load 2 = must be last on load	
PreferredResourceInd	bit	1	This is a Boolean attribute that determines if the preferred route is restricted to the preferred route only. This attribute is dependent on the PreferredResourceKey on the	

			stop table and is ignored when the PreferredResourceKey is empty. Note— The PreferredResourceKey is not defined at the FWStopTemplate level because it is almost always an Activity type attribute and a default setting would never be used.	
Profit	float	8	This is a numeric attribute used to add additional profit to the already supplied profit at the order and location levels. Please note that it is cumulative and not an override value.	
Requirements	nvarchar	255	This value can be used to set a universal requirement that will apply to all orders. Please note that this value is cumulative where it is linked to the order and location requirements. More than one value can be specified and needs to be delimited by a space. Each specific value must be validated against the FWCode table for type Requirements.	
RoutePositionInd	bit	1	This value sets the priority of certain stops positioned before other stops. For example: Stops with RoutePosition set to 1 are sequenced before activities with RoutePosition set higher than 1. A value of zero can be sequenced in any order.	
Type	varchar	50	This field is used to identify different stop template types that may require different processes.	Required
Territory	varchar	50	When entered, it must exist on the FWCode Table Type Territory. It is used as a default value for all stops.	
AllowConsolidationInd	bit	1	This is a Boolean value used to override the AllowConsolidationInd on the Schedule optimization setting for a specific stop. If specified as default, and not overwritten by the	

			order, all orders will have this setting.	
PreServiceDuration	int	4	This integer value (in seconds) is added to the pre-service duration of the order, order lines, and location pre-service duration when the activity is created. See the service duration attribute for more details.	
Status	varchar	50	This field is used to identify a different status of a given stop template over time. The current implementation is configured to support one status value "DEFAULT." As the application develops over time, additional statuses will follow.	Required
Commodities	varchar	255	This attribute is linked with the commodities from the Order, Order Lines, and Location tables. See requirements for details. Each specific value must be validated against the FWCode table for type Commodities.	
ServiceDuration	int	4	This integer value (in seconds) is added to the service duration of the order, order lines, and location values service duration when the activity is created. Note— The ServiceDuration comes with two other attributes—pre-service and post-service duration—providing the ability to specify additional time before and after any service time. These attributes can be used for any wait, setup, or cleanup time needed for a given service. All three settings are independent of each other.	
ProfitDepreciationPerHour	float	8	It is a percentage value that will depreciate the profit by the percentage value for every hour from the start of the route. It makes orders less profitable later in the route. Negative values are allowed	

			and when specified will provide the opposite effect of making later stops more profitable.	
SpecialHandling	tinyint	1	If you specify a value of 1, stops are ignored for Dynamic Zoning calculations; therefore an activity that is more than the DynamicZoningDistance from a special stop can still be inserted on the route. Routes that only contain special stops are considered empty and can be seeded.	

FWActivity

This table is used to represent a specific leg of an order (activity). An activity is a unity of work that can be performed at a specific location. Whenever a new order is created, at least one FWActivity record must be created. More than one activity record can be created and the number created is dependent on the incoming order type.

Create: Must be associated with a location and order records. The order type drives the number of activity records that need to be created. An activity cannot be created without an order being supplied and created.

***For example:** one activity is created for a pickup or drop-off order type. A pickup/drop-off order type will have two activities associated with it and both need to be created. Separate pickup and drop-off type activities is created.*

In addition, at least one stop record will need to be created that links each activity to a specified schedule (PreferredScheduleKey) value. If a schedule is not specified, the stop is created to a DefaultSchedule that should always be available. For more stop creation information please see the FWStop descriptions.

Default: Many of the Activity attributes are derived or defaulted from other table values. There are two types of defaulting types: cumulative and override.

- **Cumulative:** The cumulative attributes are added or linked from values on other tables.
- **For example:** the measure amounts are added together and the total is represented on the activity record. The StopTemplate, Location, Order, and OrderLine measure amounts are added together and stored on the Activity record. This allows the user to define appropriate values at the different table levels: product, order, location, and template.
- **Override:** The override attributes get replaced with the value at a higher level.
- **For example:** the StopTemplate is the lowest level and its value will only be used if no other table has that specific attribute set. The override priority sequence from lowest to highest is: FWStopTemplate, FWLocation, FWOrderLine, and FWOrder.

The default value for the activity table attributes is referenced with either cumulative or override type functions unless otherwise specified.

Attribute Name	Type	Field Size	Description	Required
ActivityID	int	4	Unique internal DB Identifier of the Activity record.	Required
OrderID	int	4	Unique internal DB Identifier of the Order record. System assigned to the Order ID of the FWOrder OrderKey.	Required
OrderKey	nvarchar	50	Business key to identify the Order this Activity refers to. System assigned to the Order Key of the FWOrder table. Must exist on the FWOrder table.	Required
ActivityKey	nvarchar	50	Business key to identify the activity this Activity refers to. Users may choose to identify individually every activity contained in the order by using this field.	
StopTemplateKey	nvarchar	50	Business key to identify the Stop template this Activity refers to. Defaults to the StopTemplateKey of the FWOrder. If this value is ever changed, the reset function needs to be performed so that the new template values are reflected on this activity.	
LocationID	int	4	System assigned during creation of an order.	
LocationKey	nvarchar	50	System assigned during creation of an order, but can be modified by the user. If this value is ever changed, the reset function needs to be performed so that the new location values are reflected on this activity. Note— If changed it will not automatically update	

			the associated stops but the activity status attribute is changed to Modified.	
Type	nvarchar	50	This field is used to identify different order types that may require different processes. For example: users can have pickup, drop-off, or pickup/drop-off orders, which may require different FWActivity records created. Valid Values: • DROPOFF – (default value) a drop-off type is create when the order type is drop-off or pickup/drop-off • PICKUP - a pickup type is create when the order type is pickup or pickup/drop-off • SERVICE – (currently not supported) a service type is create when the order type is service • RECHARGE – (currently no supported) a recharge type is create when the order type is recharge • BREAK – (currently not supported) a break type is create when the order type is break Note— If changed it will not automatically update the associated stops but the activity status attribute is changed to Modified.	Required
DeliveryRatio1-9	float	8	This attribute is used to reclaim capacity. A value of 1.0 will reclaim the full measure amount. A value of 0.5 will reclaim 50% of the amount, and a value of 0.0 will not reclaim any of the amounts. This setting is only	

			applicable if it is a DROPOFF or PICKUP type stop. Default is 1.0.	
InterspersableInd	bit	1	This Boolean attribute is used to set the Interspersable indicator for the related stop records that came from a DROPOFF/PICKUP order type record. It indicates if other stops can be scheduled between the PICKUP and DROPOFF type stops for that order. It is not applicable to single leg order types (PICKUP or DROPOFF). Default is False.	
Measure1-9	float	8	This numeric value is added to the other Measure1-9 amount values supplied with the activity input function. At least one measure amount must be specified for Activities types: DROPOFF, PICKUP, and RECHARGE. These values are used to manage the resource capacity. Note— The MeasureFactors must be applied before the values are added together.	
Measure1-9 Factor	float	8	This numeric value is used to adjust the actual corresponding measure amount. It is applied when the FWActivity Measures are created, but only for the OrderLine Measure portion.	

ServiceDuration	int	4	It represents the total service time in seconds. This integer value (in seconds) is added to the service duration of the order, order lines, and location values service duration when the activity is created. Default is zero (0). When the ServiceRateMx and the ServiceFactorMx fields are used, the final calculated ServiceDuration for all line items will be saved in this field. This field is an internal FW Derived field.
PreServiceDuration	int	4	This integer value (in seconds) is added to the pre-service duration of the order, order lines, and location pre-service duration when the activity is created. See the service duration attribute for more details. Default is zero (0).
PreferredResourceKey	nvarchar	50	If entered, this value must exist on the FWResource table. Default is order creation. Note— If the preferred resource record is not associated with the same schedule as this activity, then the preferred resource value is removed from the associated stop record and is ignored.
PreferredResourceInd	bit	1	This is a Boolean attribute that determines if the preferred route is restricted to the preferred route only. This attribute is dependent on the PreferredResourceKey on the FWActivity table and is ignored when if the PreferredResourceKey is empty. Default is False.

Profit	float	8	This numeric value is added to the other profit values from the FWOrder and FWStopTemplate and FWLocation.	
Requirements	nvarchar	255	This is a string value that is linked with the requirements for the FWOrder, FWStopTemplate, and FWLocation. Each value must exist on the FWCode table for type Requirements. A selection list should be provided where one or more requirements can be selected. See the StopTemplate for more details.	
Commodities	nvarchar	255	This attribute holds the additional commodities at the order line level that is linked with the commodities from the Order, Stop Template, and Location tables whenever an activity is created. Each value must exist on the FWCode table for type Commodities. A selection list should be provided where one or more commodities can be selected. See the StopTemplate for more details.	
OnLoadPosition	tinyint	1	Use to explicitly specify first or last position on a route. Valid values: • 0 = ignored • 1 = must be first on load • 2 = must be last on load Default is zero (0).	
RoutePosition	int	4	This value sets the priority of certain stops positioned before other stops. For example: <i>Stops with RoutePosition set to 1 are sequenced before activities with RoutePosition set higher than 1. A value of zero can be sequenced in any order.</i> Default is zero (0).	

Territory	nvarchar	50	When entered it must exist on the FWCode Table. Default is empty.	
SpecialHandling	tinyint	1	Currently 0 and 1 are the only values. If you specify a value of 1, stops are ignored for Dynamic Zoning calculations; therefore an activity that is more than the DynamicZoningDistance from a special stop can still be inserted on the route. Routes that only contain special stops are considered empty and can be seeded. Default is zero (0).	
AllowConsolidationInd	bit	1	This is a Boolean value used to override the AllowConsolidationInd on the Schedule optimization setting for a specific stop. If specified as default, and not overwritten by the order, all orders will have this setting. Default is false.	
ProfitDepreciationPerHour	float	8	It is a percentage value that will depreciate the profit by the percentage value for every hour from the start of the route. It makes orders less profitable later in the route. Negative values are allowed and when specified will provide the opposite effect of making later stops more profitable. Default is zero (0). Values from -1 to 1.	
AvailableDate	datetime	8	Earliest date an item (or, resource carrying it) can leave the initial depot. If this is not supplied then the Schedule date is used with AvailableTime. If there are multiple PackageItems for a Job, the latest value for the PackageItems and Job is used for the Job.	

AvailableTime	datetime	8	Earliest time of the day an item (or, resource carrying it) can leave the initial depot. If there are multiple PackageItems for a Job, the latest value for the PackageItems and Job is used for the Job.	
RestrictPreferredResource	bit	1	Determines if job is limited to its PreferredRoute setting	
RestrictRoutePosition	bit	1	Determines if a job is limited to its RoutePosition value.	
MasterRoute	int	4	Designates whether or not this activity belongs to a master order.	Required
ItineraryOrderType	int	4	Used to identify if this activity belongs to a Cross Dock Leg. Defaults are: <null> or 0 (zero).	Required
Comment	nvarchar	500	This field is used to provide additional information about a specific record.	
WorkUnits	float	8	The amount of workload an order contributes toward the maximum allowed by a resource's work units	
HazMatType	int	4	Indicate it is hazardous material	
OrigRestrictRoutePosition	bit	1	Stores the original value of RestrictRoutePosition. in an unlocking of a route event, this value will be used to reset RestrictRoutePosition	
OrgRoutePosition	int	4	Stores the original value of RoutePosition. in an unlocking of a route event, this value will be used to reset RoutePosition	
ProductTypeKeys	nvarchar	255	All the productTypeKey from all the orderliness related to this activity	
Rank			Used to sequence pickup jobs for a multi-task order. Rank values must start at 0 and increase sequentially or the setting will be ignored. For example, "0,2,3"	

and "1,2,3" are not valid but "0,1,2" is valid. This only applies to multi-task orders.

If it is not entered, backend will send -1 to rmpi so that rmpi will consider the sequence of the pickup by its engine. If the Rank is specified, then the pickup sequence will be determined by the Rank with lowest Rank value first.

FWActivityWindow

 **Note**— See Generic Window Attributes for details.

This table is used to specify specific time windows for a given activity, and when it can be serviced. It is optional in that it does not need to exist when an FWActivity table record is created.

Create: Must be associated with an activity record.

Delete: Must be deleted whenever the associated activity record is deleted.

Update: individual record only.

Attribute Name	Type	Field Size	Description	Required
ActivityWindowID	int	4	Unique internal DB Identifier of the Activity window record.	Required
ActivityID	int	4	Unique internal DB Identifier of the Activity record.	Required
Type	nvarchar	50	This field is used to identify different order types that may require different processes. For example: we can have pickup, drop-off, or pickup/drop-off orders, which may require different FWActivity records created. Valid Values: • DROPOFF – (default value) a drop-off type is create when the order type is drop-off or pickup/drop-off • PICKUP - a pickup type is create when the order type is pickup or pickup/drop-off	Required

			- SERVICE – (currently not supported) a service type is create when the order type is service - RECHARGE – (currently no supported) a recharge type is create when the order type is recharge - BREAK – (currently not supported) a break type is create when the order type is break Note— If changed it will not automatically update the associated stops but the activity status attribute is changed to Modified.	
EarliestTime	datetime	8	Beginning time for time window	
LatestTime	datetime	8	Ending time for time window	
EarliestDate	datetime	8	Beginning date for time window X. Values of 'ScheduleDate + X and 'Today + X' can be used to create relative dates.	
LatestDate	datetime	8	Ending date for time window X. Values of 'ScheduleDate + X' and 'Today + X' can be used to create relative dates.	
Frequency	bit	255	Designates which weekdays or dates the item is available to be scheduled. For weekdays, you can use the standard abbreviations, full names, or numbers 1 through 7 corresponding to days Monday through Sunday. Dates may be written as M/D/Y, M-D-Y, MMDDYY, or MMDDYYYY. For example, a setting of "135" schedules the item for Monday, Wednesday and Friday. A setting of "Mon Wed Fri 6/30/1999 7/1/1999" schedules the item for Monday, Wednesday, and Friday and for 6/30/1999 and 7/1/1999.	
ProfitFactor	float	8	Portion of the job or task's profit gained by serving it within the time window.	
VariationDuration	int	4	A "fudge factor" for time windows. Extends every time window's open and close times by the specified duration.	
VariationProfit	float	8	Portion of the task's profit gained when it is served outside the defined time	

			window, but within the allowed variation. Encourages Optimizer to assign jobs inside the variation if the time window cannot be achieved. For example, a value of 0.75 gives a job served within the variation 75% of its normal profit.	
Comment	nvarchar	500	See <i>Comment</i> under <i>Generic Attributes</i> .	
Measure1-9	float	8	The measure amount x to be delivered using this time window.	
OpenDateTime	datetime	8	Opening Date/Time of the time window.	
CloseDateTime	datetime	8	Closing Date/Time of the time window.	
FreeTextFrequency	Nvarchar	255	If the Frequency field is empty, the FreeTextFrequency field is sent instead to RMPI.	
SelectedWindow	boolean		Determines whether the Time Window selected after an optimization was performed.	

FWStop

The table is used to represent a specific leg of an order that is associated with a specific schedule. The stop table is used to link a schedule table with an activity table. It is used to break the "many-to-many" relationship between schedule and activity.

It is associated with an Activity and Schedule records. Activities with the same OrderKey must also be associated with the same schedule; therefore, the associated activity stops will all need to be created as a group.

For example: two activities with the same order and scheduler keys must be treated to the same order and need to be treated as a unit of work. They must be move, copied, created, and deleted as a unit.

Attribute Name	Type	Field Size	Description	Required
StopID	int	4	Unique internal DB Identifier of the Stop record.	Required
RouteID	int	4	Unique internal DB Identifier of the Route record. Must exist on the FWRoute table. It is important that the same ScheduleID and ScheduleKey on the specified route are the same as the ScheduleID and SccheduleKey on this stop.	

ScheduleID	nvarchar	50	Unique internal DB Identifier of the Schedule record. Must exist on the FWSchedule table. It must be updated whenever the stop is not assigned to any route. When assigned, it must be the corresponding ScheduleKey value. When this attribute is empty, it is assigned to a specific route. The route ScheduleID can be used.	Required
ScheduleKey	nvarchar	50	Business key that identifies the Schedule this Stop refers to. Must exist on the FWSchedule table. It is important that the ScheduleKey is always the same as the ScheduleKey of the associated route when assigned to a route and the corresponding ScheduleID key when unassigned. See creation notes above for more information. (types pickup and drop-off)	Required
ActivityID	int	4	Unique Identifier of the Activity record. Must exist on the FWActivity table.	
OrderID	int	4	Unique Identifier of the Order record. Must exist on the FWOrder table.	
OrderKey	nvarchar	50	It cannot be modified. Stops associated with activities with the same OrderKey: • must all be assigned to the same Schedule • must all be moved if one is moved • must all be copied if one is copied • must all be deleted if one is deleted	
LocationID	int	4	Unique Identifier of the Location record. Must exist on the FWLocation table.	
LocationKey	nvarchar	50	Business key to identify the activity this Stop refers to. Must exist on the FWLocation table.	
LocationType	nvarchar	50	The type of location, e.g., Customer, Depot, etc.	
ModifiedBy	int	4	See <i>ModifiedBy</i> under <i>Generic Attributes</i> .	
ModifyDate	datetime	8	See <i>ModifyDate</i> under <i>Generic Attributes</i> .	

CreatedBy	int	4	See <u>CreatedBy</u> under <u>Generic Attributes</u> .	
CreatedDate	datetime	8	See <u>CreatedDate</u> under <u>Generic Attributes</u> .	
Type	tinyint	1	This field is used to identify different order types that may require different processing. For example: we can have pickup, drop-off, breaks, and rest records created. Valid Values: • 0 = Initial Station Stop • 1 = Final Station Stop • 2 = Pickup Stop • 3 = Delivery Stop • 5 = Dynamic Recharge • 8 = Redispatch Stop (Dynamic) • 9 = Static Recharge • 10 = Redispatch Stop (Static) • 35 = Locationless Stop • 66 = Pickup Complex (Used for Double Ended Jobs or Multi-Task Jobs) • 67 = Delivery Complex (Used for Double Ended Jobs or Multi-Task jobs)	Required
State	tinyint	1	Valid Values: (default = active when created - Inactive when copied) • 1 = ACTIVE - An active Stop cannot be deleted unless it is the only stop associated with that Activity. A special function will need to be created so that active stops can be made temporary or cancelled. By "special," we mean that the transition is done in one transaction so as not to break the rule that there must always be one, and only one, active stop at any given time. • 2 = INACTIVE – An inactive stop is reserved for a previously active stop that is no longer active but persisted for reporting or historical reasons. As with the active stop, only one inactive stop can exist for any given schedule. • 2 = TEMPORARY – A temporary stop is used to perform route testing	

			scenarios used to perform “what-if” functionality. A temporary stop can only be made active if no other stop for that activity is already active. Multiple temporary stops can exist, but only one for a given schedule. • 3 = CANCELLED – Currently not supported - A cancelled stop can only be made active if no other stop for that activity is already active. More than one cancelled stop can exist within the same schedule. Cancelled stop should be used minimally. A stop should be deleted if it is possible to delete. Meaning that if it has not shipped, do not cancel the stop but simply delete it. The cancel feature should be reserved for situations where the goods are on the road and the customer would like to reschedule or change the order without issuing a new order.	
Status	tinyint	1	Indicates the message status. • 1 – Pending • 2 – Arrived • 3 – Completed • 4 – Delayed • 5 – En-route • 6 – Exception • 7 – Hooked • 8 – In-service • 9 – Missed • 10 – No Freight • 11 – Refused • 12 – Returned • 13 – Skipped • 14 – Spotted • 15 – Dispatched • 16 – Home • 17 – End Route • 18 – Begun • 19 – Departed • 20 – Enroute Recharge	Required

			• 99 – Info	
OnBoardMeasure1-9	float	8	Sum of OnBoard Capacity for all Activities.	
AvailableDate	datetime	8	Earliest date an item (or, resource carrying it) can leave the initial depot. If this is not supplied, then the Schedule date is used with AvailableTime. If there are multiple PackageItems for a Job, the latest value for the PackageItems and Job is used for the Job.	
AvailableTime	datetime	8	Earliest time of the day an item (or, resource carrying it) can leave the initial depot. If there are multiple PackageItems for a Job, the latest value for the PackageItems and Job is used for the Job.	
GeoStopNumber	smallint	2	This attribute represents a unique location where duplicate numbers are possible if more than one stop is at the same location. Meaning, different stops can have the same GeoStopNumber if they are at the same location. The number will range from one to the total number of stops on a given route. It is also sequenced based on unique locations. When it is unassigned, or not associated with a route, this value is 0.	Required when on a route
StopNumber	smallint	2	This attribute represents the stop number in the sequence on the route. The number will range from one to the total number of stops on a given route. All stops are accounted for. Every time a route is sequenced, this number is changed to reflect the new sequence. When it is unassigned or not associated with a route, this value is 0.	Required when on a route
PhysicalStopNumber	smallint	2	This attribute is very similar to the StopNumber, which represents the stop number in sequence on the route, except it will not include any stops that have violations. The number will range from one to the total number of stops, but can be less if some stops are in violations. This value cannot be greater than the StopNumber value. Every time a route is sequenced, this number	Required when on a route

			changes to reflect the new sequence. When it is unassigned or not associated with a route, this value is 0.	
CustomerStopNumber	smallint	2	This attribute represents a unique customer where duplicate numbers are possible if more than one stop is to service the same location for the same customer. The number will range from one to the total number of stops on a given route. It is also sequenced based on unique customer. When it is unassigned or not associated with a route, this value is 0.	
ArrivedDate	datetime	8	Projected arrival date and time. Must be empty if Previous stop DepartedDate is empty. Must be equal to or greater than Previous stop DepartedDate.	
BegunDate	datetime	8	Projected date and time begun. Must be empty if ArrivedDate is empty. Must be equal to or greater than ArrivedDate.	
CompletedDate	datetime	8	Date and time when job or task was completed. Must be empty if BegunDate is empty. Must be equal to or greater than BegunDate.	
DepartedDate	datetime	8	The actual date and time that the vehicle departed the depot. Must be empty if CompletedDate is empty. Must be equal to or greater than CompletedDate.	
ActualMeasure1-9	float	8	Actual reading reported for Measure 1...9.	
ProjectedViolation	int	4	Used to identify stop violations which will need to be translated before they are displayed.	
ProjectedProfit	float	8	This is the system calculated profit for the stop.	
ProjectedArrivalDate	datetime	8	Must be empty if not assigned to a route. Required when assigned to a route.	

ProjectedBegunDate	datetime	8	Must be empty if ProjectedArrivedDate is empty. Must be equal to or greater than ProjectedArrivedDate	
ProjectedCompletedDate	datetime	8	Must be empty if ProjectedBegunDate is empty. Must be equal to or greater than ProjectedBegunDate. Calculation: Add the pre-service duration and the service duration of the Stop and Route to the ProjectedArrivedDate. Note— This is a derived value and may not be available in the future.	
ProjectedDepartDate	datetime	8	Must be empty if ProjectedCompletedDate is empty. Must be equal to or greater than ProjectedCompletedDate. Calculation: Add the pre-service the service and the post-service duration of the Stop and Route to the ProjectedArrivedDate. Note— This is a derived value and may not be available in the future.	
ProjectedElapsedTime	int	4	Projected time in seconds from start of route to completion of this stop. Calculation: Difference between initial stop ProjectedArrivedDate to the current stop ProjectedCompletedDate. Note— This is a derived value and may not be available in the future.	
ProjectedDeltaElapsedTime	int	4	Projected time in seconds between stops. Calculation: Difference between previous stop ProjectedCompletedDate to the current stop ProjectedCompletedDate. Note— This is a derived value and may not be available in the future.	
ProjectedTimeDriven	int	4	Time in seconds from the beginning of the route to this stop. Calculation: Total of all the ProjectedDeltaTimeDriven of all the stops between the initial stop and the current stop (including the current stop).	

			Note— This is a derived value and may not be available in the future.	
ProjectedDeltaTimeDriven	float	8	Time in seconds spent driving between this and the previous stop. Calculation: Difference between current stop ArrivedDate to the previous stop DepartedDate. Note— This is a derived value and may not be available in the future.	
ProjectedDistance	float	8	Distance driven in meters from the beginning of the route to this stop. Must be equal to or greater than ProjectedDeltaDistance. Calculation: Total of all the ProjectedDeltaDistance of all the stops between the initial stop and the current stop (including the current stop). Note— This is a derived value and may not be available in the future.	
ProjectedDeltaDistance	float	8	Must be equal to or less than ProjectedDistance.	
ProjectedDeltaPenalty	float	8	Incremental cost between successive stops of a route.	
ProjectedBreaks	smallint	2	Number of breaks taken between the prior and current stops. Cannot be empty if BreakTime is not empty.	
ProjectedBreakTime	int	4	Break time in seconds taken between the prior and current stops. Cannot be empty if NumberOfBreaks is not empty. Calculation: Number of Breaks on this stop multiplied by the BreakDuration of the associated route. Note— This is a derived value and may not be available in the future.	
ProjectedRests	int	4	Number of rests taken between the prior and current stops. Cannot be empty if RestTime is not empty.	
ProjectedRestTime	int	4	Rest time in seconds taken between the prior and current stops. Cannot be empty if NumberOfRests is not empty.	

			Calculation: Number of rests on this stop multiplied by the RestDuration of the associated route. Note— This is a derived value and may not be available in the future.	
ProjectedWaitTime	int	4	Time in seconds waiting for the time window to open. Calculation: Difference between current stop ArrivedDate to the current stop WindowOpen. Note— This is a derived value and may not be available in the future.	
ProjectedWorkTime	int	4	Time in seconds to do the work. Calculation: Difference between current stop BegunDate to the current stop CompletedDate. Note— This is a derived value and may not be available in the future.	
ProjectedSlackTime	int	4	Slack time is the amount of time the beginning of the service on a job can be delayed without introducing violations or changing the net profit of the route.	
ProjectedOffset	int	4	The number of minutes the location of the Stop is offset from UTC (based on the location's TimeZone).	
ProjectedDeltaTimeDriven	float	8	The projected (current or updated) drive time in minutes, as opposed to the original planned drive time.	
Projected3rdPartyDistance	float	8	Total driving distance calculated by a 3rd party application from the starting depot to a current stop along a Rmpi-generated route. FW 7.1 and 8.1 support PCMiler for this purpose.	
PlannedArrivedDate	datetime	8	Keeps the original value of the ProjectedArrivedDate. Once the route enters into an execution state (first GPS or Status msg is received) this field will remain unchanged.	
PlannedBegunDate	datetime	8	Keeps the original value of the ProjectedBegunDate. Once the route enters into an execution state (first GPS or Status msg is received) this field will remain unchanged.	

PlannedCompletedDate	datetime	8	Keeps the original value of the ProjectedCompletedDate. Once the route enters into an execution state (first GPS or Status msg is received) this field will remain unchanged.	
PlannedDepartedDate	datetime	8	Keeps the original value of the ProjectedDepartedDate. Once the route enters into an execution state (first GPS or Status msg is received) this field will remain unchanged.	
PlannedStopNumber	int	4	Keeps the original value of the StopNumber. Once the route enters into an execution state (first GPS or Status msg is received) this field will remain unchanged.	
PlannedDistance	int	4	Keeps the original value of the ProjectedDistance. Once the route enters into an execution state (first GPS or Status msg is received) this field will remain unchanged.	
PlannedTimeDriven	int	4	Keeps the original value of the ProjectedTimeDriven. Once the route enters into an execution state (first GPS or Status msg is received) this field will remain unchanged.	
PlannedElapsedTime	int	4	Keeps the original value of the ProjectedElapsedTime. Once the route enters into an execution state (first GPS or Status msg is received) this field will remain unchanged.	
PlannedWorkTime	int	4	Keeps the original value of the ProjectedWorkTime. Once the route enters into an execution state (first GPS or Status msg is received) this field will remain unchanged.	
ActualDistance	int	4	Stores the Actual reported Distance	
ActualElapsedTime	int	4	Stores the Actual Elapsed time based on time stamps	
ActualServiceDuration	int	4	Stores the Actual Service Duration based on time stamps	
ActualTravelTime	int	4	Stores the actual travel time based on time stamps	

DeltaDistance	int	4	This is the delta distance based on the following calculation, (Actual Distance – Planned Distance).	
DeltaTravelTime	int	4	This is the delta time traveled based on the following calculation, (ActualTravelTime – PlannedTravelTime).	
DeltaServiceDuration	int	4	This is the delta service duration based on the following calculation, (ActualServiceDuration – ServiceDuration).	
DeltaElapsedTime	int	4	This is the delta elapsed time based on the following calculation, (ActualElapsedTime – PlannedElapsedTime).	
PerfElapsedTime	float	8	This is the performance elapsed time based on the following calculation, ((PlannedElapsedTime/ActualElapsedTime) * 100).	
PerfServiceDuration	float	8	This is the performance service duration based on the following calculation, ((PlannedServiceDuration/ActualServiceDuration) * 100).	
PerfTravelTime	float	8	This is the performance travel time based on the following calculation, ((PlannedTimeDriven/ActualTimeDriven) * 100).	
PerfDistance	float	8	This is the performance distance based on the following calculation, ((PlannedDistance/ActualDistance) * 100).	
MasterRoute	int	4	Indicates if this is a Master Route. Valid values are: 0 – Normal Route 1 – Master Route	Required
AVLStatus	int	4	Field reserved for AVL generated status codes. The status codes are: 1 – Pending 2 – Arrived 3 – Completed 4 – Delayed	

			• 5 – En-route • 6 – Exception • 7 – Hooked • 8 – In-service • 9 – Missed • 10 – No Freight • 11 – Refused • 12 – Returned • 13 – Skipped • 14 – Spotted • 15 – Dispatched • 16 – Home • 17 – End Trip • 18 – Departed • 19 – Dropped • 20 – Assigned • 21 – Rescheduled	
MasterRoutePosition	int		Route position in the master route	
UnassignWirelessInd	bit	1	Indicate that it is unassigned by wireless message	
ReportedLatitude	float		Contains the report latitude of the stop.	
ReportedLongitude	float		Contains the reported longitude of the stop.	
ProjectedWorkUnits	float		Projected work units	
Delayed	bit	1	Flag to indicate when a delay has been reported	
StartDate	datetime	8	StartDate for the delay	
EndDate	datetime	8	EndDate for the delay	
LastRteHistMsgID	int	4	Last route history recode ID	
LastRteHistMsgIDWActual	int	4	Last route history record ID with actual measures or time	
ArrivedDateUTC	datetime	8	Actual Arrived Date (Coordinated Universal Time)	
BegunDateUTC	datetime	8	Date and time when job or task was started. (Coordinated Universal Time)	

CompletedDateUTC	datetime	8	Date and time when job or task was completed. (Coordinated Universal Time)	
DepartedDateUTC	datetime	8	Date and time when job or task was departed. (Coordinated Universal Time)	
PlannedArrivedDateUTC	datetime	8	Keeps the original value of the ProjectedArrivedDate. Once the route enters into an execution state (first GPS or Status msg is received) this field will remain unchanged. (Coordinated Universal Time)	
PlannedBegunDateUTC	datetime	8	The planned begun date for the stop.	
PlannedCompleteDateUTC	datetime	8	The planned completed date for the stop. (Coordinated Universal Time)	
PlannedDepartedDateUTC	datetime	8	The planned departed date for the stop. (Coordinated Universal Time)	
projectedarrivedDateUTC	datetime	8	The projected arrived date for the stop. (Coordinated Universal Time)	
projectedBegunDateUTC	datetime	8	The projected begun date for the stop. (Coordinated Universal Time)	
ProjectedCompleteDateUTC	datetime	8	The projected completed date for the stop. (Coordinated Universal Time)	
ProjectedDepartedDateUTC	datetime	8	The projected departed date for the stop. (Coordinated Universal Time)	
StartDateUTC	datetime	8	Start date/time of the route (Coordinated Universal Time)	
EndDateUTC	datetime	8	End date/time of the route	
ResidualMeasure1-9				
Published	int	4	This field indicates if a Stop has been published either caused by Publishing a Route or the stop. Possible Values: Null or 0: Unpublished 1: Published 2: Changed After Published	
startupAllocCost	Float		Reserved for future development	

DistanceWeightAllocCost	float		Reserved for future development	
SequenceSameStopJobs	int		Configures the optimization sequence of multi-order stops: • 0: Do not apply • 1: Sequence deliveries before pickups by location key • 2: Sequence deliveries first and then pickups by location key	
IdleTime	float		Idle time at the stop level is calculated using GPS Status messages received between a stop's Arrive and Complete or Departed status. The GPS messages used are all those that belong to the same chain where the Ignition=true and Speed=0. Several chains may exist within the window between Arrive and Complete or Departed status.	
GPSArrival	datetime		Displays the GPS arrival time value of a stop. The GPS Arrival field will be marked in red if the value is greater than 'x' minutes different than the reported arrival.	
GPSDepart	datetime		Displays the GPS departure time value of a stop. The GPS Depart field will be marked in red if the value is greater than 'x' minutes different than the reported departure.	
GPSServiceTime	datetime		Displays the GPS service time value of a stop. The GPS Service Time field will be marked in red if the value is over a configured threshold.	
MasterBHTRel	int	1	Values: null: Stop is not a replacement for a master route generated shipment 1: Stop is a replacement for a master route generated shipment	
SelectedWindowMeasure1...9	float		Measure delivered based on the selected time window.	
PlannedGeoStopNumber			When assigning or unassigning stops, the PlannedGeoStopNumber value will be equal to the GeoStopNumber. When a statused stop is added in the middle of a	

route, the PlannedGeoStopNumber value will not change, but the GeoStopNumber value will change.

FWRouteTemplate

The FWRouteTemplate table is used to introduce the different routing settings associated with a resource record. When a route template is specified, it provides many of the default settings needed by the different optimization engines. Resource records are created by an external resource management system or API requests. These requests may already have certain attributes set and will take precedence over the route template values during the resource creation process. This provides a level of override flexibility to certain settings for a specific resource without having to create a different route template record for every possible setting combination. The intent of this table is to provide many of the optimization settings without needing to specify everything whenever a resource is created. The route template can also be used as a base template from which multiple or repetitive resources can be created. Please keep in mind that certain attributes are cumulative, not overridden, and certain templates can be used to override the resource based on the template type.

Attribute Name	Type	Field Size	Description	Required
RouteTemplateID	int	4	Unique Identifier of the Route Template record.	Required
RouteTemplateKey	nvarchar	50	Business key to identify the activity this Route template refers to. Must exist in the Route Template table. It is important that the same ScheduleID and ScheduleKey on the specified route are the same as the ScheduleID and ScheduleKey on this stop.	Required
InitialLocationKey	nvarchar	50	The initial location at which the route begins. ID of an entry in the FWLocation Table. The initial location attribute will determine the time zone for all the date saved on this record. If this value changes, all times will need to change to the new time zone.	
FinalLocationKey	nvarchar	50	The final location (Station) at which the route ends. ID of an entry in the FWLocation Table.	
ScheduleKey	nvarchar	50	If entered, it must exist on the FWSchedule table. A selection list should be provided. This field is used to link a	

			resource to a specific schedule during creation by creating a link route entry that associates this schedule and resource combination. All Route templates should default to the DefaultSchedule.	
ModifiedBy	int	4	See <i>ModifiedBy</i> under <i>Generic Attributes</i> .	
ModifyDate	datetime	8	See <i>ModifyDate</i> under <i>Generic Attributes</i> .	
CreatedBy	int	4	See <i>CreatedBy</i> under <i>Generic Attributes</i> .	
CreatedDate	datetime	8	See <i>CreatedDate</i> under <i>Generic Attributes</i> .	
Comment	nvarchar	500	See <i>Comment</i> under <i>Generic Attributes</i> .	
Name	nvarchar	255	See <i>Name</i> under <i>Generic Attributes</i> .	
Type	nvarchar	50	This field is used to identify different stop template types that may require different processes. For example: we can have a standard default type that is typically used for default value where the incoming settings take precedence. We can also have a set where the template values are used regardless of the incoming values. Currently only the default type is supported. The value must be validated against the FWCode table StopTemplateType. Valid Values: DEFAULT = This is the normal template type used for default values whenever a value is not provided with the resource creation process.	Required
Status	nvarchar	50	This field is used to identify different status' of a given route template over time. The current implementation is configured to support one status value "DEFAULT". As the application gets built out, additional statuses will follow.	Required
Territory	nvarchar	50	Must be validated against the FWCode table for a value territory. Default is empty.	

DriverKey	nvarchar	50	Must be validated against the FWCode table. As we build out the resource management system, this will represent a foreign key to a driver or user of type driver table. Default is empty.	
TrailerKey	nvarchar	50	Must be validated against the FWCode table. As we build out the resource management system, this will represent a foreign key to a Trailer or equipment of type trailer table. Default is empty.	
TractorKey	nvarchar	50	Must be validated against the FWCode table. As we build out the resource management system, this will represent a foreign key to a tractor or equipment of type tractor table. Default is empty.	
Measure1-9	float	8	These values are applicable to the corresponding stop occurrence values used to manage capacity. Default is zero (0). For example: If the stop is a Drop-off type stop then this amount is reduced by the corresponding stop amounts (plus factor) to reflect additional capacity on this resource and save it in the ProjectedMeasureCapacity on the route record. If the stop is a pickup then the corresponding amount is added to reduce this capacity.	
Measure1-9Factor	float	8	When entered, the Measures1-9 occurrence must also be entered. This value provides the ability to reduce or increase the Measure values by this factor. This attribute is a percentage value used to adjust the measure value. Default is one (1). For example: If the factor is less than one then the measure is reduced. If it is equal to one, then the measure will stay the same; and if the factor is more than one, then the measure is more. A negative value is not allowed.	
Measure1-9RechargeRate	float	8	When entered, the Measures1-9 occurrence must match the location supplied measure values; otherwise,	

			recharge is ignored. The recharge rate is measured in units per second and is associated with the MeasureRechargeRate and Measure values of the location. Default is zero (0).	
MaxStops	smallint	2	This value is used to set the absolute threshold of the total number of stops that can be associated with this route. The stop types must be PICKUP (1), DROPOFF(0) or RECHARGE(4). Default is zero (0).	
FullThreshold	float	8	Used as a percentage of the capacity (measure amount) to apply the CompleteFullBonus and PartialFullPenalty values. Please see complete full bonus and partial full penalty for details. A value of 1.0 will require all measures to be at capacity before the Complete Full Bonus is applied and to avoid incurring the Partial Full Penalty. A value of 0.5 will require at least 50% capacity utilization on all measure to receive the Complete Full Bonus. If one of the measures is less than 50%, then the Partial Full Penalty is applied. A value of 0.0 (default) indicates that the bonus and penalty values is ignored. Default is zero (0).	
CompletedFullBonus	float	8	This attribute works in conjunction with the FullThreshold value. The CompleteFullBonus is added to the overall route profit when all measures are equal to or greater than the FullThreshold values. If not, then this value is ignored. Default is zero (0).	
PartialFullPenalty	float	8	This attribute works in conjunction with the FullThreshold value. The PartialFullPenalty is subtracted from the overall route profit when one or more measures are less than the FullThreshold values. If not, then this value is ignored. Default is zero (0).	
Requirements	nvarchar	255	Each specified code must exist on the FWCode table for the Requirement type. Multiple entries are possible and delimited by a comma or space. These	

			values are accumulated with the requirements supplied with the input request to create a resource. They are used to match the requirement values on the stop records, and each specified stop requirement must be defined on the route requirements field or the stop will not be serviced by that resource. Additional requirements can be specified at the resource and route level and are ignored, but additional requirements cannot be specified at the stop level. All stop requirements must be specified by the route record to be considered for possible service on that route. Default is empty.	
Profit	float	8	This attribute is set to a base profit value. It is cumulative where the template value is added to the incoming resource creation call and saved on the Resource record. Default is zero (0).	
SameStopBonus	float	8	This value is added to the overall profit of the route for every additional stop at a specific location. For example: If we have two stops at the same location, then the second stop will have the bonus applied once. If a third stop was also served at the same stop, then the bonus is applied twice, once for the second stop and again for the third stop. If we have two different locations with two serviceable stops at each location (type pickup or drop-off), then the bonus is applied twice, once for the second stop of the first pair and again for the second stop of the second pair. And so on.	
DiscountInitialLeg	float	8	This value is used to discount the penalty of the initial leg of the route. It is a percent value used to reduce the cost of that leg. For example: a value of 1.0 will discount the full penalty of this leg and no penalty is incurred, a value of .6 will discount 60% of the penalty and apply 40% of the penalty. A	

			<i>value of .15 will discount 15% and apply 85% of the penalty.</i>	
DiscountFinalLeg	float	8	This value is used to discount the penalty of the final leg of the route. See the discount initial leg for details.	
PenaltyPerOrder	float	8	This value is used to apply a penalty for every order (type Pickup or Drop-off) serviced. Total penalty is multiplied by this value and the number of orders serviced. It is deducted from the overall route profit. If less than the MinPaidOrders have been assigned to this route, then the MinPaidOrders is used as the multiplier. See the MinPaidOrders for details. Default is zero (0).	
PenaltyPerMeter	float	8	Only used when the Schedule is set to StraightLine distances. This value is used to apply a penalty for every meter driven. Previous setting was in miles, so care should be taken when updating or using this value. This amount is multiplied by the distance and deducted from the overall route profit. Default is zero (0).	
MinPaidOrders	smallint	2	This value is used to set a minimum threshold of the number of orders that the PenaltyPerOrder is applied against. Default is zero (0). For example: if the number of orders serviced is three, and this value is set to 5, then the PenaltyPerOrder is applied against the minimum setting. If seven orders were serviced, then the penalty is applied against the actual value serviced and this setting is ignored.	
ServicePenalty	float	8	This attribute is used to set a base penalty value that is issued to deploy this resource. It is deducted from the overall route profit. Default is zero (0).	
MissedWindowPenalty	float	8	Per second penalty for a missed time window. With values greater than zero, the Optimizer tends to keep unserved	

			jobs close to their time windows instead of placing them geographically.	
LatestEndPenalty	float	8	This value is used to penalize a route that missed its LatestEndDate time. Regardless of how much the time window was missed, the same flat value is used. Default is 1000.	
WaitTimePenalty	float	8	This value is applied for every second that the resource waits to service the order. It is applied against every second of the route. WaitTime values are used.	
Work1-7Penalty	float	8	This attribute is based on using occurrence one first, then two, three, and so on. It is not possible to skip any occurrences. Occurrences two through seven must be empty if one is empty. Occurrences three through seven must be empty if two is empty and so on. This value is used in conjunction with the WorkDuration attribute. These values are used to either increase or decrease the penalty as the route executes over time at different rates. This value is applied for every second of duration. For example If only Work1Penalty is set then that will apply for the whole elapsed time of the route. If Work1duration is set as well then Work1Penalty will apply until the Elapsed time of the route equals Work1Duration at which point Work2Penalty will be applied. The Work Duration fields are cumulative. If Work1Duration is 36000 secs and Work2Duration is 54000 seconds then Work1Penalty will be applied until the elapsed time is 36000 seconds and Work2Penalty will be applied when the elapsed time is >36000 and <= 54000. Work3Penalty will be applied when the elapsed time is >54000 Please note that the duration must be in seconds.	
Work1-7Duration	int	4	This attribute is directly related to the Work1-7Penalty attribute. These values	

			are used to either increase or decrease the penalty as the route executes over time at different rates. See Work penalty for more details.	
BreakDuration	int	4	This attribute specifies the length of the break to be taken. If this attribute is empty the following must also be empty: • MaxElapsedTimeBeforeBreak • MaxDrivingTimeBeforeBreak • MaxNumberOfBreaks	
MaxElapsedTimeBeforeBreak	int	4	If entered, the BreakDuration must also be greater than 0. This attribute indicates how much time must pass before a break can be scheduled. Time is measured in seconds. Default is zero (0).	
MaxDrivingTimeBeforeBreak	int	4	If entered, the BreakDuration must also be greater than 0. This attribute indicates how much driving time must pass before a break can be scheduled. Time is measured in seconds.	
MaxNumberOfBreaks	tinyint	1	If BreakDuration was entered, then this attribute must be greater than 0. This attribute indicates the maximum number of breaks allowed for a given work unit. A work unit is dependent on whether rests are being used. If rests are not being used, then the number of breaks is allowed for the duration of the route. If rests are specified, then this number of breaks is allowed between every rest. For example: When a rest occurs, the number of breaks taken is reset and the count starts again. The default value is zero, which represents an unlimited number of breaks.	
BreakTimeOnDutyInd	bit	1	Determines if time spent on break is counted as elapsed time between rests.	
RestDuration	int	4	This attribute specifies the length of the rest to be taken. If this attribute is empty the following must also be empty: • MaxElapsedTimeBeforeRest • MaxDrivingTimeBeforeRest	

			FirstStopTimeThreshold	
MaxElapsedTimeBeforeRest	int	4	If entered, then RestDuration must be greater than 0. This attribute indicates how much time must pass before a break can be scheduled. Time is measured in seconds. Default is zero (0).	
MaxDrivingTimeBeforeRest	int	4	If entered, then RestDuration must be greater than 0. This attribute indicates how much driving time must pass before a break can be scheduled. Time is measured in seconds. Default is zero (0).	
FirstStopTimeThreshold	float	8	If entered, then RestDuration must be greater than 0. This attribute is very similar to the Early Rest Time Threshold attribute in that its value is applied against the maximum elapsed or driving time settings with the exception that it is only considered for the first stop on the route. This attribute allows users to have the first stop treated differently from other stops (e.g., only allow early rests to be scheduled for the first stop). The First Stop Time Threshold attribute is only evaluated if the Allow Early Rests setting is true. It is also represented as a percentage of the maximum elapsed or drive time. Default is zero (0). Note— Both the First Stop Time Threshold and the Early Rest Wait Threshold must be met for an early rest to be taken.	
UseFullWaitAsRestInd	bit	1	This attribute determines how the recuperation time is considered. If enabled, the entire recuperation time is considered to be off the clock and the clock is reset when work begins again. If disabled, then the clock will restart as soon as the recuperation time has elapsed. This is only significant if the idle time before a stop is larger than the Recuperation Time attribute. Default is true. For example: If Use Full Wait As Rest were false, the idle time before a particular stop were eight (8) hours,	

			and the Recuperation Time for the resource were six (6) hours, then the clock would restart two (2) hours before the idle time were completed (when work at the stop begins).	
AllowEarlyRest	bit	1	If set to true the following must also be greater than 0: • EarlyRestTimeThreshold • EarlyRestWaitThreshold This attribute indicates whether early rests are allowed. If yes, then rests can be scheduled before the maximum elapsed or drive time threshold values have been reached. If no, then the maximum elapsed or drive threshold values must be reached before a rest can be scheduled.	
EarlyRestTimeThreshold	float	8	If greater than 0, then the AllowEarlyRests must be set to true. This attribute indicates how much of the maximum elapsed or drive time threshold must be used before an early rest can be scheduled. The early rest time threshold is only evaluated when the Allow Early Rest indicator is enabled. The Early Rest Time Threshold value is represented as a percentage of the maximum elapsed or drive time. For example: if the Maximum Elapsed Time Before Rest attribute is set to four (4) hours and the Early Rest Time Threshold is set to .50 (50%), then the earliest that the rest can be scheduled would be two (2) hours into the route. Note— Both the Early Rest Time Threshold and the Early Rest Wait Threshold must be met for an early rest to be taken.	
EarlyRestWaitThreshold	float	8	If greater than 0, then the AllowEarlyRests must be set to True. This attribute indicates how much of the recuperation time must be available to schedule an early rest. It is represented as a percentage of the total recuperation time. In other words, the Early Rest Wait	

			Threshold attribute works in conjunction with the Recuperation Time attribute. For example: if the Recuperation Time is eight (8) hours and the Early Rest Wait Threshold is set to .75 (75%) then an early rest can be taken only if there is a period of idle time that is greater than six (6) hours, 75% of the total recuperation time. The early rest wait threshold is only evaluated when the Allow Early Rest indicator is enabled. Note— Depending on the stop in question, both the Early Rest Time Threshold or First Stop Time Threshold and the Early Rest Wait Threshold must be met for an early rest to be taken.	
OpportunisticEarlyRest	bit	1	Specifies whether or not an early rest should be taken whenever the wait time to serve a job is larger than the Resource's RecuperationTime. Default is 1 (true). If false, early rests are entirely controlled by FirstStopTimeThreshold, EarlyRestTimeThreshold, and EarlyRestWaitThreshold.	
RechargeDuration	int	4	This attribute is used to set a consistent recharge duration in seconds for this resource. Default is zero (0).	
MaxDwellDuration	int	4	Restricts the "dwell" time for a route, which is the total elapsed time from the beginning of service for the first job (initial drive and parking time are ignored) through the final return to depot. A value of 00:00 places no restriction on the dwell time. Default is zero (0).	
MaxElapsedDuration	int	4	Maximum elapsed time for the entire route. Elapsed time is measured from	

			start of route through the return to final depot. Note— Routes that violate this attribute incur the LatestEndingTimePenalty. Stops (including the final return to depot) that cause violations are marked with the letter code "T" and colored red.	
InitialServiceDuration	int	4	This attribute is used to set an initial service duration for the first job for the same customer at the same location. It is added to the standard service duration once for each grouping of same stop orders. Default is zero (0).	
ServiceDuration	int	4	This attribute is used to set a consistent service duration for this resource.	
PreServiceDuration	int	4	This attribute is used mostly to model parking time at each stop, but users can include other activity times here like item unloading time which is not reflected in ServiceDuration. When it's used to model parking time, it differentiates between resources that are easy to park and those that are difficult to park. For example, parking a large truck usually takes longer than parking a van. The main difference between PreServiceDuration and ServiceDuration is that PreServiceDuration can be scheduled before a time window starts whereas ServiceDuration must be served within the time window.	
ServiceSpeedFactor	float	8	This attribute can be used to adjust and control how quickly a resource serves stops. It is used to calculate how long it takes a resource to service a stop by dividing the stop's total service duration (including pre and post values) by the resource's ServiceSpeedFactor value. A value less than 1.0 increases the time the resource requires to service stops, and a value greater than 1.0 decreases the time. For example: If stop X has a total service duration of 100 seconds.	

			• A resource with a ServiceSpeedFactor of 1.0 needs 100 seconds to serve it ($100/1.0 = 100$). • A resource with a ServiceSpeedFactor of .50 needs 200 seconds to serve it ($100/0.5 = 200$). • A resource with a ServiceSpeedFactor of 1.5 needs 67 seconds to serve it ($100/1.5 = 67$).	
SpeedFactor	float	8	Used to offset the RoutingParameter speeds. Changes travel times without reconfiguring the Routing Parameter attributes or the speed attributes. Numbers greater than one create faster travel times; numbers less than one create slower travel times. The calculation is: adjusted time = calculated time divided by speed factor. For example: calculated time = 100, Speed factor = .75, resulting in an adjusted time of 133.34 ($100 \div .75 = 133.34$).	
MaxDistance	float	8	This attribute restricts the distance that this resource can travel in meters, not miles.	
MinTravelDistance	float	8	This attribute is used to determine the distance at which two stops are considered separate stops. If the distance between two sequential stops is less than this value, it is assumed that the vehicle parks once to service both stops. Please note that the unit is meters not feet.	
Speed1-7	float	8	This is used when the Schedule setting UseRoadDistance =0 meaning straight line rather than along road routing. This attribute is based on using occurrence one first. Once one is used, two is available for use. Occurrences two through seven must be empty if one is empty. Occurrences three through seven must be empty if two is empty, and so on. This attribute is used in conjunction with the SpeedThreshold attribute. It is used to control the speed the resource travels	

			between different SpeedThreshold distances.	
Speed1-7Threshold	float	8	This is used when the Schedule setting UseRoadDistance =0 meaning straight line rather than along road routing. This attribute is directly related to the Speed1-7 attribute and each Speed occurrence must have a corresponding Speed1-7Threshold value This attribute works in conjunction with the Speed Attribute where the speed is used for the corresponding speed threshold distance. Once the distance is reached, the second occurrence speed and distance is used.	
DynamicZoningDistance	float	8	Distance in Meters: Allows Optimizer to make appropriate geographic decisions based on the current set of orders in the system. Unlike "static" zones (which must be pre-determined in the data set using the "Zone" attributes), Dynamic Zoning allows the set of Resources/Jobs allowed for any given optimization to change over time based on the current assignment of jobs and Distance value.	
ServicePenalty	float	8	Startup Penalty for this Resource	
MinPaidOrders	smallint	2	Minimum number of orders the driver will be paid for. Multiplied by the DriverPayPerOrder value.	
RoutingParameterKey	nvarchar	50	Routing parameter key to use during optimization	
DriverID	int	4	Unique Identifier of the Driver record. Must exist in the Driver table.	
EarliestEndDay	int	4	Used for generation of resources using this template. A value greater than 0, will indicate that the EarliestEndDate for the generated resource will be the date that falls within the date range and matches its frequency value plus this field value. For example, if resources are generated for a week date range and its frequency is Monday, Wednesday and Friday, then the EarliestEndDate will be based on Monday's date plus this field	

			value, Wednesday's date plus this field value, etc.	
EarliestServiceDay	int	4	Used for generation of resources using this template. A value greater than 0, will indicate that the EarliestServiceDate for the generated resource will be the date that falls within the date range and matches its frequency value plus this field value. For example, if resources are generated for a week date range and its frequency is Monday, Wednesday and Friday, then the EarliestServiceDate will be based on Monday's date plus this field value, Wednesday's date plus this field value, etc.	
EarliestStartDay	int	4	Used for generation of resources using this template. A value greater than 0, will indicate that the EarliestStartDate for the generated resource will be the date that falls within the date range and matches its frequency value plus this field value. For example, if resources are generated for a week date range and its frequency is Monday, Wednesday and Friday, then the EarliestStartDate will be based on Monday's date plus this field value, Wednesday's date plus this field value, etc.	
EarliestEndTime	datetime	8	EarliestEndTime to set to the generated resources using this template	
EarliestServiceTime	datetime	8	EarliestServiceTime to set to the generated resources using this template	
EarliestStartTime	datetime	8	EarliestStartTime to set to the generated resources using this template	
Frequency	int	4	Bitmask used by FW to generate resources from this template. This field specifies what days of the week should be used to generate resources given an input date range.	
LatestEndTime	datetime	8	LatestEndTime to set to the generated resources using this template	
LatestEndDay	int	4	Used for generation of resources using this template. A value greater than 0, will indicate that the LatestEndDate for	

			the generated resource will be the date that falls within the date range and matches its frequency value plus this field value. For example, if resources are generated for a week date range and its frequency is Monday, Wednesday and Friday, then the LatestEndDate will be based on Monday's date plus this field value, Wednesday's date plus this field value, etc.	
LatestServiceDay	int	4	Used for generation of resources using this template. A value greater than 0, will indicate that the LatestServiceDate for the generated resource will be the date that falls within the date range and matches its frequency value plus this field value. For example, if resources are generated for a week date range and its frequency is Monday, Wednesday and Friday, then the LatestServiceDate will be based on Monday's date plus this field value, Wednesday's date plus this field value, etc.	
LatestStartDay	int	4	Used for generation of resources using this template. A value greater than 0, will indicate that the LatestStartDate for the generated resource will be the date that falls within the date range and matches its frequency value plus this field value. For example, if resources are generated for a week date range and its frequency is Monday, Wednesday and Friday, then the LatestStartDate will be based on Monday's date plus this field value, Wednesday's date plus this field value, etc.	
LatestServiceTime	datetime	8	LatestServiceTime to set to the generated resources using this template	
LatestStartTime	datetime	8	LatestStartTime to set to the generated resources using this template	
ScheduleID	int	4	Unique Identifier of the Schedule record. Must exist on the FWSchedule table.	
Prefix	nvarchar	50	Prefix utilized to in the Resource Key of the records generated using this template	

Suffix	nvarchar	50	Suffix utilized to in the Resource Key of the records generated using this template	
MaxDrivingTime	int	4	Maximum Driving Time before a break	
LastDateWorked	datetime	8	Date of the final day supplied in WorkWeekHistory.	
WorkWeekLength	int	4	Number of days in a work week (e.g. 6 or 7).	
WorkWeekLimit	int	4	Number of hours that can be worked (e.g., 70:00).	
WeeklyRestLength	int	4	Idle time required to reset the work week (e.g, 34:00).	
WorkWeekResetTime	int	4	Time of day marking the end of one 'day' and the beginning of the next (e.g., midnight, entered as 00:00).	
WorkWeekHistory1-14	int	4	Hours worked per day for each day from 1-WorkWeekLength. For any day without a history supplied zero (0) will be assumed.	
DaysFromToday	int	4	The number of days to add to today to compute the first date on which to generate resources based on this route template.	
NoDays	int	4	The number of days from the first date for which to generate resources based on this route template.	
NoResources	int	4	Number of Resources to generate when using this route template.	
AdapiDrivingTimeFactor	float	8	Multiplier that the advise will use to calculate nominal driving time.	
MaxRedispatches	int	4	Maximum number of recharges	
MaxRedispatchTime	int	4	Maximum recharge time	
TruckKey	nvarchar	50	Business key to identify the activity this Route template refers to. Must exist exist in the FWTruck table.	
TruckID	int	4	Unique Identifier of the Truck record. Must exist in the FWTruck table.	

TrailerID	int	4	Unique Identifier of the Trailer record. Must exist in the FWTrailer table.
FWResourceTemplateSetID	int	4	Internal unique database record ID.
FWResourceTemplateSetKey	nvarchar	4	Business Key identifier for this Resource Template Set Key. This key is used for the optimizer.
Reload/Threshold	int	4	Allows the optimizer to consider a recharge for this resource whenever its capacity is less than or equal to this number.
UnloadThreshold	int	4	Allows the optimizer to consider a recharge for this resource whenever its capacity is greater than or equal to this number.
RechargeToCapacity	bit	1	Tells the optimizer if the recharge should be done to capacity or not. If this value is set to zero, then the recharge amount will only be required to fulfill the planned stops in the route. 1=Recharge to Capacity, 0=Recharge
EndVehicleFullEmpty	bit	1	Allows Route Planner to schedule a recharge or unload at the end of the route when the vehicle returns to its final location. If active, then Route Planner will calculate the time required to execute this task based on the Recharge rates defined for this resource.
Published	Int		This field indicates if a Stop has been published either caused by Publishing a Route or the stop. Possible Values: Null or 0: Unpublished 1: Published 2: Changed After Published
UDFDateTime1-3	nvarchar		User-defined date/time fields
RequirementSetID	int	4	Unique internal DB Identifier for the Requirement set record.
UDFString1-9	nvarchar	500	User Defined Fields for strings 1...9

UDFInteger1-3	int	4	User Defined Fields for integers 1...9	
UDFNumber1-3	float	8	User Defined Fields for numbers 1...9	
CompleteFullBonus	float	8	This attribute works in conjunction with the FullThreshold value. The CompleteFullBonus is added to the overall route profit when all measures are equal to or greater than the FullThreshold values. If not, then this value is ignored. Default is zero (0).	
ResourceKeyWizardID	int	4	ResourceKey wizard record ID used for generating Resourcekey	
WorkUnits	float	8	The maximum workload a resource can handle during the entire route duration, no matter how many recharge occurs in it.	
StartUpTime	int	4	Allows RMPI to add a startup time at the beginning of a route. This time can be specified by resource and/or by location. If both exist, RMPI adds them together.	
WindUpTime	int	4	Allows RMPI to add a shut down time at the end of a route. This time can be specified by resource and/or by location. If both exist, RMPI adds them together.	
ReloadThreshold	float	8	Used to determine when to perform reloads	
RechargeToCapacity	bit	1	Tells the optimizer if the recharge should be done to capacity or not. If this value is set to zero, then the recharge amount will only be required to fulfill the planned stops in the route. 1=Recharge to Capacity, 0=Recharge	
EndVehicleFullEmpty	bit	1	Allows Route Planner to schedule a recharge or unload at the end of the route when the vehicle returns to its final location. If active, then Route Planner will calculate the time required to execute this task based on the Recharge rates defined for this resource.	
HazMatType	int	4	Indicate the resource can handle hazardous material	
Weight	float	8	Asset weight	

Height	float	8	Asset height	
Length	float	8	Asset length	
Width	float	8	Asset width	
SpeedCatID	int		Speed category record ID	
AllowEmptyStart	bit	1	Enables Resources to start empty and allows them to load at their home or remote location. To activate this functionality on a schedule, the AllowEmptyStart property must be set to "1".	
EndVehicleMeasuresRequired	int	4	Integer that encodes indices of measures that a resource needs to fill up before closing out the route in preparation for next day's route. MeasureN is encoded into 2N-1. For example, the requirement to fill up Measure1, Measure3, and Measure6 are represented by EndVehicleMeasuresRequired=37	
AllowRecharges	bit	1	If the resource is allowed to recharge	
CommodityTransitionFactor	float	8	Used to calculate the time required to change commodities in a reload.	
FreezeInd	bit	1	This field indicates if the route is frozen. If this field has a value of 1, then the route cannot be changed at all. It can only take status messages.	
MaxDistanceFromDepot	float	8	Maximum as-the-crow-flies distance away from initial depot of resource	
RedispatchRest	int	4	The rest time for redispatch (reserved for future development)	

FWResource

The FWResource table is used to introduce the different resources that are available for building routes. The resource table provides specific instances of a physical unit that can actually perform the work (stops – the work that needs to be done). The resource creation process uses the specified route template to set many of the default settings needed by the different optimization engines. An external resource management system, or API request to create a resource record, may have certain attributes already set which will take precedence over the route template values.

Create: Resources can be created by themselves or several at a time. If a PreferredScheduleKey is specified, then the schedule and resource link (FWRoute) record needs to be created.

A special creation process will need to be created where one or more resources can be generated from a specific template.

Delete: When deleted, all associated FWRoute records are also deleted. There may be a need to delete location records of type temporary if they have no activities or templates associated with them.

Update: Individual record. Updates certain attributes that could have an impact on the schedule. It updates the FWRoute record associated to this Resource.

Attribute Name	Type	Field Size	Description	Required
ResourceID	int	4	Unique internal DB Identifier of the Resource table.	Required
ResourceKey	nvarchar	50	Business key to identify the activity this Resource refers to. Must exist in the FWResource table.	Required
RouteTemplateID	int	4	Unique internal DB Identifier of the Route Template table.	
RouteTemplateKey	nvarchar	50	Business key to identify the Route template this Resource refers to. Must exist in the FWRouteTemplate table.	Required
InitialLocationKey	nvarchar	50	The initial location at which the route begins. ID of an entry in the FWLocation Table. The initial location attribute will determine the time zone for all the date saved on this record. If this value changes, all times will need to change to the new time zone.	Must exist on the FWLocation table.
FinalLocationKey	nvarchar	50	The final location (Station) at which the route ends. ID of an entry in the FWLocation Table.	Must exist on the FWLocation table.
RoutingParmameterKey	nvarchar	50	Business key to identify the Routing Parameter this Resource refers to.	Must exist on the FWRoutingParameter table (default =Default).

ModifiedBy	int	4	See <i>ModifiedBy</i> under <i>Generic Attributes</i> .	
ModifyDate	datetime	8	See <i>ModifyDate</i> under <i>Generic Attributes</i> .	
CreatedBy	int	4	See <i>CreatedBy</i> under <i>Generic Attributes</i> .	
CreatedDate	datetime	8	See <i>CreatedDate</i> under <i>Generic Attributes</i> .	
Comment	nvarchar	500	See <i>Comment</i> under <i>Generic Attributes</i> .	
Company	nvarchar	50	Company Name of this resource	
ScheduleKey	nvarchar	50	Business key to identify the Schedule this Resource refers to. This field is used to link a resource to a specific schedule during creation.	If entered, it must exist on the FWSchedule table.
Name	nvarchar	255	This is a descriptive field used to identify this resource in a more user friendly way. It is not used for internal processing, but typically as output only. If parameter is not zero, displays blank if name and ID are identical.	
Status	nvarchar	50	This attribute is used in the future to identify different resource status changes so that different processing can be performed based on status values, such as maintenance.	
Territory	nvarchar	50	The Preferred territory (zone). Must be validated against the FWCode table for a value territory.	
DriverKey	nvarchar	50	Must be validated against the FWCode table.	
TrailerKey	nvarchar	50	Currently used as reference but is allocated for future processing. Must be validated against the FWCode table.	
TractorKey	nvarchar	50	Currently used as reference, but is allocated for future processing. Must be validated against the FWCode table.	

EarliestStartDate	datetime	8	Earliest date and time resource can leave the base at the start of a route. Values of "ScheduleDate + X" and "Today + X" can be used to create relative dates. The default value comes from the RouteTemplateWindow EarliestDate and EarliestStartTime. Must be less than or equal to the LatestStartDate	
LatestStartDate	datetime	8	Latest date and time resource can leave the base at the start of a route. Values of "ScheduleDate + X" and "Today + X" can be used to create relative dates. The default value comes from the RouteTemplateWindow LatestDate and LatestStartTime. Must be greater than or equal to the EarliestStartDate and must be within the RouteTemplate time window range and frequency.	Required
EarliestServiceDate	datetime	8	Earliest date the resource can begin serving a job. Values of "ScheduleDate + X" and "Today + X" can be used to create relative dates. The default value comes from the RouteTemplateWindow EarliestDate and EarliestServiceTime. Must be less than or equal to the LatestServiceDate. Must be greater than LatestStartDate.	
LatestServiceDate	datetime	8	Latest date the resource can begin serving a job. Values of "ScheduleDate + X" and "Today + X" can be used to create relative dates. The default value comes from the RouteTemplateWindow LatestDate and LatestServiceTime. Must be greater than or equal to the EarliestServiceDate. Must be greater than LatestStartDate.	

EarliestEndDate	datetime	8	Earliest date and time that a resource can end its route. If the resource reaches its final depot earlier, it must stop and wait, increasing the total elapsed time. The default value comes from the RouteTemplateWindow EarliestDate and EarliestEndTime. Must be less than or equal to the LatestEndDate. Must be greater than LatestStartDate. Must be greater than LatestServiceDate.	
LatestEndDate	datetime	8	Latest date and time that the resource can return to the final depot. Values of "ScheduleDate + X" and "Today + X" can be used to create relative dates. The default value comes from the RouteTemplateWindow LatestDate and LatestEndTime. Must be greater than or equal to the EarliestEndDate. Must be greater than LatestStartDate. Must be greater than LatestServiceDate. Must be within the RouteTemplateWindow time window range and frequency.	
Measure1-9	Float	8	These values are applicable to the corresponding stop occurrence values used to manage capacity. For example: If the stop is a Drop-off type stop then this amount is reduced by the corresponding stop amounts (plus factor) to reflect additional capacity on this resource and save it in the ProjectedMeasureCapacity on the route record. If the stop is a pickup then the corresponding amount is added to reduce this capacity.	

Measure1-9Factor	Float	8	When entered, the Measures1-9 occurrence must also be entered. This value provides the ability to reduce or increase the Measure values by this factor. This attribute is a percentage value used to adjust the measure value. For example: If the factor is less than one then the measure is reduced. If it is equal to one, then the measure will stay the same; and if the factor is more than one, then the measure is more. A negative value is not allowed.	
Measure1-9RechargeRate	Float	8	When entered, the Measures1-9 occurrence must match the location supplied measure values; otherwise, recharge is ignored. The recharge rate is measured in units per second and is associated with the MeasureRechargeRate and Measure values of the location.	
MaxStops	smallint	2	This value is used to set the absolute threshold of the total number of stops that can be associated with this route. The stop types must be PICKUP (1), DROPOFF(0) or RECHARGE(4).	
Requirements	nvarchar	255	Each specified code must exist on the FWCode table for the Requirement type. Multiple entries are possible and delimited by a comma or space. See route template for details.	
FullThreshold	float	8	Used as a percentage of the capacity (measure amount) to apply the CompleteFullBonus and PartialFullPenalty values. Please see complete full bonus and partial full penalty for details. A value of 1.0 will require all measures to be at capacity before the Complete Full Bonus is applied and to avoid incurring the Partial Full Penalty. A value of 0.5 will require at least 50% capacity utilization on all measure to	

			receive the Complete Full Bonus. If one of the measures is less than 50%, then the Partial Full Penalty is applied. A value of 0.0 (default) indicates that the bonus and penalty values is ignored. Percentage of full capacity that is acceptable for applying CompleteFullBonus and PartialFullPenalty.	
CompletedFullBonus	float	8	This attribute works in conjunction with the FullThreshold value. The CompleteFullBonus is added to the overall route profit when all measures are equal to or greater than the FullThreshold values. If not, then this value is ignored. Bonus for a route that has all measures filled above its FullThreshold.	
PartialFullPenalty	float	8	This attribute works in conjunction with the FullThreshold value. The PartialFullPenalty is subtracted from the overall route profit when one or more measures are less than the FullThreshold values. If not, then this value is ignored. Penalty applied to a route that has one or more measures filled more than the FullThreshold percentage, and one or more measures filled less than the FullThreshold percentage.	
Profit	float	8	This attribute is a set base profit value. It is cumulative where the template value is added to the Resource value whenever a route is created from this resource. Tells the Optimizer how important it is to schedule the job or item.	
SameStopBonus	float	8	This value is added to the overall profit of the route for every additional stop at a specific location. For example: If we have two stops at the same location, then the second stop will have the bonus applied once. If a third	

			stop was also served at the same stop, then the bonus is applied twice, once for the second stop and again for the third stop. If we have two different locations with two serviceable stops at each location (type pickup or drop-off), then the bonus is applied twice, once for the second stop of the first pair and again for the second stop of the second pair. And so on. Bonus for a stop that has all measures filled above its FullThreshold.	
DiscountInitialLeg	float	8	This value is used to discount the penalty of the initial leg of the route. It is a percent value used to reduce the cost of that leg. For example: a value of 1.0 will discount the full penalty of this leg and no penalty is incurred, a value of .6 will discount 60% of the penalty and apply 40% of the penalty. A value of .15 will discount 15% and apply 85% of the penalty.	
DiscountFinalLeg	float	8	This value is used to discount the penalty of the final leg of the route. See the discount initial leg for details.	
PenaltyPerMeter	float	8	This value is used to apply a penalty for every meter driven. Previous setting was in miles, so care should be taken when updating or using this value. Straight-line distance is used for this calculation. This amount is multiplied by the distance and deducted from the overall route profit.	
PenaltyPerOrder	float	8	This value is used to apply a penalty for every order (type Pickup or Drop-off) serviced. Total penalty is multiplied by this value and the number of orders serviced. It is deducted from the overall route	

			profit. If less than the MinPaidOrders have been assigned to this route, then the MinPaidOrders is used as the multiplier. See the MinPaidOrders for details.	
MinPaidOrders	smallint	2	This value is used to set a minimum threshold of the number of orders that the PenaltyPerOrder is applied against. For example: if the number of orders serviced is three, and this value is set to 5, then the PenaltyPerOrder is applied against the minimum setting. If seven orders were serviced, then the penalty is applied against the actual value serviced and this setting is ignored.	
MissedWindowPenalty	float	8	Per minute penalty for a missed time window. With values greater than zero, the Optimizer tends to keep unserved jobs close to their time windows instead of placing them geographically.	
LatestEndPenalty	float	8	This value is used to penalize a route that missed its LatestEndDate time. Regardless of how much the time window was missed, the same flat value is used. Should be set to a value roughly equal to the profit value for a job.	
WaitTimePenalty	float	8	This value is applied for every second that the resource waits to service the order. It is applied against every second of the route. WaitTime values are used.	
Work1-7Penalty	float	8	This attribute is based on using occurrence one first, then two, three, and so on. It is not possible to skip any occurrences. Occurrences two through seven must be empty if one is empty. Occurrences three through seven must be empty if two is empty and so on.	

			This value is used in conjunction with the WorkDuration attribute where they must be applied in pairs. If one is entered, the other is required. These values are used to either increase or decrease the penalty as the route executes over time at different rates. This value is applied for every second duration. For example: if the WorkDuration is 7,200 seconds and the penalty value is 1.00, this route will insure a penalty of 7,200 points. If the second occurrence duration is 3600 and the penalty is 1.5, then the total penalty is 7,200 points from first occurrence and 5,400 points for the second occurrence and so on. Please note that the duration must be in seconds.	
Work1-7Duration	int	4	This attribute is directly related to the Work1-7Penalty attribute and each Work1-7Penalty occurrence must have a corresponding Work1-7Duration value and must be in seconds. These values are used to either increase or decrease the penalty as the route executes over time at different rates. See Work penalty for more details.	
BreakDuration	int	4	This attribute specifies the length of the break to be taken. If this attribute is empty the following must also be empty: • MaxElapsedTimeBeforeBreak • MaxDriveTimeBeforeBreak • MaxNumberOfBreaks	
MaxElapsedTimeBeforeBreak	int	4	If entered, the BreakDuration must also be greater than 0. This attribute indicates how much time must pass before a break can be scheduled. Time is measured in seconds.	

			If BreakDuration was entered, then this attribute must be greater than 0.	
MaxDrivingTimeBeforeBreak	int	4	If entered, the BreakDuration must also be greater than 0. This attribute indicates how much driving time must pass before a break can be scheduled. Time is measured in seconds. If BreakDuration was entered, then this attribute must be greater than 0.	
MaxNumberOfBreaks	tinyint	1	If BreakDuration was entered, then this attribute must be greater than 0. This attribute indicates the maximum number of breaks allowed for a given work unit. A work unit is dependent on whether rests are being used. If rests are not being used, then the number of breaks is allowed for the duration of the route. If rests are specified, then this number of breaks is allowed between every rest. For example: When a rest occurs, the number of breaks taken is reset and the count starts again. The default value is zero, which represents an unlimited number of breaks. If BreakDuration was entered, then this attribute must be greater than 0.	
BreakTimeOnDutyInd	bit	1	Determines if time spent on break is counted as elapsed time between rests.	
RestDuration	int	4	This attribute specifies the length of the rest to be taken. If this attribute is empty the following must also be empty: - MaxElapsedTimeBeforeRest - MaxDrivingTimeBeforeRest	

			FirstStopTimeThreshold	
MaxElapsedTimeBeforeRest	int	4	This attribute indicates how much time must pass before a break can be scheduled. Time is measured in seconds. If entered, then RestDuration must be greater than 0.	
MaxDrivingTimeBeforeRest	int	4	This attribute indicates how much driving time must pass before a break can be scheduled. Time is measured in seconds. If entered, then RestDuration must be greater than 0.	
FirstStopTimeThreshold	float	8	If entered, then RestDuration must be greater than 0. This attribute is very similar to the Early Rest Time Threshold attribute in that its value is applied against the maximum elapsed or driving time settings with the exception that it is only considered for the first stop on the route. This attribute allows users to have the first stop treated differently from other stops (e.g., only allow early rests to be scheduled for the first stop). The First Stop Time Threshold attribute is only evaluated if the Allow Early Rests setting is true. It is also represented as a percentage of the maximum elapsed or drive time. Note— Both the First Stop Time Threshold and the Early Rest Wait Threshold must be met for an early rest to be taken.	
UseFullWaitAsRestInd	bit	1	This attribute determines how the recuperation time is considered. If enabled, the entire recuperation time is considered to be off the clock and the clock is reset when work begins again. If disabled, then the	

			clock will restart as soon as the recuperation time has elapsed. This is only significant if the idle time before a stop is larger than the Recuperation Time attribute. For example: <i>If Use Full Wait As Rest were false, the idle time before a particular stop were eight (8) hours, and the Recuperation Time for the resource were six (6) hours, then the clock would restart two (2) hours before the idle time were completed (when work at the stop begins).</i>	
AllowEarlyRest	bit	1	If set to true the following must also be greater than 0: • EarlyRestTimeThreshold • EarlyRestWaitThreshold This attribute indicates whether early rests are allowed. If yes, then rests can be scheduled before the maximum elapsed or drive time threshold values have been reached. If no, then the maximum elapsed or drive threshold values must be reached before a rest can be scheduled.	
EarlyRestTimeThreshold	float	8	If greater than 0, then the AllowEarlyRests must be set to true. This attribute indicates how much of the maximum elapsed or drive time threshold must be used before an early rest can be scheduled. The early rest time threshold is only evaluated when the Allow Early Rest indicator is enabled. The Early Rest Time Threshold value is represented as a percentage of the maximum elapsed or drive time. For example: <i>if the Maximum Elapsed Time Before Rest attribute is set to four (4) hours and the Early Rest Time Threshold is set to .50 (50%), then the earliest that the rest</i>	

			can be scheduled would be two (2) hours into the route. Note— Both the Early Rest Time Threshold and the Early Rest Wait Threshold must be met for an early rest to be taken.	
EarlyRestWaitThreshold	float	8	If greater than 0, then the AllowEarlyRests must be set to True. This attribute indicates how much of the recuperation time must be available to schedule an early rest. It is represented as a percentage of the total recuperation time. In other words, the Early Rest Wait Threshold attribute works in conjunction with the Recuperation Time attribute. For example: if the Recuperation Time is eight (8) hours and the Early Rest Wait Threshold is set to .75 (75%) then an early rest can be taken only if there is a period of idle time that is greater than six (6) hours, 75% of the total recuperation time. The early rest wait threshold is only evaluated when the Allow Early Rest indicator is enabled. Note— Depending on the stop in question, both the Early Rest Time Threshold or First Stop Time Threshold and the Early Rest Wait Threshold must be met for an early rest to be taken.	
OpportunisticEarlyRest	bit	1	Specifies whether or not an early rest should be taken whenever the wait time to serve a job is larger than the Resource's RecuperationTime. If false, early rests are entirely controlled by FirstStopTimeThreshold, EarlyRestTimeThreshold, and EarlyRestWaitThreshold.	

RechargeDuration	int	4	This attribute is used to set a consistent recharge duration in seconds for this resource.	
MaxDwellDuration	int	4	Restricts the "dwell" time for a route, which is the total elapsed time from the beginning of service for the first job (initial drive and parking time are ignored) through the final return to depot. A value of 00:00 places no restriction on the dwell time.	
MaxElapsedDuration	int	4	Maximum elapsed time for the entire route. Elapsed time is measured from start of route through the return to final depot.	
InitialServiceDuration	int	4	This attribute is used to set an initial service duration for the first job for the same customer at the same location. It is added to the standard service duration once for each grouping of same stop orders.	
ServiceDuration	float	8	Service time "overhead" for the customer or depot. Added to the customer's first job. Useful in operations such as couriers, where several jobs may be performed in the same amount of time as a single job. This attribute is used to set a consistent service duration for this resource.	
PreServiceDuration	int	4	This integer value (in seconds) is added to the post-service duration of the order, order lines, and location post duration when the activity is created. See the service duration attribute for more details.	
ServiceSpeedFactor	float	8	This attribute can be used to adjust and control how quickly a resource serves stops. It is used to calculate how long it takes a resource to service a stop by dividing the stop's total service duration (including pre and post values) by the resource's ServiceSpeedFactor value. A value	

			less than 1.0 increases the time the resource requires to service stops, and a value greater than 1.0 decreases the time. For example: If stop X has a total service duration of 100 seconds. - A resource with a ServiceSpeedFactor of 1.0 needs 100 seconds to serve it ($100/1.0 = 100$). - A resource with a ServiceSpeedFactor of .50 needs 200 seconds to serve it ($100/0.5 = 200$). - A resource with a ServiceSpeedFactor of 1.5 needs 67 seconds to serve it ($100/1.5 = 67$).	
SpeedFactor	float	8	Used to offset the RoutingParameter speeds. Changes travel times without reconfiguring the RoutingParameter attributes or the speed attributes. Numbers greater than one create faster travel times; numbers less than one create slower travel times. The calculation is: adjusted time = calculated time multiplied by speed factor. For example: calculated time = 100, Speed factor = .75, resulting in an adjusted time of 75 ($100 \times .75 = 75$).	
MaxDistance	float	8	This attribute restricts the distance that this resource can travel in meters, not miles.	
MinTravelDistance	float	8	This attribute is used to determine the distance at which two stops are considered separate stops. If the distance between two sequential stops is less than this value, it is assumed that the vehicle parks once to service both stops. Please note that the unit is meters not feet.	
Speed1-7	float	8	This attribute is directly related to the Speed1-7 attribute and each	

			Speed occurrence must have a corresponding Speed1-7Threshold value This attribute works in conjunction with the Speed Attribute where the speed is used for the corresponding speed threshold distance. Once the distance is reached, the second occurrence speed and distance is used.	
Speed1-7Threshold	float	8	This attribute is directly related to the Speed1-7 attribute and each Speed occurrence must have a corresponding Speed1-7Threshold value.	
DynamicZoningDistance	float	8	Distance in Meters: Allows Optimizer to make appropriate geographic decisions based on the current set of orders in the system. Unlike "static" zones (which must be pre-determined in the data set using the "Zone" attributes), Dynamic Zoning allows the set of Resources/Jobs allowed for any given optimization to change over time based on the current assignment of jobs and Distance value.	
RequirementSetID	int	4	The Requirement Set ID taken from the FWRequirementSet table.	
ServicePenalty	float	8	This attribute is used to set a base penalty value that is issued to deploy this resource. It is deducted from the overall route profit. Default is zero (0).	
UDFString1-9	nvarchar	500	User Defined Fields for Strings 1...9	
UDFInteger1-3	int	4	User Defined Fields for integers 1..3	
UDFNumber1-3	float	8	User Defined fields for numbers 1...3	
Criteria	nvarchar	512	This attribute is used to convey selection criteria from the Route Planner UI or other external applications.	

ScheduleID	int	4	Unique Identifier of the Schedule record. Must exist on the FWSchedule table.	
MaxDrivingTime	int	4	Maximum Driving Time before a break.	
AdapiDrivingTimeFactor	float	8	Multiplier that the advise will use to calculate nominal driving time.	
LastDateWorked	datetime	8	Date of the final day supplied in WorkWeekHistory.	
WorkWeekLength	int	4	Number of days in a work week (e.g., 6 or 7).	
WorkWeekLimit	int	4	Number of hours that can be worked (e.g., 70:00).	
WeeklyRestLength	int	4	Idle time required to reset the work week (e.g., 34:00).	
WorkWeekResetTime	int	4	Time of day marking the end of one 'day' and the beginning of the next (e.g., midnight, entered as 00:00).	
WorkWeekHistory1-14	int	4	Hours worked per day for each day from 1-WorkWeekLength. For any day without a history supplied zero (0) will be assumed.	
MasterRoute	int	4	Indicates if this route is a Master Route. Valid values are: 0 – Normal Route 1 – Master Route	
MaxRedispatches	int	4	Maximum number of redispatches that can occur during a route.	
MaxRedispatchTime	int	4	Maximum elapsed time that can pass between visits to the depot.	
DriverID	int	4	Unique Identifier of the Driver record. Must exist on the FWDriver table.	
TruckID	int	4	Unique Identifier of the Truck record. Must exist on the FWTruck table.	
TrailerID	int	4	Unique Identifier of the Trailer record. Must exist on the FWTrailer table.	

MasterFrequency	int	4	This field specifies the day of the week to use when generating routes from this master route record. For example, this master route may apply to Monday/Wednesdays and Fridays
UserMatchingField	nvarchar	50	This field is used to generate routes from master routes and use this field name to match the orders that come with this field. If no value is specified in this record, then the value will come from the MasterOrders assigned to this Master Route.
UserMatchingFieldValue	nvarchar	512	This field is used when generating routes from this master route. This field value will be used as an extra criterion to match real orders to this route.
TruckKey	nvarchar	50	Business key to identify the Truck this Resource refers to. Must exist in the FWTruck table.
AllowEmptyStart	bit	1	Enables Resources to start empty and allows them to load at their home or remote location. To activate this functionality on a schedule, the AllowEmptyStart property must be set to "1".
StartupTime	nvarchar		Allows RMPI to add a startup time at the beginning of a route. This time can be specified by resource and/or by location. If both exist, RMPI adds them together.
WindupTime	nvarchar		Allows RMPI to add a shut down time at the end of a route. This time can be specified by resource and/or by location. If both exist, RMPI adds them together.
DynamicZoningIncludeDepot			Determines whether the Initial or Final Depot should be included in the dynamic zoning distance test. The attribute can have one of the following values: 0: do not include initial or final depot in dynamic zoning (this

			was the behavior in previous versions) • 1: include only the initial depot in dynamic zoning • 2: include only the final depot in dynamic zoning • 3: include the initial and final depots in dynamic zoning This setting is defaulted to '0'.	
UDFDateTime1-3	datetime	8	User-defined date/time fields	
OrigMasterReskey	nvarchar	50	The resource key of the master route which a resource was created from. If a resource was not created from a master route, this value will be null	
RteGeoFencingThreshold	float	8	The route geofencing threshold	
StopGeoFencingThreshold	float	8	The stop geofencing threshold	
StartDate	datetime		Date and time resource can leave the base at the start of a route.	
UDFURL1-3	nvarchar	500	User-defined URL fields	
CompleteFullBonus	float	8	This attribute works in conjunction with the FullThreshold value. The CompleteFullBonus is added to the overall route profit when all measures are equal to or greater than the FullThreshold values. If not, then this value is ignored. Default is zero (0).	
WorkUnits	float	8	The maximum workload a resource can handle during the entire route duration, no matter how many recharge occurs in it.	
ReloadThreshold	float	8	Used to determine when to perform reloads	
RechargeToCapacity	bit	1	Tells the optimizer if the recharge should be done to capacity or not. If this value is set to zero, then the recharge amount will only be required to fulfill the planned stops in the route. 1=Recharge to Capacity, 0=Recharge	

EndVehicleMeasuresRequired	int	4	Integer that encodes indices of measures that a resource needs to fill up before closing out the route in preparation for next day's route. MeasureN is encoded into 2N-1. For example, the requirement to fill up Measure1, Measure3, and Measure6 are represented by EndVehicleMeasuresRequired=37	
AllowRecharges	bit	1	Resource is allowed to recharge	
CommodityTransitionFactor	float	8	Used to calculate the time required to change commodities in a reload.	
TimeZone	nvarchar	100	Time zone of the resource	
UDFDateTime1-3	datetime	8	User-defined date/time fields	
MaxDistanceFromDepot	float	8	Maximum as-the-crow-flies distance away from initial depot of resource	
RedispatchRest	int	4	The rest time for redispatch (reserved for future development)	
RoutePositionInd	bit	1	RMPI has been configured to support this setting through the UseRoutePosition attribute. RMPI handles this new setting as follows: • If the setting is disabled at the schedule level, it will not be used by RMPI • If the setting is enabled at the schedule level, it will be used based on the value of resource setting • If the resource setting is null, RMPI will use the schedule setting • If the setting is enabled at the resource level, RMPI will use the resource setting • If the value of the setting at the resource level is 0, RMPI will not apply the UseRoutePosition setting to the particular resource	

AlwaysHazardous	int	1	When enabled, the resource will always be routed as if carrying hazardous materials. When disabled, the system will evaluate the onboard orders for hazardous materials.
MinInterstopTravelTime			This field allows Descartes Route Planner to handle "Interstop" service time values from Descartes Route Planner RS.
RechargePolicy	int	4	Select one of the following options from the drop-down menu to control recharges at depots: • Recharge at any depot • Recharge only at initial depot • Recharge only at final depot • Recharge at either initial or final depot

FWRoute

The FWRoute table is used to link a specific schedule with a specific resource. All the Resource attributes are duplicated on each FWRoute record so that different settings can be used to build different routes within different schedules. A FWRoute record cannot exist without a schedule or resource record. In addition, there are other required table attributes and they are specified below.

Create: The route record can be created in several ways:

- The resource creation process (based on the preferred schedule key)
- The route copy process (user selection)
- The route move process (user selection)
- The route new process (user input)



Note—Please see the status updates when a route is created.



Note—The Route is not automatically updated when the related resource or schedule is updated.

Delete: Delete selected record(s) – see resource for more delete details.

Update: Update as needed.

Reset: This function can be used to rebuild the selected route based on the current resource and schedule records.

Attribute Name	Type	Field Size	Description	Required
RouteID	int	4	Unique internal DB Identifier of the Route record.	Required
ScheduleID	int	4	Unique internal DB Identifier of the Schedule record. Must exist on the FWSchedule table.	Required
ResourceID	int	4	Unique internal DB Identifier of the Resource record. Must exist on the FWResource table.	Required
ResourceKey	nvarchar	50	Business key to identify the Resource this Route refers to. Must exist on the FWResource table.	Required
ScheduleKey	nvarchar	50	Business key to identify the Schedule this Route refers to. Must exist on the FWSchedule table.	Required
ModifiedBy	int	4	See <u>ModifiedBy</u> under <u>Generic Attributes</u> .	
ModifyDate	datetime	8	See <u>ModifyDate</u> under <u>Generic Attributes</u> .	
CreatedBy	int	4	See <u>CreatedBy</u> under <u>Generic Attributes</u> .	
CreatedDate	datetime	8	See <u>CreatedDate</u> under <u>Generic Attributes</u> .	
RoutingParmameterKey	nvarchar	50	Business key to identify the Routing parameter this Route refers to. Must exist on the FWRoutingParameter table (default =Default).	
Type	tinyint	1	This field is used to identify different order types that may require different processing. Valid Values: • 0 = Single day • 1 = Multi-day • 3 = Rechargeable • 4 = Service • 5 = ...	
Status	tinyint	1	Default to the state of the Schedule. Valid Values:	Required

			• 0 = Open • 1 = Closed • 2 = Reservation Open • 3 = Reservation Closed • 4 = Planning Open • 5 = Planning Closed • 6 = Dispatch Open • 7 = Dispatch Released • 8 = Dispatch En Route • 9 = Dispatch Exception • 10 = Dispatch Closed	
ProjectedTerritory	nvarchar	50	A territory assigned to an order by Rmpi.	
ProjectedViolation	int	4	System assigned - display only value. Please note this value will need to be rendered before it can be displayed.	
ProjectedProfit	int	4	System assigned - display only value. Please note this is the output profit value (calculated by the optimizer) and not the input profit value.	
ProjectedDistance	float	8	System assigned. Note— System Assigned. Same as the final stop ProjectedDistance.	
ProjectedOnBoard(1-9)	float	8	System assigned. Note— This value is determined from the stop measure amounts.	
ProjectedElapsedTime	int	4	System assigned - Elapsed time in the route and considers durations.	
ProjectedSlackTime	int	4	System assigned - Estimated Slack Time on the Route.	
ProjectedTimeDriven	int	4	System assigned - Estimated Driving Time.	
ProjectedWorkTime	int	4	System assigned - Considers only duration fields in the Route.	
LastReportedDate	datetime	8	This field will contain the Date and Time of the last reported status.	
LastReportedLatitude	float	8	Only specified when different from any associated stops. This field contains the Last Reported Latitude for this route.	

LastReportedLongitude	float	8	Only specified when different from any associated stops. This field contains the Last Reported Latitude for this route.	
LastReportedStatus	int	4	Same as the stop status and contains the last Status reported to any of the Stops of this Route.	
NumberOfStops	int	4	Total number of stops	
ProjectedBreaks	tinyint	2	Projected Number of Breaks calculated by the optimization engine	
ProjectedRests	int	4	Projected number of rests calculated by the optimization engine	
ProjectedDepartedDate	datetime	8	Projected Departed Date	
ProjectedArrivedDate	datetime	8	Projected Arrived Date	
PlannedDepartedDate	datetime	8	Keeps the original value of the ProjectedDepartedddDate. Once the route enters into an execution state (first GPS or Status msg is received) this field will remain unchanged.	
PlannedArrivedDate	datetime	8	Keeps the original value of the ProjectedArrivedDate. Once the route enters into an execution state (first GPS or Status msg is received) this field will remain unchanged.	
ActualDepartedDate	datetime	8	Stores the Actual Departed Date reported by the status message	
ActualArrivedDate	datetime	8	Stores the Actual Arrived Date reported by the status message	
ProjectedBreakTime	int	4	Projected Break Time calculated by the optimization engine	
ProjectedRestTime	int	4	Projected Rest Time	
ProjectedWaitTime	int	4	Projected Wait Time.	
LastReportedStopID	int	4	Database ID of the last reported Stop.	
ModifyCount	int	4	Everytime a route record is updated ModifyCount is increased by 1. The Route Planner backend uses it for concurrency control among competing transactions.	

PublishRoute	int	4	Field to indicate the publishing status of this Route. 0 – Has not been published 1 – Published 2 – Changed after published	
DriverID	int	4	Unique Identifier of the Driver record. Must exist on the FWDriver table.	
ProjectedCompletedDate	datetime	8	Projected Completed Date	
ProjectedBegunDate	datetime	8	Projected Begun Date.	
MasterRoute	int	4	Indicates if this route is a Master Route. Valid values are: 0 – Normal Route 1 – Master Route	Required
AVLStatus	int	4	Field reserved for AVL generated status codes.	
FreezeInd	bit	1	This fields indicates if the route is frozen. IF this field has a value of 1, then the route cannot be changed at all. It can only take status messages.	
PlannedNumberOfStops	int	4	Keeps the original value of the Number of Stops. Once the route enters into an execution state (first GPS or Status msg is received) this field will remain unchanged.	
PlannedTimeDriven	int	4	Keeps the original value of the ProjectedTimeDriven. Once the route enters into an execution state (first GPS or Status msg is received) this field will remain unchanged.	
PlannedDistance	int	4	Keeps the original value of the ProjectedDistance. Once the route enters into an execution state (first GPS or Status msg is received) this field will remain unchanged.	
StartOdometer	int	4	Start odometer reading. This field can be set by GPS or Status messages.	
LastReportedOdometer	int	4	This field will be reported in the last gps or status message received.	
PlannedElapsedTime	int	4	Keeps the original value of the ProjectedElapsedTime. Once the route	

			enters into an execution state (first GPS or Status msg is received) this field will remain unchanged.	
RemainingNoOfStops	int	4	This field will be updated based on the reporting of completed stops.	
RemainingDropOffStops	int	4	This field will be updated based on the reporting of completed stops of type Drop off.	
RemainingPUStops	int	4	This field will be updated based on the reporting of completed stops of type Pickup.	
ActualTravelTime	int	4	Actual Travel Time calculated using GPS and Status messages	
DeltaTravelTime	int	4	Route (actual minus planned) drive time in minutes. Note— Delta is calculated so that positive values indicate that the actual time exceeded the planned time, i.e. negative values are good.	
PerfTravelTime	float	8	Route (planned divided by actual) drive time as a percent. Note— The percent is calculated so that a value greater than 100% indicates the actual time exceeded the planned time, i.e. values less than 100% are good.	
ActualServiceDuration	int	4	Route current cumulative service time in minutes.	
DeltaServiceDuration	int	4	Route (actual minus planned) service time in minutes.	
PerfServiceDuration	float	8	Route (planned divided by actual) service time as a percent.	
ActualDistance	int	4	Route current cumulative distance traveled miles or kilometers (km).	
DeltaDistance	int	4	Route (actual minus planned) distance in miles or km.	
PerfDistance	float	8	Route (planned divided by actual) distance as a percent.	

ActualElapsedTime	int	4	Route current cumulative total time in minutes.	
DeltaElapsedTime	int	4	Route (actual minus planned) total time in minutes.	
PerfElapsedTime	float	8	Route (planned divided by actual) total time as a percent.	
PlannedWorkTime	int	4	Keeps the original value of the ProjectedWorkTime Once the route enters into an execution state (first GPS or Status msg is received) this field will remain unchanged.	
ProfileCost	float	8	Total cost of this route using Profile Costs	
Projected3rdPartyDistance	float	8	Stores the distance calculated using a 3 rd party distance calculator (not rmpi)	
InTransitInd	bit	1	Indicate the vehicle is in transit mode (not at a customer site or depot)	
EndOdometer	float	8	Odometer reading at the end of a route	
ProjectedWorkUnits	float	8	Projected work units	
StartDate	datetime	8	Start date of the route	
EndDate	datetime	8	End date of the route	
wGLNSyncInd	bit	1	Can be set to true or false depending on the response from wGLN. 0 – Failure. wGLN does not have the latest changes. 1 – wGLN is in sync with Route Planner.	
wGLNResponse	nvarchar	512	This field holds the response from wGLN	
PlannedProfCost	float	8	Planned profile cost	
LastReportedTimeZone	nvarchar	8	The time zone of last reported location	
LastReportedDateUTC	datetime	8	This field will contain the Date and Time of the last reported status. (Coordinated Universal Time)	

ActualArrivedDateUTC	datetime	8	Stores the Actual Arrived Date reported by the status message (Coordinated Universal Time)	
ActualDepartedUTC	datetime	8	Stores the Actual Departed Date reported by the status message. (Coordinated Universal Time)	
PlannedArrivedDateUTC	datetime	8	Keeps the original value of the ProjectedArrivedDate. Once the route enters into an execution state (first GPS or Status msg is received) this field will remain unchanged. (Coordinated Universal Time)	
PlannedDepartedDateUTC	datetime	8	Keeps the original value of the ProjectedDepartedDate. Once the route enters into an execution state (first GPS or Status msg is received) this field will remain unchanged. (Coordinated Universal Time)	
ProjectedArrivedDateUTC	datetime	8	Projected Arrived Date (Coordinated Universal Time)	
ProjectedDepartedDateUTC	datetime	8	Projected Departed Date (Coordinated Universal Time)	
StartDateUTC	datetime	8	Start date of the route (Coordinated Universal Time)	
EndDateUTC	datetime	8	End date of the route (Coordinated Universal Time)	
LockInd	bit	1	A Boolean value indicating whether a route was locked or not. When a route is locked, the preferred resource and the route position of the assigned orders are set to the current resource and their current stop number.	
OnBoardMeasure1-9	float	8	Sum of OnBoard Capacity for all routes.	
LastPublishedDate	datetime	8	The date/time at which the route was last published.	
IdleTime	float	8	Idle time at the route level is the sum of all stops' idle time, including initial and final stations.	

FirstIgnition/LastIgnition	datetime	8	Displays the time recorded between the First Ignition=true value reported after the Start of Route and the Last Ignition=false value reported before the End of Route status messages.	

FWRouteHist

Attribute Name	Type	Field Size	Description	Required
RouteHistID	int	4	Unique internal DB Identifier of the Route History record.	Required
RouteID	int	4	Unique internal DB Identifier of the Route record. Must exist on the FWRoute table.	Required
StopID	int	4	Unique internal DB Identifier of the Stop record. Must exist on the FWStop table.	
Latitude	float	8	Reported GPS Latitude. In conjunction with Longitude attribute, provides geographic position of address. For use with non-geocodable addresses, or in long haul situations where the map does not support geocoding.	
Longitude	float	8	Reported GPS Longitude In conjunction with Latitude attribute, provides geographic position of address. For use with non-geocodable addresses, or in long haul situations where the map does not support geocoding.	
MessageDate	datetime	8	Time stamp of message	
Status	tinyint	1	Status reported	
ActualMeasure1-9	float	8	Actual Measure 1..9 reported in status messages	
TrailerKey	nvarchar	50	Trailer Key reported in Status message	
TractorKey	nvarchar	50	Tractor key reported in status message	

DriverKey	nvarchar	50	Driver Key reported in Status message	
Comment	nvarchar	500	See <i>Comment</i> under <i>Generic Attributes</i> .	
UDFString1-9	nvarchar	500	UDF String 1..9 reported in the Status message	
UDFInteger1-3	int	4	UDF Integer 1..3 reported in the status message	
UDFNumber1-3	float	8	UDF Number 1..3 reported in the status message	
ModifiedBy	int	4	See <i>ModifiedBy</i> under <i>Generic Attributes</i> .	
ModifyDate	datetime	8	See <i>ModifyDate</i> under <i>Generic Attributes</i> .	
CreatedDate	datetime	8	See <i>CreatedDate</i> under <i>Generic Attributes</i> .	
OrderLineID	Int	4	Foreign Key. Links this record with an order line	
ActualQuantity	Float	8	This fields will hold the actual quantity reported by mobile devices	
Source	Int	4	Source of the status message. Either UI or wireless devices	
ProductTypeKey	Nvarchar	50	Product Type Key for which the status is reported	
AlertID	Int	4	Internal ID to link this record with an alert ID when custom alerts are reported during actuals	
TruckKey	Nvarchar	50	This field holds the TruckKey used to report this status update	
ActualTotalMeasure1-9	float	8	Actual total measure reported	
ScannedIndActualMeasure1-9	bit	1	Indicate that actual measure is scanned in	
ScannedIndDriverKey	bit	1	Indicate the driverKey is scanned in	
ScannedIndTruckKey	bit	1	Indicate the truckKey is scanned in	
ScannedIndTrailerKey	bit	1	Indicate the TrailerKey is scanned in	
ScannedTractorKey	bit	1	Indicate the TractorKey is scanned in	

ScannedIndUDFString1-9	bit	1	Indicate the UDFString is scanned in	
ScannedIndUDFInteger1-3	bit	1	Indicate the UDFInteger is scanned in	
ScannedIndUDFNumber1-3	bit	1	Indicate the UdfNumber is scanned in	
ScannedIndActualQuantity	Bit	1	Indicate the Actual Quantity is scanned in	
ScannedIndProductTypeKey	Bit	1	Indicate the ProductTypeKey is scanned in	
ScannedProductKey	Bit	1	Indicate the ProductKey is scanned in	
ScannedIndOrderKey	bit	1	Indicate the Orderkey is scanned in	
ScannedIndLocationKey	Bit	1	Indicate the LocationKey is scanned in	
StartDate	datetime		Start time for the delay	
EndDate	datetime		End time for the delay	
MsgID	Int	4	Associated message record ID	
LastReportedTimeZone	nvarchar	100	Time zone of last reported time	
MessageDateUTC	datetime	8	Time stamp of message (Coordinated Universal Time)	
StatusDesc	nvarchar	50		
WirelessSource	nvarchar	20	Allows wGLN to detect whether the submitted status information was sent by driver, black box or ASU	

FWSchedule

The FWSchedule table is used to define different schedules with different optimization settings.

Create: A schedule can be created without any other table. It can be a stand-alone table. A schedule can be created in several ways:

- The system ships with one schedule predefined (DefaultSchedule)
- Create new schedule (user input)
- Copy an existing schedule

Delete: When a schedule is deleted, all associated routes and stops must be deleted as well. In addition, all associated buckets and schedule windows must also be deleted.

Update: Update individual records via UI or one or more records via the XML API.

Reset: A schedule can be reset to reflect the schedule template values.

➡ **Note**—Changes to the schedule will not automatically update all associated routes and stops. Individual route and stop reset functions can be called, or application optimization calls will need to be called to reflect the new schedule changes.

Default setting: For the most part, the Schedule Template is used to determine most of the schedule default values. Once a schedule is created, all subsequent schedule template changes are ignored.

Attribute Name	Type	Field Size	Description	Required
ScheduleID	int	4	Unique Identifier of the Schedule record.	Required
ScheduleKey	nvarchar	50	Business key to identify the activity this Schedule refers to.	Required
ScheduleTemplateKey	nvarchar	50	Business key to identify the Schedule template this Schedule refers to.	Required
RoutingParameterKey	nvarchar	50	Business key to identify the Routing parameter this Schedule refers to. Must exist on the FWRoutingParm table.	
ModifiedBy	int	4	See <i>ModifiedBy</i> under <i>Generic Attributes</i> .	
ModifyDate	datetime	8	See <i>ModifyDate</i> under <i>Generic Attributes</i> .	
CreatedBy	int	4	See <i>CreatedBy</i> under <i>Generic Attributes</i> .	
CreatedDate	datetime	8	See <i>CreatedDate</i> under <i>Generic Attributes</i> .	
Comment	nvarchar	500	See <i>Comment</i> under <i>Generic Attributes</i> .	
Name	nvarchar	50	Schedule Name	

Type	tinyint	1	Schedule Type. Values: 1 = Reservations 2 = Planning 3 = Execution (important for AVL and execution type of schedules. If type 3, then GPS and Status messages will apply to schedules of this type only) 4 = Simulation	
State	tinyint	1	Valid Values: (default = active when created - Inactive when copied) 1 = Active – An active schedule could have active, test, and cancelled routes and stops. 2 = Test – A test schedule cannot have any active routes or stops. This is strictly used for temporary optimizations.	Required
Status	tinyint	1	The value in the Schedule template record can be copied over here unless it's set explicitly by a user. Valid Values: 0 = Open 1 = Closed 2 = Reservation Open 3 = Reservation Closed 4 = Planning Open 5 = Planning Closed 6 = Dispatch Open 7 = Dispatch Closed	
ScheduleStartDate	datetime	8	The format of the date is yy:mm:dd:hh:mm:ss. This date is used to restrict any routes and stops that are not after the schedule start date.	Required
ScheduleEndDate	datetime	8	When specified must be in yy:mm:dd:hh:mm:ss. This date is used to restrict the inclusion of all routes and stops if their end dates are not within the schedule date. If not specified, all future routes and stops is accepted.	

ConsolidateStops	smallint	2	This attribute provides the ability to check time windows for the first stop only when there are multiple stops at the same location with identical time windows. If the first stop meets its time windows, then all of the subsequent stops will also meet their time windows. Note— Only stops with identical time windows can be consolidated. Stops with different time windows, even if they overlap, will always be evaluated independently. Current values are: • 0 - Do not consolidate any jobs. All stop time windows are checked individually regardless of the AllowConsolidation (stop) attribute value. • 1 - Consolidate all stops. All stop time windows are consolidated regardless of the AllowConsolidation (Job) attribute value. • 2 - Only consolidate stops with the AllowConsolidation (stop) attribute set to True (1).	
CountStops	smallint	2	This attribute is used to enforce the MaxStops attribute functionality on the route record. Currently the CountStop attribute supports two possible values as follows: • 0 - False - Ignores the MaxStops attribute on the route table and continues to assign stops past the maximum setting. • 1 - True (default) - Uses the MaxStops attribute on the route table to flag the route and affected stops as being in violation when the MaxStops value is exceeded. Note— For more information about how Scheduler determines a new stop, please reference	

			the NewStop (Scheduler) definition.	
IgnoreTerritoryInd	bit	1	This attribute is used to disable the Territory attributes on the route and stop records. When this attribute is enabled, the route and stop territories is ignored.	
NewStop	smallint	2	The NewStop attribute determines what constitutes a new stop. Each activity is associated with one or more stops, and each stop has a location. It is worth noting that starting and ending depot locations are not considered Stops. Valid values for the same location: • 0 - Every stop is always a new stop. • 1 - Every stop for a different location key is a new stop. Stops for the same location key are assigned the same Stop number. • 2 - Every stop for the same latitude and longitude, even for different location keys, is assigned the same stop number. • 3 - Stops for locations within the route's MinTravelDistance (in meters) setting are considered the same stop regardless of location key values.	
NumberOfMeasures	smallint	2	Determines the maximum number of measures used for capacity violation calculations. Maximum allowed is 9.	Required
PreferredResourceWarningInd	smallint	2	Determines whether the PreferredResource attribute on the stop table triggers warnings or violations. If false (0) – all stops must be assigned to their preferred resource or not serviced at all. If true (1) – The preferred resource setting on the stop is considered but is not limited to that resource and can be serviced by other resources.	

RechargeInd	bit	1	This attribute is used to disable the recharge capability of all depots that have recharge capability	
ServiceViolations	int	4	Valid values: 0, 1 and 2. Determines rmpi handling of time, capacity, and distance when representing Schedule Violations. • 0=Resources will ignore stops that cannot be served, driving from the last servable stop directly to the next servable stop. There is no impact to capacity or the number of stops. • 1=Resource will drive to the location of the task or job that cannot be served, but will not spend time there. There is no impact to capacity or the number of stops. • 2=Resource will drive to the location of the task or job and spend the time that would be required to serve the stop. Capacity is consumed and the number of stops is increased as if the stop had been served.	
UseHardWindows	smallint	2	Valid values: 0, 1 and 2. Determines rmpi handling of Jobs' WindowLatestTime values. • 0=Job must be started by WindowLatestTime • 1=Job must be completed by WindowLatestTime • 2=Handling is determined by Job. WindowType value (field determines Job-specific settings for UseHardWindows on individual basis, if applied) • 3=Handling is determined by the job's location time and order date	
UsePreferredResource	smallint	2	Valid values: 0, 1 and 2. Determines how rmpi uses Job.PreferredRoute and Job.RestrictPreferredRoute values during optimization.	

			• 0=Ignores Job.PreferredRoute attributes and overrides Jobs.RestrictPreferredRoute attributes. Equivalent to selecting all jobs and setting RestrictPreferredRoute to false (0). • 1=Uses Job.PreferredRoute attributes and overrides Job.RestrictPreferredRoute attributes. Equivalent to selecting all jobs and setting RestrictPreferredRoute to true (1). • 2=Use Job.PreferredRoute attributes on jobs that have Job.RestrictPreferredRoute set to true (1).	
UseRoadDistance	smallint	2	Valid values: 0 and 1. Determines if Router uses along-road distances or approximate distances when evaluating the schedule. • 0=Use straight-line distances. • 1=Use along-road distances.	
UseRoutePosition	smallint	2	Valid values: 0, 1, and 2. Determines how rmpi uses Job. [Type]RoutePosition and Job.RestrictRoutePosition values during Optimization. • 0=Ignores Job. [Type]RoutePosition attributes and overrides Job.RestrictRoutePosition attributes. Equivalent to selecting all jobs and setting RestrictRoutePosition to false (0). • 1=Uses Job. [Type]RoutePosition attributes and overrides Job.RestrictRoutePosition attributes. Equivalent to selecting all jobs and setting RestrictRoutePosition to true (1). • 2=Uses Job. [Type]RoutePosition attributes on jobs that have RestrictRoutePosition set to true (1).	
UseTimeWindows	smallint	2	Valid values 0, 1, and 2.	

			Determines how rmpi respects Window EarliestTime or LatestTime values. • 0=Apply any job time windows. Otherwise use customer/depot time windows. • 1=Apply customer/depot time windows only. Ignore any job time windows. • 2=Ignore all Job or Customer.Window[X] settings. Only apply Resource.Earliest/LatestStartingTime or Earliest/LatestServiceTime values.	
AutoAssignCloseOrderInd	bit	1	Valid values: 0 and 1. Determines if close jobs are automatically assigned to a single resource. • True (1) = When Optimizer assigns a job, it automatically assigns all jobs at that stop to the same resource regardless of violations. What constitutes a new stop is determined by the NewStop schedule setting. This can speed up the Optimizer run time, but there can be a noticeable delay before Optimizer begins assigning jobs. • False (0) = Optimizer will assign one job at a time. If there are multiple jobs for the same customer, Optimizer will usually, but not always, put all of these jobs on the same resource.	
AssignUnassignedOrdersInd	bit	1	Valid values: 0 and 1. • True (1) = Optimizer will run the Assignment Process and attempt to put unassigned jobs onto resources. • False (0) = Optimizer will not run the Assignment Process. Useful when you are running the Improvement Process only and do	

			not want to assign any of the remaining unassigned jobs.	
DistanceAssessmentInd	int	4	Valid values: 0, 1, and 3. Determines whether Router uses along-road distances or approximations. 0 = Always use Router setting; Use schedule's UseRoadDistance setting. 1 = Use "straight line" for entire run; Always use straight-line approximations. 3 = Use "straight line" for entire run, then final pass with "road distance". Use straight-line approximations throughout. After optimization is complete, if the schedule's UseRoadDistance setting is true (1) and the IntraRouteOptimization setting is true (1), then runs a final intra-route pass with full road distances. Note— Leave at 0 unless instructed by Descartes.	
DynamicZoningInd	bit	1	Valid values: 0 and 1 Allows Optimizer to make appropriate geographic decisions based on the current set of orders in the system. Unlike "static" zones (which must be pre-determined in the data set using the "Zone" attributes), Dynamic Zoning allows the set of Resources/Jobs allowed for any given optimization to change over time based on the current assignment of jobs. - True (1) = Enable Dynamic Zoning. - False (0) = Disable Dynamic Zoning.	
DynamicZoningEntireRoute Ind	bit	1	When Dynamic Zoning Entire Route is enabled, all the stops on a route as determined by the DynamicZoningStrategy setting must be within the DynamicZoningDistance value of	

			each other. So if DynamicZoningStrategy is set to include pickup and dropoff stops and not Initial and Final Depots (default setting) then all the pickups and dropoffs must be within the DynamicZoningDistance of each other calculated straight line. When Dynamic Zoning Entire Route is disabled, each stop on a route, as determined by the DynamicZoningStrategy setting, must be within the DynamicZoningDistance value of one other stop on the route. So if DynamicZoningStrategy is set to include pickup and dropoff stops and not Initial and Final Depots (default setting) then all the pickups and dropoffs must be within the DynamicZoningDistance of at least one other pickup or dropoff, calculated straight line.	
ExchangeRouteInd	bit	1	Valid values: 0 and 1 Determines if Optimizer will attempt to swap entire routes between resources. • True (1) = Optimizer will attempt to swap entire routes between any pair of eligible resources (in accordance with zone files, capabilities, etc.). Useful when resources have different cost structures (such as different driver pay) or are based at different depots. • False (0) = Optimizer will not attempt to swap entire routes.	
IgnoreEmptyRoutesInd	bit	1	Valid values: 0 and 1. Determines if resources without assigned jobs are considered during inter-route swapping. • True (1) = Optimizer will only consider resources that already have jobs assigned. • False (0) = Optimizer will consider all resources.	

InterRouteChainSize	int	4	Maximum number of jobs Optimizer can move as a group within a route during InterRoute Optimization. Recommend setting of at least 10% of the average number of stops per route, and at least 20% if the jobs are double-ended. Note— Larger values give longer running times, and often, but not always, better results. Larger values are recommended if the jobs are double-ended, or the routes are large.	
InterRouteOptimizationInd	bit	1	Valid values: 0 and 1 Determines if Improvement Process attempts to optimize each route individually by running InterRoute Optimization (swapping between available routes, based upon swapping settings). • True (1) = Runs InterRoute Optimization. • False (0) = Does not run InterRoute Optimization.	
IntraRouteChainSize	int	4	Maximum number of jobs Optimizer can move as a group within a route during IntraRoute Optimization. Recommend setting of at least 10% of the average number of stops per route, and at least 20% if the jobs are double-ended. Note— Larger values give longer running times, and often, but not always, better results. Larger values are recommended if the jobs are double-ended, or the routes are large.	
IntraRouteOptimization	bit	1	Valid values: 0 and 1 Determines if Improvement Process attempts to optimize each route individually by running IntraRoute Optimization (swapping within a route, based upon swapping settings).	

			• True (1) = Runs IntraRoute Optimization. • False (0) = Does not run IntraRoute Optimization.	
MaxAssignedBetweenIntra	int	4	When non-zero, during the Assignment Process, Optimizer runs an IntraRoute Optimization pass every time the specified number of jobs is assigned to a route. For example: If setting is 10, IntraRoute Optimization runs after Optimizer has assigned 10 jobs, 20 jobs, 30 jobs, etc. If you have a tightly constrained problem, such as narrow time windows or capacity constraints, using this setting can help generate better assignment answers. Slows down Assignment Process.	
MaxTimeWindowWait	int	4	This attribute is used determine how long a resource will wait for the time window to open at a given stop. It is set in seconds.	
MaxUnservedOrders	int	4	Maximum number of unserved jobs that can be assigned. A value of -1 places no restriction on unserved jobs. This is an approximate setting. Under certain conditions generally related to cost structure, the Optimizer can assign more unserved jobs.	
Merges	smallint	2	Valid values: 0 and 1 Determines if Optimizer attempts to merge routes to create a single route. • True (1) = Optimizer will attempt to merge two routes if the resulting route would have a total length less than the MergeThreshold (see below). • False (0) = Optimizer will not attempt to merge routes. Note— Merging should only be used when routes are small.	

MergeThreshold	int	4	Maximum route length that can be created by merging two routes. See Optimization.Merges (above). Note— When set to 0, Optimizer will not attempt to merge routes. Merging should only be used when routes are small.
PreferredResource	smallint	2	Valid values: 0, 1, and 2. Determines how rmpi uses Job.PreferredRoute and Job.RestrictPreferredRoute values during optimization. • 0=Ignores Job.PreferredRoute attributes and overrides Jobs.RestrictPreferredRoute attributes. Equivalent to selecting all jobs and setting RestrictPreferredRoute to false (0). • 1=Uses Job.PreferredRoute attributes and overrides Job.RestrictPreferredRoute attributes. Equivalent to selecting all jobs and setting RestrictPreferredRoute to true (1). • 2=Use Job.PreferredRoute attributes on jobs that have Job.RestrictPreferredRoute set to true (1).
PreRouteInd	bit	1	Valid values: 0, 1, and 2. • True (1) = Optimizer uses pre-routing. • False (0) = Optimizer does not use pre-routing. Note— Requires Version 7 maps and a valid file specified in the DistanceCacheFile setting. Should only be set to false (0) when the cache file contains all (or the vast majority) of the required road distances. In this case, pre-routing slows optimization slightly by wasting time determining that no routings are needed.

SeedDesiredAreaInd	bit	1	Valid values: 0 and 1. Enables improved handling for seeding with multiple depots. • True (1) = Prevents usable seeds from being assigned to the wrong area. • False (0) = Does not affect assignment. Note— Especially useful in multi-depot settings where seed jobs might otherwise get assigned to a route with an inappropriate depot. Generally should be set to true (1) unless Assignment Process is producing bad answers.	
SeedRouteInd	bit	1	Valid values: 0 and 1. Seeding routes is the process of identifying the difficult jobs (jobs with tight constraints), and assigning these jobs ahead of easier jobs. The Optimizer can then try to fill in around the difficult jobs. • True (1) = Optimizer seeds routes. • False (0) = Optimizer does not seed routes.	
SeedZonedOrdersInd	bit	1	Valid values: 0 and 1. Determines if seeding is restricted to zoned jobs. • True (1) = Restrict seeding to zoned jobs. Un-zoned jobs are never used as seeds. • False (0) = Allow seeding with un-zoned jobs. Optimizer will never seed with more than one un-zoned job in a single pass. Note— Should be set to true if all jobs are expected to be zoned. Should be set to false if there is a mix of zoned and un-zoned jobs.	
SwapsToNeighborsInd	bit	1	Valid values: 0 and 1.	

			Limits InterRoute Optimization to swapping between <i>Routes</i> that are in close proximity. Can improve Optimizer performance. • True (1) = Optimizer will only examine pairs of routes that are close together. • False (0) = Optimizer will examine all pairs of routes, subject to zone definitions. Note— Should be set to true (1) if zones are not well-defined. May miss swap opportunities if capacities are included.	
SwapsToNeighborsPointsInd	bit	1	Valid values: 0 and 1. Limits InterRoute Optimization to swapping Jobs that are in close proximity. Can improve Optimizer performance. Only used if RestrictSwapsToNeighbors is set to true (1). • True (1) = Optimizer will only examine pairs of jobs that are close together and on neighboring routes. • False (0) = Optimizer will examine all pairs of jobs. Note— May miss swap opportunities if capacities are included.	
SwapsWithinZoneInd	bit	1	Valid values: 0 and 1. • True (1) = Job swapping is limited to same zone only. • False (0) = Job swapping is controlled by the [Swaps] section of the zone file. If no Zone file, or no [Swaps], [NoSwaps] section defined in Zone file, all jobs are open for swapping between all capable available Resources.	
UnassignUnservicedOrdersInd	bit	1	Valid values: 0 and 1. Determines if jobs that cannot be serviced are removed from routes. • True (1) = After optimization is complete, Optimizer will remove	

			jobs that cannot be serviced, leaving such jobs unassigned. False (0) = Optimizer will not remove jobs that cannot be serviced. A non-serviced job is violation (visible in RoutePlanner in Red). Allows user to judge if the Optimizer's estimates are accurate.	
UseCluster	smallint	2	Valid values: 0 and 1. Turns clustering on and off. Clustering encourages the Optimizer to assign jobs that are near each other to the same route. Nearness is decided by a number of factors. Two jobs for the same client at the same location are usually near, as are two jobs for clients across the street from each other. Using clustering can speed up the Assignment Process. - True (1) = Optimizer attempts to keep jobs that are near to each other on the same route. - False (0) = Optimizer does not cluster nearby jobs. Note— Clusters can be used only if there are no more than 65,535 jobs. Clustering is known to cause issues with routing the following "genres" of Jobs, and should not be used for: Double-ended Job scenarios (JOB.JobType=1), Recharge scenarios, Redispatch Scenarios, Multi-Task Job scenarios. Use AutoAssignCloseJobs to put all jobs for a single customer on the same route regardless of whether they are served.	
ViableOrdersInd	bit	1	Valid values: 0 and 1. True (1) = Jobs will not be assigned if they cannot be served by any resource. For example: a job is excluded from assignment if (a) its time window cannot be met by any resource; (b) its volume is	

			greater than the capacity of any resource; (c) no resource can drive directly from the pickup to the delivery and still meet the delivery time window; or (d) none of the resources have the capabilities required for the job. False (0) = Optimizer will assign jobs even if they violate rules. You may want to assign such jobs if you do not completely trust the data, or if the constraints in the problem were not absolute.	
ZonableOrdersInd	bit	1	Valid values: 0 and 1. True (1) = Jobs that have a zone that cannot be served by any resource will not be assigned. False (0) = All jobs are assigned, even in violation of zoning rules. Note— If true (1), jobs with blank zones will not be assigned to any resource. If any resource has a wildcard zone (*), then every job can be served by that resource, so this setting is effectively ignored. If false, zones are still preserved as much as possible.	
ZoneFile	nvarchar	255	Entry should contain path and filename of rmpi Zone file (.LZA file) that contains any or all Zone definitions and/or [Swaps][NoSwaps] rules. Can be left blank or omitted if not used.	
UseCandidateList	smallint	2	Determines if the Optimizer uses candidate lists, which can speed up the Assignment Process. True (1) = Optimizer uses candidate lists. False (0) = Optimizer does not use candidate lists. Note— Candidate lists introduce an element of randomness into the Assignment Process. Re-running optimization on the	

			same data set can yield different answers.	
DynamicZoningDistance	float	8	Radius (in meters) around Job or Route (See DynamicZoningEntireRouteInd) that determines distance threshold for inclusion in DynamicZoning.	
DispatchMode	bit	1	Determines a new schedule's DispatchMode setting.	Required
UseRmpiAdvise	bit	1	This field will determine what advise engine to use. Valid values: 0 = false, use adapi 1 = true, use rmpi Rmpi advise is slower when higher volume of orders are required. Default value is 0.	
ImprovementThreshold	float	8	During the Improvement Process, Optimizer will not consider any route change if the net profitability improvement is smaller than this number. Default setting of approximately 0.1 allows Optimizer to find improvements that save miles while ignoring changes that save feet. Note— Smaller numbers may mean longer run times.	
UseTerritoryAssignment	int	4	This field is used to determine when a schedule should use the Territory polygon definition. 0 - Will not use territory polygon assignment/definition 1 - Will always use territory assignment. Use when data is imported using business documents, this value will allow data imported into this schedule to be assigned a territory based on the Territory Definition 2 - Assign territory only if territory is null. This value will assign a territory to orders imported into this schedule only	

			when their territory field value is null	
MaxSuggestions	int	4	Maximum number of suggestions returned in suggest calls	
MaxSuggestPerRoute	int	4	Maximum number of suggestions returned per route	
MaxOptRunTime	int	4	Maximum optimization time. This field limits the optimization running time.	
MinAssignmentProfit	float	8	Minimum incremental profit that a job must provide in order to be assigned. Number set too small may allow for job violations to be assigned. Number set too high will run longer and may lose potential negative net Job profit assignments, even when net Route profit is positive. Note— A value of zero (0) will usually, but not always, keep all unserved jobs from being assigned. Default value is -10000	
LowLevelReroute	int	4	Valid values: 0 = Never use Low-Level Rerouting 1 = Use Low-Level Rerouting when normal routing fails 2 = Always use Low-Level Rerouting. Low-Level refers to the layers (or levels) of Road Classifications within the .LMB Map File Road Networks.	
ThirdPartyDistanceEngine	int	4	Engine to use for distance calculation.	
StartRouteWGPS	bit	1	Valid values: 1 - any GPS message that arrives first for a route is marked as dispatched. 0 - a route is marked as dispatched ONLY when the first status message with explicit status code 5 (DISPATCHED) is received.	

UseAVLInd	bit	1	If it's 0, the following steps are omitted from wireless message processing: • AVL status update • Actual distance/time update.	
UseAutoPublish	bit	1	This field will allow users to set this schedule to use the Autopublish routes functionality defined in the AVL parameters for this schedule. 0 – Do not use Autopublish 1 – Use Autopublish	
AccountName	nVarchar	255	Account name for Autopublish functionality. This value will be included in the published xml	
ConnectionType	nVarchar	255	Type of connection to publish data from FW. Supported Connection Types - FTP - HTTP - Local Path	
LocalPublishPath	nVarchar	255	If the connection type is Local Path, this field will store the directory path where files will be published.	
SenderID	nVarchar	255	This field holds the SenderID value to publish in the xml. Normally used to identify where the data is sent from.	
ReceiverID	nVarchar	255	This field holds the ReceiverID value to publish in the xml. Normally used to identify where the data is sent to.	
FTPURL	nVarchar	255	This field stores the ftp address for publishing data	
FTPUserName	nVarchar	255	This field stores the User name for the ftp connection	
FTPPassword	nVarchar	255	This field stores the Password for the ftp connection	
HttpURL	nVarchar	255	This field stores the HTTP url used to publish data	
CalcualteProfileCost	Bit	1	This field allows FW to calculate costs of routes based on Cost Profiles.	

			0 – Off 1 – On	
TerritorySwapSetKey	nVarchar	50	This field stores the name of the Territory Swap logic in FW	
MaxRechargeDepotsToTry	Int	4	This field stores the maximum number of recharging locations to try when evaluating recharges	
InfeasibilityTaggedConstraints	Int	4	This field will allow Rmpi to impose a large penalty whenever any of the supported constraints gets violated. • The following character codes will apply InfeasiblePenalty for: • C - for Capacity • D - for Maximum Distance • S - for Maximum Stops • R - for Requirements • T - for Resource Time Window • W - for Task Time Window • Q – for Route Position Violation • M – for Commodity Constraints	
SkipInitialLoadTime	Bit	1	Whenever users have double ended jobs where pickups take place at the initial station of the resource and do not want to consider either the location or order service durations just for the pickup portion, then this flag needs to be set to 1. 0 – Default Value. It will consider service durations of pickup portion taking place at initial station of resource of Double ended orders 1 – Will skip service durations of the pickup portion of a double ended order when it takes place at the initial station of the Resource.	
RtePosWithLockInd	Bit	1	This field indicates if a Route Position should be set to the order when a route is locked. 0 – Do not set Route Position when locking 1 – Set the Route position when locking	

ReschePartialComplOrd	Int	4	This field indicates whether or not to allow rescheduling of multi-task orders (double or multi task job orders) when only a portion of this order has been completed. No – Do not allow rescheduling on partially completed orders Yes – Allow rescheduling of orders that have been partially completed	
InfeasiblePenalty	nvarchar		Penalty only used in combination with the Infeasible constraints checked on in this section.	
ForceEndRouteInd	Bit	1	Force the end of route even if pending stops are still present in the route	
SameStopSlots	Int	4	Encourages the return of the same slots if there is already a stop in the same location as the requested advise	
TurboAdvise	Bit	1	Indicator for turbo advise is used when it is true	
TurboAdviseMaxRoutesPerDay	int	4	Maximum number of routes per day to be considered for turbo advise	
TurboAdviseStemDistance	float	8	Radius from the order where available routes will be considered for turbo advise	
UseSchPublishingSettings	bit	1	Use schedule publishing settings instead of application publishing settings	
AllowEarlyRecharge	bit	1	Enables opportunistic recharge	
RequirementsMatchBonusFactor	float	8	Multiplier ranging between 0 and 1 which, when multiplied to an order's profit, produces a bonus profit to be added to a route profit. The bonus gets effective only when the order's requirements are met by the resource's capabilities. Default is 0.01.	
UpdateDropoffQtyWActualPUQty	Bit	1	Update dropoff measures with actual pickup measures	

DefRescheduleKey	nvarchar	50	Default scheduleKey when a reschedule is called without a destination scheduleKey
SetStatRechOnDisp	bit	1	When this is true, all dynamic recharge will be converted to static recharge when dispatched
MaxDistanceFromDepot	float	8	Maximum as-the-crow-flies distance away from initial depot of resource
MaxDistanceFromCustomer	float	8	Maximum distance to path between any two customers. This setting only applies to customers using external pathing and is different from dynamic zoning. If both dynamic zoning and this setting are specified, the maximum distance between customers should be larger than the dynamic zoning distance; otherwise the optimizer will only plan routes for the lesser of the two.
PUComplexSeq	int	4	This field allows users to specify to RMPI the sequence of pickups for double-ended jobs at the same depot based on the delivery sequence: null: Last in, first out (LIFO) 1: Last in, first out (LIFO) 2: First in, first out (FIFO)
SetBegunToArrived	int	4	Values: null or 0: flag is ignored (default) 1: Set Begun = Arrived Timestamp
Disable AutAssign/Optimize All	bit	1	This setting has two options. • Disable Optimize/AutAssign All: When selected, the Optimize All and Auto Assign All options will be hidden from right-click menus. • Disable Optimize/AutAssign All/Selected: When selected, the following options will be hidden from right-click menus: ○ Optimize All ○ Auto Assign All ○ Optimize Selected on the Unassigned Stops quadrant

			○ Optimize Selected, when more than one non-frozen routes are selected on the Route Summary page or Route Detail quadrants ○ Optimize Selected, when multiple unassigned stops are selected on the map or multiple non-frozen routes are selected on the Map quadrant ○ Auto Assign on the Unassigned Stops quadrant or when multiple unassigned stops are selected on the Map quadrant	
UseCallOutNotifyInd	int	4	This field will activate or deactivate Callouts/Notifications for this org 0: use the higher level setting 1: use the local setting 2: do not call	
CallOutNotifyProfileId	int	4	This field will identify the primary key of the Callout/Notification Profile to use	
ScheduleGroup	nvarchar	50	Schedules sharing the same ScheduleGroup value are treated as a single group and users can apply the Suggest function against the group. Each schedule group can be served by a dedicated instance of LNOS Pathing Service for best performance and utilization of computing resources.	
StraightLineSpeed	float	8	Used when RMPI cannot find a legal road to navigate from one point to another (m/s).	
RouteAsCommercialTruck	bit	1	Supports integer entries as follows: 0: Passenger Vehicle 1: Straight Truck	

			2: Articulated Truck	
CheckMeasureLimitsOnLocations	bit	1	When enabled, performs a capacity check at each delivery location to ensure weight and security restrictions are met.	
UseAutoPublish	bit	1	Enables autopublishing for the associated schedule.	
ViolationPenaltyFromProfitFactor	float	8	(default value of 0.0): This field accepts a value between 0.0 and 1.0: When set to 0.0, no additional penalty is incurred (missed time window violations, excessive number of stops and stop order will each have their own penalty factors) When set to 1.0, all profit is added on as an additional penalty (missed time window violations, excessive number of stops or stop order will each have their own penalty factors) When set to 0.5, all profit is added as an additional penalty equal to half the task profit. Essentially, the profit for that stop increases by $(1 - 0.5)$, multiplied by the task_profit.	
IgnoreServiceProfitOnViolation	bit	1	When enabled, regardless of the value of the Violation Penalty From Profit Factor field, violated stops will not contribute profit and no additional penalty is picked up other than missed time window violations, excessive number of stops or stop order issues. When disabled, the task profit is incurred completely for stops in violation.	
OnEndRoute	int	4	0 or null: Do not apply	

			• 1: Set Missed to Pending Stops on End of Trip • 2: Unassign Pending Stops on End of Trip. When set to 2, the system will unassign any pending or non-complete equivalent status stops of the route and then end the trip. This process will trigger a RMPI update.	
MinInterstopTravelTime	int	4	Allows users to enter a minimum value for the travel time to be used between two GeoStops. If RMPI calculates the travel time of a route between GeoStops and the resulting time value is less than the entered Min InterStop Travel Time, the Min InterStop Travel Time is used instead.	
MaxNumberOfOrdersFor OptAll	int	4	Allows users to set a maximum number of assigned and unassigned orders to be sent to RMPI via Optimize All. If a set of orders or routes exceeds the set maximum, the system returns a warning message.	
MaxNumberOfRoutesFor OptAll	int	4	Allows users to set a maximum number of routes to be sent to RMPI via Optimize All. If a set of orders or routes exceeds the set maximum, the system returns a warning message.	
ExternalRoadRouter	nvarchar	250	Map edit server URL used for RMPI routing.	
AltExternalRoadRouter	nvarchar	250	Alternate map edit server URL used for RMPI routing.	
BackhaulDynamicZoningDist	float	8	Provides distance entry for advise calls with LastNStopForDZ values. If Backhaul Dynamic Zoning Distance is not configured per schedule, the configured CtySysValue equivalent is used. If the distance value entered is zero or is not configured anywhere in the system, backhaul dynamic zoning logic is not included in advise filter.	
SetTWtoGenTW	bit	1	When this setting is enabled, the system will replace existing time windows with generated time	

			windows based on projected arrival and projected departure values.	
LockGenTimeWindow	bit	1	Locks generated time windows in place so that, when this setting is enabled, the system will return a violation if users attempt to update time windows for a route that fall outside the initial values.	
LockGenTimeWindow AllowanceX and LockGenTimeWindow AllowanceY	int	4	Allows users to round down the Projected Arrival Time value to the closest hour within a configurable time interval (X) and round up to the closest hour within another configurable time interval (Y).	
AutoPublishLock	bit	1	When enabled, this setting will lock the route when the autopublish command XML is sent from Descartes wGLN. By default, the Use Auto Publish Lock setting is disabled.	
DispatchBGO_FreezeWindowSize	int	4	Using the time value (N) entered in this field, any improvement that would assign an order within the next 'N' minutes will be rejected so as to avoid infeasible routes being created when running an execution BGO	
DispatchBGO_MinStopsToFreeze	int	4	Using the stops value (N) entered in this field, any improvement that would assign an order to the next 'N' stops will be rejected so as to avoid infeasible routes being created when running an execution BGO	
DispatchBGO_AllowDeliverySwap	bit	1	When enabled, this setting allows the BGO to swap orders between routes in an execution schedule.	
CreateReturnOnMiss	bit	1	With this setting is enabled, when a stop is marked as "Missed", the system creates a new order, which moves goods from the customer back to the depot	
AllowPartialDelivery	bit	1	When enabled, allows multi-task orders to be delivered even if one of the pickups is missed	

IncludePickupInAdviseFilter	bit	1	When enabled, pickups will be included in the Advise filter.	
MaxEmptyRoutes	int	4	If Max Empty Routes is set, then the BGO can include up to this number of empty routes while searching for compatible used routes for batch assign.	
Suggest_MaxRoutesPerDay	int	4		
SequenceSameStopJobs	int	4	In cases where a customer has two orders at a stop, one pickup and one delivery, RMPI sequences the orders with the pickup first. With the SequenceSameStopJobs setting, users can modify the default behavior. 0: Disabled 1: Place Deliveries before Pickups by LocationKey	
AMR_Enabled	bit	1	When enabled, the system will attempt to assign unassigned orders to routes generated from master routes before assignment to other routes.	
TerritoryBase	int	4	TerritoryBase has the following options: • Base Territory on Delivery • Base Territory on Pickup • Base Territory on Location: When this option is selected, an additional dropdown menu appears with a description field, allowing users to select a City, Country, LocationKey, Postal Code and various UDF fields to define the territory. Use the operators used in the filters to match the value.	
MeasureForWeight	smallint	2	Specifies the measure to use for onboard weight calculations.	
MeasureToWeightFactor	float	8	Specifies a conversion ratio. The conversion needs to result in a weight in kilograms (kg).	

Dynamic Zoning Strategy			Applies dynamic zoning granularly to Pickup Tasks, Delivery Tasks, Initial Depot and/or Final Depot.	

FWScheduleTemplate

The ScheduleTemplate table is used to create a series of templates that will serve as a basis for creating new schedules.

Create: Individual records as needed.

Delete: A schedule template can be deleted even if there is a schedule associated with it.

Attribute Name	Type	Field Size	Description	Required
ScheduleTemplateID	int	4	Unique internal DB Identifier of the Schedule template record.	Required
ScheduleTemplateKey	nvarchar	50	Business key to identify the activity this Schedule template refers to.	Required
RoutingParameterKey	nvarchar	50	Business key to identify the Routing parameter this Schedule template refers to. Must exist in the FWRouting Parameter table.	
ModifiedBy	int	4	See <i>ModifiedBy</i> under <i>Generic Attributes</i> .	
ModifyDate	datetime	8	See <i>ModifyDate</i> under <i>Generic Attributes</i> .	
CreatedBy	int	4	See <i>CreatedBy</i> under <i>Generic Attributes</i> .	
CreatedDate	datetime	8	See <i>CreatedDate</i> under <i>Generic Attributes</i> .	
Comment	nvarchar	500	See <i>Comment</i> under <i>Generic Attributes</i> .	
Company	varchar	50	See <i>Company</i> under <i>Generic Attributes</i> .	
Name	nvarchar	255	Schedule Template Name	
Type	nvarchar	50	Schedule Template Type. Values: 1 = Reservations	

			• 2 = Planning • 3 = Execution (important for AVL and execution type of schedules. If type 3, then GPS and Status messages will apply to schedules of this type only) • 4 = Simulation	
Status	nvarchar	50	Valid Values: • 0 = Open • 1 = Closed • 2 = Reservation Open • 3 = Reservation Closed • 4 = Planning Open • 5 = Planning Closed • 6 = Dispatch Open • 7 = Dispatch Closed	
ConsolidateStops	smallint	2	This attribute provides the ability to check time windows for the first stop only when there are multiple stops at the same location with identical time windows. If the first stop meets its time windows, then all of the subsequent stops will also meet their time windows. Note—Only stops with identical time windows can be consolidated. Stops with different time windows, even if they overlap, will always be evaluated independently. Current values are: • 0 – Do not consolidate any jobs. All stop time windows are checked individually regardless of the AllowConsolidation (stop) attribute value. • 1 – Consolidate all stops. All stop time windows are consolidated regardless of the AllowConsolidation (Job) attribute value.	

			• 2 – Only consolidate stops with the AllowConsolidation (stop) attribute set to True (1).	
CountStops	smallint	2	This attribute is used to enforce the MaxStops attribute functionality on the route record. Currently the CountStop attribute supports two possible values as follows: • 0 – False – Ignores the MaxStops attribute on the route table and continues to assign stops past the maximum setting. • 1 – True (default) – Uses the MaxStops attribute on the route table to flag the route and affected stops as being in violation when the MaxStops value is exceeded. Note— For more information about how Scheduler determines a new stop, please reference the NewStop (Scheduler) definition.	
IgnoreTerritoryInd	bit	1	This attribute is used to disable the Territory attributes on the route and stop records. When this attribute is enabled, the route and stop territories is ignored.	
NewStop	smallint	2	The NewStop attribute determines what constitutes a new stop. Each activity is associated with one or more stops, and each stop has a location. It is worth noting that starting and ending depot locations are not considered Stops. Valid values for the same location: • 0 – Every stop is always a new stop. • 1 – Every stop for a different location key is a new stop. Stops for the same location key	

			are assigned the same Stop number. • 2 – Every stop for the same latitude and longitude, even for different location keys, is assigned the same stop number. • 3 – Stops for locations within the route's MinTravelDistance (in meters) setting are considered the same stop regardless of location key values.	
NumberOfMeasures	smallint	2	Determines the maximum number of measures used for capacity violation calculations. Maximum allowed is 9.	Required
PreferredResourceWarningInd	smallint	2	Determines whether the PreferredResource attribute on the stop table triggers warnings or violations. If false (0) – all stops must be assigned to their preferred resource or not serviced at all. If true (1) – The preferred resource setting on the stop is considered but is not limited to that resource and can be serviced by other resources.	
RechargeInd	bit	1	This attribute is used to disable the recharge capability of all depots that have recharge capability.	
ServiceViolation	int	4	Valid values: 0, 1 and 2. Determines rmpi handling of time, capacity, and distance when representing Schedule Violations. • 0=Resources will ignore stops that cannot be served, driving from the last servable stop directly to the next servable stop. There is no impact to capacity or the number of stops. • 1=Resource will drive to the location of the task or job that cannot be served, but will not	

			spend time there. There is no impact to capacity or the number of stops. • 2=Resource will drive to the location of the task or job and spend the time that would be required to serve the stop. Capacity is consumed and the number of stops is increased as if the stop had been served.	
UseHardWindows	smallint	2	Valid values: 0, 1 and 2. Determines rmpi handling of Jobs' WindowLatestTime values. • 0=Job must be started by WindowLatestTime • 1=Job must be completed by WindowLatestTime • 2=Handling is determined by Job. WindowType value (field determines Job-specific settings for UseHardWindows on individual basis, if applied) • 3=Handling is determined by the job's location time and order date.	
UsePreferredResource	smallint	2	Valid values: 0, 1 and 2. Determines how rmpi uses Job.PreferredRoute and Job.RestrictPreferredRoute values during optimization. • 0=Ignores Job.PreferredRoute attributes and overrides Jobs.RestrictPreferredRoute attributes. Equivalent to selecting all jobs and setting RestrictPreferredRoute to false (0). • 1=Uses Job.PreferredRoute attributes and overrides Job.RestrictPreferredRoute attributes. Equivalent to selecting all jobs and setting RestrictPreferredRoute to true (1). • 2=Use Job.PreferredRoute attributes on jobs that have	

			Job.RestrictPreferredRoute set to true (1).	
UseRoadDistance	smallint	2	Valid values: 0 and 1. Determines if Router uses along-road distances or approximate distances when evaluating the schedule. • 0=Use straight-line distances. • 1=Use along-road distances.	
UseRoutePosition	smallint	2	Valid values: 0, 1, and 2. Determines how rmpi uses Job.[Type]RoutePosition and Job.RestrictRoutePosition values during Optimization. • 0=Ignores Job.[Type]RoutePosition attributes and overrides Job.RestrictRoutePosition attributes. Equivalent to selecting all jobs and setting RestrictRoutePosition to false (0). • 1=Uses Job.[Type]RoutePosition attributes and overrides Job.RestrictRoutePosition attributes. Equivalent to selecting all jobs and setting RestrictRoutePosition to true (1). • 2=Uses Job.[Type]RoutePosition attributes on jobs that have RestrictRoutePosition set to true (1).	
UseTimeWindows	smallint	2	Valid values 0, 1, and 2. Determines how rmpi respects Window EarliestTime or LatestTime values. • 0=Apply any job time windows. Otherwise use customer/depot time windows. • 1=Apply customer/depot time windows only. Ignore any job time windows.	

			• 2=Ignore all Job or Customer.Window[X] settings. Only apply Resource.Earliest/LatestStartingTime or Earliest/LatestServiceTime values. • 3= The system will use the location time windows if no order time windows are entered, but the date of the order is used to create a legitimate time window	
AutoAssignCloseOrderInd	bit	1	Valid values: 0 and 1. Determines if close jobs are automatically assigned to a single resource. • True (1) = When Optimizer assigns a job, it automatically assigns all jobs at that stop to the same resource regardless of violations. What constitutes a new stop is determined by the NewStop schedule setting. This can speed up the Optimizer run time, but there can be a noticeable delay before Optimizer begins assigning jobs. • False (0) = Optimizer will assign one job at a time. If there are multiple jobs for the same customer, Optimizer will usually, but not always, put all of these jobs on the same resource.	
AssignUnassignedOrdersInd	bit	1	Valid values: 0 and 1. • True (1) = Optimizer will run the Assignment Process and attempt to put unassigned jobs onto resources. • False (0) = Optimizer will not run the Assignment Process. Useful when you are running the Improvement Process only and do not want to assign any	

			of the remaining unassigned jobs.	
DistanceAssessmentInd	int	4	Valid values: 0, 1, and 3. Determines whether Router uses along-road distances or approximations. 0 = Always use Router setting; Use schedule's UseRoadDistance setting. 1 = Use "straight line" for entire run; Always use straight-line approximations. 3 = Use "straight line" for entire run, then final pass with "road distance". Use straight-line approximations throughout. After optimization is complete, if the schedule's UseRoadDistance setting is true (1) and the IntraRouteOptimization setting is true (1), then runs a final intra-route pass with full road distances. Note— Leave at 0 unless instructed by Descartes.	
DynamicZoningInd	bit	1	Valid values: 0 and 1 Allows Optimizer to make appropriate geographic decisions based on the current set of orders in the system. Unlike "static" zones (which must be pre-determined in the data set using the "Zone" attributes), Dynamic Zoning allows the set of Resources/Jobs allowed for any given optimization to change over time based on the current assignment of jobs. - True (1) = Enable Dynamic Zoning. - False (0) = Disable Dynamic Zoning.	
DynamicZoningEntireRouteInd	bit	1	When Dynamic Zoning Entire Route is enabled, all the stops on a route as determined by the	

			DynamicZoningStrategy setting must be within the DynamicZoningDistance value of each other. So if DynamicZoningStrategy is set to include pickup and dropoff stops and not Initial and Final Depots (default setting) then all the pickups and dropoffs must be within the DynamicZoningDistance of each other calculated straight line. When Dynamic Zoning Entire Route is disabled, each stop on a route, as determined by the DynamicZoningStrategy setting, must be within the DynamicZoningDistance value of one other stop on the route. So if DynamicZoningStrategy is set to include pickup and dropoff stops and not Initial and Final Depots (default setting) then all the pickups and dropoffs must be within the DynamicZoningDistance of at least one other pickup or dropoff, calculated straight line.	
ExchangeRouteInd	bit	1	Valid values: 0 and 1 Determines if Optimizer will attempt to swap entire routes between resources. • True (1) = Optimizer will attempt to swap entire routes between any pair of eligible resources (in accordance with zone files, capabilities, etc.). Useful when resources have different cost structures (such as different driver pay) or are based at different depots. • False (0) = Optimizer will not attempt to swap entire routes.	
IgnoreEmptyRoutesInd	bit	1	Valid values: 0 and 1.	

			Determines if resources without assigned jobs are considered during inter-route swapping. • True (1) = Optimizer will only consider resources that already have jobs assigned. • False (0) = Optimizer will consider all resources.	
InterRouteChainSize	int	4	Maximum number of jobs Optimizer can move as a group within a route during InterRoute Optimization. Recommend setting of at least 10% of the average number of stops per route, and at least 20% if the jobs are double-ended. Note— Larger values give longer running times, and often, but not always, better results. Larger values are recommended if the jobs are double-ended, or the routes are large.	
InterRouteOptimizationInd	bit	1	Valid values: 0 and 1 Determines if Improvement Process attempts to optimize each route individually by running InterRoute Optimization (swapping between available routes, based upon swapping settings). • True (1) = Runs InterRoute Optimization. • False (0) = Does not run InterRoute Optimization.	
IntraRouteChainSize	int	4	Maximum number of jobs Optimizer can move as a group within a route during IntraRoute Optimization. Recommend setting of at least 10% of the average number of stops per route, and at least 20% if the jobs are double-ended. Note— Larger values give longer running times, and often, but not always, better	

			results. Larger values are recommended if the jobs are double-ended, or the routes are large.	
IntraRouteOptimization	bit	1	Valid values: 0 and 1 Determines if Improvement Process attempts to optimize each route individually by running IntraRoute Optimization (swapping within a route, based upon swapping settings). • True (1) = Runs IntraRoute Optimization. • False (0) = Does not run IntraRoute Optimization.	
MaxAssignedBetweenIntra	int	4	When non-zero, during the Assignment Process, Optimizer runs an IntraRoute Optimization pass every time the specified number of jobs is assigned to a route. For example: If setting is 10, IntraRoute Optimization runs after Optimizer has assigned 10 jobs, 20 jobs, 30 jobs, etc. If you have a tightly constrained problem, such as narrow time windows or capacity constraints, using this setting can help generate better assignment answers. Slows down Assignment Process.	
MaxTimeWindowWait	int	4	This attribute is used determine how long a resource will wait for the time window to open at a given stop. It is set in seconds.	
MaxUnservdOrders	int	4	Maximum number of unserved jobs that can be assigned. A value of -1 places no restriction on unserved jobs. This is an approximate setting. Under certain conditions generally related to cost structure, the Optimizer can assign more unserved jobs.	

Merges	smallint	2	Valid values: 0 and 1 Determines if Optimizer attempts to merge routes to create a single route. • True (1) = Optimizer will attempt to merge two routes if the resulting route would have a total length less than the MergeThreshold (see below). • False (0) = Optimizer will not attempt to merge routes. Note— Merging should only be used when routes are small.
MergeThreshold	int	4	Maximum route length that can be created by merging two routes. See Optimization.Merges (above). Note— When set to 0, Optimizer will not attempt to merge routes. Merging should only be used when routes are small.
PreferredResource	smallint	2	Valid values: 0, 1, and 2. Determines how rmpi uses Job.PreferredRoute and Job.RestrictPreferredRoute values during optimization. • 0=Ignores Job.PreferredRoute attributes and overrides Jobs.RestrictPreferredRoute attributes. Equivalent to selecting all jobs and setting RestrictPreferredRoute to false (0). • 1=Uses Job.PreferredRoute attributes and overrides Job.RestrictPreferredRoute attributes. Equivalent to selecting all jobs and setting RestrictPreferredRoute to true (1). • 2=Use Job.PreferredRoute attributes on jobs that have Job.RestrictPreferredRoute set to true (1).

PreRouteInd	bit	1	Valid values: 0, 1, and 2. • True (1) = Optimizer uses pre-routing. • False (0) = Optimizer does not use pre-routing. Note— Requires Version 7 maps and a valid file specified in the DistanceCacheFile setting. Should only be set to false (0) when the cache file contains all (or the vast majority) of the required road distances. In this case, pre-routing slows optimization slightly by wasting time determining that no routings are needed.	
SeedDesiredAreaInd	bit	1	Valid values: 0 and 1. Enables improved handling for seeding with multiple depots. • True (1) = Prevents usable seeds from being assigned to the wrong area. • False (0) = Does not affect assignment. Note— Especially useful in multi-depot settings where seed jobs might otherwise get assigned to a route with an inappropriate depot. Generally should be set to true (1) unless Assignment Process is producing bad answers.	
SeedRouteInd	bit	1	Valid values: 0 and 1. Seeding routes is the process of identifying the difficult jobs (jobs with tight constraints), and assigning these jobs ahead of easier jobs. The Optimizer can then try to fill in around the difficult jobs. • True (1) = Optimizer seeds routes.	

			False (0) = Optimizer does not seed routes.	
SeedZonedOrdersInd	bit	1	Valid values: 0 and 1. Determines if seeding is restricted to zoned jobs. - True (1) = Restrict seeding to zoned jobs. Un-zoned jobs are never used as seeds. - False (0) = Allow seeding with un-zoned jobs. Optimizer will never seed with more than one un-zoned job in a single pass. Note— Should be set to true if all jobs are expected to be zoned. Should be set to false if there is a mix of zoned and un-zoned jobs.	
SwapsToNeighborsInd	bit	1	Valid values: 0 and 1. Limits InterRoute Optimization to swapping between <i>Routes</i> that are in close proximity. Can improve Optimizer performance. - True (1) = Optimizer will only examine pairs of routes that are close together. - False (0) = Optimizer will examine all pairs of routes, subject to zone definitions. Note— Should be set to true (1) if zones are not well-defined. May miss swap opportunities if capacities are included.	
SwapsToNeighborsPointsInd	bit	1	Valid values: 0 and 1. Limits InterRoute Optimization to swapping <i>Jobs</i> that are in close proximity. Can improve Optimizer performance. Only used if <i>RestrictSwapsToNeighbors</i> is set to true (1). True (1) = Optimizer will only examine pairs of jobs that are close together and on neighboring routes.	

			False (0) = Optimizer will examine all pairs of jobs. Note— May miss swap opportunities if capacities are included.	
SwapsWithinZoneInd	bit	1	Valid values: 0 and 1. - True (1) = Job swapping is limited to same zone only. - False (0) = Job swapping is controlled by the [Swaps] section of the zone file. If no Zone file, or no [Swaps], [NoSwaps] section defined in Zone file, all jobs are open for swapping between all capable available Resources.	
UnassignUnservicedOrdersInd	bit	1	Valid values: 0 and 1. Determines if jobs that cannot be serviced are removed from routes. - True (1) = After optimization is complete, Optimizer will remove jobs that cannot be serviced, leaving such jobs unassigned. - False (0) = Optimizer will not remove jobs that cannot be serviced. A non-serviced job is violation (visible in RoutePlanner in Red). Allows user to judge if the Optimizer's estimates are accurate.	
UseCluster	smallint	2	Valid values: 0 and 1. Turns clustering on and off. Clustering encourages the Optimizer to assign jobs that are near each other to the same route. Nearness is decided by a number of factors. Two jobs for the same client at the same location are usually near, as are two jobs for clients across the street from each other. Using clustering can speed up the Assignment Process.	

			• True (1) = Optimizer attempts to keep jobs that are near to each other on the same route. • False (0) = Optimizer does not cluster nearby jobs. Note— Clusters can be used only if there are no more than 65,535 jobs. Clustering is known to cause issues with routing the following “genres” of Jobs, and should not be used for: Double-ended Job scenarios (JOB.JobType=1), Recharge scenarios, Redispatch Scenarios, Multi-Task Job scenarios. Use AutoAssignCloseJobs to put all jobs for a single customer on the same route regardless of whether they are served.	
ViableOrdersInd	bit	1	Valid values: 0 and 1. • True (1) = Jobs will not be assigned if they cannot be served by any resource. For example: a job is excluded from assignment if (a) its time window cannot be met by any resource; (b) its volume is greater than the capacity of any resource; (c) no resource can drive directly from the pickup to the delivery and still meet the delivery time window; or (d) none of the resources have the capabilities required for the job. • False (0) = Optimizer will assign jobs even if they violate rules. You may want to assign such jobs if you do not completely trust the data, or if the constraints in the problem were not absolute.	

ZonableOrdersInd	bit	1	Valid values: 0 and 1. • True (1) = Jobs that have a zone that cannot be served by any resource will not be assigned. • False (0) = All jobs are assigned, even in violation of zoning rules. Note— If true (1), jobs with blank zones will not be assigned to any resource. If any resource has a wildcard zone (*), then every job can be served by that resource, so this setting is effectively ignored. If false, zones are still preserved as much as possible.	
ZoneFile	nvarchar	255	Entry should contain path and filename of rmpi Zone file (.LZA file) that contains any or all Zone definitions and/or [Swaps][NoSwaps] rules. Can be left blank or omitted if not used.	
Profit	float	8	The schedule profit that is to be copied over to a newly created schedule.	
UseCandidateList	smallint	2	Determines if the Optimizer uses candidate lists, which can speed up the Assignment Process. • True (1) = Optimizer uses candidate lists. • False (0) = Optimizer does not use candidate lists. Note— Candidate lists introduce an element of randomness into the Assignment Process. Re-running optimization on the same data set can yield different answers.	
DynamicZoningDistance	float	8	Radius (in meters) around Job or Route (See <i>DynamicZoningEntireRoute</i> in <i>FWSchedule</i> table) that	

			determines distance threshold for inclusion in DynamicZoning.	
LowLevelReroute	int	4	Valid values: • 0 = Never use Low-Level Rerouting • 1 = Use Low-Level Rerouting when normal routing fails • 2 = Always use Low-Level Rerouting. Low-Level refers to the layers (or levels) of Road Classifications within the .LMB Map File Road Networks.	
ThirdPartyDistanceEngine	int	4	Engine to use for distance calculation.	
StartRouteWGPS	bit	1	Valid values: • 1 – any GPS message that arrives first for a route is marked as dispatched. • 0 – a route is marked as dispatched ONLY when the first status message with explicit status code 5 (DISPATCHED) is received.	
UseAVLInd	bit	1	If it's 0, the following steps are omitted from wireless message processing: • AVL status update • Actual distance/time update.	
MaxDistanceFromDepot	float	8	Maximum as-the-crow-flies distance away from initial depot of resource	
MaxDistanceFromCustomer	float	8	Maximum distance to path between any two customers. This setting only applies to customers using external pathing and is different from dynamic zoning. If both dynamic zoning and this setting are specified, the maximum distance between customers should be larger than the dynamic zoning distance; otherwise the optimizer will only	

			plan routes for the lesser of the two.	
TerritorySwapSetKey	nvarchar	50	This field stores the name of the Territory Swap logic in FW	
MaxRechargeDepotsToTry	int	4	This field stores the maximum number of recharging locations to try when evaluating recharges	
InfeasibilityTaggedConstraints	int	4	This field will allow Rmpi to impose a large penalty whenever any of the supported constraints gets violated. • The following character codes will apply InfeasiblePenalty for: • C – for Capacity • D – for Maximum Distance • S – for Maximum Stops • R – for Requirements • T – for Resource Time Window • W – for Task Time Window • Q – for Route Position Violation • M – for Commodity Constraints	
SkipInitialLoadTime	bit	1	Whenever users have double ended jobs where pickups take place at the initial station of the resource and do not want to consider either the location or order service durations just for the pickup portion, then this flag needs to be set to 1. 0 – Default Value. It will consider service durations of pickup portion taking place at initial station of resource of Double ended orders 1 – Will skip service durations of the pickup portion of a double ended order when it takes place at the initial station of the Resource.	

RtePosWithLockInd	bit	1	This field indicates if a Route Position should be set to the order when a route is locked. 0 – Do not set Route Position when locking 1 – Set the Route position when locking	
ForceEndRouteInd	bit	1	Force the end of route even if pending stops are still present in the route	
ReschePartialCmplOrd	int	4	This field indicates whether or not to allow rescheduling of multi-task orders (double or multi task job orders) when only a portion of this order has been completed. No – Do not allow rescheduling on partially completed orders Yes – Allow rescheduling of orders that have been partially completed	
SameStopSlots	int	4	Encourages the return of the same slots if there is already a stop in the same location as the requested advise	
TurboAdvise	bit	1	Indicator for turbo advise is used when it is true	
TurboAdviseMaxRoutesPerDay	int	4	Maximum number of routes per day to be considered for turbo advise	
TurboAdviseStemDistance	float	8	Radius from the order where available routes will be considered for turbo advise	
InfeasiblePenalty	float	8	Penalty only used in combination with the Infeasible constraints checked on in this section.	
AllowEarlyRecharge	bit	1	Enables opportunistic recharge	
RequirementsMatchBonusFactor	float	8	Multiplier ranging between 0 and 1 which, when multiplied to an order's profit, produces a bonus profit to be added to a route profit. The bonus gets effective only when the order's	

			requirements are met by the resource's capabilities. Default is 0.01.	
UpdateDropoffQtyWActualPUQty	bit	1	When this is on, dropoff measures will be updated with actual pickup measure of a double-ended job	
UseAutoPublish	bit	1	This field will allow users to set this schedule to use the Autopublish routes functionality defined in the AVL parameters for this schedule. 0 – Do not use Autopublish 1 – Use Autopublish	
AccountName	nvarchar	255	Account name for Autopublish functionality. This value will be included in the published xml	
ConnectionType	nvarchar	255	Type of connection to publish data from FW. Supported Connection Types - FTP - HTTP - Local Path	
LocalPublishPath	nvarchar	255	If the connection type is Local Path, this field will store the directory path where files will be published.	
SenderID	nvarchar	255	This field holds the SenderID value to publish in the xml. Normally used to identify where the data is sent from.	
ReceiverID	nvarchar	255	This field holds the ReceiverID value to publish in the xml. Normally used to identify where the data is sent to.	
FTPURL	nvarchar	255	This field stores the ftp address for publishing data	
FTPUserName	nvarchar	255	This field stores the User name for the ftp connection	
FTPPassword	nvarchar	255	This field stores the Password for the ftp connection	

HttpURL	nvarchar	255	This field stores the HTTP url used to publish data	
CalculateProfileCost	bit	1	When this is true, profile cost will be calculated	
DefRescheduleKey	nvarchar	50	Default scheduleKey when a reschedule is called without a destination scheduleKey	
UseSchPublishingSettings	bit	1	Use schedule publishing settings instead of application publishing settings	
SetStatRechOnDisp	bit	1	When this is true, all dynamic recharge will be converted to static recharge when dispatched	
MaxDistanceFromDepot	float	8	Maximum as-the-crow-flies distance away from initial depot of resource	
StraightLineSpeed	float	8	Used when RMPI cannot find a legal road to navigate from one point to another (m/s)	
OnEndRoute	int	1	• 0 or null: Do not apply • 1: Set Missed to Pending Stops on End of Trip • 2: Unassign Pending Stops on End of Trip. When set to 2, the system will unassign any pending or non-complete equivalent status stops of the route and then end the trip. This process will trigger a RMPI update.	
MinInterstopTravelTime	int	4	Allows users to enter a minimum value for the travel time to be used between two GeoStops. If RMPI calculates the travel time of a route between GeoStops and the resulting time value is less than the entered Min InterStop Travel Time, the Min InterStop Travel Time is used instead.	
MaxNumberOfOrdersForOptAll	int	4	Allows users to set a maximum number of assigned and unassigned orders to be sent to RMPI via Optimize All. If a set of	

			orders or routes exceeds the set maximum, the system returns a warning message.	
MaxNumberOfRoutesForOptAll	int	4	Allows users to set a maximum number of routes to be sent to RMPI via Optimize All. If a set of orders or routes exceeds the set maximum, the system returns a warning message.	
ExternalRoadRouter	nvarchar	250	Map edit server URL used for RMPI routing.	
AltExternalRoadRouter	nvarchar	250	Alternate map edit server URL used for RMPI routing.	
BackhaulDynamicZoningDist	float	8	Provides distance entry for advise calls with LastNStopForDZ values. If Backhaul Dynamic Zoning Distance is not configured per schedule, the configured CtySysValue equivalent is used. If the distance value entered is zero or is not configured anywhere in the system, backhaul dynamic zoning logic is not included in advise filter.	
SetTWtoGenTW	bit	1	When this setting is enabled, the system will replace existing time windows with generated time windows based on projected arrival and projected departure values.	
LockGenTimeWindow	bit	1	Locks generated time windows in place so that, when this setting is enabled, the system will return a violation if users attempt to update time windows for a route that fall outside the initial values.	
LockGenTimeWindow AllowanceX and LockGentimeWindow AllowanceY	int	4	Allows users to round down the Projected Arrival Time value to the closest hour within a configurable time interval (X) and round up to the closest hour within another configurable time interval (Y).	

AutoPublishLock	bit	1	When enabled, this setting will lock the route when the autopublish command XML is sent from Descartes wGLN. By default, the Use Auto Publish Lock setting is disabled.	
DisplatchBGO_FreezeWindowSize	int	4	Using the time value (N) entered in this field, any improvement that would assign an order within the next 'N' minutes will be rejected so as to avoid infeasible routes being created when running an execution BGO	
DispatchBGO_MinStopsToFreeze	int	4	Using the stops value (N) entered in this field, any improvement that would assign an order to the next 'N' stops will be rejected so as to avoid infeasible routes being created when running an execution BGO	
DispatchBGO_AllowDeliverySwap	bit	1	When enabled, this setting allows the BGO to swap orders between routes in an execution schedule.	
CreateReturnOnMiss	bit	1	With this setting is enabled, when a stop is marked as "Missed", the system creates a new order, which moves goods from the customer back to the depot	
AllowPartialDelivery	bit	1	When enabled, allows multi-task orders to be delivered even if one of the pickups is missed	
IncludePickupInAdviseFilter	bit	1	When enabled, pickups will be included in the Advise filter.	
MaxEmptyRoutes	int	4	If Max Empty Routes is set, then the BGO can include up to this number of empty routes while searching for compatible used routes for batch assign.	
Suggest_MaxRoutesPerDay	int	4		
SequenceSameStopJobs	int	4	In cases where a customer has two orders at a stop, one pickup and one delivery, RMPI sequences the orders with the pickup first.	

			With the SequenceSameStopJobs setting, users can modify the default behavior. 0: Disabled 1: Place Deliveries before Pickups by LocationKey	
AMR_Enabled	bit	1	When enabled, the system will attempt to assign unassigned orders to routes generated from master routes before assignment to other routes.	
TerritoryBase	int	4	TerritoryBase has the following options: • Base Territory on Delivery • Base Territory on Pickup • Base Territory on Location: When this option is selected, an additional dropdown menu appears with a description field, allowing users to select a City, Country, LocationKey, Postal Code and various UDF fields to define the territory. Use the operators used in the filters to match the value.	
MeasureForWeight	smallint	2	Specifies the measure to use for onboard weight calculations.	
MeasureToWeightFactor	float	8	Specifies a conversion ratio. The conversion needs to result in a weight in kilograms (kg).	
Disable AutAssign/Optimize All	bit	1	This setting has two options. • Disable Optimize/AutAssign All: When selected, the Optimize All and Auto Assign All options will be hidden from right-click menus. • Disable Optimize/AutAssign All/Selected: When selected, the following options will be hidden from right-click menus:	

			○ Optimize All ○ Auto Assign All ○ Optimize Selected on the Unassigned Stops quadrant ○ Optimize Selected, when more than one non-frozen routes are selected on the Route Summary page or Route Detail quadrants ○ Optimize Selected, when multiple unassigned stops are selected on the map or multiple non-frozen routes are selected on the Map quadrant ○ Auto Assign on the Unassigned Stops quadrant or when multiple unassigned stops are selected on the Map quadrant	

FWBucket

➡ **Note**—Currently this table is here to support old version and is replaced by ScheduleWindow table.

Attribute Name	Type	Field Size	Description	Required
BucketID	int	4	Unique Identifier of the Bucket record.	Required
ScheduleID	int	4	Unique Identifier of the Schedule record. Must exist on the FWSchedule table.	Required
ScheduleKey	nvarchar	50	Business key to identify the Schedule this Bucket refers to. Must exist on the FWSchedule table.	Required

TimeZoneKey	nvarchar	50	Business key to identify the Time zone this Bucket refers to. Must exist on the FWSchedule table.	
Description	nvarchar	255	This attribute is assigned a specific string value by the user when created. Valid values must be validated against a predefined list of valid values. The predefined list needs to be externalized for possible extension during implementations.	
EarliestDate	datetime	8	This attribute is assigned a specific string value by the user when created. Valid values must be validated against a predefined list of valid values. The predefined list needs to be externalized for possible extension during implementations.	Required
LatestDate	datetime	8	This attribute is assigned a specific string value by the user when created. Valid values must be validated against a predefined list of valid values. The predefined list needs to be externalized for possible extension during implementations.	
Type	nvarchar	50	Custom-defined type of a bucket.	
Territory	nvarchar	300	Territory to which this bucket applies. Only buckets with territory code matching to that of an order are sent to Adapi or Rmpi for advise.	
Comment	nvarchar	500	See <u>Comment</u> under <u>Generic Attributes</u> .	
ModifiedBy	int	4	See <u>ModifiedBy</u> under <u>Generic Attributes</u> .	
ModifyDate	datetime	8	See <u>ModifyDate</u> under <u>Generic Attributes</u> .	
CreatedBy	int	4	See <u>CreatedBy</u> under <u>Generic Attributes</u> .	

CreatedDate	datetime	8	See <i>CreatedDate</i> under <i>Generic Attributes</i> .	
BucketKey	nvarchar	50	Business key to identify the activity this Bucket refers to.	
UseFrequency	bit	1	Flag to indicate if frequency is used. If used, then the Frequency field will be used instead of the date portion of the StartDate and EndDate fields.	
Frequency	int	4	Flag to indicate what days of the week in a date range should be used. Buckets will be loaded into the advise engine for each of the frequency days that fall within the input date range of the advise call. For Example: Suppose there is a daily bucket, only Monday through Friday, to offer between 8:00 am to 10:00 am. Then just creating a Bucket with a Frequency of M/T/W/Th/F on from 8:00 am to 10:00 am will suffice. The date portion of the start/end is ignored. This way there is no need to generate buckets for specific dates. When an advise call is sent with a date range from say 23-May to 26-May, Route Planner will send the following buckets to adapi: • Bucket 1 ○ Start Date/Time: 23-May-2008 08:00 am ○ End Date/Time: 23-May-2008 10:00 pm • Bucket 2 ○ Start Date/Time: 26-May-2008 08:00 am ○ End Date/Time: 26-May-2008 10:00 pm	

FWBucketTemplate

Attribute Name	Type	Field Size	Description	Required
BucketTemplateID	int	4	Unique internal DB Identifier of the Bucket Template record.	Required
ScheduleID	int	4	Unique internal DB Identifier of the Schedule record. Must exist on the FWSchedule table.	Required
ScheduleKey	nvarchar	50	Business key to identify the Schedule this Bucket template refers to. Must exist on the FWSchedule table.	
Description	nvarchar	255	Description of this bucket	
EndDay	int	4	Used for generation of buckets using this template. A value greater than 0, will indicate that the EndDate for the generated bucket will be the date that falls within the date range and matches its frequency value plus this field value. <i>For example: if buckets are generated for a week date range and its frequency is Monday, Wednesday and Friday, then the EarliestEndDate will be based on Monday's date plus this field value, Wednesday's date plus this field value, etc.</i>	
StartTime	datetime	8	Start time for this bucket	
EndTime	datetime	8	End time for this bucket	
Type	nvarchar	50	Field to designate a type of Bucket. <i>For example: Platinum, gold and silver. This string will guide the interfaces as to what slots to offer given customer loyalty.</i>	
Territory	nvarchar	300	Territory to which this bucket applies	
[Size]	int	4	Size of this bucket. Used when generating buckets based. This size field value will determine the Start and End Time	

SizeTimeUnits	int	4	Time units of the size (hours or minutes)	
NoOfBucketsPerDay	int	4	Number of buckets to generate per day.	
Comment	nvarchar	500	See <i>Comment</i> under <i>Generic Attributes</i> .	
Company	nvarchar	50	See <i>Company</i> under <i>Generic Attributes</i> .	
CreatedBy	int	4	See <i>CreatedBy</i> under <i>Generic Attributes</i> .	
CreatedDate	datetime	8	See <i>CreatedDate</i> under <i>Generic Attributes</i> .	
ModifiedBy	int	4	See <i>ModifiedBy</i> under <i>Generic Attributes</i> .	
ModfiyDate	datetime	8	See <i>ModifyDate</i> under <i>Generic Attributes</i> .	
BucketTemplateKey	nvarchar	50	Business key to identify the activity this Bucket template refers to.	
Name	nvarchar	50	Bucket Template name	
MultiDayInd	bit	1	Designates if this bucket template is a multi-day. If it is, then	
UseFrequency	bit	1	Flag to indicate if frequency is used. If used, then the Frequency field will be used instead of the date portion of the StartDate and EndDate fields.	
Frequency	int	4	Flag to indicate what days of the week in a date range should be used. Buckets will be loaded into the advise engine for each of the frequency days that fall within the input date range of the advise call. <i>For Example: Suppose there is a daily bucket, only Monday through Friday, to offer between 8:00 am to 10:00 am. Then just creating a Bucket with a Frequency of M/T/W/Th/F on from 8:00 am to 10:00 am will suffice. The date portion of the start/end is ignored.</i>	

			<i>This way there is no need to generate buckets for specific dates.</i> <i>When an advise call is sent with a date range from say 23-May to 26-May, Route Planner will send the following buckets to adapi:</i> Bucket 1 Start Date/Time: 23-May-2008 08:00 am End Date/Time: 23-May-2008 10:00 pm Bucket 2 Start Date/Time: 26-May-2008 08:00 am End Date/Time: 26-May-2008 10:00 pm	
TimeZone	nvarchar	100	Time zone	

FWAlert

Attribute Name	Type	Field Size	Description	Required
AlertID	int	4	Unique internal DB identifier of the Alert record.	Required
RouteID	int	4	Unique internal DB identifier of the Route record. Must exist on the FWRoute table.	
StopID	int	4	Unique identifier of the Stop record. Must exist on the FWStop table.	
DateTime	datetime	8	The Date/Time stamp reported in the Alert.	
ResourceKey	nvarchar	50	Business key to identify the Resource this Alert refers to.	
AlertCodeID	int	4	Unique Identifier of the Alert code record.	
HiddenInd	bit	1	This field indicates if this alert is to be hidden in the user interface (UI). 0 - hidden	

			1 - shown - Null - shown	
HiddenByID	int	4	User ID that hid this field.	
ModifiedBy	int	4	See <i>ModifiedBy</i> under <i>Generic Attributes</i> .	
ModifyDate	datetime	8	See <i>ModifyDate</i> under <i>Generic Attributes</i> .	
AlertCodeValue	nvarchar	250	This alert code may have a default value that if reported by the wireless device and no value comes in, then this value will be used to display in the UI.	
Type	int	4	Type of the alert. 1 for custom alert, 2 for reason code and 3 for system alert	
Comment	nvarchar	500	It is used to provide additional information about a specific record.	
HiddenTime	datetime	8	Time when the alert is changed to hidden	
OrderLineID	int	4	Unique internal DB Identifier of the Order Line record.	
DateTimeUTC	datetime	8	Timestamp of the alert (Coordinated Universal Time)	
wGLNSyncInd	bit	1	Can be set to true or false depending on the response from wGLN. 0 – Failure. wGLN does not have the latest changes. 1 – wGLN is in sync with Route Planner.	
wGLNResponse	nvarchar		This field holds the response from wGLN	

FWAlertCode

Attribute Name	Type	Field Size	Description	Required

AlertCodeID	int	4	Unique internal DB Identifier of the Alert code record.	Required
AlertCodeKey	nvarchar	50	Business key to identify the activity this Alert code refers to. It is used to identify the alert reported by wireless devices	Required
Description	nvarchar	250	Description for this alert code	
AlertCodeValue	nvarchar	250	This alert code may have a default value that if reported by the wireless device and no value comes in, then this value will be used to display in the UI.	
ColorCode	nvarchar	50	AlertCode that can be used to map wireless alerts to this code.	
ModifiedBy	int	4	See <i>ModifiedBy</i> under <i>Generic Attributes</i> .	
ModifyDate	datetime	8	See <i>ModifyDate</i> under <i>Generic Attributes</i> .	
wGLNSyncInd	bit	4	This field holds the indicator value for wGLN synchronization: 0-Not in synch with wGLN 1-In synch with wGLN	
Category	nvarchar	50	Category value For wGLN purposes	
Type	int	4	Type of the alert. 1 for custom alert, 2 for reason code and 3 for system alert	
wGLNResponse	nvarchar		This field holds the response from wGLN	
SnoozeDuration	int	4	Minimum duration after the acknowledgment that recurrences of the same alert are ignored. Certain types of system alerts can occur repeatedly (e.g. speeding alert); once a user acknowledges it, the alert is hidden from the UI, but will re-appear when active again. In a case where users no longer want that alert to re-appear, they can use this field to specify a duration to keep the alerts silenced.	

ExtDocControlID	nvarchar	250	External business document ID	

FWDataSetGroup

The table is used to identify a problem set group used to differentiate between different problem types. It is typically used to define the problem for Background optimization, planning, and dispatch.

Attribute Name	Type	Field Size	Description	Required
DataSetGroupID	int	4	Unique internal DB Identifier of the DataSetGroup record.	Required
DataSetGroupKey	nvarchar	50	Business key to identify the activity this DataSetGroup refers to.	Required
AreaKey	nvarchar	50	Business key to identify the Area this DataSetGroup refers to. Must exist on the FWArea table.	Required
Type	nvarchar	50	This attribute is assigned a specific string value by the user when created. Valid values must be validated against a predefined list of valid values on the FWCode table. Fixed values: • BGO = Background Optimizer • DEDICATED_BGO = Background optimizer fixed on one group • PLANNER = Route Planner • DISPATCH = Dispatch	Required
ServiceName	nvarchar	255	The service name is set to "-None-" when the status is set to "Free." It is system assigned to a service name of the service receiving the dataset for processing. It is typically set to a unique service scheduler machine name and port.	Required
Status	nvarchar	50	This attribute is assigned a specific string value by the system. It is not modifiable by the user. Valid values must be validated against a predefined list of valid values. The predefined list needs to be externalized for possible extension during implementations.	Required

			Fixed values: System Assigned • Planning • Dispatching • Frozen • Optimizing • Free • Busy • Done	
LastEvaluated	datetime	8	When the status is set to "Processing" or "Done." Should be blank when the status is set to "Free."	
Comment	nvarchar	500	See <i>Comment</i> under <i>Generic Attributes</i> .	
CreatedBy	int	4	See <i>CreatedBy</i> under <i>Generic Attributes</i> .	
CreatedDate	datetime	8	See <i>CreatedDate</i> under <i>Generic Attributes</i> .	
ModifiedBy	int	4	See <i>ModifiedBy</i> under <i>Generic Attributes</i> .	
ModifyDate	datetime	8	See <i>ModifyDate</i> under <i>Generic Attributes</i> .	
Company	nvarchar	50	See <i>Company</i> under <i>Generic Attributes</i> .	
LastRetrieved	datetime	8	Last time the record is retrieved.	
UseDispatchBGO	bit	1	Specifies whether or not this data slice selects from a Dispatching schedule	
Active	bit	1	When this setting is enabled, the BGO will optimize the data slice. This functionality applies to both types of data slices: dedicated and dynamic. The Active setting is disabled by default in order to allow users to configure the data slice first and then activate it when configuration is complete. In cases where the data slice is active and being optimized and this setting is	

			turned off, the system will behave as follows: • The data slice will not be assigned to a BGO instance. • Or, if a BGO instance is currently optimizing the data slice, the next time that the BGO requests a refresh of the data slice, the data slice will not be returned, causing BGO instance to move on to another data slice. The refresh interval is controlled by the RefreshData optimizer property: com.descartes.escheduler.optimizer.refresh_inosfw_data	

FWDataSetNesting

The table is used to specify the table and attributes used to define and extract a specific problem set. It provides the ability to define up to six different attributes in a nested format that can be used to filter different attributes within attributes.

For example: If you specify Territory at level one and Driver at level two, then the problem is restricted to that driver within that territory, instead of the whole territory.

Attribute Name	Type	Field Size	Description	Required
DataSetNestingID	int	4	Unique internal DB of the DataSetNesting record.	Required
DataSetNestingKey	nvarchar	50	Business key to identify the activity this DataSetNesting refers to.	Required
TableName	nvarchar	50	The value must be a valid Descartes Route Planner table name. All tables with a FW prefix are valid. The table name is validated against the FWCode table of type NestedTable.	Required
Level1Attribute	nvarchar	50	Level one is always required, whereas the other Level Attributes are optional. The concept here is to start at level one and work deeper into the table by being more specific with each level using	Required

			a different attribute. It is possible to use the same attribute at different levels. The actual attribute name is provided by the user and must be on the table specified in the TableName attribute. If an attribute is used that does not exist, then it is ignored. The attributes must exist on the FWCode table of type NestedTableAttribute.	
Level2Attribute	nvarchar	50	If Level two is specified, then level one must also be specified. If level two is empty, then levels three, four, five, and six must also be empty. See Level1Attribute for more details.	
Level3Attribute	nvarchar	50	If Level three is specified, then levels one and two must also be specified. If level three is empty, then levels four, five, and six must also be empty. See Level1Attribute for more details.	
Level4Attribute	nvarchar	50	If Level four is specified, then levels one, two, and three must also be specified. If level four is empty, then levels five and six must also be empty. See Level1Attribute for more details.	
Level5Attribute	nvarchar	50	If Level five is specified, then levels one, two, three, and four must also be specified. If level five is empty, then level six must also be empty. See Level1Attribute for more details.	
Level6Attribute	nvarchar	50	If Level six is specified, then levels one, two, three, four, and five must also be specified. See Level1Attribute for more details.	
Comment	nvarchar	255	See <i>Comment</i> under <i>Generic Attributes</i> .	
CreatedBy	int	4	See <i>CreatedBy</i> under <i>Generic Attributes</i> .	

CreatedDate	datetime	8	See <u>CreatedDate</u> under <u>Generic Attributes</u> .	
ModifiedBy	int	4	See <u>ModifiedBy</u> under <u>Generic Attributes</u> .	
ModifyDate	datetime	8	See <u>ModifyDate</u> under <u>Generic Attributes</u> .	
Company	nvarchar	50	See <u>Company</u> under <u>Generic Attributes</u> .	

FWDDataSetValue

This table is related to both the FWDDataSetGroup and the FWDDataSetNesting tables. It bridges the gap between the problem set definition and the different tables and their nested attributes by providing the actual attribute values used to glue them together. The DataSetValue table records link the other two DataSet tables together. It provides the link between “many to many” relationships and provides the ability to specify value-level definitions needed to perform specific problem set extractions.

Attribute Name	Type	Field Size	Description	Required
DataSetValueID	int	4	Unique internal DB Identifier of the DataSetValue record.	Required
DataSetGroupID	int	4	Unique internal DB Identifier of the DataSetGroup record. Must exist on the FWDDataSetGroup table.	Required
DataSetNestingID	int	4	Unique internal DB Identifier of the DataSetNesting record. Must exist on the FWDDataSetNesting table.	Required
DataSetGroupKey	nvarchar	50	Business key to identify the DataSetGroup this DataSetValue refers to. Must exist on the FWDDataSetGroup table.	Required
DataSetNestingKey	nvarchar	50	Business key to identify the DataSetNesting this DataSetValue refers to. Must exist on the FWDDataSetNesting table.	Required
EndLevel	tinyint	1	Since we only have six levels, the valid values are 1 through 6 only. A value of zero (0) is not allowed. Once a value has been entered, it needs to be validated against the FWDDataSetNesting table Level 1 – 6 Attributes.	Required

			<i>For example: If the EndLevel is set to two, then the Level one and two Attributes must be defined on the FWDDataSetNesting table. If it is set to five, then one through five must be set, and so on.</i> The EndLevel attribute allows you to define a fully nested definition at the nesting table level and control the execution level at the value table definition. The default is 1. For example: You can specify all six levels but only extract to level 1 if this value is set to 1. This provides a certain reuse capability with fewer definitions. See the FWDDataSetNesting table for additional level details.	
Level1Value	nvarchar	10000	Level one is always required, whereas the other Level Attributes are optional. The concept here is to start at level one and work deeper into the table by being more specific with each level using different attribute values. The user will provide the actual attribute value. It should be a valid data setting for that attribute, otherwise no results is returned.	
Level2Value	nvarchar	1000	If the EndLevel attribute is one.	Required - If the EndLevel attribute is two or higher. If level two is specified, then level one must also be specified.
Level3Value	nvarchar	1000	If the EndLevel attribute is two or lower.	Required - If the EndLevel attribute is three or

				higher. If level three is specified, then level one and two must also be specified.
Level4Value	nvarchar	1000	If the EndLevel attribute is three or lower.	Required - If the EndLevel attribute is four or higher. If level four is specified, then level one, two, and three must also be specified.
Level5Value	nvarchar	1000	If the EndLevel attribute is four or lower.	Required - If the EndLevel attribute is five or higher. If level five is specified, then level one, two, three, and four must also be specified.
Level6Value	nvarchar	1000	If the EndLevel attribute is five or lower.	Required - If the EndLevel attribute is six. If level six is specified, then level one, two, three, four,

				and five must also be specified.
Comment	nvarchar	255	See <i>Comment</i> under <i>Generic Attributes</i> .	
CreatedBy	int	4	See <i>CreatedBy</i> under <i>Generic Attributes</i> .	
CreatedDate	datetime	8	See <i>CreatedDate</i> under <i>Generic Attributes</i> .	
ModifiedBy	int	4	See <i>ModifiedBy</i> under <i>Generic Attributes</i> .	
ModifyDate	datetime	8	See <i>ModifyDate</i> under <i>Generic Attributes</i> .	

FWDDataSetGroupSetSchedule

Attribute Name	Type	Field Size	Description	Required
DataSetGroupSetScheduleID	int	4	Unique internal DB Identifier of the DataSetGroupSetSchedule record.	Required
DataSetGroupSetID	int	4	Unique internal DB Identifier of the DataSetGroupSet record.	Required
ProfileSetID	int	4	Unique internal DB Identifier of the data set properties record used for this schedule	Required
StartTime	datetime	8	Start time for the data set schedule	
EndTime	datetime	8	End time for the data set schedule	
CreatedDate	datetime	8	See <i>CreatedDate</i> under <i>Generic Attributes</i> .	
ModifiedBy	int	4	See <i>ModifiedBy</i> under <i>Generic Attributes</i> .	
ModfiyDate	datetime	8	See <i>ModifyDate</i> under <i>Generic Attributes</i> .	
DataSetGroupName	nvarchar	50	Name of the DataSetGroup Must exist on the FWDDataSetGroup table.	Required

ProfileSetName	nvarchar	50	Name of the data set properties used for this schedule	Required
Frequency	bit	255	Designates which weekdays or dates the item is available to be scheduled. For weekdays, you can use the standard abbreviations, full names, or numbers 1 through 7 corresponding to days Monday through Sunday. Dates may be written as M/D/Y, M-D-Y, MMDDYY, or MMDDYYYY. For example, a setting of "135" schedules the item for Monday, Wednesday and Friday. A setting of "Mon Wed Fri 6/30/1999 7/1/1999" schedules the item for Monday, Wednesday, and Friday and for 6/30/1999 and 7/1/1999.	

FWDriver

Attribute Name	Type	Field Size	Description	Required
DriverID	int	4	Unique internal DB Identifier of the driver record.	Required
DriverKey	nvarchar	50	Business key to identify the activity this driver refers to.	Required
DriverFirstName	nvarchar	255	First name of the driver.	Required
DriverLastName	nvarchar	255	Last name of the driver.	
ModifiedBy	int	4	See <i>ModifiedBy</i> under <i>Generic Attributes</i> .	
ModifyDate	datetime	8	See <i>ModifyDate</i> under <i>Generic Attributes</i> .	
DeviceKey	nvarchar	50	Unique Business Key that identifies the handheld device assigned to this driver. Used for wireless purposes.	
LocationID	int	4	Unique internal DB Identifier. Sets the relationship between the FWLocation and the FWDriver	

			tables. This field defines the driver's address.	
LicenseNumber	nvarchar	50	The license number of the driver.	
LicenseIssueDate	datetime	8	The issue date of the license.	
LicenseExpDate	datetime	8	The expiration date of the license.	
EarliestStartTime	datetime	8	The earliest start time for the driver. Important Notice— To the extent that the application is being used in connection with total and or elapsed driver hours it is the sole responsibility of the user to ensure compliance with any and all applicable statutory regulations and applicable laws relating to driver hours.	
LatestStartTime	datetime	8	The latest start time for the driver. Important Notice— To the extent that the application is being used in connection with total and or elapsed driver hours it is the sole responsibility of the user to ensure compliance with any and all applicable statutory regulations and applicable laws relating to driver hours.	
EarliestEndTime	datetime	8	The earliest end time for the driver. Important Notice— To the extent that the application is being used in connection with total and or elapsed driver hours it is the sole responsibility of the user to ensure compliance with any and all applicable statutory regulations and applicable laws relating to driver hours.	
LatestEndTime	datetime	8	The latest end time for the driver. Important Notice— To the extent that the application is being used in connection with total and or elapsed driver hours it is the sole responsibility of the user to ensure compliance with any and all	

			applicable statutory regulations and applicable laws relating to driver hours.	
Active	int	4	When this attribute is 'Active' the driver is active or available. When it is 'Inactive' the driver is inactive or not available.	
CostProfileID	int	4	Identifies the cost profile of the driver	
Email	nvarchar	50	The email address of the driver.	
Certificates	nvarchar	250	Certificates that this driver has to drive asset types.	
wGLNSyncInd	bit		Can be set to true or false depending on the response from wGLN. 0 – Failure. wGLN does not have the latest changes. 1 – wGLN is in sync with Route Planner.	
wGLNResponse	nvarchar		This field holds the response from wGLN	
Comment	nvarchar	500	It is used to provide additional information about a specific record.	
ExtDocControlID	nvarchar	250	External business document ID	
CostProfileKey	nvarchar	50	Unique Business Key Identifier for the the profile.	
Role	nvarchar	256	Stores the role of the driver in according to the crew assignment	
ServiceSpeedFactor	float	8	Service Speed Factor is driver skill factor or efficiency. It works together with FwResource.ServiceSpeedFactor. If Driver factor is set up, it replaces Resource factor. Driver factor is used in current service duration formula as a divider. If it is less than 1, the resulting service duration is larger than the standard service duration. If	

greater than 1, the result is shorter (that is, more efficient).

FWCrewAssignment

Attribute Name	Type	Allow Nulls	Description
CrewAssignmentID	int	No	Internal Primary Key ID for this crew assignment
DriverID	int	No	DriverID value that refers to the FWDriver Table
DriverKey	nvarchar	No	Business Driver Key identifier
ResourceID	int	No	Resource ID to which this Driver is assigned to
ResourceKey	nvarchar	No	Business Resource Key identifier

FWCostProfile

Attribute Name	Type	Field Size	Description	Required
CostProfileID	int	4	Unique internal DB Identifier for the cost profile.	
CostProfileKey	nvarchar	50	Unique Business Key Identifier for the profile.	
CostProfileDescription	nvarchar	500	A description of the cost profile.	
StartupCost	float	8	The start up cost for the cost profile. Note— The startup cost is not used in optimization.	

FWCostDefinition

Attribute Name	Type	Field Size	Description	Required
CostDefinitionID	int	4	Unique internal DB Identifier for the cost profile definition.	

CostDuration	float	8	Defines the parameters for calculating incrementally increasing or decreasing costs on the duration of routes. The cost per second of route time is CostPerSecond until the time of the route exceeds the CostDuration value. <i>For example: if the first hour costs 100 and the second hour costs 200, then the CostDuration for the first hour should be 3600 and the CostDuration for the 2nd hour should be 7200.</i> Note— CostDuration is not use in optimization.	
LowerBoundHourOfDay	datetime	8	Lower bound of the Time of the day at which this CostPerSecond will be incurred. <i>For example: a route can have a LowerBoundHourOfDay=7:00 am, then the CostPerSecond will begin incurring at 7:00 am until the UpperBoundHourOfDay is reached.</i> Note— LowerBoundHourOfDay is not use in optimization.	
UpperBoundHourOfDay	datetime	8	Upper bound of the Time of day at which this CostPerSecond will be incurred. <i>For example: a route can have a UpperBoundHourOfDay=7:00 pm, then the CostPerSecond will stop incurring at 7:00 pm.</i> Note— UpperBoundHourOfDay is not use in optimization.	
CostPerSecond	float	8	Cost per second of work Note— CostPerSecond is not used in optimization if CostDuration is used.	
CostPerDistance	float	8	Cost Per Meter. This cost will allow users to define a cost per meter so that a total cost per distance driven can be calculated. Note— CostPerDistance is not used in optimization.	

LowerBoundDistance	float	8	Lower bound of the Distance interval at which the Cost Per Distance will be incurred. Note— LowerBoundDistance is not used in optimization.	
UpperBoundDistance	float	8	Upper bound of the Distance interval at which the Cost Per Distance will be incurred. Note— UpperBoundDistance is not used in optimization.	
CostProfileID	int	4	Unique internal DB Identifier of the Cost profile record. Must exist on the FWCostProfile table.	

FWTrailer

Attribute Name	Type	Field Size	Description	Required
TrailerID	int	4	Unique internal DB Identifier of the trailer record.	Required
TrailerKey	nvarchar	50	Business key to identify the activity this Trailer refers to.	
FinalLocationID	int	4	Defines the initial station at which the trailer should go back at the end of a journey.	
InitialLocationID	int	4	Defines the initial station at which the trailer should start its journey.	
ModifiedBy	int	4	See <i>ModifiedBy</i> under <i>Generic Attributes</i> .	
ModifyDate	datetime	8	See <i>ModifyDate</i> under <i>Generic Attributes</i> .	
Comment	nvarchar	500	See <i>Comment</i> under <i>Generic Attributes</i> .	
Company	nvarchar	50	The company name of the trailer.	
Measure1-9	float	8	Capacity x in units of Measure x of this trailer. Amount of single-dimension measure X.	
Measure(1-9)Factor	float	8	Factor for single-dimension measure X. Typically used to	

			convert Measure[X]. Amount attribute to a different unit of measure.	
Measure(1-9)RechargeRate	float	8	This attribute field is currently not used, but is a place holder for future use.	
Name	nvarchar	255	Name of the trailer.	
PreServiceDuration	int	4	Pre Service Duration in seconds for the trailer. This time will be incurred at every stop before it starts working on the job.	
Requirements	nvarchar	255	Capabilities of the trailer.	
ServiceDuration	float	8	This time will be incurred at every stop.	
RechargeDuration	int	4	Time to do a recharge.	
UDFString(1-9)	nvarchar	500	User defined field string (1-9)	
UDFInteger (1-3)	int	4	User defined field integer (1-3)	
UDFNumber (1-3)	float	8	User defined field number (1-3)	
Commodities	nvarchar	255	Commodities that the trailer can handle. Expressed in a string format (i.e. AA BB).	
LicenseNumber	nvarchar	50	The license number of the trailer.	
LicenseIssueDate	datetime	8	The issue date of the license.	
LicenseExpDate	datetime	8	The expiration date of the license.	
CostProfileID	int	4	Field to identify the cost profile for the trailer.	
MobileNumber	float	8	The mobile number for the trailer.	
Owner	nvarchar	50	The owner of the trailer.	
Height	float	8	Asset Height	
Length	float	8	Asset Length	
Width	float	8	Asset Width	
VehicleType	nvarchar	50	Asset Type	
HeightLoadCapacity	float	8	Asset Height Load Capacity	
WidthLoadCapacity	float	8	Asset Width Load Capacity	

LengthLoadCapacity	float	8	Asset Length Load Capacity	
RequirementSetID	int	4	Requirements Set identifier for this asset	
wGLNSyncInd	bit		Can be set to true or false depending on the response from wGLN. 0 – Failure. wGLN does not have the latest changes. 1 – wGLN is in sync with Route Planner.	
wGLNResponse	nvarchar		This field holds the response from wGLN	
TareWeight	int	4	Tare weight trailer	
MaxWeight	int	4	Maximum weight of the trailer	
TotalWeight	int	4	Total weight of the trailer	
Cooler	bit	1	Indicate the trailer has a cooler	
Lift	bit	1	Indicate the trailer has a lift	
CostProfileKey	nvarchar	50	Unique Business Key Identifier for the the profile.	
ExtDocControlID	nvarchar	250	External business document ID	

FWTruck

Attribute Name	Type	Field Size	Description	Required
TruckID	int	4	Unique internal DB Identifier of the truck record.	Required
TruckKey	nvarchar	50	Business key to identify the activity this Truck refers to.	
FinalLocationID	int	4	Defines the initial station at which the truck should go back at the end of a journey.	
InitialLocationID	int	4	Defines the initial station at which the truck should start its journey.	
Comment	nvarchar	500	See <i>Comment</i> under <i>Generic Attributes</i> .	

Company	nvarchar	50	The company name of the truck.	
ModifiedBy	int	4	See <i>ModifiedBy</i> under <i>Generic Attributes</i> .	
ModifyDate	datetime	8	See <i>ModifyDate</i> under <i>Generic Attributes</i> .	
Measure1-9	float	8	Capacity x in units of Measure x of this truck. Amount of single-dimension measure X.	
Measure(1-9)Factor	float	8	Factor for single-dimension measure X. Typically used to convert Measure[X]. Amount attribute to a different unit of measure.	
Measure(1-9)RechargeRate	float	8	Specifies how quickly a truck can deliver a Measure.	
Name	nvarchar	255	Name of the truck.	
PartialFullPenalty	float	8	Penalty applied to a route that has one or more measures filled more than the FullThreshold percentage, and one or more measures filled less than the FullThreshold percentage.	
Requirements	nvarchar	255	Capabilities of the truck.	
ServiceDuration	float	8	This time will be incurred at every stop.	
RechargeDuration	int	4	Time to do a recharge.	
Territory	nvarchar	50	The territory for the truck.	
Commodities	nvarchar	255	Commodities that the trailer can handle. Expressed in a string format (i.e. AA BB).	
DiscountInitialLeg	float	8	Field to define whether or not to discount the initial leg of the journey off the total cost of the route.	
DiscountFinalLeg	float	8	Field to define whether or not to discount the final leg of the journey off the total cost of the route.	
UDFString(1-9)	nvarchar	500	User defined field string (1-9)	

UDFInteger(1-3)	int	4	User defined field integer (1-3)	
UDFNumber(1-3)	float	8	User defined field number (1-3)	
DynamicZoningDistance	float	8	Dynamic zoning distance in meters.	
LicenseNumber	nvarchar	50	The license number of the trailer.	
LicenseIssueDate	datetime	8	The issue date of the license.	
LicenseExpDate	datetime	8	The expiration date of the license.	
CostProfileID	int	4	Field to identify the cost profile for the trailer.	
MobileNumber	nvarchar	50	The mobile number for the trailer.	
DriverID	int	4	Unique internal DB Identifier of the Driver record. Must exist on the FWDriver table.	
TrailerID	int	4	Unique internal DB Identifier of the Trailer record. Must exist on the FWTrailer table.	
Owner	nvarchar	250	The owner of the trailer.	
VehicleType	nvarchar	50	Free text identifying the vehicle type. User specified.	
FirstStopTimeThreshold	float	8	If early rests are allowed, prior to serving the first job on a route, if either of the clocks (elapsed or driving) have higher utilization that this threshold, an EarlyRest is inserted, possibly backing up the initial departure time.	
FullThreshold	float	8	Percentage of full capacity that is acceptable for applying CompleteFullBonus and PartialFullPenalty.	
WaitTimePenalty	float	8	This attribute field is currently not used, but is a place holder for future use.	
MaxDistance	float	8	This attribute field is currently not used, but is a place holder for future use.	
MaxStops	smallint	2	Maximum number of stops the resource can perform. For this	

			attribute to be applied, the schedule's CountStops setting must be set to true (1). If two jobs are in close proximity they are counted as one stop. You can also set a maximum number of jobs	
TerritoryID	int	4	Unique internal DB identifier of the Territory record. Must exist on the FWTerritory table. Used to identify the assets territory.	
Height	float	8	Asset Height	
Length	float	8	Asset Length	
Width	float	8	Asset Width	
HeightLoadCapacity	float	8	Asset Type	
WidthLoadCapacity	float	8	Asset Height Load Capacity	
LengthLoadCapacity	float	8	Asset Width Load Capacity	
RequirementSetID	int	4	Asset Length Load Capacity	
wGLNSyncInd	bit		Can be set to true or false depending on the response from wGLN. 0 – Failure. wGLN does not have the latest changes. 1 – wGLN is in sync with Route Planner.	
wGLNResponse	nvarchar		This field holds the response from wGLN	
TareWeight	int	4	Tare weight of the truck	
MaxWeight	int	4	Maximum weight of the truck	
TotalWeight	int	4	Total weight of the truck	
Cooler	bit	1	Indicate the truck has a cooler	
Lift	bit	1	Indicate the truck has a lift	
OnBoardComputer	bit	1	Indicate the truck has an onboard computer	
OnBoardComputerKey	nvarchar	50	Business key for the onboard computer	

OnBoardComputerActive	bit	1	Indicate the onboard computer is active/inactive	
CostProfileKey	nvarchar	50	Unique Business Key Identifier for the the profile.	
ExtDocControlID	nvarchar	250	External business documentation ID	

FWOrderLeg

Attribute Name	Type	Field Size	Description	Required
OrderLegID	int	4	Unique internal DB Identifier for the Order leg record.	
FromLocationID	int	4	Unique Origin Location Identifier for the cross dock leg.	
ToLocationID	int	4	Unique Destination Location Identifier for the cross dock leg.	
LegNumber	int	4	The leg number for this leg.	
OrderID	int	4	The original order ID of this cross dock leg.	
ProjectedDrivingTime			Estimated driving time calculated by the optimization engine	
ProjectedDistance			Estimated Distance calculated by the optimization engine	

FWProductType

Attribute Name	Type	Field Size	Description	Required
ProductTypeId	int	4	Unique internal DB Identifier for the Product type record.	Required
ProductTypeKey	nvarchar	50	Business key to identify the activity this Product type refers to.	Required
ProductTypeName	nvarchar	255	The name of the product type.	
Measure(1-9)	float	8	Amount that this product uses of measure x (1-9) per Unit.	

Measure(1-9)Factor	float	8	Factor for single-dimension measure X. Typically used to convert Measure[X]. Amount attribute to a different unit of measure.
ServiceRate	float	8	Factor used to Calculate the ServiceDuration. This rate will be used to calculate it against the Measure specified in the ApplyToMeasure field.
ApplyToMeasure	tinyint	4	Specifies the Measure number to which the specified ServiceRate should be applied.
PreServiceDuration	int	4	Pre-Service Duration before servicing this Item.
ServiceDuration	int	4	Service Duration for this Product Type. If defined, then the Service Rate is ignored.
PUPreServiceDuration	int	4	Pre Service Duration for the pickup portion of an order.
PUServiceDuration	int	4	Service Duration for the pickup portion of an order.
DeliveryPreServiceDuration	int	4	Pre Service Duration for the Delivery portion of an order.
DeliveryServiceDuration	int	4	Service Duration for the Delivery portion of an order.
Profit	float	8	The item's profit.
Requirements	nvarchar	255	The requirements for the Product key.
Commodities	nvarchar	255	The commodities for the Product key.
Comment	nvarchar	500	See <i>Comment</i> under <i>Generic Attributes</i> .
ModifiedBy	int	4	See <i>ModifiedBy</i> under <i>Generic Attributes</i> .
ModifyDate	datetime	8	See <i>ModifyDate</i> under <i>Generic Attributes</i> .
RequirementSetID	int	4	The Requirement Set ID taken from the FWRequirementSet table.

wGLNSyncInd	bit	1	Can be set to true or false depending on the response from wGLN. 0 – Failure. wGLN does not have the latest changes. 1 – wGLN is in sync with Route Planner.	
wGLNResponse	nvarchar	512	This field holds the response from wGLN	
WorkUnits	float	8	The amount of workload an product type contributes toward the maximum allowed by a resource's work units	
HazMatType	int	4	Indicate the product is hazardous material.	
Category	nvarchar	100	Captures category for product types	
ProdCategory	nvarchar	100	Captures product category for product types	

FWRequirementSet

Attribute Name	Type	Field Size	Description	Required
RequirementSetID	int	4	Unique internal DB Identifier for the Requirement set record.	
RequirementSetKey	nvarchar	50	Business key that identifies the activity this Requirement set refers to.	Required
Name	nvarchar	50	The name of the requirement set.	
Description	nvarchar	50	The description of this requirement set.	
Comment	nvarchar	500	See <i>Comment</i> under <i>Generic Attributes</i> .	

FWRequirement

Attribute Name	Type	Field Size	Description	Required

RequirementID	int	4	Unique internal DB Identifier for the requirement.	
RequirementKey	nvarchar	50	Unique Business Key Identifier for the requirement. This key will be used for the optimizer.	Required
Name	nvarchar	50	The name of the requirement.	
Description	nvarchar	50	The description of this requirement.	
Comment	nvarchar	500	See <i>Comment</i> under <i>Generic Attributes</i> .	

FWAVLParm

Attribute Name	Type	Field Size	Description	Required
AVLParmID	int	4	Unique internal DB identifier of the AVLParm record.	Required
ScheduleID	int	4	Unique internal DB identifier of the Schedule record. Must exist in the FWSchedule table.	Required
RouteThreshold	float	8	Distance for Route Geofence	
RouteThresholdUnit	nvarchar	10	Distance units for Route Geofence Distance	
StopThreshold	float	8	Distance for Stop Geofence	
StopThresholdUnit	nvarchar	10	Distance units for Stop Geofence distance	
DistanceAssessment	bit	1	Field to determine if along the road is used to calculate the geofence distance for both, route and stops. 0 – Do not use along the road 1 - use along the road	
AVLStatusRefreshCycle	int	4	Number of GPS messages to recalculate the route	
MobilityWindow	int	4	Time interval to check when a resource has not reported gps or status messages	

MinAvgSpeed	float	8	Specifies the minimum speed allowed in a vehicle before reporting a mobility alert (vehicle is not moving or moving very slow)	
SpeedUnit	nvarchar	10	Speed units	
MinTemperature	float	8	Specifies the minimum temperature that the temperature is allowed before an alert is triggered.	
MaxTemperature	float	8	Specifies the maximum temperature that the temperature is allowed before an alert is triggered.	
AutomaticEnrouteonGPS	Bit	1	This field defines whether or not the application will set the status to Enroute automatically once the first GPS message is received	
ServiceDurationTolerance	int	4	Specifies the maximum time a ServiceDuration can be exceeded before raising an alert. A value of 0 will turn off this notification.	
DepartStopTolerance	int	4	Specifies the maximum time a route stays in a stop that has been completed. A value of 0 will turn off this notification.	
StartTimeTolerance	int	4	Specifies the maximum exceeding time a route is allowed to delay its start before raising an alert. A value of 0 will turn off this notification.	
AutoCorrectGeocodeInd	bit	8	This field will turn on or off Autocorrection on Geocoding information based on actual. Possible values: 0 – Off 1 – On	
AutoCorrectGeocodeAfter	int	8	The field will define the number of successful GPS readings before doing the correction of the latitude/longitude.	
AutoCorrectDistance	int	8	This field will define the distance threshold that will decide if a	

			latitude/longitude reading is a candidate of autocorrection. If all the readings (defined by the AutoCorrectGeocodeAfter value) are within this distance from each other, then the latitude/longitude saved in the database will be corrected only for those locations where the LocLatLong field is off.	
DepartDistanceThreshold	Int	4	This field will specify the distance threshold to consider when a vehicle has left the stop	
SpeedTolerance	Int	4	This field will specify the speed tolerance allowed by a vehicle before a Speeding alert is created. Meters/Second A value of 0 will turn off this notification	
RemainingRoutingTime	int		This field defines the number of maximum number of seconds before a route ends. If the remaining time of the route is less than this number, then an alert will be raised.	
RemainingRoutingDistance	float		This field will specify the distance threshold to consider when a vehicle has left the stop	
RemainingStops	int	4	This field defines the number of maximum number of stops before a route ends. If the remaining stop of the route is less than this number, then an alert will be raised.	
AlertNoStopsPending	int		This field defines the number (integer) of stops that are in pending before the route is completed. Pending equivalent status include: • Pending • Delayed • Enroute • Skipped	

RemainingRoutingTime	int		This field defines the number of maximum number of seconds before a route ends. If the remaining time of the route is less than this number, then an alert will be raised.	
RemainingRoutingDistance	int		This field defines the number of maximum number of meters before a route ends. If the remaining distance of the route is less than this number, then an alert will be raised.	

FWCommodityConstraint

Attribute Name	Type	Field Size	Description	Required
CommodityConstraintID	int	4	Unique internal DB Identifier of the Commodity constraint record.	Required
CommodityKey	nvarchar	255	Business key to identify the activity this Commodity key refers to.	Required
CommodityDescription	nvarchar	255	The description of this commodity.	
CommodityConflictKey	nvarchar	255	Business key to identify the Commodity conflict this Commodity refers to.	
ScheduleKey	nvarchar	50	Business key to identify the Schedule this Commodity refers to. Must exist on the FWSchedule table.	
Active	bit	1	Turns off/on this constraint	

FWGPSStatus

Attribute Name	Type	Field Size	Description	Required
GPSStatusID	int	4	Unique internal DB Identifier of the GPS status record.	Required

RouteID	int	4	Unique internal DB Identifier of the Route record. Must exist on the FWRoute table.	
DateTime	datetime	8	The Date/Time stamp reported in the GPS message.	
Latitude	float	8	In conjunction with Longitude attribute, provides geographic position of address. For use with non-geocodable addresses, or in longhaul situations where the map does not support geocoding.	
Longitude	float	8	In conjunction with Latitude, provides geographic position of address. For use with non-geocodable addresses, or in longhaul situations where the map does not support geocoding.	
ModifiedBy	int	4	See <i>ModifiedBy</i> under <i>Generic Attributes</i> .	
ModifyDate	datetime	8	See <i>ModifyDate</i> under <i>Generic Attributes</i> .	
ResourceKey	nvarchar	50	Business key to identify the Resource this GPS status refers to. Must exist on the FWResource table.	
Ignition	bit	1	Indicates if this GPS reports Ignition on or off. 0 - Off 1 - On Null – not reported	
Odometer	float	8	Stores the odometer reading reported in the incoming gps status message	
UnscheduledStop	bit	1	Indicates if this GPS reports a unscheduled stop on or off 0 - Off 1 - On Null – not reported	
Speed	float	8	Stores the speed reported in the GPS Status message	

Direction	nvarchar	50	Stores the direction reported in the GPS Status message	
DriverKey	nvarchar	50	Stores the DriverKey reported in the GPS Status message	
TractorKey	nvarchar	50	Stores the TractorKey reported in the GPS Status message	
TrailerKey	nvarchar	50	Stores the TrailerKey reported in the GPS Status message	
Temperature	float	8	Stores the Temperature reported in the GPS Status message	
AVLCounter	int	4	Stores the counter for AVL parameters	
StopID	int	4	StopID the gps message is related to	
StartOdometer	int	4	Beginning odometer of the vehicle	
EndOdometer	float	8	Ending odometer of the vehicle	
LastReportedTimeZone	nvarchar	100	Time zone of last reported time	
DateTimeUTC	datetime	8	Time stamp of the gps status in UTC	
WirelessSource	nvarchar	20	Allows wGLN to detect whether the submitted status information was sent by driver, black box or ASU	

FWAutoPublish

Attribute Name	Type	Field Size	Description	Required
AutoPublishLogID	int	4	Unique Log ID for this record.	Required
RouteID	int	255	Route ID	Required
ScheduleID	int	255	Schedule ID	Required
ProjDepartedDate	date	255	Route Projected Departed Date	
ProjArrivedDate	date	255	Route Projected Arrived Date	
ProjDistance	float	255	Route Projected Distance	
PotAvailCap1	float	255	Route Projected Availability for Measure 1	

PotAvailCap2	float	255	Route Projected Availability for Measure 2	
PotAvailCap3	float	255	Route Projected Availability for Measure 3	
PotAvailCap4	float	255	Route Projected Availability for Measure 4	
PotAvailCap5	float	255	Route Projected Availability for Measure 5	
PotAvailCap6	float	255	Route Projected Availability for Measure 6	
PotAvailCap7	float	255	Route Projected Availability for Measure 7	
PotAvailCap8	float	255	Route Projected Availability for Measure 8	
PotAvailCap9	float	255	Route Projected Availability for Measure 9	
NumberOfStops	int	255	Route Number of Stops	

FWExtAcntConfig

Attribute Name	Type	Field Size	Description	Required
ExtAcntConfigID	decimal	9	DB Identifier for this Account	Required
ApplicationKey	nvarchar	256	Application identifier. For example, wGLN	
CompanyName	nvarchar	32	Company name to log into external application	
GroupName	nvarchar	32	Group name to log into external application	
UserID	nvarchar	32	DCF User ID	
UserName	nvarchar	256	External application user name	
Password	nvarchar	256	External application password. Encrypted using blowfish encryption algorithm	

ModifiedBy	decimal	9	User ID used to create/modify this record	
ModifyDate	datetime	8	Modify date	

FWCalloutNotifyProfile

ColumnName	Type	Allow Nulls	Description
CalloutNotifyProfileId	int	No	This field will be used as the primary internal record identifier in this table.
CalloutNotifyProfileKey	nvarchar	No	This field will help users identify the different Call out and Notification profiles defined in the database.
CalloutNotifyProfName	nvarchar	Yes	Descriptive Name for this Profile Key
UseCallOutInd	int	Yes	Its default value should be 0. This field will indicate whether or not call outs will be active in this profile.
CallMinsBeforeArrive	int	Yes	This field will indicate wGLN the number of minutes it should trigger a call before Arriving to customer. If null, it will not call the customer.
CallMinsAfterComplete	int	Yes	This field will indicate wGLN the number of minutes it should trigger a call after completing the delivery. If null, it will not call the customer.
CallOutPrefLangDesc	nvarchar	Yes	Indicates the preferred language to use during a callout. If null, it will use English as the default language. CallOutPrefLangDesc will accept language name values ("English", "Spanish", etc.). The following is the complete list of language codes: • 1 English • 2 Espanol • 3 Francais

			• 4 Deutsch • 5 Italiano • 6 Portugueses • 7 Chinese • 8 Japanese • 11 Danish • 12 Swedish • 14 Korean • 20 Polish Note— For callouts, only English, French and Spanish are supported.
UseNotificationInd	int	Yes	Its default value should be 0. This field will indicate whether or not notifications will be active in this profile.
NotifyMinsBeforeArrive	int	Yes	This field will indicate wGLN the number of minutes it should trigger a notification before Arriving to customer. If null, it will not notify the customer.
NotifyMinsAfterComplete	Int	Yes	This field will indicate wGLN the number of minutes it should trigger a notification after completing the delivery. If null, it will not notify the customer.
NotifyEvents	bitmask	Yes	Events to notify (subscribe to events): 1-EnRoute 2-Arrived 3-Completed 4-Delayed 5-Skipped 6-NoFreight 7-Missed 8-Returned 9-Hooked 10-Spotted

OtherEmailToNotify	nvarchar	Yes	Other emails to notify in addition to the one contained in the location email field. This field may contain multiple email addresses separated by a ',' or ';'.
UseSMSNotificationInd	bit	Yes	Enables SMS messages for notifications
UseSMSLocationField	nvarchar	Yes	Used to specify which FWLocation field should be used for SMS messages. Valid values include ContactPhone and ContactMobilePhone
UseCallOutLocationField	nvarchar	Yes	Used to specify which FWLocation field should be used for call outs. Valid values include ContactPhone and ContactMobilePhone

FWAppConfig

Attribute Name	Type	Allow Nulls	Description
UseCallOutNotifyInd	int	no	This Keyword will activate or deactivate Callouts/Notifications for this org
DefaultCallNotifyProfKey	nvarchar	yes	This keyword will indicate the default Call Out/Notification Profile to use for this Organization

FWFieldData

Attribute Name	Type	Field Size	Description
Class	nvarchar	50	0: User 1: System Command XML status allows passing of this field when setting a status.

FWScheduleGroup

Attribute Name	Type	Field Size	Description	Required
ScheduleGroupKey	nvarchar		Business key identifier for the schedule group record. This must be an alphanumeric value without interleaving spaces.	Required
ExternalPatherURL			The URL of the LNOS Pathing Service that will be used for the group.	
AltExternalRoadRouter			Allow users to specify an alternate External Pather URL.	
ExtPathingViolationRadius	int		Specify a radius value for a circular boundary on the map in which violations will be ignored at the origin or destination when those violations would prevent the External Pather from determining a path between locations. When converting the entered distance value to time, the system uses a static conversion of 35 miles per hour. If no value for Ext Pathing Violation Radius is specified, Descartes Route Planner will send a default of 300 seconds (approximately 2.9 miles) to RMPI.	
UnconstrainedVehicleHazMat			Minimum hazmat value. 0: No hazardous type 1: All hazardous types	
UnconstrainedVehicleWeight			Minimum truck weight for pathing.	
UnconstrainedVehicleHeight			Minimum truck height for pathing.	
UnconstrainedVehicleLength			Minimum truck length for pathing.	
UnconstrainedVehicleWidth			Minimum truck width for pathing.	
UnconstrainedVehicleType			0: Passenger Vehicle 1: Straight Truck 2: Articulated Truck	
ConstrainedVehicleHazMat			Maximum hazmat value. 0: No hazardous type	

			1: All hazardous types	
ConstrainedVehicleWeight			Maximum truck weight for pathing.	
ConstrainedVehicleHeight			Maximum truck height for pathing.	
ConstrainedVehicleLength			Maximum truck length for pathing.	
ConstrainedVehicleWidth			Maximum truck width for pathing.	
ConstrainedVehicleType			0: Passenger Vehicle 1: Straight Truck 2: Articulated Truck	
NumTruckWeightSlots			Maximum number of vehicle weight slots that will be pathed between the minimum and the maximum.	
NumTruckLengthSlots			Maximum number of vehicle length slots that will be pathed between the minimum and the maximum.	
NumTruckHeightSlots			Maximum number of vehicle height slots that will be pathed between the minimum and the maximum.	
NumTruckWidthSlots			Maximum number of vehicle width slots that will be pathed between the minimum and the maximum.	

FWResTemplateSet

Attribute Name	Type	Field Size	Description	Required
ResourceTemplateSetID	int	4	Internal unique database record ID	
ResourceTemplateSetKey	nvarchar	50	Business Key identifier for this Resource Template Set Key (Required)	
Name	nvarchar	50	Name of this Resource Template Set	
Description	nvarchar	50	Description of this Resource Template Set	
Comment	nvarchar	500	Comment field	

Appendix B: System Value Definitions

Descartes Route Planner system values are described in the following table.

Keyword	Description
AdapiINI	File path for the Adapi ini file.
AdviseReturnAssetInfo	When enabled, Advise calls will return asset field data including: • CrewCount • TruckType • TrailerType
AllowableProfitDrop	Profit threshold value. When set, the system will not reject suggested assignments. Used to enable the BGO to force the assignment of unassigned orders that would otherwise be in violation when <code>com.descartes.escheduler.optimizer.force_assign</code> is set to "1".
AllowResDateChangeOnAssign	With <code>AllowResDateChangeOnAssign</code> enabled, the assignment process updates the route's Earliest Start Time to the user's current time when users assign an order to an empty route. For example, if a route is created at 06:00 and an order is not assigned to the route until 07:30, the route's Earliest Start Time value will be updated to 07:30, the user's system time.
AltExternalRoadRouter	Alternate map edit server URL used for RMPI routing.
ApplyDockDoorConstraint	Select the Loading Batches value to enable the Dock Door Planning feature. Select Assign Docks to have the optimization engine automatically schedule route starting times and recharges when the number of loading docks for a depot is limited. For example, if 20 vehicles are scheduled to begin at a depot but the depot only has 10 doors (<code>MaxDockDoors</code> set to "10" for the depot location), the system will now schedule only 10 of the 20 vehicles at the doors at the given time. The remaining 10 will be assigned to doors as they become available. The Assign Docks functionality considers fixed loading times at the vehicle and location levels as well as variable loading times based on the product loaded on the vehicle.

AssetOverlapAllowanceInMin	Sets a minimum allowance threshold for the asset overlap check. The asset overlap check excludes any routes added to an "Archive" schedule type. Archived routes do not impact the creation or handling of production routes. The asset overlap check also displays the ScheduleKey when executed for easier location of the conflicting asset.
AsyncUpdateStatus	When AsyncUpdateStatus is enabled (set to "1"), the UpdateStatus call saves the input message in the FWUpdateStatus database table and immediately returns a message ID that the caller can use to look up the processing result. Saved messages are processed in the background by the new DocRouteEditTask.AsyncUpdateStatus API, which is invoked by the Batch Processor. The Batch Processor configuration file provides a new task type of AsyncUpdateStatus as shown below. <pre><?xml version="1.0"?> <configuration> <applicationSettings> <FWBatchProcessor.Properties.TaskSettings> <taskSetting name="AsyncUpdateStatus" enable="true" pollingInterval="5" threadPoolSize="1" staggerFactor="0"/> </pre> The processing result of each message is saved in the FWUpdateStatus.Result column. If a message contains multiple DocRouteEditTask nodes, each processing result will be concatenated and up to 1000 characters of the result will be saved in the Result column. The Batch Processor will save the complete, non-truncated processing result in its log. Once processing is complete, a message in FWUpdateStatus will not be reprocessed even if it failed. This functionality aims to prevent cases where new messages continue to queue while older messages are retried. ManualStatus values entered in the user interface will be processed synchronously regardless of the AsyncUpdateStatus system value setting. If autopublish is enabled, once all status messages for a given route have been processed, a route update will be sent to Descartes wGLN automatically if the

	configuration thresholds have been exceeded. In asynchronous mode, a route will be autopublished regardless of the system value setting as long as the publish criteria is met.  Note— To use this feature, autopublishing in Descartes wGLN should be disabled.
AutoPublishOnUpdateStatus	AutoPublishOnUpdateStatus enables or disables autopublishing of routes after processing a status update message. • 0: Do not auto-publish route at the end of UpdateStatus. • 1: Auto-publish route at the end of UpdateStatus. Route is auto-published after a dispatch message is applied and the route ETAs are refreshed.
BackhaulDZDistance	When BackhaulDZDistance is set to "1", enables Backhaul Dynamic Zoning Distance logic for the advise filter.
BatchProcessorLogin	BatchProcessor organization login record.
BL_*	If SQL Parser is selected in the SQLScanPolicy CtySysValue, BL_* sysvalues are used to configure blacklists. These values are not intended to be modified by users; edits will cause certain Descartes Route Planner APIs to fail.
BGOTimeout	Sets the number of minutes until the system will can assign a data slice to the BGO once a route is flagged as completed.
CheckValue	If Blacklist is selected in the SQLScanPolicy CtySysValue, the CheckValue sysvalue is used to configure blacklists.
ConversionXSLPath	File path of the XSL Transform.
DefaultTimeZone	Sets the default time zone for the system from a drop-down menu.
DisableUILatLong	When DisableUILatLong is set to "1", all latitude and longitude columns are hidden in quadrants, location forms and map displays.
DispatchBGO_FreezeWindowSize	Value set in seconds. Pending stops scheduled before the LastReportedDate value plus the FreezeWindowSize value will not be moved within the route or to another route. Default 7200.

DispatchBGO_MinStopsToFreeze	Sets the minimum number of pending stops that will be frozen after the LastReportedDate. Default 0.
DispatchBGO_AllowDeliverySwap	Allows deliveries to be moved to another route when set to "1". Default disabled.
DSGGeocoderThreshold	In cases where the GCPI or GeoMicro engines do not return a high enough geocode score, the system will use DSG Geocoder services when the DSGGeocoderThreshold score is below the set value. Default Value: 75
DumpPublishToWGLNInfo	When DumpPublishToWGLNInfo is set to "1", the system will log Descartes wGLN HTTP posts in the directory defined in the DumpPublishToWGLNInfoPath sysvalue.
DumpPublishToWGLNInfoPath	Directory for the Descartes wGLN dump publish log.
DumpRmpiDebugInfo	When DumpRmpiDebugInfo is set to "1", the system will log RMPI debug information in the directory defined in the DumpRmpiDebugInfoPath sysvalue.
DumpRmpiDebugInfoPath	Directory for the RMPI dump publish log.
EncryptGPS	When set to "1", latitude and longitude values in the FWGPSSStatus database table will be encrypted when GPS points are saved.
ExternalRoadRouter	The URL that determines which LNOS Pathing Service will be used to process requests for a Schedule. The ExternalRoadRouter sysvalue will be considered for a schedule only when no External Road Router value is set at the schedule or schedule group level. Click the Test button to ensure successful connectivity to the external pather via the entered URL. The version number of the external pather displays in a dialog.
FastPublish	When enabled, a read will be taken of the database without waiting for all other database transactions to finish. For routes that are actively modified, this may return incorrect data, so use this setting with caution. FastPublish works best for routes that are no longer being updated.
FilterWithScheduleKey	When enabled, Advanced Find will function across schedules (ScheduleKeys).
FormatXSLPath	File path for the XSL Formatting Object.
GcpiINI	File path for the Gcpi ini file.

GeoScoreScaleFactor	Sys value to set a multiplier for geocode score output.
GrpPickupOrdFromUnassign	When set to "1", the system will group the stops of unassigned double-ended orders added to the end of a route. By default, this setting is disabled.
IntraForBatchProc	In the [Save Update] Order APIs, orders are auto-assigned to their preferred routes if they are either: • Unassigned orders where the preferred resources are newly set or changed to a new resource • Assigned orders where the schedule key has been changed and preferred resource is not null When IntraForBatchProc is set to "1", routes are resequenced after auto-assignment. Orders that cannot be auto-assigned are force-assigned.
IntraOnReturns	Previously, Descartes Route Planner resequenced a route in response to a "Missed" message for a complex order when the CreateReturnOnMiss schedule setting was set to "1". How the system handles this case is now configurable via the new IntraOnReturns system value. This system value has the following settings: • 0: Default Value. The system will not resequence a route in response to a "Missed" message for a complex order when the CreateReturnOnMiss schedule setting is enabled. Descartes Route Planner will find the best position by inserting the return delivery next to the pending stop that shares the same location. If no stop is found that shares the same location, the return is placed at the end of the route. • 1: Descartes Route Planner will perform an intra so that RMPI finds the best place to position it.
IntraOnStatusForBatchProc	Allows the resequence of a route upon the reception of a status (complete or equivalent of complete). • 0: Not active • 1: Creates a record in the OptTask table to resequence the route. The resequence is performed by the batch processor service. Default=1.
LanguageCulture	Sets the language to be used by the geocoder and resolvegeocoder. Valid input values can be found in this table .

LastNStopForDZ	When LastNStopForDZ is set to a value greater than zero, dynamic zoning will only include routes with the last "N" stops falling within the dynamic zone. Empty routes are not returned in this case.
MaxEmptyRoutes	If MaxEmptyRoutes is set, then the BGO can include up to this number of empty routes while searching for compatible used routes for batch assign.
MaxIdleTime	Sets the maximum allowable idle time. Idle time at the stop level is calculated using GPS Status messages received between a stop's Arrive and Complete or Departed status. The GPS messages used are all those that belong to the same chain where the Ignition=true and Speed=0. Several chains may exist within the window between Arrive and Complete or Departed status.
MaxRetriesForBatchProcessor	Sets the maximum number of automatic retries that will be performed for the Batch Processor on errors. When sending data to wGLN, if no retries are specified, the system will try to resend the data three times on deadlocks, COM timeouts and similar internal errors.
MaxRetriesForRmpiCalls	Sets the maximum number of automatic retries that will be performed for RMPI calls in error. When sending data to wGLN, if no retries are specified, the system will try to resend the data three times on deadlocks, COM timeouts and similar internal errors.
MaxRetriesForUpdateStatus	Sets the maximum number of automatic retries that will be performed for UpdateStatus errors. When sending data to wGLN, if no retries are specified, the system will try to resend the data three times on deadlocks, COM timeouts and similar internal errors.
MaxScoreToAutoCorrect	The system will automatically correct the latitude/longitude of a customer site based on the reported GPS latitude/longitude contained in the status messages if the current Geocode Score for the customer is less than the value specified. If the reported latitude/longitude in a status message (see AVLParms) is to be considered for updating a customer, then FWLocation.GeocodeLatitude, GeocodeLongitude and the AutoCorrectLatitude/Longitude is updated with the last reported status GPS If the FWLocation.GeocodeScore value is less than or equal to the MaxScoreToAutoCorrect value and the latitude/longitude is not locked, then

	FWLocation.GeocodeLatitude, GeocodeLongitude and the AutoCorrectLatitude/Longitude is updated with the last reported status GPS Otherwise, the latitude/longitude reported in status GPS is saved in the AutoCorrectLatitude/Longitude. Route directions and the optimization will use the FWLocation.GeocodeLatitude and Longitude values.
OptimizeTimeout	Sets the maximum amount of time allowed before an optimize operation times out.
ProcessAsync	When ProcessAsync is set to "1", all Save, Update, Create, Remove Order and Update Resource commands are processed asynchronously by the batch processor. This option is also available when importing master routes, but asynchronous processing will not occur if a resequence is expected or there is no update to planned values.
PublishBatchSize	Allows users to specify the number of routes the system will publish to Descartes wGLN at one time when the Publish, Auto Publish or Publish to wGLN operations are selected for multiple routes. For example, if PublishBatchSize is set to "5" and the user selects 20 routes, the system will send publish routes in four batches of five routes.
RechargeDepotRadius	Works in conjunction with the UseDepotLimit sysvalue. When the UseDepotLimit sysvalue is enabled, the RechargeDepotRadius value (in meters) is used to create a bounding box of all latitudes and longitudes of all locations involved in a database set. Only the depots contained in this bounding box are sent to RMPI.
RmpiINI	Path for the RMPI ini file.
RmpiINI1_[type]_Q1	The installer updates the sysvalues rmpiINI[#]_[type]_Q1 from rmpi*.ini to rmpi.ini, allowing all RMPI settings to be consolidated into one file.
RTraceFilePath	File path to the RTraceLevel log.
RTraceLevel	When RTraceLevel is set to "1", all calls to Descartes Transportation Manager will be logged.
SendMessageToWGLN	SendMessageToWGLN allows text messages to be sent directly to Descartes wGLN when the send options are selected from the New Messages page. This setting has four possible values:

	• 0: do not publish • 1: publish to the publish setting only (default) • 2: publish to wGLN only • 3: publish to both publish setting and wGLN setting
SingleRoutePerTemplateForAdapi	When SingleRoutePerTemplateForAdapi is set to "1", the system selects empty routes based on ResourceClass, Date and requirements. This setting eliminates duplicate empty resources loaded to the Adapi component.
SQLScanPolicy	Descartes Route Planner version 16.03 introduces two methods to protect data from SQL injection: Blacklist and SQL Parser. Users can configure which method to use by entering one of the following values in the SQLScanPolicy CtySysValue. This setting has four possible values: • 0: Do not scan SQL statements. • 1: Blacklist • 2: (Reserved) • 3: SQL Parser (default for new installations) Both methods are described in detail below. Blacklist: Descartes Route Planner designates a set of SQL keywords, table, view, column and function names and special symbols as a blacklist and scans an externally provided SQL string to ensure the string does not contain matching values. If one or more blacklisted names are found in the string, the system fails the associated API. Please note that since the scan is a simple string search without considering the structure of a SQL statement, the scan can be too restrictive in that it denies the possibility of allowing a name in the selection list but not in the where-clause. Another defect of this method is that the blacklist can become large and difficult to manage. The blacklist is configured in the CheckValue CtySysValue. SQL Parser: The SQL Parser is a parser made of a lexical analyzer and a SQL grammar parser. Currently the parser accepts only a subset of Microsoft® T-SQL SELECT statements. Since the parser understands the SQL grammar, it can easily extract object names used in a given SQL string and also identify their types (i.e. table, column, function, string literals among others). The system checks such extracted names and types to determine if the given string is safe to use as a part of an embedded SQL. For this, Descartes Route Planner uses whitelists of tables, views, functions and

	columns and blacklists of columns and string literals. For an input string to be considered safe, all the names and string literals in the string should belong to whitelists but not blacklist. If the system finds otherwise, it fails the API. Whitelists are configured with WL_* CtySysValues. Blacklists are configured with the BL_* CtySysValues.
Suggest_MaxRoutesPerDay	Maximum number of routes per day to be considered for suggest.
TurboAdvise	Activates turbo advise functionality when set to "1".
TurboAdviseStem Distance	Maximum distance (in meters) that the initial location of an empty route can be from the orphan
TurboAdviseMaxRoutesPerDay	Maximum number of used routes that can be sent to the Adapi component per day. This does not include empty routes.
UseAPITracking	Enables the API tracking feature.
UseDepotLimit	Works in conjunction with RechargeDepotRadius. When enabled, the number of depots sent to RMPI can be limited with a value in the RechargeDepotRadius sysvalue.
UseExternalRoadRouter	Enables the set it to '1' to use the Telogis-based router. Default is 0.
UseParty	Enables Party User functionality.
UseRmpiAdvise	Determines which advise engine will be used: • 0 : Use Adapi • 1 : Use RMPI Rmpi advise is slower when higher volume of orders are required. Default value is 0.
UseSPInAdviseFilter	Determines whether or not the FWAdvTopNResClass stored procedure is used with the Advise filter query for selecting empty routes. • 0: the system uses the dynamic query for the advise call • 1: the system uses the FWAdvTopNResClass stored procedure
WL_*	If SQL Parser is selected in the SQLScanPolicy CtySysValue, WL_* sysvalues are used to configure whitelists. These values are not intended to be modified by users; edits will cause certain Descartes Route Planner APIs to fail.

Appendix C: Advise Component Settings

Assumptions:

- Single and multi-day routes
- Locationless jobs supported.
- Driver breaks supported based on time window range per resource
- Rests supported
- Re-charging supported
- Supports interspersed pickup and delivery
- Supports source/follow-up jobs, interspersed or not (multiple source pickups followed by one delivery)
- Single time window on stops
- (3-14-12) Multiple Imbs with optional map edit files. Adapi will choose the best to use on each Advise call and use estimated driving times and distances where needed for any stops not on the best map.
- One set of routing parameters, but rush-hour speed degradation factors allowed
- Stage cache database table available for performance
- Static zones supported including blank zone on resource
- Dynamic zoning supported
- Supports commodity conflicts per route and per trip
- Supports specification of stop as first or last on route, and stop sequence priority
- (7-22-11) Supports implied pre-route/post-route time (i.e. start-up/wind-down time) on pre-existing routes. Latest arrival time at end of route is assumed to be completion-based.

Depots (currently needed only if re-charging is allowed)

Data item	Type	Size/range	Description
Name	Char	50	Id information. If depot appears as a recharge stop on a pre-existing route, this name must be used for the associated stop name.
ID	Int	4 byte	Unique identifier for debugging only.
Allow recharging	Boolean	Yes/no	If value is yes, the following re-charging fields apply. Otherwise those fields are ignored. (Default=no)
Latitude, longitude	Float	Valid coordinates	Latitude-longitude in decimal degrees.
Requirements	Char	255	Blank-delimited strings. Resource capabilities must include all of the strings in this field if this depot is to be used for re-charging the resource.

Earliest recharge time	Time of day	00:00-23:59	Earliest arrival time at depot for re-charging. (not yet implemented)
Latest recharge time	Time of day	00:00-23:59	Latest arrival time at depot for re-charging. (not yet implemented).
Measures supplied[9]	Boolean	Yes/no	Element value is yes if the associated measure can be recharged at this depot
Recharge duration	Int	>=0	Fixed time in seconds for re-charging at this depot (to be added to recharge rate per unit and additional resource recharge duration specifications) (Default=0)
Recharge rate[9]	Float	>=0	Recharge rate in seconds per unit for each rechargeable measure. (Default=0) As of 12-01-10, this is units per second and is floating point.

Resources

Data item	Type	Size/range	Description
Name	Char	50	Id information for debugging only
ID	Int	4 byte	Unique identifier
Capabilities	Char	255	Blank-delimited strings for matching job requirements
Earliest service date/time	DateTime	Valid date/time on or before latest end date/time	Earliest date/time on which servicing a job can start.
Latest service date/time	DateTime	Valid date/time on or after dispatch date/time	Latest date/time on which servicing a job can start.
Dispatch date/time	DateTime	Valid date/time	Earliest dispatch date/time
Latest end date/time	DateTime	Valid date/time	Latest ending date/time of route
Zone	Char	50	If honoring zones, only wildcard jobs or jobs in same zone will be assigned to this resource (should have been filtered by database call). If zone is '*' (wildcard), any

			stops may be assigned to the resource regardless of zone. If the zone is blank, Advise will assume the zone of the resource is the most frequently occurring zone of the stops assigned to the route. If the resource has no stops assigned, the initial zone will assumed to be '*' (wildcard) in this case.
Parking seconds	Int	≥ 0	Parking time at each stop
Maximum seconds duration	Int	≥ 0	Maximum total time for route
Maximum meters distance	Float	≥ 0	Maximum total driving distance for route
Maximum meters non-stem distance	Float	≥ 0	Maximum total driving distance for route excluding stems to and from the depot
Maximum meters radius	Float	≥ 0	Maximum distance between any two stops (i.e. dynamic zone radius override if differs from global dynamic zone radius and is between 0 and 3000 miles) This value is ignored if the DynamicZoning parameter is equal to 0.
Dynamic Zone Flag	Int	-1 to 3	Determines how depots are used to compute the dynamic zone of a route. -1 (default) – use the DynamicZoneFlag parameter to determine the value 0 – Do not include depots 1 – Include initial depot only 2 – Include final depot only 3 – Include both initial and final depot
Maximum inter-stop distance	Float	≥ 0	Maximum distance between any two consecutive stops
Maximum number of stops	Int	> 0	Advise will not assign more stops than this maximum to a route. Each job is a stop for this computation.
Maximum number of distinct stops	Int	> 0	Advise will not assign more pickup and delivery locations (as identified by their associated latitude-longitude) than this maximum to a route.

Missed time window penalty cost per minute	Float	≥ 0	Not used at this time. Advise won't intentionally break windows or increase amount of pre-existing broken windows
Minimum driver pay	Float	≥ 0	Assumed minimum cost of route
Driver pay per order	Float	≥ 0	Additional cost per stop
Minimum number of paid orders	Int	≥ 0	Use this value together with preceding value instead of Minimum driver pay field.
Cost per hour[9]	Float	≥ 0	Incremental route cost per hour, used with following field
Cost per hour step time[9]	Float	≥ 0	Increments for preceding field
Driving time adjustment factor	Float	> 0	Multiplier to nominal driving time
Capacity adjustment factor	Float	> 0	Multiplier to size array
Capacity[9]	Float	≥ 0	Size of vehicle in up to 9 measures
Minimum seconds before break	Int	≥ 0	Minimum elapsed time between breaks
Maximum seconds before break	Int	≥ 0	Maximum elapsed time between breaks
Break seconds	Int	≥ 0	Break duration
Maximum number of breaks	Int	≥ 0	Maximum number of breaks per day
Minimum seconds before rest	Int	≥ 0	Minimum elapsed time between rests
Maximum seconds before rest	Int	≥ 0	Maximum elapsed time between rests

Maximum driving seconds before rest	Int	≥ 0	Maximum total driving time between rests
Rest seconds	Int	≥ 0	Rest duration
Recharge duration	Int	≥ 0	Fixed time in seconds for re-charging this resource (to be added to additional recharge duration specifications for the recharging depot) (Default=0)
Recharge rate[9]	Float	≥ 0	Recharge rate in seconds per unit for each rechargeable measure. (Default=0) – Not used as of 3/7/05 for recharging. As of 11-30-10, used to compute service time for stops on the resource. Now is 'units per second' and is floating point.
Work units	Float	8 bytes	Maximum number of work units per route (default = 25000.)
Rush hour time windows[6]	Time of Day	00:00-23:59	Time window specified by an earliest and latest time for which a corresponding rush hour degradation factor will apply.
Rush hour degradation factor[6]	Float	> 0	Multiplicative factors corresponding to the rush hour time windows. Any point-to-point driving time that begins within a rush hour time window will be multiplied by the corresponding factor.
Initial stem cost factor	Float	0.-1.	Multiplicative factor applied to cost of driving from depot to first stop on a route. (default=1.)
Final stem cost factor	Float	0-1.	Multiplicative factor applied to cost of driving depot last stop on a route to the depot. (default=1.)
Maximum distance to the depot	Float	≥ 0 .	3-5-12 This value is ignored if it is 0. If it is greater than 0. and the global maximum distance to the depot is 0, then a stop will not be assigned to a resource unless it is estimated to be within this distance from the originating depot (initial stop). (default=0.)

Resource (route) stops

Data item	Type	Size/range	Description
Customer id	Int	4 byte	Id for debugging only
Name	Char	50	Id for debugging only if not recharging. For recharge stops, must match the name of one of the supplied depots.

Stop type	Int	0-5	Initial, Final, Delivery, Pickup, Locationless, Recharge, Source, Follow-up.
Latitude, longitude	Float	Decimal degrees	Location of stop (unused for locationless)
Size[9]	Float	>=0	Order size at stop in up to 9 measures
Revenue	Float	>=0	Gross profit for servicing the stop
Open time	Date/Time	Valid date/time	Earliest arrival date/time at stop
Close time	Date/Time	Valid date/time	Latest arrival date/time at stop
Service seconds	Int	>=0	Service time at stop
Arrival time	Date/Time	Valid date/time	Arrival date/time at stop
Begun time	Date/Time	Valid date/time	Service start date/ time at stop
Completion time	Date/Time	Valid date/time	Service completion date/time at stop
Departure time	Date/Time	Valid date/time	Departure date/time from stop
Wait seconds	Int	>=0	Waiting time required between arrival and stop open time
Break seconds	Int	>=0	Number of seconds for break at the stop (not yet used)
Rest seconds	Int	>=0	Number of seconds for rest at the stop(not yet used)
Delta driving seconds	Int	>=0	Driving time from previous stop
Delta driving meters	Float	>=0	Driving distance from previous stop

Delta driving cost	Float	>=0	Driving cost from previous stop
Freed space fraction (DeliveryRatio in RMPI)	Float		For delivery orders, fraction of delivery order size space that will become available for subsequent pickup orders after serving this job. If negative, will imply that more was picked up than delivered. Size on truck after delivery will be original order size - freed space fraction * original order size. (1.)
Revenue depreciation percent per hour	Float	-100. - +100.	Percent of revenue received will depend on total work time of the route so far at the arrival time for this job. If this value is 'p', and total work time at arrival is n hours, then revenue received = $((100.- n*p) * .01) * \text{revenue}$.
Is Interspersable	Boolean	Yes or no.	Applies to source/follow-up stops when the global 'OverrideInterspersable' setting is 0. If value is no, source and related follow-ups must appear consecutively on a route. (default=yes)
Closing time is latest completion	Boolean	Yes or no	If value is no, closing time represents latest arrival time at the stop. If value is yes, closing time represents latest service completion time at a stop. Will be ignored unless global closing time type has a value of 2. (default=no)
Work units	Float	8 byte	Number of work units associated with this stop.

			Will be used to limit maximum number of work units per resource. (Default = 0.)
Commodities	Char	255	Space-delimited string of commodity identifiers associated with the jobs at the stop.
Sequence priority	Int	>=0	If this is a positive number, determines the priority of stop in the service sequence. Stops must be serviced in sequence priority order. Zero is ignored. (Default=0)
Load position	Int	0,1,2	0=Ignore 1=Stop must be first in service sequence 2=Stop must be last in service sequence (Default=0)
Attributes	Int	4 byte	Bit map. Bit definitions: 1 – Recharge required before this stop. 2 – Stop is not included in dynamic zone computation (Default=0)

New Job for which to find best time slots

Data item	Type	Size/range	Description
ID	Int	4 byte	ID for debugging
Name	Char	50	Name for debugging
Customer Id	Int	4 byte	ID for debugging
Customer name	Char	50	Name for debugging
Zone	Char	50	Job will only be considered for

			assignment to resources with matching zone(should have been filtered before input by database call). If zone is blank or equal to '*', all resources will be considered for assignment regardless of zone. For jobs with multiple source orders, the zone of the source orders is ignored.
Latitude, longitude	Float	Decimal degrees	Latitude-longitude of delivery location
Requirements	Char	255	Blank-delimited strings used for matching to depot and resource. Job will only be considered for assignment to resource and depot with matching capabilities
Size[9]	Float	>=0.	Order size in up to 9 measures
Type	Int	2,3,6 or 7	2= Delivery , 3=pickup ,6=Follow-up delivery,7=source pickup
Revenue	Float	>=0.	Gross profit for serving this location
Freed space fraction (DeliveryRatio in RMPI)	Float		For delivery orders, fraction of delivery order size space that will become available for subsequent pickup orders after serving this job. If negative, will imply that more was picked up than delivered. Size on truck after delivery will be original order size - freed space fraction * original order size. (1.)
Revenue depreciation percent per hour	Float	-100.- +100.	Percent of revenue received will depend on total work time of the

			route so far at the arrival time for this job. If this value is 'p', and total work time at arrival is n hours, then revenue received = $((100.- n*p) * .01) * \text{revenue}$. (Default=0.)
Interspersable	Boolean	Yes or no	Applies to source/follow-up jobs when the global 'OverrideInterspersable' setting is 0. If value is no, sources and follow-up must appear consecutively on a route. Otherwise, sources may be interspersed with other stops on the route preceding the follow-up. (default=yes)
Closing time is latest completion	Boolean	Yes or no	If value is no, closing time represents latest arrival time at the stop. If value is yes, closing time represents latest service completion time at a stop. Will be ignored unless global closing time type has a value of 2. (default=no)
Work units	Float	8 byte	Number of work units associated with this job. Will be used to limit maximum number of work units per resource. (Default = 0.)
Commodities	Char	255	Space-delimited string of commodity identifiers associated with the job.
Sequence priority	Int	>=0	If this is a positive number, determines the priority of stop in the service sequence. Stops must be serviced in

			sequence priority order. Zero is ignored. (Default=0)
Load position	Int	0,1,2	0=Ignore 1=Stop must be first in service sequence 2=Stop must be last in service sequence (Default=0)

Time window buckets

Data item	Type	Size/range	Description
Id	Int	4 byte	Bucket id
Date	Date		Date of delivery
Start time	Time of day	00:00-23:59	Earliest arrival
End time	Time of day	00:00-23:59	Latest arrival
Profit Adjustment Factor	Float	8 byte, > 0	Multiplicative factor applied to incremental profit for inserting a stop at a particular slot and offering this bucket.. (Default=1.)

Commodity Restrictions

Data item	Type	Size/range	Description
Commodity	Char	255	Commodity ID
Incompatibilities	Char	255	Space-delimited list of commodities that may not be on the same route/trip (depending on setting) with Commodity above.

Settings from Current Scheduler Advise (defaults in [Settings] section of adapi initialization file)

- ReturnMaxSlotProfit
- SameStopSlots – Revised values, 8-18-08, see in table below
- TopSlotProfit
- BottomSlotProfit
- TopSlots
- BottomSlots

Settings from RMPI (defaults in [Settings] section of adapi initialization file)

- NumMeasures
- NumHourlyCosts

New Advise Settings (defaults in [Settings] section of adapi initialization file)

Setting	Description
MaximumCloseSites	Maximum number of nearest sites to orphan for which an along the road path will be computed (default=30)
InitialProximityFactor	Fraction of diagonal distance of bounding box enclosing route candidates for selecting close stop candidates. (default=.3)
MaximumSitesMultipleForPathInput	MaximumCloseSites multiplier for limit on number of input sites to the one-to-many path routine. (default=2)
MaximumSecondsInPathComputation	Limit on path calculation time (default=.3)
ScheduleBreaksBasedOnResourceSettings	If value is 1, break parameters defined for each resource will be used to schedule driver breaks. If value is 0, no driver breaks will be scheduled. (default=1)
AllowPickupsBeforeDeliveries	If value is 1, pickup orders may be inserted before any deliveries on the route. (default=1)
HonorZones	If value is 1, resource and job zone designations will be considered in selected routes for assignment. If value is 0, zones will be disregarded. (default=1)
DynamicZoning	If value is 1 or 2, DynamicZoningRadiusInMeters will be used to

	determine feasibility of assigning a job to a resource. See description below. (default=0)
DynamicZoningRadiusInMeters	When DynamicZoning=1 , a resource will be considered for assignment if the estimated driving distance between the stop to be assigned and all of the non-source, pre-existing stops is less than the value of this setting. That is, all stops on a route must be close to each other. DynamicZoning=2 , a resource will be considered for assignment if the estimated driving distance between the stop to be assigned and any one of the pre-existing non-source stops is less than the value of this setting. That is, each stop on a route must be close to an adjacent stop in the route sequence. (default = 3000. miles)
DynamicZoneFlag	Determines how depots are used to compute the dynamic zone of a route. 0 – Do not include depots (default) 1 – Include initial depot only 2 – Include final depot only 3 – Include both initial and final depot
PreserveSourceOrder	If set to 1, engine assumes that a route which services a job consisting of multiple source orders and one follow-up will service the source orders in the sequence presented in the Advise call. (default=0)
NeighboringSlotsOnly	If set to 1, Advise will only consider inserting a stop on a route at a position where the preceding or following stop is 'nearest' neighbor of the stop to be inserted. This means the engine will not return a slot that depended on a 'crow flies' estimate for driving time to both the preceding and following stop on the suggested route. (default = 0)
OverrideInterspersable	Ignored if value is 0. If value is 1, then all source/follow-up jobs will be considered to be interspersable regardless of setting on the individual jobs. If value is 2, then all source/follow-up jobs will be considered to be non-interspersable regardless of the setting on individual jobs. (default=0)

ClosingTimeType	Ignored if value is 2. If value is 0, all closing times will be interpreted as the latest arrival time at a stop. If value is 1, all closing times will be interpreted as the latest service completion time at a stop. (default=0)
MaximumStopsPerRoute	If value is positive, it represents the maximum number of stops (each job is one stop) on any route. If the maximum number of stops defined for a resource is less than this value, it will be honored. (default=0)
MaximumDistinctStopsPerRoute	If value is positive, it represents the maximum number of distinct pickup and delivery locations (as defined by their associated latitude-longitude) on any route. If the maximum number of distinct stops defined for a resource is less than this value, it will be honored. (default=0)
MaximumTotalRouteMeters	If value is positive, it represents the maximum total driving distance for any route. If the maximum distance defined for a resource is less than this value, it will be honored. (default=0)
MaximumNonStemRouteMeters	If value is positive, it represents the maximum total driving distance for any route excluding the drive time from the depot to the first stop and from the final stop to the depot. If the maximum non stem distance defined for a resource is less than this value, it will be honored. (default=0)
ServiceTimeAdjustmentFactor	Nominal service time for orphans will be multiplied by this factor for the purposes of estimating schedules. (default=1.) Included in versions after 8-21-04
CommoditiesPerTrip	If value is non-zero, commodity restrictions apply per trip (stops between re-charges). Otherwise restrictions apply per route.
FixedDispatchTime	Start times on pre-existing routes may not be modified. (Default=0)
AlwaysReturnASlot	If value is non-zero, the Advise call will always return a slot even if there are no feasible insertion positions on any route. In the case of no feasible insertions, the resource with the closest stop to the job to be assigned will be returned with a position at

	the closest stop. The bucket will be one that includes the arrival time at the closest stop. If no such bucket exists, the earliest bucket will be chosen. The score returned will be 0. (default=0)
SameStopsSlots	If value is non-zero, determines special handling for cases where the orphan has the same latitude-longitude of a pre-existing stop on a route. The following non-zero values are acceptable: 1-Ignore infeasibilities for slots that precede or follow the routed stop at the same location. 2-Same as 1, but will respect an infeasibility for the case where the order size of the orphan exceeds the capacity of the vehicle in any measure. 3-Same as 1, but will respect all vehicle capacity infeasibilities.
DefaultResourceRechargeRates	=1 to allow specification of default recharge rates on resources in the .ini file . (default=0) (12-2-10)
DefaultResourceRechargeUnitsPerSecondm	If above setting is equal to 1, specifies the default units per second of incremental service time for a resource that has no recharge rates associated. (12-2-10)
ResourceCostPerHourAdjustmentFactor	Nominal resource costs per hour will be multiplied by this factor. (default=1.) This is to accommodate a problem where the input was not scaled correctly in earlier releases. Included in versions after 6-23-11
SameStopSlotsWorktimeWithBreaks	If this setting is 1, Adapi will include the an approximate driver break time in the calculation of total work time for determining the incremental cost of adding an orphan at the same location as an existing stop on a route. (Default=0) 6-29-11
MaximumDistanceToDepotInMeters	If this setting is greater than 0 (default=0.) a stop will not be assigned to a resource unless it is estimated to be within this distance from the originating depot (initial stop). 3-14-11

Debug settings (defaults in [Settings] section of adapi initialization file)

DebugLevel	Controls detail level of debugging information stored in an ASCII text file. Integer from -1 to 5. If the value is -1, no debug file will be created. The higher the value, the greater the level of debugging detail generated. (default = 1)
DebugFileName	Complete path and file name of debugging text file. (default = AdviseEngineLog.txt)
AppendProcessId	6-16-11 If this number is 1, debug log files created by the engine will have the current process ID appended to the file name to ensure uniqueness among currently running processes. The process ID will be separated from the name with an underscore (_). If the number is 0, the process ID will not be appended to the file names. Default is 0 for compatibility with previous versions.
DebugFileOpenOptions	String passed as argument to fopen() function. In general, use 'w' to create a new file at initialization time and use 'a' to append to the pre-existing file. Do not include the surrounding apostrophes (default='w')
AlwaysFlushDebugFile	Set to 1 and the debug file buffer will be flushed after every write. Should only be used for debugging due to performance degradation when this is in effect. (default=0)
Debug Customer	Customer name for which intense debugging should be done. When an Advise call is made for a job with a matching customer name, very detailed debugging output will be written to the debug text file. (default is empty string)
DisplayScheduleForTopBucket	If value is 1 and DebugLevel is ≥ 2 , at the end of the 'advise' call, the schedule for the resource representing the best 'bucket' will be displayed on the debugging file as it would appear after inserting the jobs to be assigned. This can be used to compare the schedule obtained after a confirm.

MessageBoxOK

If value is 1 and a certain system errors occur, the component will display a Windows message box with an error message. If the value is 0, no message box will be displayed for these errors. An error message will be written to the debug file and an error return code will be send to the calling routine. (default = 0)

Path engine (papi) settings (defaults in [Papi] section of adapi initialization file)

File	Full path name of the .lmb file
Filenn	Additional lmb file specifications. Replace <i>nn</i> with a sequence number, leading zeros optional. 3-14-12
UseMapEdits	Value = 1 if a set of map edits should be applied to the .lmb, 0 otherwise (default=0)
MapEditPath	Path to folder where map edit files are stored. If this is specified, adapi will look in this folder first for map edit files. If not found, adapi will look in the same folder as the map (.lmb) files. Map edit files will have the extensions .lpe and lme with the same file name as the associated lmb. 3-14-12
LPE	Full path name of the portable map edit file to use if UseMapEdits is 1. (Obsolete as of 3-14-12)
LME	Full path name of the map edit file to use if UseMapEdits is 1. (Obsolete as of 3-14-12)
TopMapLevel	Like current RMPI
AccessFromBothSides	Value is 1 if a route leg may start and finish on either side of the street. Value is 0 if a route leg must start on the same side of the street as the origin and finish on the same side of the street as the destination. Note that the current engine does not do geo-coding of addresses and uses latitude-longitude as the sole identification of a location. Until this is changed, this switch will only offer partial functionality.
BufferSize	Like current RMPI
MetersPerSecnnmmmm	Routing parameter speed for road type nnn and population density mmm. (Road types: 1=Limited access, 2=Primary connecting, 3=Secondary connecting, 4=Local, 5=Access ramp, 6=Service road, 7=Ferry, 8=Jeep trail, 9=ally, 10=walkway, 11=traffic circle)

	(Population densities: 0=Rural, 1=Suburban, 2=Semi-urban, 3=Urban, 4=High density) Alternate keywords: 1) MilesPerHour 2) KMPerHour
CostPerMeternnnnmm	Routing parameter cost per meter for road type nnn and population density mmm. Alternate keywords: 1) CostPerKMnnnnmm to enter cost per kilometer and 2) CostPerMile to enter cost per mile.
CostPerSecond	Routing parameter cost per second. Alternate keyword: DrivingCostPerHour to enter driving cost per hour.
UseRoadDistance	If no, will estimate all driving times and distances
StandardCurvature	CrowFlies multiplier (default = 1.18)
Acceleration	Controls rate of acceleration for computation of driving time estimates. The higher the value, the slower the rate of acceleration. Must be greater than 0. (default = 7.)
MaxMPH	Maximum driving speed (miles per hour) for computation of driving time estimates. (default = 55.) Alternate keywords: 1) MaxKPH to enter kilometers per hour and 2) MaxMPS to enter meters per second
MinMPH	Minimum driving speed (miles per hour) for computation of driving time estimates. (default = 3.75.) Alternate keywords: 1) MinKPH to enter kilometers per hour and 2) MinMPS to enter meters per second
CostPerMile	Driving cost per mile used for estimates and to simplify profit calculations (default=1.) Alternate keywords: 1) CostPerKM to enter driving cost per kilometer and 2) CostPerMeter to enter driving cost per meter.
CostPerHour	Driving cost per hour used for estimates and to simplify profit calculation (default=18.)
CostPerTurnnnn	Cost of turn type nnn (1=U-turn, 2=Left, 3=Shart left, 4=Right, 5=Sharp right, 6=Bear left, 7=Bear right)
SecondsPerTurnnnn	Time for turn type nnn
Restrictednnn	Value is 1 if vehicles may drive on restricted road type nnn. Value is 0 otherwise. (default = 1 for all types) (0=walkways, 1=trails, 2=ferries, 3=military, 4=buses, 5=private roads, 6=non-commercial only)

EstimatedDrivingTimeAdjustmentFactor	Multiplier used to fine tune standard estimates
UseStageCache	Value = 1 if router stages are to be saved and retrieved from a Microsoft® Corporation's SQL Server® database table. Value is 0 otherwise. (default=1)
StageDSN	ODBC datasource name of router stage cache. (default=StageCache)
StageTable	Name of database table in which router stages are stored (default = Stage)
StageDB	Name of database table in which router stages are stored. May be omitted if this is default database associated with ODBC DSN.
StagePassword	Optional password for accessing router stage cache database
StageUserId	Optional database user id for accessing router stage cache database
AllowStemEstimates	Value =1 if returned slots from the advise call may be based on estimated driving times and distances between the depot and the orphan. Value = 0 if such driving times and distances must be calculated using the road network. This setting provides a trade-off computational speed for accuracy. This computation will be done regardless of the value for MaximumSecondsInPathComputation. The default value is 1.
AllowEstimates	3-5-12 Value=1 means if a valid .lmb file is not specified or Adapi cannot find or open the specified .lmb file, Adapi will use all estimates for the driving time and distance calculations and will not return an error message during initialization. If the user wants the error message and does not want this new behavior, then set AllowEstimates to 0. The default value is 1.

Appendix D: Housekeeping Information

Monitoring Descartes Route Planner Components

Descartes Route Planner has a utility application that allows users to manually reset the Descartes Route Planner components; or users can choose to have Windows Schedule Tasks watch over these components and trigger an automatic maintenance task to the Descartes Route Planner Components.

This utility is typically used as a monitoring tool to make sure the Descartes Route Planner Servers are up and running and performing at optimal levels.

Manually Running the RestartCom.exe

This utility is located in the c:\LNOS\bin directory and it is called *RestartCOM.exe*. This is a small C# application used to restart LNOS FW Application components.

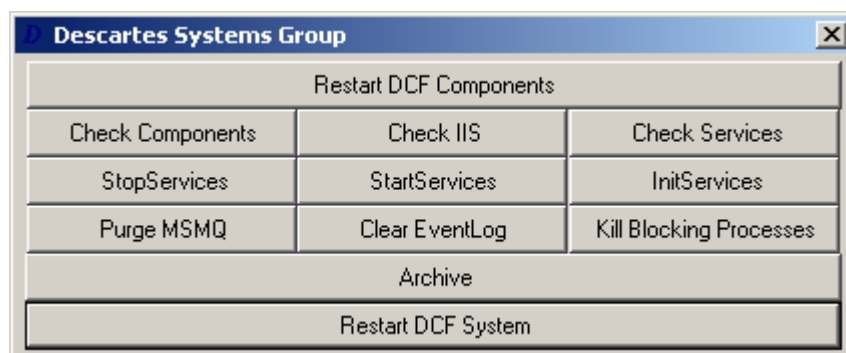
These components include

- DCF Component
- IIS Server
- DCF Services

To manually run the RestartCom.exe:

- 1 From Explorer, double-click the RestartCom.exe from c:\LNOS\bin.

The following popup appears.



- 2 Click one of the following buttons depending on what action you want to perform.



Note— For all actions, users will not see any changes on screen.

- **RestartComponents** – allows the DCF component application to be restarted. When finished, the application will close.

- **CheckComponents** – checks the DCF component connections to make sure they are working correctly. This application sends a light command xml to the Descartes Route Planner backend components. If successful, the application closes. If the call fails, then it will try to start the DCF components and will close when finished. It will also check the number of objects created in DCF and if they exceed a preconfigured number (found in the registry key *HKEY_LOCAL_MACHINE\SOFTWARE\DCF\RestartMaxDCFObjects* -default 100 objects), then it will restart DCF as well.
- **CheckIIS** – does the same thing as **CheckComponents**, but does it via post to ensure IIS works properly. Also, it will start IIS web service if it is down.
- **Check Services** – checks every DCF service for memory usage and status. It will generate a message in the Event Log if any service is restarted.
 - The application will go through all DCFProcessSvc windows processes and check how much memory each uses. Correspondent service will be restarted if it uses (percent wise of *VisiblePhysicalMemory*) more then in *HKEY_LOCAL_MACHINE\SOFTWARE\DCF\MemoryUsagePercentage*
 - This option will also find all pending services (stuck in starting example) and terminate (using taskkill.exe) processes related to the service.
 - At the end application checks if DCF services are not disabled, but not started and will start them.
- **StopServices** – stops every DCF Service. When finished, the application will close.
- **StartServices** – starts every DCF Service. When finished, the application will close.



Note— Prior to starting, all DCF and LNOS MSMQ files that are currently on the system will be deleted. Starting the DCF Services will recreate these files.

- **InitServices** – initializes all services by calling dummy functions. This will load/initialize necessary dlls.
- **Purge MSMQ** – purges every DCF MSMQ queue. This will get rid of any messages left in the queues. It will also clear journal(s) if this feature is enabled.
- **Clear EventLog** – clears the event log for the LNOS DCF category. When finished, the application will close.
- **Kill Blocking Processes** – kills any blocking Process from in the SQL database. For instance, if you have a process that is holding up another process, and you click **Kill Blocking Processes**, the process that is holding up the other process is killed. Only the blocking process is killed, not any consecutive processes.
- **Archive** – runs archive procedure and will archive data for all organizations keeping only 60 days data. It will generate a message in the Event Log.

- **Restart System** – restarts any Descartes Route Planner related components in the following order:
 - check what services are running and stop DCF and Descartes Route Planner services in that order
 - restart IIS and MSMQ services
 - kill blocking processes in the DBs, if there are any
 - purge Descartes Route Planner private queues
 - restart DCF components
 - start only services that were not running
 - clean event log and write to EventLog that system has been restarted
 - initialize all services
 - generate a message in the Event Log

Automatically Running the RestartCom.exe

Using Windows® Scheduled Tasks (**Start menu > Accessories > System Tools > Scheduled Tasks**), this utility can be run using the following parameters:

- RestartCOM.exe no arguments

The following form will be shown on screen every time this task gets executed.

Restart DCF Components		
Check Components	Check IIS	Check Services
StopServices	StartServices	InitServices
Purge MSMQ	Clear EventLog	Kill Blocking Processes
Archive		
Restart DCF System		

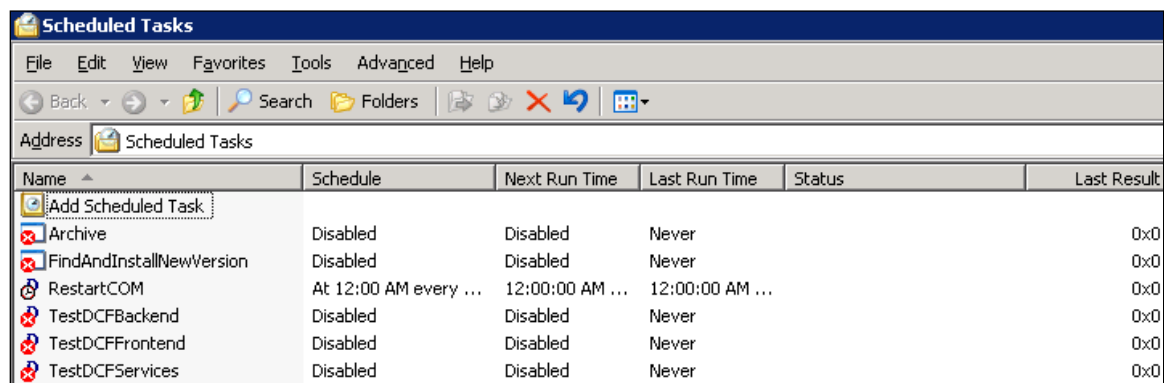
- RestartCOM.exe restart
No form will be shown on the screen. The Descartes Route Planner DCF Components will be restarted.
- RestartCOM.exe trydcfbackend
No form will be shown on the screen, but a command xml for Listing Organizations will be sent.
 - If it fails, Descartes Route Planner DCF Components will be restarted.
 - If the call takes longer than the value specified in
 - *HKEY_LOCAL_MACHINE\SOFTWARE\DCF\RestartTimeOut* (milliseconds default 1000 = 1 sec); or if the response time is longer than RestartTimeOut, then the entire system will be restarted.

- If it works and everything is all right it will check how many components are activated and if it is more than the maximum specified in *HKEY_LOCAL_MACHINE\SOFTWARE\DCF\RestartMaxDCFObjets* (default 100 objects), then it will check if there are any blocking process in the SQL Server and will kill the blocking processes and cycles again.
- If the problem still persists, it will restart the system.
- RestartCOM.exe trydcfbackendandsavestat
This option will do the same as the previous option (trydcfbackend), but will save statistics to a debug/log file found in *c:\LNOS\bin\[ComputerName]RestartCOM.log*.
- RestartCOM.exe trydcffrontend
No form will be shown on the screen, but the application will try list Orgs via UI post and if result is not successful it will:
 - Try to restart ASP process and web services
 - then try to get Orgs again and if it still does not work
 - then application will stop and start IIS and WWW services
- RestartCOM.exe trydcfservices
No form will be shown on the screen, but the application will go through all DCFProcessSvc Windows® processes and check how much memory each uses, Correspondent service will be restarted if it uses (percent wise of *VisiblePhysicalMemory*) more then value in *HKEY_LOCAL_MACHINE\SOFTWARE\DCF\MemoryUsagePercentage*. This option will also find all pending services (stuck in starting for example) and terminate (using taskkill.exe) processes related to the services. At the end, the application checks if DCF services are not disabled, but not started and will start them.
- RestartCOM.exe stopservices
No form will be shown on screen, but all DCF services (Queue Services) will be stopped, as well as any Descartes Route Planner Services (BGO services).
- RestartCOM.exe startservices
No form will be shown on the screen, but all DCF services (Queue Services) will be started, as well as any Descartes Route Planner Services (BGO services). It will wait two seconds between services.
- RestartCOM.exe init services
No form will be shown on the screen but all services will be initialized and all necessary dlls will be loaded.
- RestartCOM.exe purgemsmq
No form will be shown on screen, but all Descartes Route Planner Queues will be purged.
- RestartCOM.exe cleareventlog

No form will be shown on screen, but all event messages from "LNOS DCF" event log will be cleared.

- RestartCOM.exe archive
No form will be shown on screen, but the Descartes Route Planner tables and DCF messagelog tables will be archived (only 60 days data will be saved in Descartes Route Planner tables). This does not archive orphan locations.
- RestartCOM.exe restartsystem
No form will be on screen, but all above described events will be executed in the following order:
 - check what services are running and stop DCF and Descartes Route Planner services in that order
 - restart IIS and MSMQ services
 - kill blocking processes in the DBs, if there are any
 - purge Descartes Route Planner private queues
 - restart DCF components
 - start only services that were running
 - clean event log and write to EventLog that system has been restarted

To create scheduled tasks refer to the Windows® Help documentation (screenshot below):



Name	Schedule	Next Run Time	Last Run Time	Status	Last Result
Add Scheduled Task					
Archive	Disabled	Disabled	Never		0x0
FindAndInstallNewVersion	Disabled	Disabled	Never		0x0
RestartCOM	At 12:00 AM every ...	12:00:00 AM ...	12:00:00 AM ...		0x0
TestDCFBackend	Disabled	Disabled	Never		0x0
TestDCFFrontend	Disabled	Disabled	Never		0x0
TestDCFServices	Disabled	Disabled	Never		0x0

These scheduled tasks will be created automatically after Descartes Route Planner (LNOS FW) installation. It is recommended that users enable TestDCFBackend, TestDCFFrontend, and TestDCFServices tasks.

Configuring RestartCom.exe to Monitor the Memory of Services and Check IIS

Monitoring the Memory of Services

If users have DCFProcess.exe process consuming lots of memory, the RestartCom.exe utility can be modified to make it monitor the processes. FW v 2.7.1.32 installation creates it automatically and it only has to enable a scheduled task. For older versions the user has to configure it manually.

To configure RestartCom.exe manually:

- 1 Copy new version of RestartCom.exe from the FTP:

ftp://rst_ftp.demoservices.Inos.com/ ... /Dmitriy/RestartCOM_x/

Login:

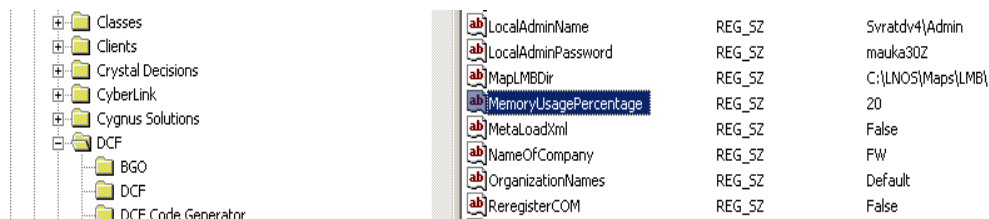
InstallFTPOut

InstallFTPOut

- 2 Place the files to c:\Inos\bin of target machine (replace old if necessary).

This version has a new option called 'trydcfservices'. No form is on the page, but the app will go through all DCFProcessSvc Windowsv processes and check how much memory it consumes. DCFProcessSvc.exe will be terminated if it uses more then value in HKEY_LOCAL_MACHINE\SOFTWARE\DCF\MemoryUsagePercentage (percentwise of VisiblePhysicalMemory). This registry entry will be created automatically after first run of the 'trydcfservices' command.

The default value will be 20, which means that if a given process uses $(dProcessMemoryUsage * 100) / dTotalVisibleMemorySize$ more then 20 this process will be restarted

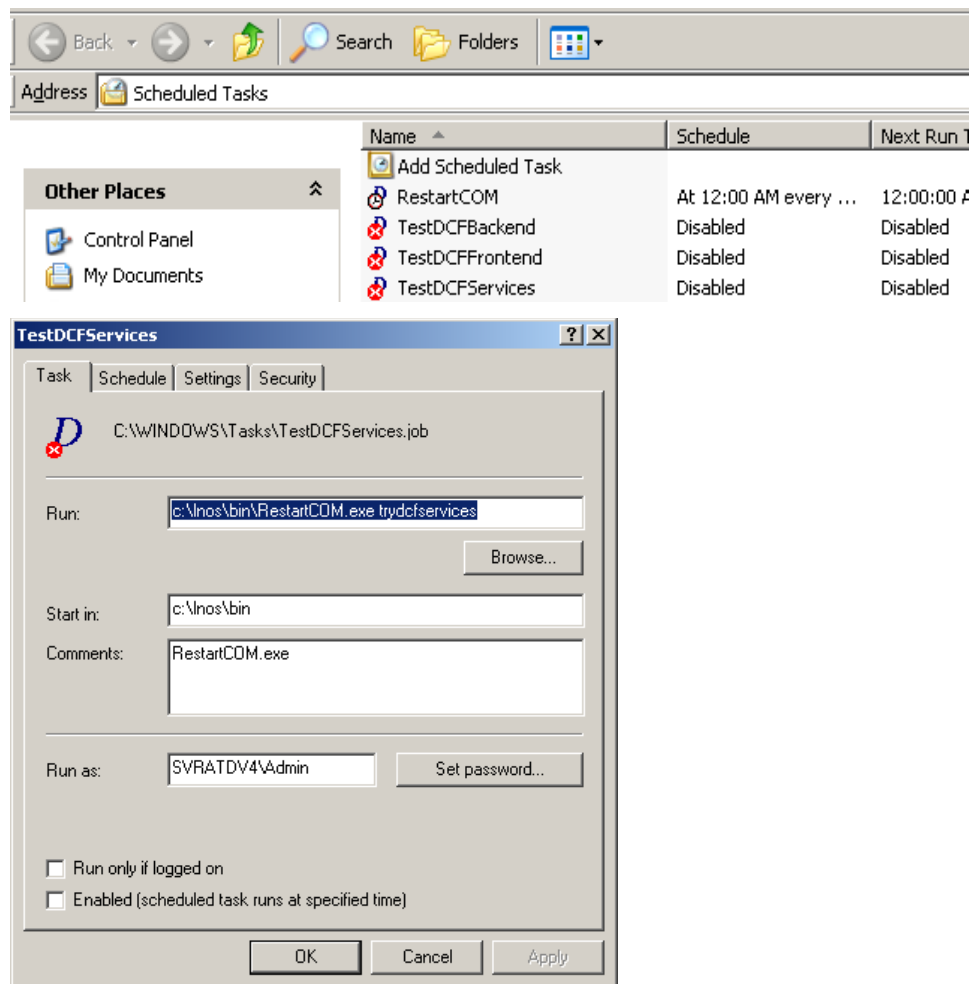


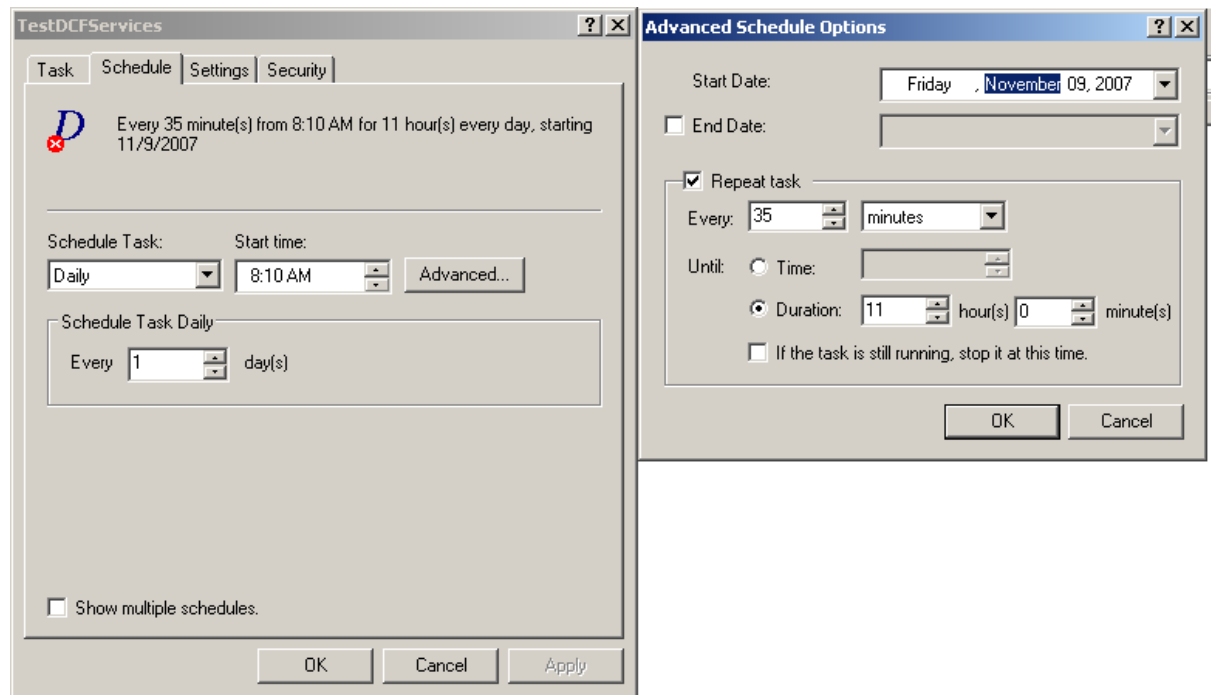
At the end, the application checks to see if DCF services are running if the services are not disabled.

- 3 Go to the Scheduled Tasks of the target machine and create a new 'TestDCFServices'. The name is important for future upgrades. Command line will be:

c:\Inos\bin\RestartCOM.exe trydcfservices

It can be set to run every 35min from 8am-7pm every day.





- 4 Enable the Scheduled Task and, please, monitor the services for one to two hours.
- 5 Run scheduled task at least ones to see it is operational.
- 6 Test and make sure the scheduled task is working correctly:
 - Change HKEY_LOCAL_MACHINE\SOFTWARE\DCF\MemoryUsagePercentage to a value of two.
 - Check event log (DCF LNOS) for these entries.
 - Restart DCF\$Optimize Svc 3 service because it uses 1.93587114201391% of physical memory 1039532Kb, max 1.93%.
 - Make sure services was restarted
 - Change HKEY_LOCAL_MACHINE\SOFTWARE\DCF\MemoryUsagePercentage to 20.

Checking IIS

Checking IIS is basically the same process and installation most likely created this scheduled task already. Instructions are similar to those above, but schedule task name and command line are different.

Go to "Scheduled Tasks" of the target machine and create new 'TestDCFFrontend'. The name is important for future upgrades. Command line will be:

c:\LNOS\bin\RestartCOM.exe trydcffrontend

It can be to run every 35min from 8am-7pm every day.

Setting up Archiving

Archiving allows users to archive data. There are three archiving options that are available. They include:

- Archive
- Archive Orphan Locations
- Clean Up MessageLog



Note— The process of moving data to the archive tables will take some time, especially the first time this action is performed. It is strongly recommended that archiving is only run during non-operational hours. The procedure is transactional, so if anything happens in between, all changes will be rolled back.

Archiving

When **Archive** is selected, Descartes Route Planner will prepare a new table inside the Descartes Route Planner database based on the following tables:

- FWStop
- FWRoute
- FWResource
- FWResRteParm
- FWOrder
- FWOrderLine
- FWActivity

The archived data will not be accessible from the UI, to see the data, you will have to go to the matching LNOSFW database table. For example, if you archive Routes from the FWRoute table, you will be able to view the archived data from the FWRouteArchive table.



Note— Archiving will not clean/archive any temporary files.

To set up archiving:

- 1 From the main menu, select **AppSetup > Maintenance > FW Archive**.

The **Archiving** page appears.

- 2 Enter the number of days and the schedule key that you want archived. If you leave the Schedule Key field blank, the data from all available schedules will be archived.



Note— Descartes Route Planner will archive all data older than the current date to the number of days specified. So, if you put in thirty days, it will archive all records starting with the previous day and continue until it reaches the thirtieth day.

- 3 Click **Archive**.

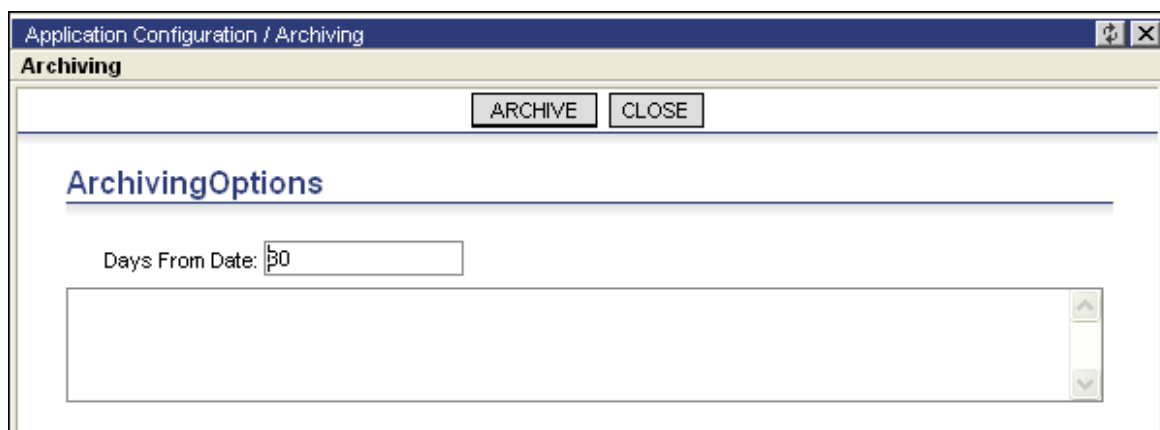
Archiving Orphan Locations

When **Archive Orphan Locations** is selected, Descartes Route Planner will archive all locations in the FWLocation table.

To set up archiving of orphan locations:

- 1 From the main menu, select **AppSetup > Maintenance > FW Archive orphan locations**.

The **Archiving** page appears.



- 2 Enter the number of days to archive.



Note— Descartes Route Planner will archive all data older than the current date to the number of days specified. So, if you put in thirty days, it will archive all records starting with the previous day and continue until it reaches the thirtieth day.

- 3 Click **Archive**.

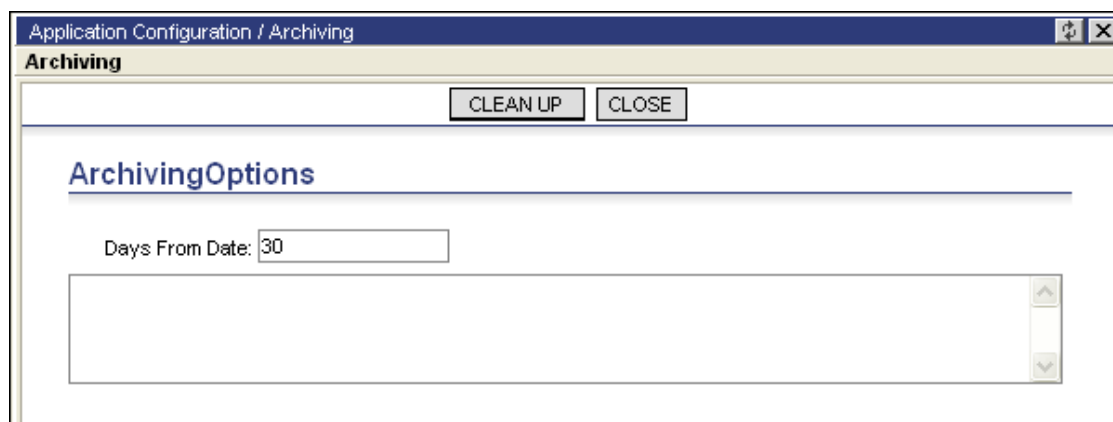
Cleaning Up the MessageLog

When **Clean Up MessageLog** is selected, Descartes Route Planner will clean up all messages from CtyMsgLog, CtyMsgTrace, CtyMsgLogParm from any DCF%, DCFSec%, or Sec% databases.

To clean up the message log:

- 1 From the main menu, select **AppSetup > Maintenance > Clean Up MessageLog**.

The **Archiving** page appears.



- 2 Enter the number of days.



Note— Descartes Route Planner will delete (clean up) the message log if the records are older than the current date to the number of days specified. So, if you put in thirty days, it will clean up all records starting with the previous day and continue until it reaches the thirtieth day.

- 3 Click **Archive**.

Extracting Data from the LNOSFW Database

To begin the process, users must create functions and procedures in the LNOSFW database representing data to be extracted. Follow the guidelines of on the following table.

Setup Steps	What to run
Create the function 'FWTransformValue'	FWTransformValue.sql
Create the function 'getFieldList'	getFieldList.sql
Create procedures 'FWCopy1'~'FWCopy7'	FWCopy1.sql ~ FWCopy7.sql
Create the procedure 'FWExtractRoutes'	FWExtractRoutes.sql
Create the procedure 'FWExtractSchedule'	FWExtractSchedule.sql

Extract data based on resources/routes

exec FWExtractRoutes *resources*, *outputDB*, *templateDB*, *prefix*

- **Resources:** a comma-separated list of resource keys which are to be copied to outputDB
- **OutputDB:** the name of a new or an existing database to which data are copied.
- **templateDB:** a path to an Inosfw-version-specific database backup file. Usually /LNOS/DatabaseBackup/DCFFWSetup.BAK can serve the purpose of this parameter.
- **Prefix:** (optional) a string which is to be prefixed to business keys of copied data.

e.g. `exec FWExtractRoutes 'AVL_D1_V1_001,AVL_D1_V2_002', 'OutputDB', 'C:\LNOS\DatabaseBackup\DCFFWSetup.bak', ''`

Extract data based on a schedule

exec FWExtractSchedule *scheduleKey*, *outputDB*, *templateDB*, *prefix*

- **ScheduleKey:** the schedule whose data are to be copied to outputDB
- **OutputDB:** the name of a new or an existing database to which data are copied.
- **TemplateDB:** a path to an Inosfw-version-specific database backup file. Usually /LNOS/DatabaseBackup/DCFFWSetup.BAK can serve the purpose of this parameter.
- **Prefix:** (optional) a string which is to be prefixed to business keys of copied data.

e.g. `exec FWExtractSchedule 'RechargeDemo', 'OutputDB', 'C:\LNOS\DatabaseBackup\DCFFWSetup.bak', ''`

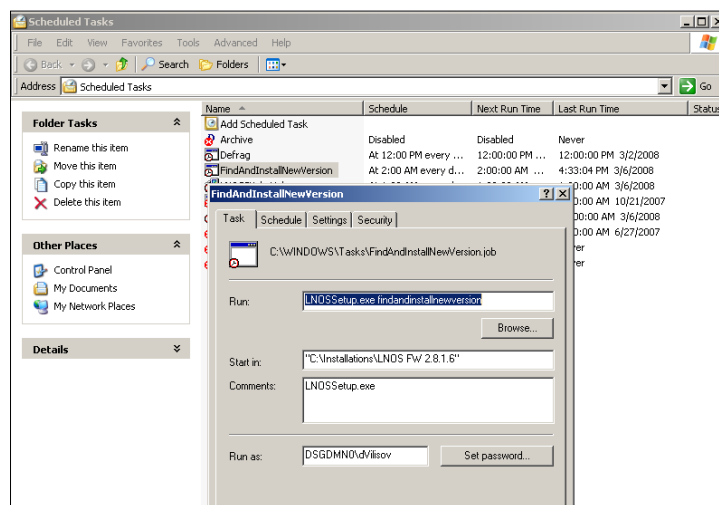
Silent Installation Using LNOSSetup.exe

LNOSSetup.exe can be run via command line using the following options:

- **installation:** Installs everything
- **registercom:** Only registers components/dlls
- **findandinstallnewversion:** Finds new package at the same directory and install it
- **findandinstalllatestversion:** Finds and install find new package at the same directory and install/reinstall it. This option will allow to reinstall same package over and over. For example you have installed LNOS FW 2.7.1.35 and last package you have LNOS FW 2.7.1.35. This option will install it again, good for test environment

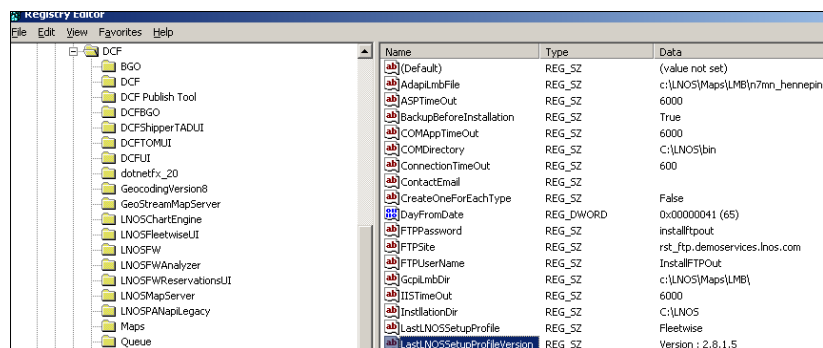
How It Works

LNOSSetup.exe v 1.2.22.41 and older will create the scheduled task "FindAndInstallNewVrsion."



This schedule task can be run every night and search new packages at the same directory were last installation run.

For example: If run C:\Descartes\Installations\LNOS FW 2.8.1.6 is run, it will create scheduled task which will search every directory under C:\Descartes\Installations and it will find C:\Descartes\Installations\{LNOS FW 2.8.1.X} folder with LNOSSetupProfile.xml and LNOSSetup.exe files in it and LNOSSetupProfile.xml has an older version then value in HKEY_LOCAL_MACHINE\SOFTWARE\DCF\LastLNOSSetupProfileVersion registry entry.



The system will run the installation backup dlls at c:\inos\bin and install all products at that computer. At the end of installation, the system will update the "FindAndInstallNewVrsion" schedule task to run from the new location.

Appendix E: Adding Generic Reports to LNOS Integrated Reporting Services/Route Planner

Preconditions

- RPS is installed and the customer database is linked
- Customer organization is active and RPS users are created
- RPS functions are restored
- Additional RPS database tables are restored

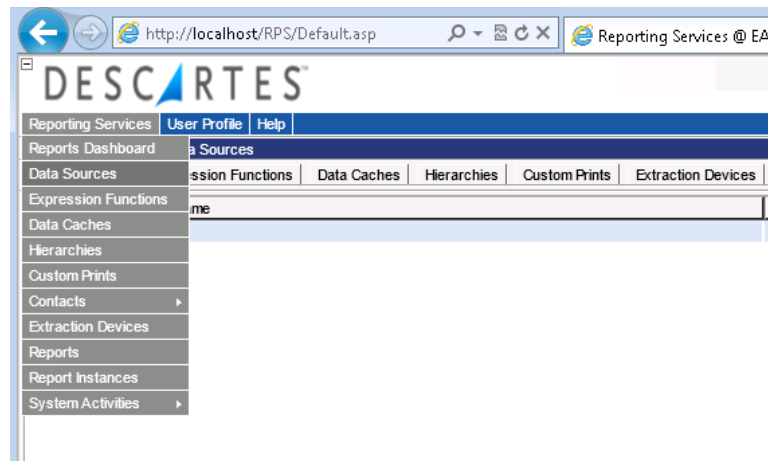
Reports Import

- 1 Prepare import files for import.
 - a Copy the "RPS RP Generic Reports" package locally.
 - b Open the Data Source XML files and replace "CustomerABCD" with the real customer name.
 - ExtDataSource.DS Route Detail (21 replacements)
 - ExtDataSource.DS Route Summary (23 replacements)
 - c Open the "FunctionsScript" file and replace "CustomerABCD" with the real customer name (8 replacements).
 - d Open the "jm_fwtypestatusmatch" file and replace "CustomerABCD" with the real customer name (1 replacement).
 - e Open the "jm_routechanges" file and replace "CustomerABCD" with the real customer name (1 replacement).
- 2 Login to <http://localhost/RPS/Default.asp> as the organization created during activation. The link and login may be different for Route Planner/RPS installation.

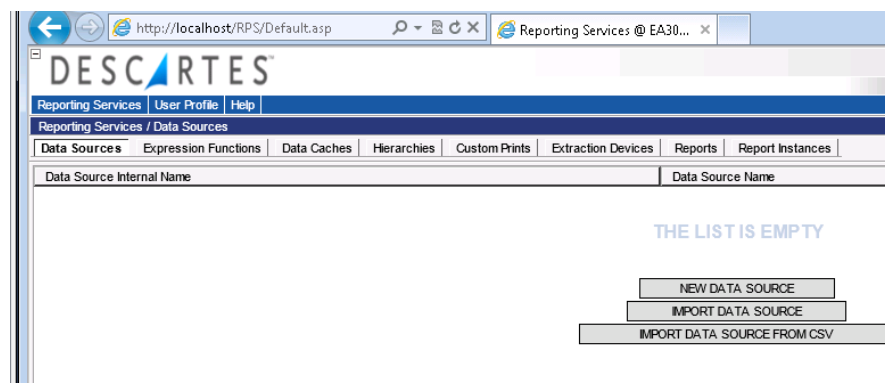
Example:

Only the RPSSystem user has rights to create all report components, specifically Data Source. Other RPS users have specific rights. Please see the RPS User's Guide located under the **Help** tab for more information.

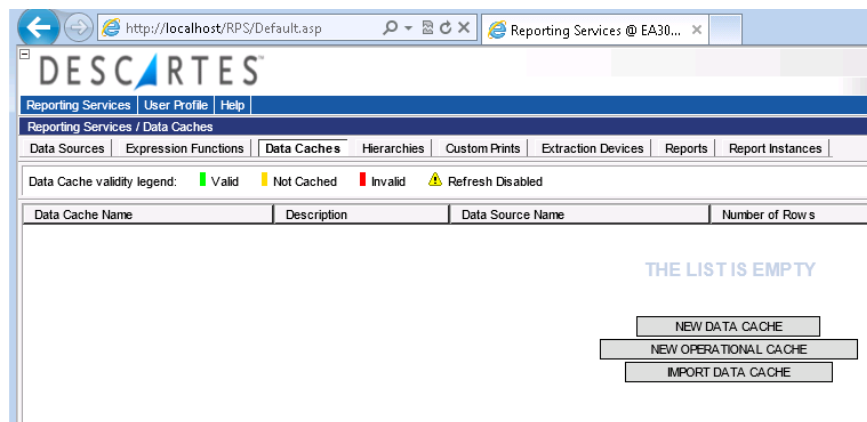
- 3 After logging in as the RPSSystem user, select **Reporting Services > Data Source** from the main menu.



- 4 Select the **Import Data Source** button or if there are items in the list, right-click on the page and select the **Import Data Source** option from the right-click menu. Browse to the location where the Data Source XML is stored and select it. In the "RPS RP Generic Reports" package, there are two Data Source XML files. Import both of these files:
- ExtDataSource.DS Route Detail
 - ExtDataSource.DS Route Summary

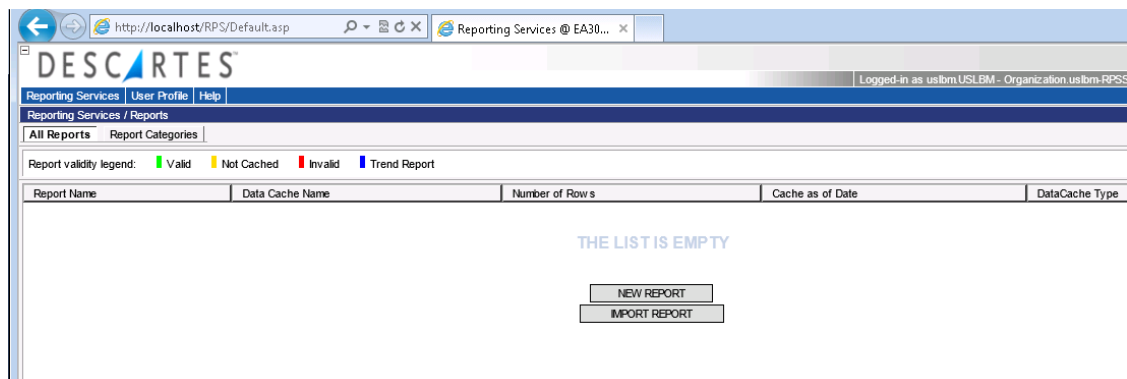


- 5 Click the **Data Caches** tab and select **Import Data Cache**. Browse to the location where the Data Cache XML is stored and select it. In the "RPS RP Generic Reports" package, there are two Data Caches XML files. Import both of these files:
- ExtDataCache.DC Route Detail
 - ExtDataCache.DS Route Summary



After the import, right-click each new Data Cache and select **Refresh Data Cache**. A valid data cache will be marked green in the list.

- 6 Select the **Reports** tab and select **Import Report**. Browse the location where Reports XMLs are located. In the "RPS RP Generic Reports" package, there are 14 Report XML files. Import all of these files.

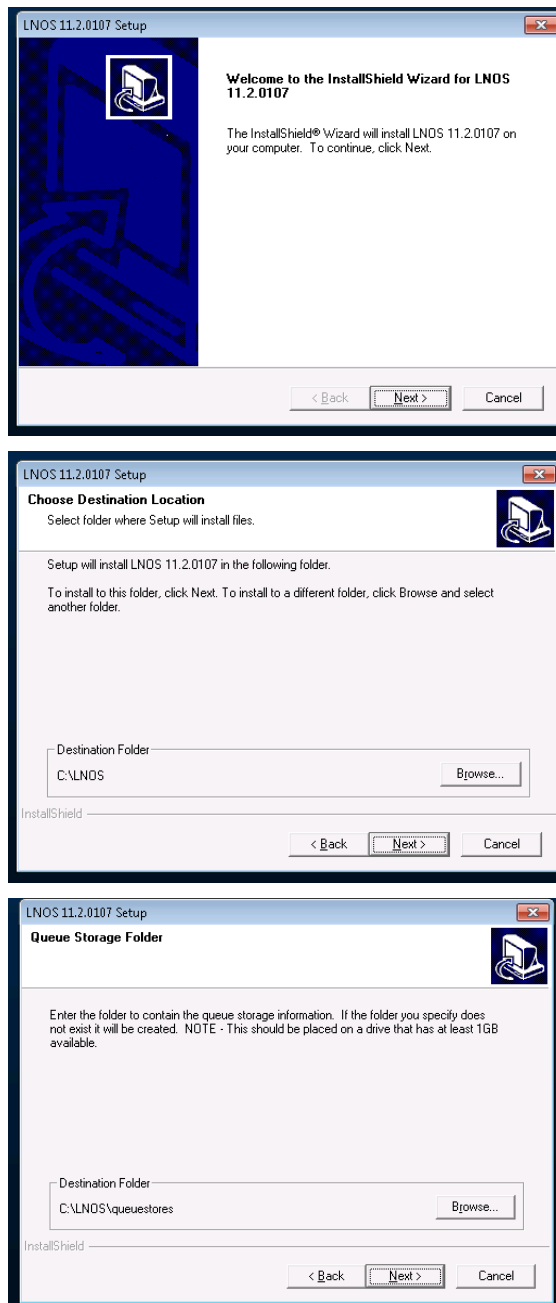


After the Reports have been imported, the user can run and view them.

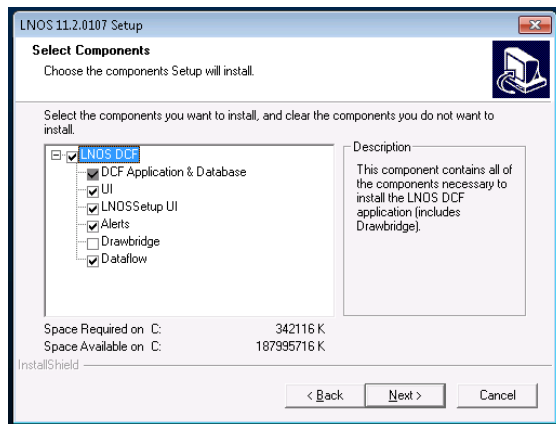
RPS Installation

To install RPS:

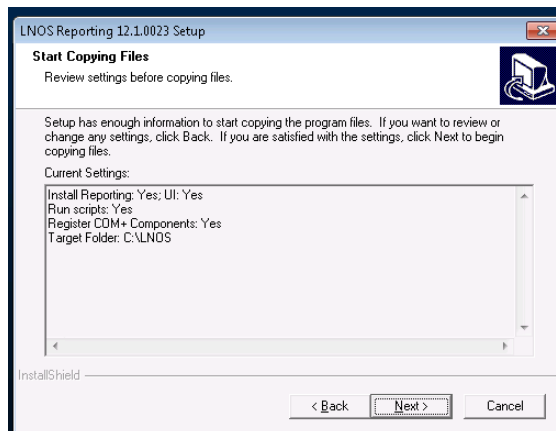
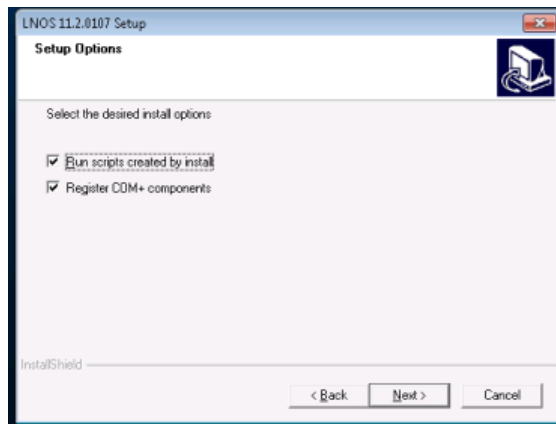
- 1 Download the RPS package.
- 2 Run **Setup.exe**.



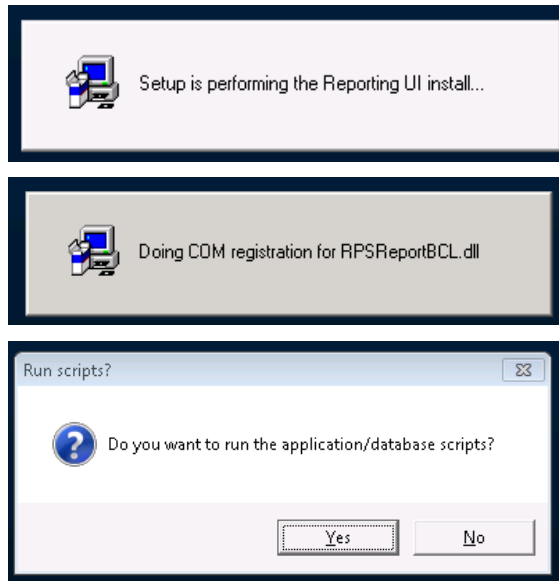
3 Select the following components:



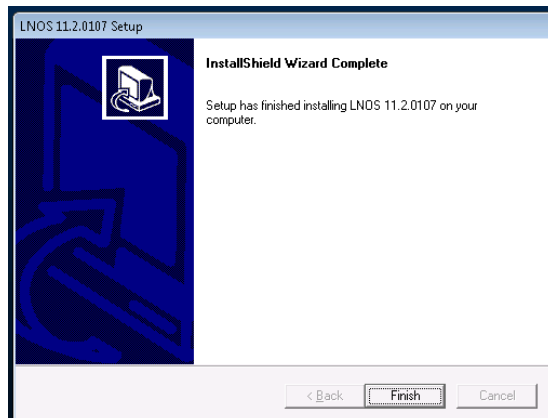
4 Select the following options.

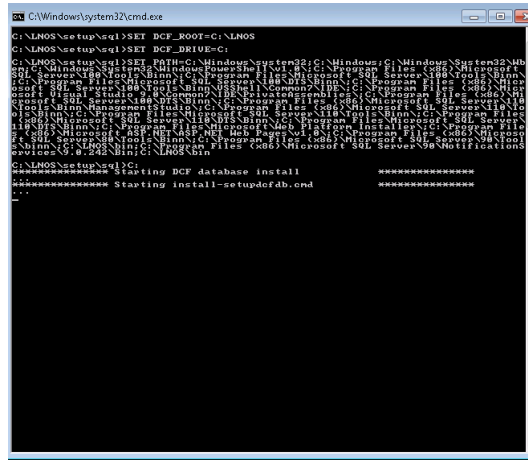


Installation will begin.



5 Run the database script.





The script can be run manually from: c:\LNOS\Setup\SQL\

- 6 Validate RPS installation. Login to <http://localhost/DCF/Default.asp> using the credentials below.

Company Name: Descartes Systems Group

Login Name: SecAdmin

Password: ●●

Remember Me: ☐ [Check to Auto-Login Next Time](#)

Login

Activating Customer Organization

- 1 Login to: <http://localhost/lnos%20setup/>.
- 2 Right-click in the page and select **New Organization** from the right-click menu.
- 3 Use the customer name as the Organization name. Click **Save**.
- 4 After organization is saved, from the **Application** drop-down menu, select **SecAdmin**.

DESCARTES™

ORGANIZATIONS

LNOS Setup / Setup / User Profile / Help

LNOS Setup / Organizations

Organizations

Organization Name	Application	Is Host Organization
CustomerName	1. SecAdmin	False

Rows 1 - 3 of 3 / Page 1 of 1

Rows Per Page: 10 / 20 / 50

Page 1

LNOS Setup / Organizations / Activate Application

Activate Application

ACTIVATE CANCEL

Organization Information

Organization Name: CustomerName

Security Application: SecAdmin

Application: SecAdmin

Resecure Rights: RPS

Database Server Information

Change: ☐

Name: EA3001

Extra Task Parameters

Name	Value
Parameter Name:	
Parameter Value:	

NEW DELETE

- 5 Click **Activate**. The activation process can take a few minutes.
- 6 When activation is complete, select **RPS** from the **Application** drop-down menu.

DESCARTES™

ORGANIZATIONS

LNOS Setup / Setup / User Profile / Help

LNOS Setup / Organizations

Organizations

Organization Name	Application	Is Host Organization
CustomerName	1. SecAdmin	False

Rows 1 - 3 of 3 / Page 1 of 1

Rows Per Page: 10 / 20 / 50

Page 1

LNOS Setup / Organizations / Activate Application

Activate Application

ACTIVATE CANCEL

Organization Information

Organization Name: CustomerName

Security Application: SecAdmin

Application: RPS

Resecure Rights: RPS

Database Server Information

Change: ☐

Name: EA3001

Extra Task Parameters

Name	Value
Parameter Name:	
Parameter Value:	

NEW DELETE

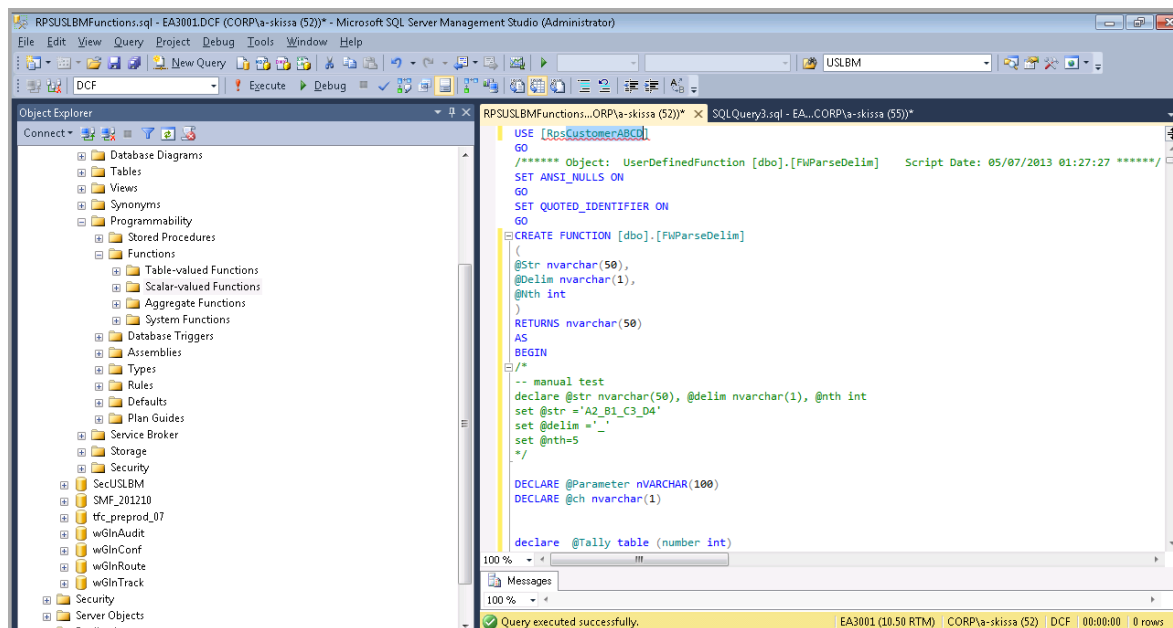
- 7 Click **Activate**. The activation process can take a few minutes. Activation creates the users listed below. Each user has different rights. For more information, please see the RPS User's Guide found in the **Help** menu.

User Management / Active Users		
New Users Active Users Inactive Users Logged-In Users		
Login Name	Name	Organization Name
CustomerName-DataAdmin	CustomerName DataAdmin	CustomerName
CustomerName-SecAdmin	CustomerName SecAdmin	CustomerName
CustomerName-UsrAdmin	CustomerName UsrAdmin	CustomerName

Restoring RPS Functions

Upon setup, some RPS functions will need to be added.

- 1 Open Microsoft® Corporation's SQL Server® Management Studio.
- 2 Go to **File/Open/File**. Navigate to the location of the "RPS RP Generic Reports" package and select the "FunctionsScript" file. The script will appear in the application.

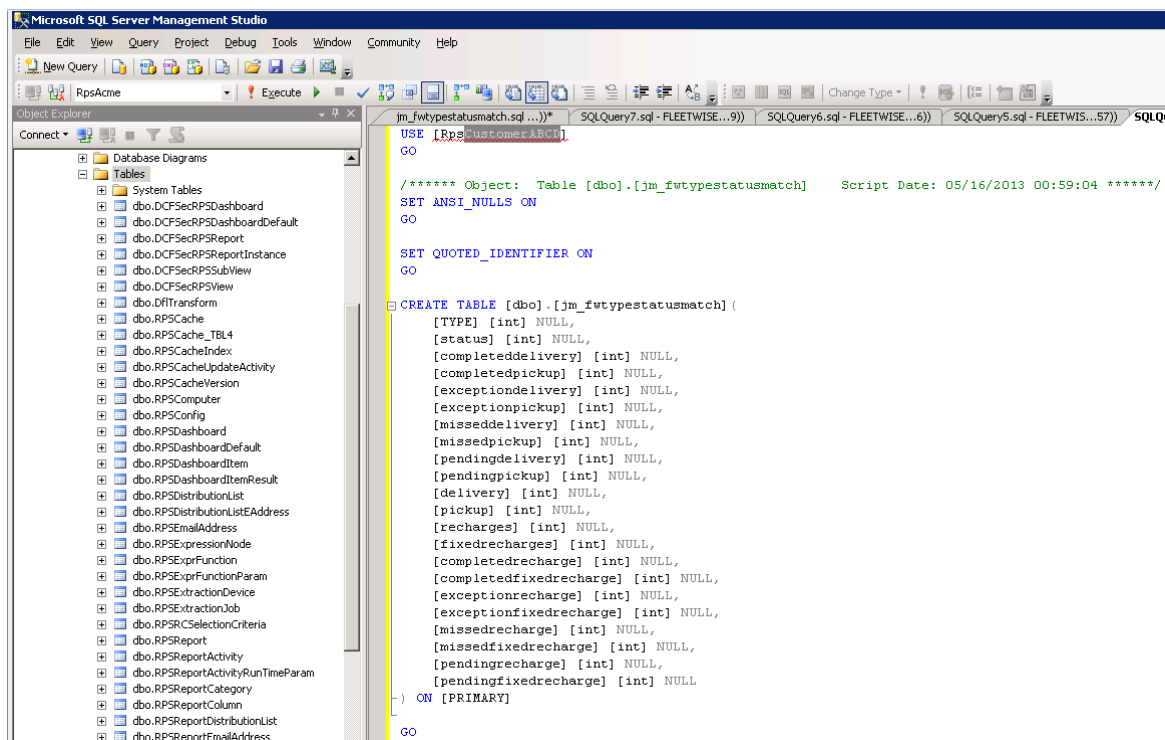


- 3 Click **Execute**. the script will run and create RPS functions in the RPS customer database.
- 4 Check the Customer DB/Programmability/Functions/Scalar-valued Functions folder to make sure it contains the additional functions.

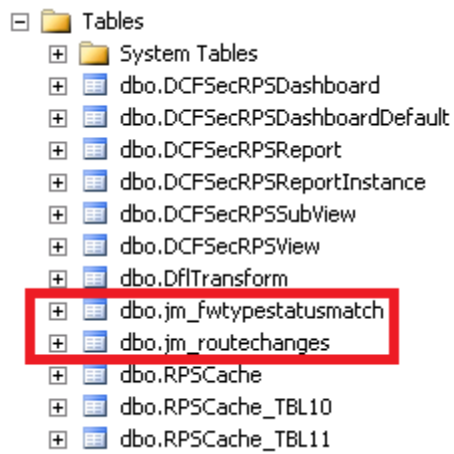
Generate Additional RPS Database Tables

Two additional RPS tables will need to be added manually since they are not part of the standard RPS installation. These tables should be added for advanced analytics.

- 1 Open Microsoft® Corporation's SQL Server® Management Studio.
- 2 Go to **File/Open/File**. Navigate to the location of the "RPS RP Generic Reports" package and select the "jm_fwtypestatusmatch" file. The script will appear in the application.



- 3 Click **Execute**.
- 4 The script will run and create the required database table in RPS customer database.
- 5 Repeat the steps above for the second table. Select the "jm_routechanges" file instead.
- 6 Check the RPS customer database to make sure the new tables are installed:



Appendix F: Federated SSO with LNOS and SAML 2

General

Federated SSO allows users who are authenticated against one directory to access additional applications and services without re-authenticating when a trust relationship has been established.

This document describes how Federated Single Sign On (SSO) can be implemented with Descartes Logistics Technology Platform's LNOS applications and meet the request from customers for SSO and adherence to the SAML 2 standard. The design is based on content provided by Microsoft® available here:

<http://www.microsoft.com/en-us/download/details.aspx?id=19909>

Federated SSO is a claims-based solution where some service "Claims" the validity of the user is correct. The claim is submitted to the desired application; the information in the claim provides information about the user – email address, organization association or company role.

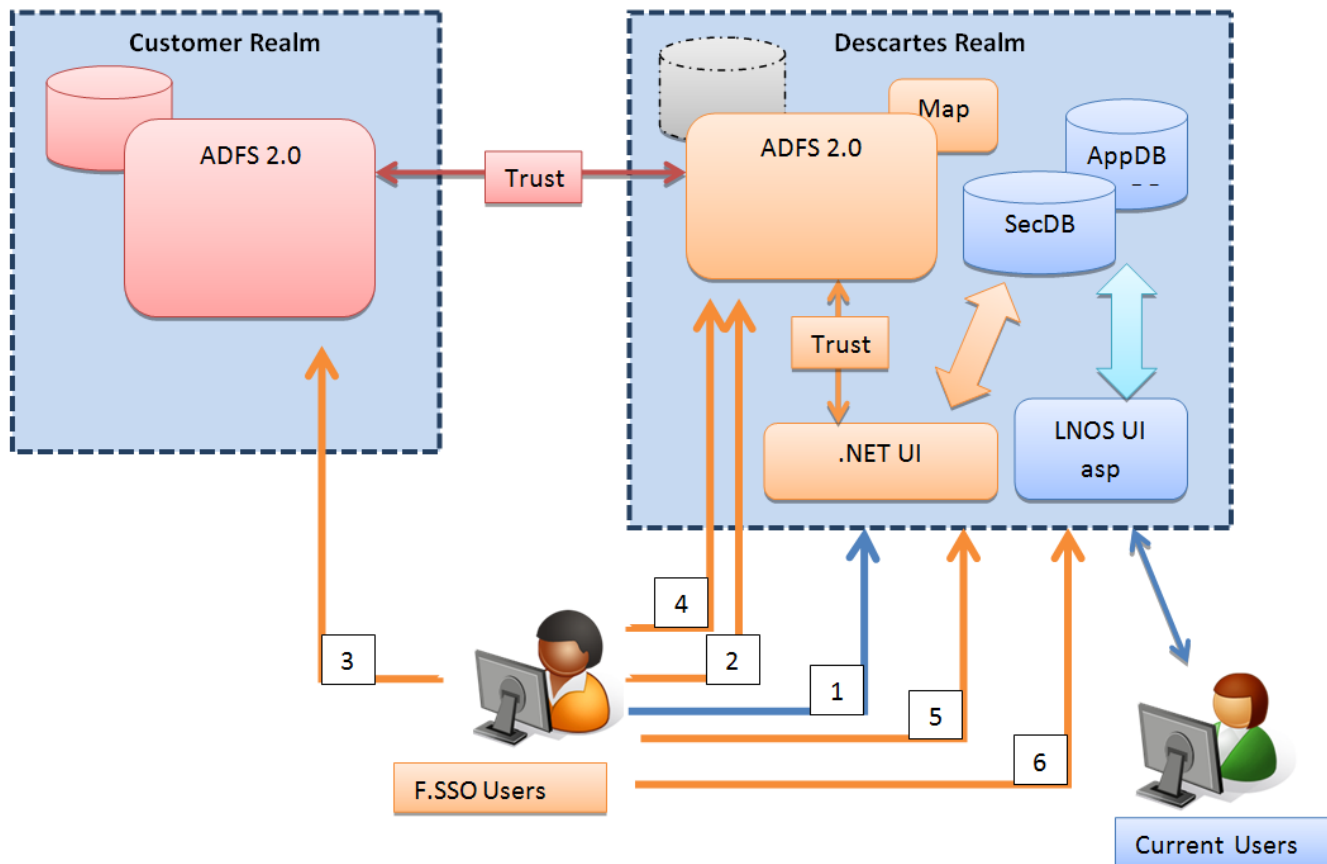
Key Benefits

- Reduces the cost of password management as users are required to only remember one set of credentials
- Strengthens security and reduces the risk of a breach by providing a single centralized authentication point for applying stronger policies
- Frees constrained technical resources from unneeded direct participation by securely delegating to users and privileged staff
- Speeds deployment by enabling a single password management strategy for the customer's corporate applications including Descartes Cloud based Logistics Technology Platform

There are numerous identity solutions available to companies and most support SAML and SAML 2 methods of identity management between realms. This solution is expected to meet the needs of companies desiring to implement Federated SSO with Descartes LNOs applications.

How to Support F.SSO

The image below explains how F.SSO will work with LNOS. All calls are HTTPS and most of them are caused by a redirect from the end users browser.



- 1 The users of a Federated SSO enabled customer will connect to a different URL, which will be an ASP.NET-based site. This site has a trust with a locally hosted ADFS 2.0.
- 2 The .NET UI does not find any claims/tokens and will redirect the user to its trusted token provider. At this point the user will be prompted a screen showing all trusted claims providers; this is called "Home Realm Discovery".
- 3 After the user selected their proper authentication service (usually their own company) the user's browser will be redirected to the URL of the customers token/claims provider. At this point it's the responsibility of the customer to authenticate the user (Windows® authentication, Forms based).
- 4 After positive authentication of the user, the token provider will redirect back to the DSG token provider, this time the request will have a set of customer claims about the user that want to use the LNOS application. Later in this document, the minimal content of expected claims will be discussed.

- 5 After mapping the customer's claims to LNOS claims the user will be redirected back to the URL he/she originally connected to. This time the LNOS claims are passed on with the redirect.

The .NET UI will read the claims and execute the proper Login API (i.e. Login.login) without the need of a password. An LNOS Session object that will be populated accordingly with the user information returned by Login.login.

- 6 The user will finally be redirected to the traditional LNOS URL, but it will contain the LNOS SessionID which will bypass the login and will go straight to the user start page.

➡ **Note—** Step 1 to 4 can be skipped with use of a browser cookie with a limited life span.

➡ **Note—** Home Realm discovery could be skipped by use of specific browser cookie with medium life span.

➡ **Note—** On-Demand Provisioning: In case the Login.login fails with step 5, because the user does not exist the .NET UI will create a user record with the minimal information it has from the LNOS claims, when done it will retry the login logic.

Pros/Cons of Solution

Customers do not need to add LNOS-specific claims into the Microsoft® Corporation's Active Directory®. The mapping rules at the Descartes ADFS 2.0 will map to the proper LNOS claims, primarily to UserGrp/Roles.

A change on either customer or LNOS App site will require modification of the mapping rules used.

Created users from claims do not have a password and cannot login using the traditional LNOS URL. If user leaves the customer, he or she is not able to connect to the DSG hosted application.

Setup of trusts on the DSG ADFS 2.0 is a simple process.

Setup

Non-code related

Setup ADFS 2.0 on a Windows® 2008 R2 server.

Setup a trust with the customer.

Code Related

Create a claims enabled web site that will be used to create the LNOS Session ID.

ASP LNOS UI change

The ASP LNOS UI's will have to switch to the SessionStoreDCF.asp which will find/save the session using a DCF component called DCFSessionD.

Proof of concept

Create POC where the claims enabled web site is configured to trust a little test STS (token provider).

Requirements

Windows® 2008 R2 must be joined to a domain.

ADFS 2.0 must installed on the server

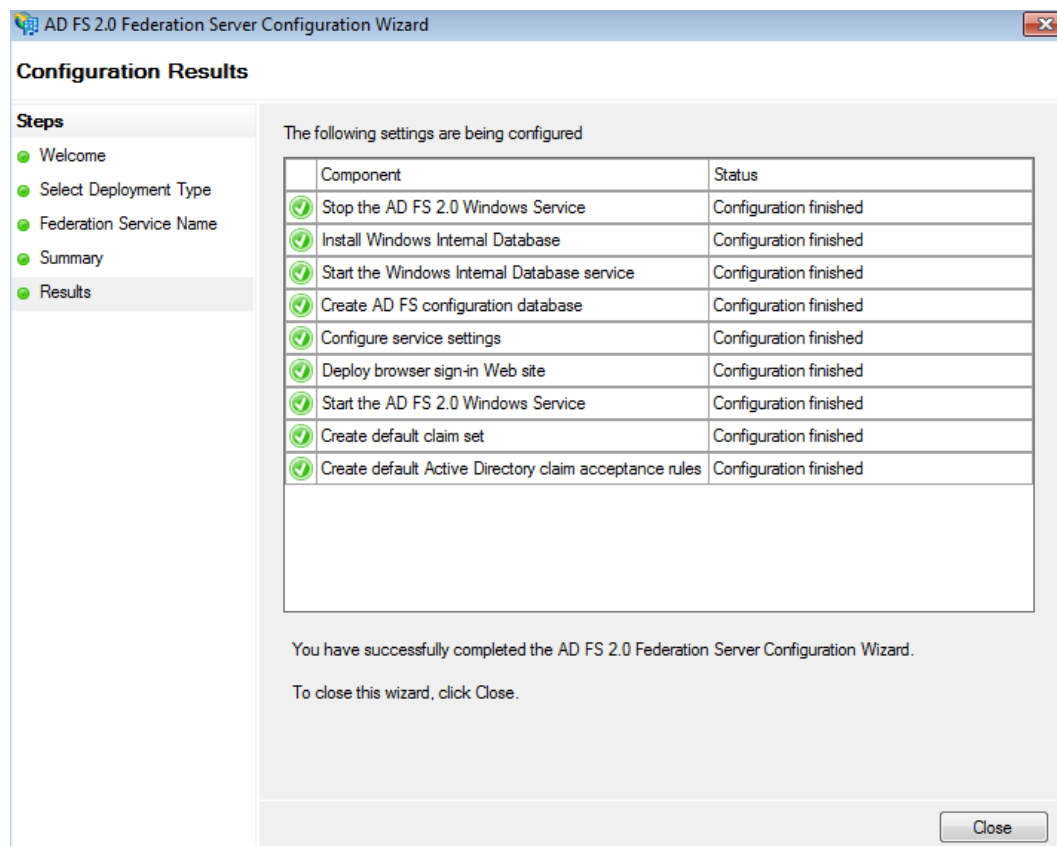
LNOS Core UI.NET

LNOS DCF 11.2.0110

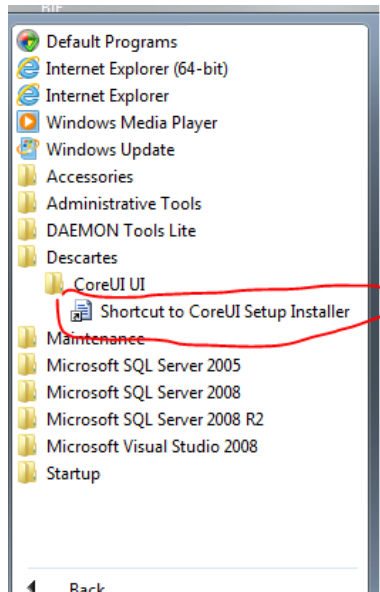
Steps and Downloads

- 1 Download ADFS 2.0 (if not installed): <http://www.microsoft.com/en-us/download/details.aspx?id=10909>.
- 2 Create a self-signed certificate with a friendly name within ISS (ex: fedss0.dsglab.net).

- 3 Add site binding for HTTPS and setup with certificate created.
- 4 Configure ADFS 2.0 as a standalone server:
 - a Open ADFS 2.0 Manager.
 - b Click the ADFS FES Configuration Wizard.
 - i Create new FED Service.
 - ii Select Standalone Server.
 - iii Certificate added to IIS port 433 should show.
 - iv Review summary.
 - v End result.



- 5 Join the second virtual machine to DSGLAB domain
 - a Ensure the machine name is unique for the domain (DSGWA01).
 - b Join DSGWA01 to DSGLAB domain.

6 Run Core UI.NET v12.1.4:**a Run the Vdir Setup.****b Click Next.****c Provide a Vdir Name (FEDSSO).****d Assign Application pool with .net 4.0 to the FEDSSO.****7 Open GeneratedFiles\config.xml. Add the following:**

```
<Assembly value="FedSSO, Version=12.1.1.1, Culture=neutral,
PublicKeyToken=9ea4b23f98029da1" key="FedSSO" HasResources="True"/>
```

8 Install the WIF SDK 4.0**9 Rename the folder "FederationMetadata" under the FEDSSO folder****10 Run the Federated Setup Utility. This tool will setup web.config and federatedmetdata.xml.****11 Add the following claims to the metadata.xml file at the proper xml node.**

```
<auth:ClaimTypeUri="http://descartes.com/claims/2011/01/organization" Optional="true" xmlns:auth="http://docs.oasis-open.org/wsrfed/authorization/200706"/>
<auth:ClaimTypeUri="http://descartes.com/claims/2011/01/usergroup" Optional="true" xmlns:auth="http://docs.oasis-open.org/wsrfed/authorization/200706"/>
<auth:ClaimTypeUri="http://descartes.com/claims/2011/01/role" Optional="true" xmlns:auth="http://docs.oasis-
```

```
open.org/wsfed/authorization/200706"/> <auth:ClaimTypeUri="http://descartes.com/claims/2011/01/redirectURL" Optional="true" xmlns:auth="http://docs.oasis-open.org/wsfed/authorization/200706"/>
```

- 12** On STS add a relying party. Use the location option and browsed to [\\dsglabwa01\c\\$\inetpub\wwwroot\FEDSSO\FederationMetadata\2007-06\FederationMetadata.xml](\\dsglabwa01\c$\inetpub\wwwroot\FEDSSO\FederationMetadata\2007-06\FederationMetadata.xml)

- 13** Cd C:\inetpub\wwwroot\DCF

- 14** Add the following to the config.xml: SessionCache="true"
SessionStoreDCF="true"

- 15** If ASP.NET returns a problem regarding the web.config it would be helpful to run:

```
%SystemRoot%\system32\inetsrv\appcmd migrate config "Default Web Site/UI Framework NET/"
```

Step 1: Upgrade and Re-configure LNOS Servers

- 1** Run the DCF 11.2.0110 Upgrade. Make sure DCF and SecAdmin scripts are running at the end.
- 2** Establish a working database session between ASP and ASPX:
 - a** Open DCF UI.
 - b** Login as DSG Developer.
 - c** Select **DCF > Components**.
 - d** Right click and select **Import Components** from the right-click menu.
- 3** After import you should see the following

DCF Setup Dataflow System Monitor UIF Tools User Profile Help			
Component Management / Components / Component APIs			
Edit Component Component APIs Component API List Shape Browser			
NEW API REFRESH OPTIMIZE DE-OPTIMIZE			
APIs for DCFSessionD			
	Name/Description	Command/Service/Handler	Optimized
+	Name: CDCFSessionDSCmd::find Description:	Command: CDCFSessionDSCmd Service: DCFSession Data Service Handler: CDCFSessionDS::CDCFSessionDSCmd::fiFind	✓
+	Name: CDCFSessionDSCmd::list Description:	Command: CDCFSessionDSCmd Service: DCFSession Data Service Handler: CDCFSessionDS::CCTyDSCmd::fiList	✓
+	Name: CDCFSessionDSCmd::purge Description:	Command: CDCFSessionDSCmd Service: DCFSession Data Service Handler: CDCFSessionDS::CDCFSessionDSCmd::fiPurge	✓
+	Name: CDCFSessionDSCmd::remove Description:	Command: CDCFSessionDSCmd Service: DCFSession Data Service Handler: CDCFSessionDS::CCTyDSCmd::fiRemove	✓
+	Name: CDCFSessionDSCmd::save Description:	Command: CDCFSessionDSCmd Service: DCFSession Data Service Handler: CDCFSessionDS::CDCFSessionDSCmd::fiSave	✓
+	Name: CDCFSessionDSCmd::listForVector Description:	Command: CDCFSessionDSCmd Service: DCFSession Data Service Handler: CDCFSessionDS::CDCFSessionDSCmd::fiListForVector	✓

- 4** Select **DCF > Menus**.

- 5 Right click and select **Import Menus** from the right-click menu.
- 6 Browse for the file on the local machine: c:\lnos\setup\xml\Menu-System.xml.
- 7 Select **DCF > Function Groups**.
- 8 Right-click and select **Import Function Groups** from the right-click menu.
- 9 Browse for the file on the local machine: c:\lnos\setup\xml\FnGrp-System.xml.

Configure LNOS UI (ASP versions)

- 1 Setup ALL LNOS UI's to use the SessionStoreDCF logic.
- 2 Open GeneratedFiles\Config.xml in Notepad.
- 3 Add SessionCache="true" SessionStoreDCF="true"
- 4 Remove SessionSQLDataSource, SessionSQLUserID and SessionSQLPassword if present.
- 5 The new SessionStore logic requires the file CaprockDCT.dll to be registered. A 32 bit version should be installed by the DCF UI installer and can be found under inetpub\wwwroot\DCF and it's not registered.

To register, do the following:

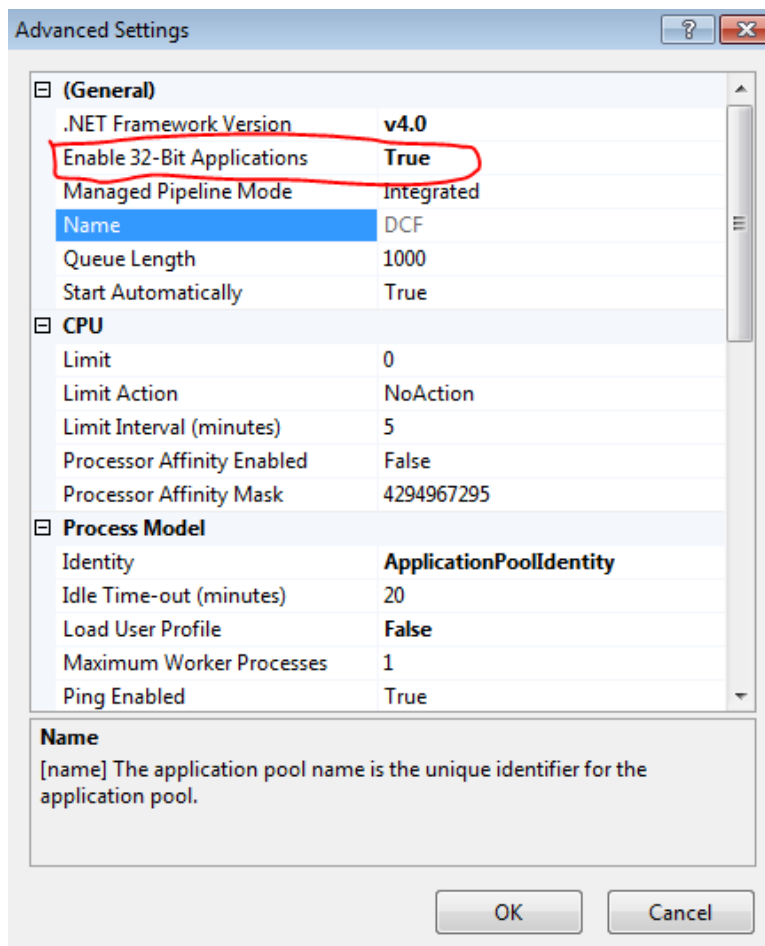
- 1 Open a command window.
- 2 cd c:\inetpub\wwwroot\dcf.
- 3 Regsvr32 CaprockDCT.dll.

If the DCF UI is not installed get a copy of the dll from another server and register it.

If the environment is a 64bit OS like Windows® 2008 R2 you will need make sure that all IIS Application pools used by the LNOS UI's are set to run in 32Bit mode.

- 1 Open IIS Service Manager
- 2 Navigate to the list of Application Pools. Generally you should find several Application Pools with similar names as the VDir of the application.

For each open the "Advanced Settings..." window and make sure "Enable 32bit Applications=true"



Verification

At this point the existing environments should work as before, with the only difference that the DCFSession component is used to manage session records. The table used is called DCFSession and exists in the DCF database of the environment.

- 1 Open Microsoft® Corporation's SQL Server® Management Studio.
- 2 Start a new query.
- 3 Execute the following:
 - USE DCF
 - SELECT * FROM DCFSession (nolock)

Records should show up once a user has success fully logged on.

Step 2: Setup of a Core UI.NET

This section describes the setup process for the Microsoft® .NET-based LNOS UI, which will serve as the entry point for certain users. This URL grants access to FW by using the DOMAIN/LDAP information.

When running a WebFarm of webapp servers the following has to be repeated on each webapp server

Install Core UI.NET

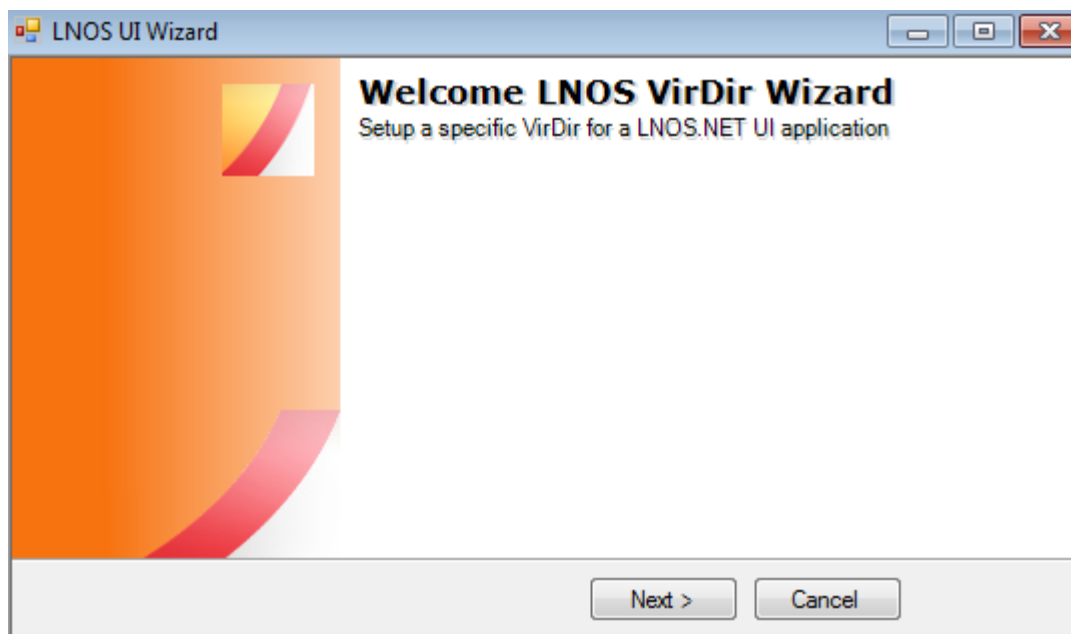
For this installation, v12.1.6 is needed. If an earlier version is installed please uninstall it from the control panel.

Run the installer found @ <\\svratbuild1\Project Builds\CoreUI.NET\12.1.5>

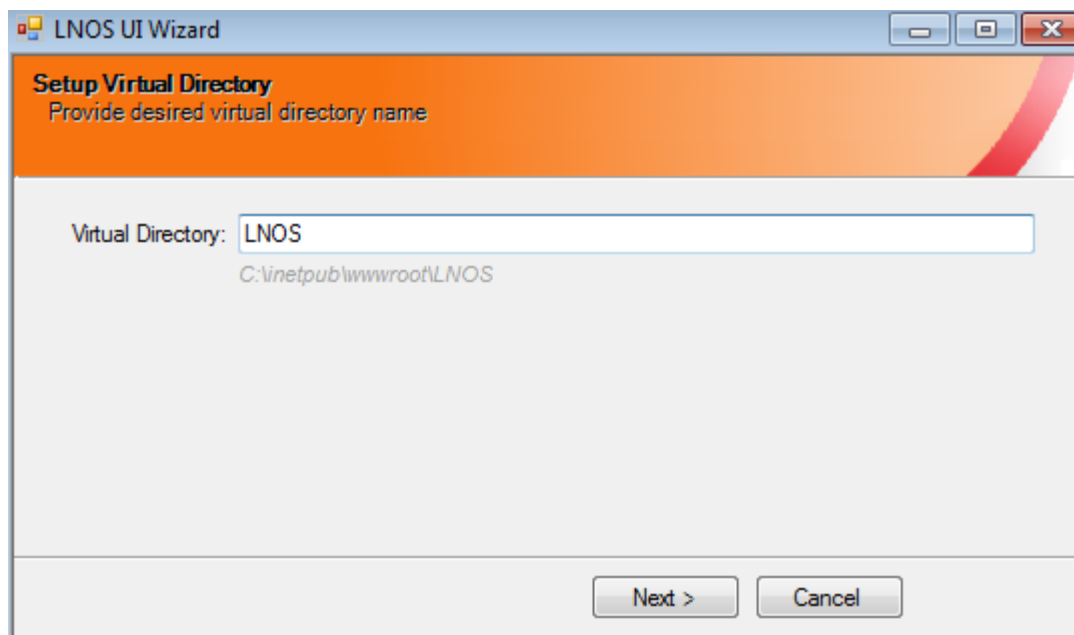
Setup VirDir Instance

Next we will be creating a LNOS VDir under the local IIS server.

- 1 Select **Start > All Programs > Descartes > Core UI UI > Short cut to UI Installer.**



- 2 Click **Next.**

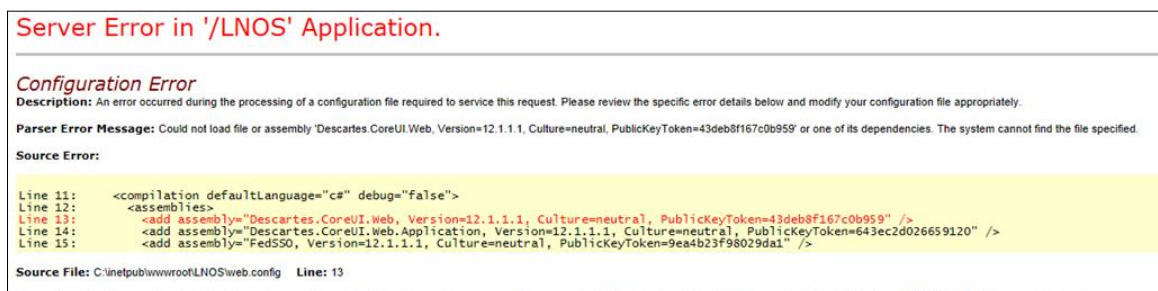


- 3 Provide the name of the Virtual Directory and click **Next**.
- 4 Wait until the progress bar is completed and click **Next**.
- 5 Click **Next** again.

At this point you will have a familiar looking folder under c:\inetpub\wwwroot\LNOS

Quick First Time Test

- 1 Open the browser and navigate to <http://localhost/LNOS/Default.aspx>.
- 2 A configuration error may appear.

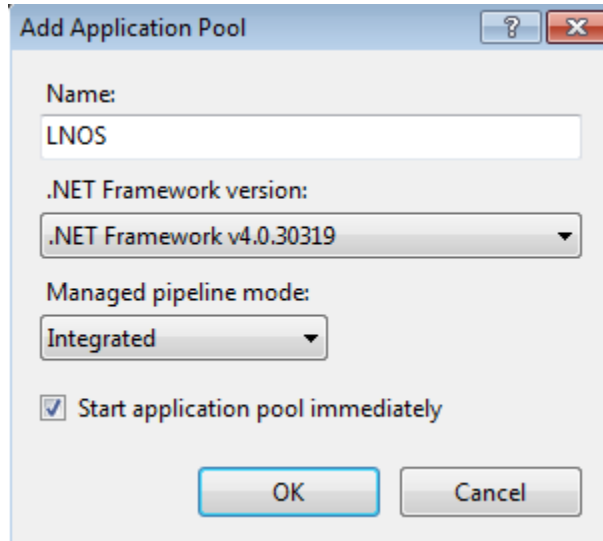


Resolution:

Setup a new Application Pool "LNOS", using Microsoft® .NET 4.0 and Integrated. Link the new app-pool to the LNOS Virtual Directory.

- 1 Open IIS Manager

- 2 Got to list of Application Pools, right click and select **Add App Pool**.



Add Application Pool

Name:
LNOS

.NET Framework version:
.NET Framework v4.0.30319

Managed pipeline mode:
Integrated

☒ Start application pool immediately

OK Cancel

- 3 Click **OK**.
- 4 Expand the "Default Web Site" node.
- 5 Select "LNOS" Virtual Directory. From the action panel, click **Basic Settings...**
- 6 Click the **Select** button and select the "LNOS" app pool. Click **OK**.

7 Click **OK**.

8 Try <http://localhost/LNOS/Default.aspx> again.

Next error is probably an access error:

Server Error in '/'LNOS' Application.

Access to the path 'C:\inetpub\wwwroot\LNOS\Config.xml' is denied.

Resolution:

Grant full rights to the root folder of LNOS. The Microsoft® .NET version needs full access to all folders.

- 1 Open Windows® Explorer.
- 2 Navigate to c:\inetpub\wwwroot\LNOS.
- 3 Right click the folder and select properties.
- 4 Select the security tab. Ensure that LOCALMACHINE\IUSR has FULL Control. Ensure that LOCALMACHINE\ISS_IUSRS have FULL Control.
- 5 Try <http://localhost/LNOS/Default.aspx> again.

If the Login Screen appears, try to login with a Client-SecAdmin user.
OrgName/OrgName-SecAdmin/cs.

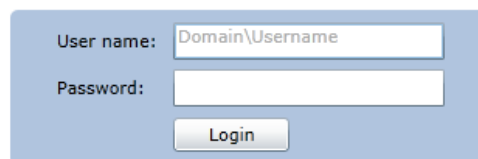
Reconfigure .NET UI to use the DCFSession

Session is still managed "in memory". In order to have a successful working Federated SSO environment, the LNOS VirDir needs to be changed to use the same Session manager as the ASP application.

- 1 Open <http://localhost/LNOS/Default.aspx>.
- 2 Login with "local/configurationmanager/change".
- 3 Select **UI Tools > Settings**.
- 4 If you see the "Install Silverlight Logo" please do so.

If you see the Domain/Password app, provide a valid domain-user/password.

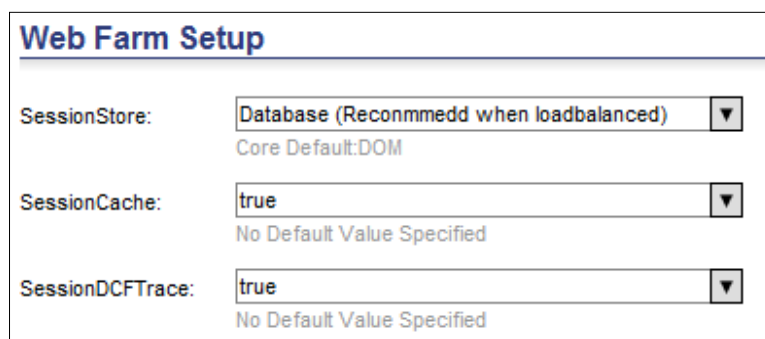
Domain Authentication



A login form with a light blue background. It contains two input fields: "User name:" with the text "Domain\Username" and "Password:". Below the password field is a "Login" button.

Use a domain-user known within the domain of the webapp server.

- 5 Upon successful verification of the user, you will be able to see the UI Configuration screen. Similar the "ASP" UI Tools --> Settings options.
- 6 Set the section "Web Farm Setup"



A configuration screen titled "Web Farm Setup". It contains three settings, each with a dropdown menu and a default value:

- SessionStore:** Database (Recommedd when loadbalanced) [dropdown arrow]
Core Default:DOM
- SessionCache:** true [dropdown arrow]
No Default Value Specified
- SessionDCFTrace:** true [dropdown arrow]
No Default Value Specified

➡ **Note—** SessionDCFTrace is for test purposes and should be false when working. Under GenFiles folder there will be a specific log folder/files for the session calls.

Setup the FedSSO Application into LNOS URI

So far the setup includes the use of Default.aspx, which normally be used to present the login page and after login the navigation/pages. The Default.aspx is not WIF aware. Instead a specific module has been created that provides a new page called DefaultSSO.aspx; the WIF logic and the new module will create the new users.

- 1 Open Windows® Explorer.
- 2 Navigate to C:\inetpub\wwwroot\LNOS\GeneratedFiles.
- 3 Open Config.xml with notepad.
- 4 Add the following xml within the <Config></Config> :


```
<Assemblyvalue="FedSSO, Version=12.1.1.1, Culture=neutral, PublicKeyToken=9ea4b23f98029da1" key="FedSSO" HasResources="True"/>
```
- 5 Save the changes, and recycle the LNOS application pool.
- 6 The DefaultSSO.aspx page will be created with the first page request.

Step 3. Enable Claims Authentication

Required software

Microsoft® .NET Framework 3.5, Windows® Identification Foundation (WIF) libraries:

<http://www.microsoft.com/en-us/download/details.aspx?id=17331#Requirements>

Download "6.1 x64" for 64bit OS or "6.1 x86" for 32 bit OS

Windows® Identification Foundation SDK Primarily to get the FedUtil.exe installed

<http://www.microsoft.com/en-us/download/details.aspx?id=4451>

Download and install the 4.0 version.

IIS SSL certificate issued by (CA) used by application URL

IIS SSL certificate issued by (CA) used by the STS (ADFS 2.0)

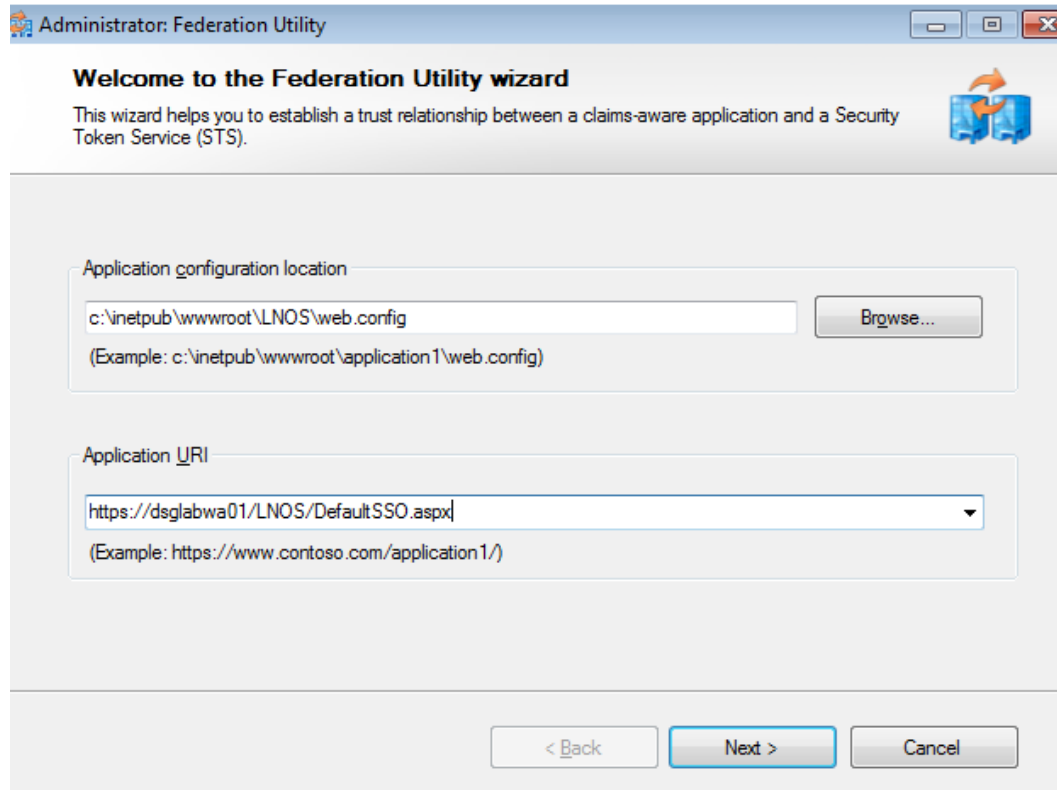
General Concept

To enable claims authentication it requires both the Application (claims consumer) and STS to be configured so that it will setup a trust relationship.

Enable Application with WIF

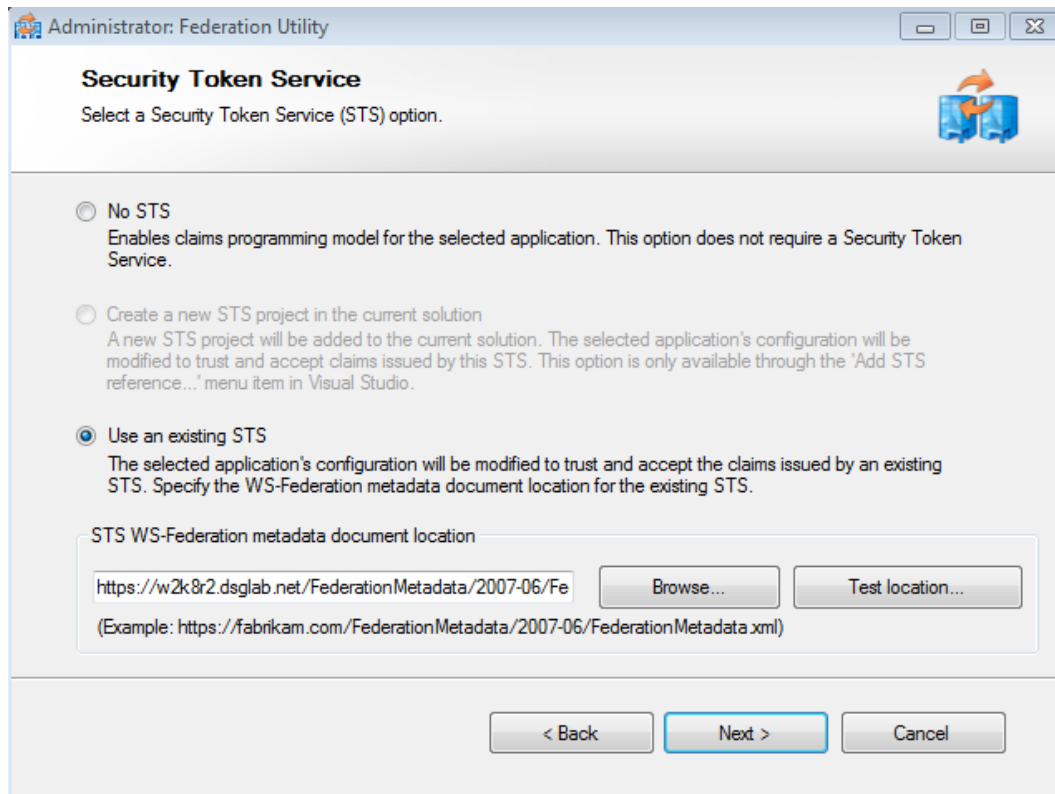
- 1 Open Windows® Explorer.
- 2 Browse to c:\inetpub\wwwroot\LNOS.
- 3 Disable "readonly" on web.config; this file will be modified.
- 4 Delete folder "FederationMetadata"; a new version will be created.

- 5 Select **Start > All Programs > Administrative Tools > Windows Identity Foundation Federation Utility**.
- 6 Provide a full path the web.config of the application.
- 7 Specify the Application UI. HTTPS is suggested.
- 8 Click **Next**.

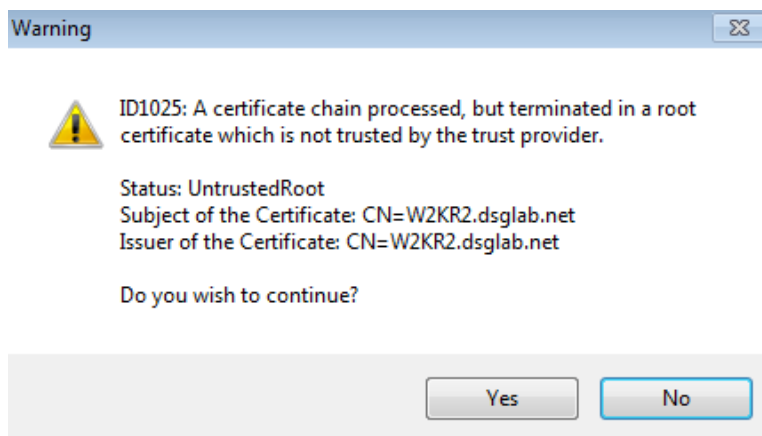


The image shows a screenshot of the 'Administrator: Federation Utility' wizard window. The title bar reads 'Administrator: Federation Utility'. The main heading is 'Welcome to the Federation Utility wizard'. Below this, a description states: 'This wizard helps you to establish a trust relationship between a claims-aware application and a Security Token Service (STS)'. To the right of the text is an icon of two blue cubes with an orange arrow pointing from one to the other. The wizard has two main input sections. The first section is labeled 'Application configuration location' and contains a text box with the path 'c:\inetpub\wwwroot\LNOS\web.config' and a 'Browse...' button. Below the text box is an example: '(Example: c:\inetpub\wwwroot\application1\web.config)'. The second section is labeled 'Application URI' and contains a dropdown menu with the value 'https://dsglabwa01/LNOS/DefaultSSO.aspx'. Below the dropdown is an example: '(Example: https://www.contoso.com/application1/)'. At the bottom of the window are three buttons: '< Back', 'Next >', and 'Cancel'.

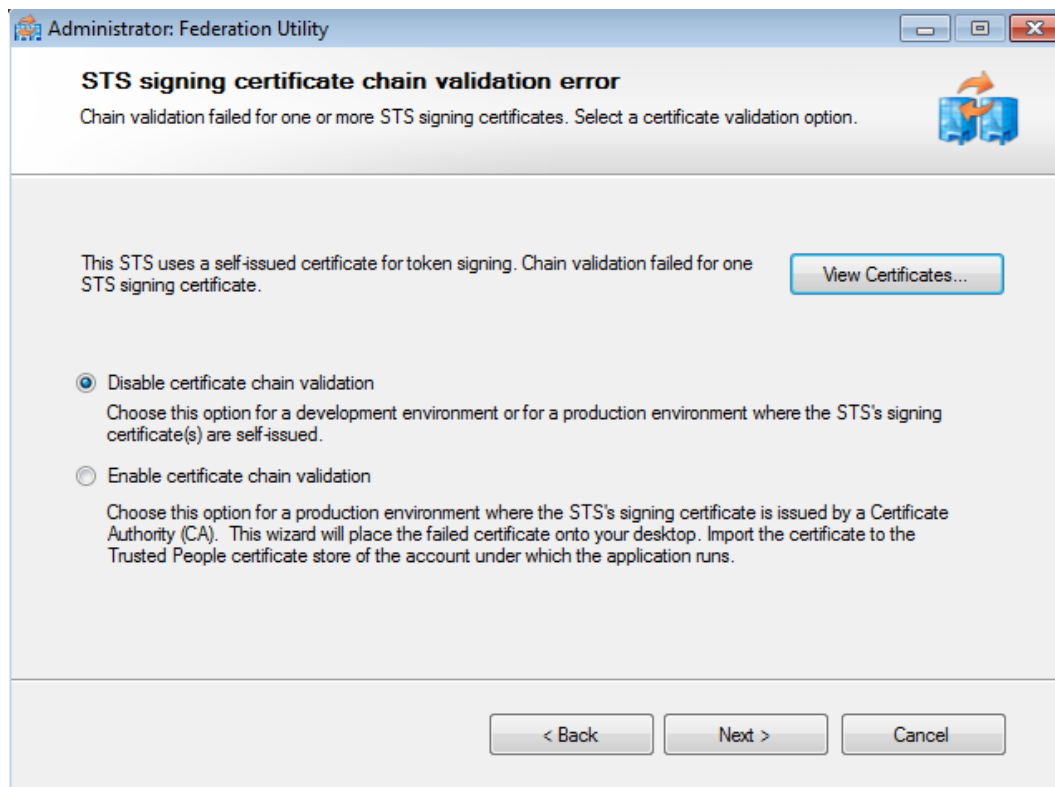
- 9 Select an existing STS and provide a full URL to the FederationMetadata.xml file.
- 10 Click **Next**.



If you receive the certificate warning, click **Yes**.

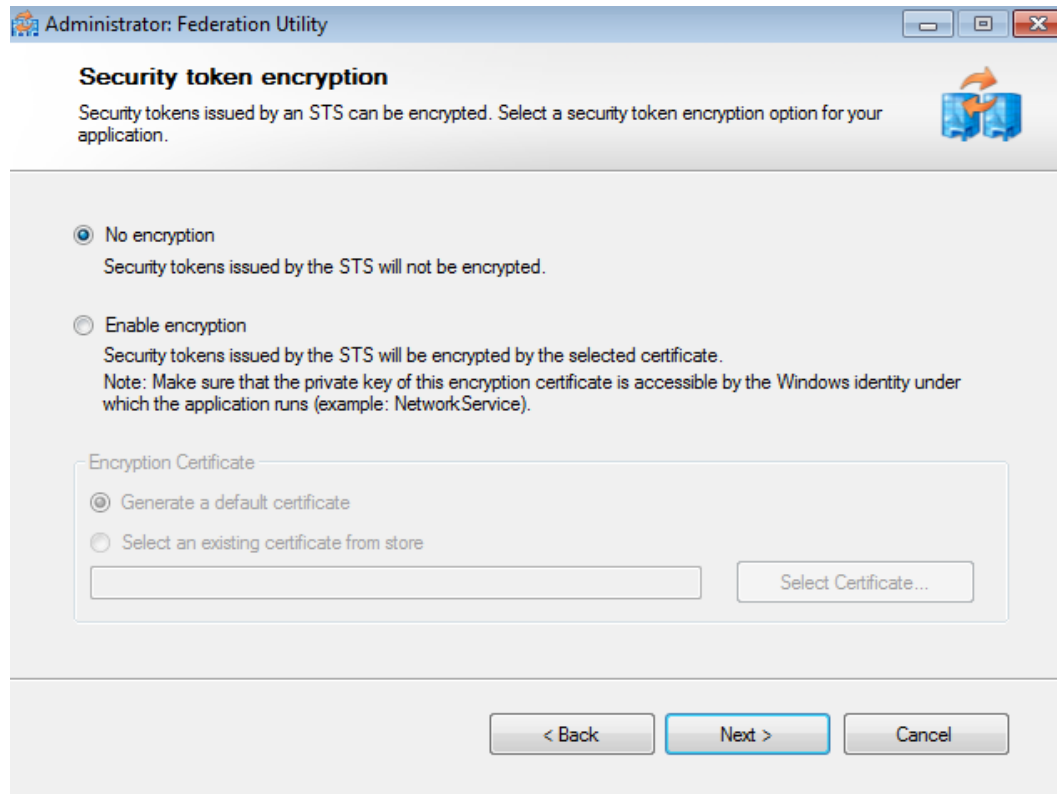


11 Select **Disable certificate chain validation.**



Enable certificate chain validation in a production environment.

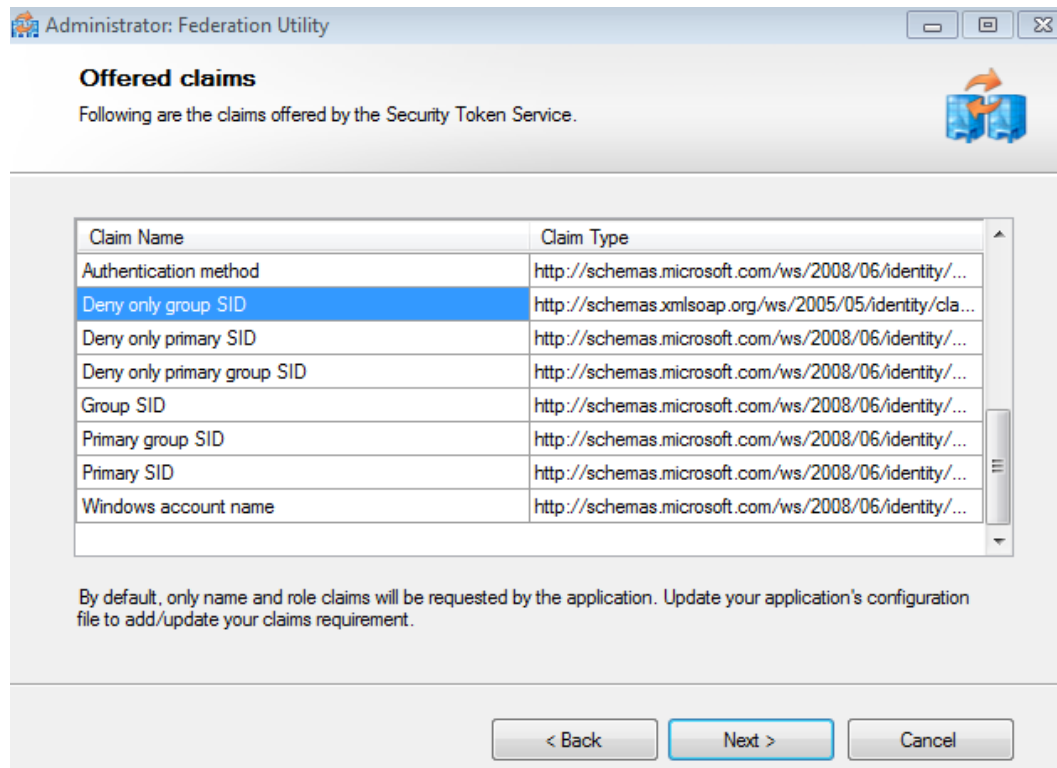
12 Select **No Encryption**. Click **Next**.



The image shows a screenshot of the 'Administrator: Federation Utility' window. The title bar reads 'Administrator: Federation Utility'. The main heading is 'Security token encryption'. Below the heading, there is a descriptive text: 'Security tokens issued by an STS can be encrypted. Select a security token encryption option for your application.' To the right of this text is a small icon of two blue boxes with an orange arrow pointing from one to the other. There are two radio button options: 'No encryption' (selected) and 'Enable encryption'. Below 'No encryption' is the text 'Security tokens issued by the STS will not be encrypted.' Below 'Enable encryption' is the text 'Security tokens issued by the STS will be encrypted by the selected certificate.' followed by a note: 'Note: Make sure that the private key of this encryption certificate is accessible by the Windows identity under which the application runs (example: NetworkService).' Below these options is a section titled 'Encryption Certificate' containing two radio button options: 'Generate a default certificate' (selected) and 'Select an existing certificate from store'. Below the 'Select an existing certificate from store' option is a text box and a button labeled 'Select Certificate...'. At the bottom of the window are three buttons: '< Back', 'Next >' (highlighted with a blue border), and 'Cancel'.

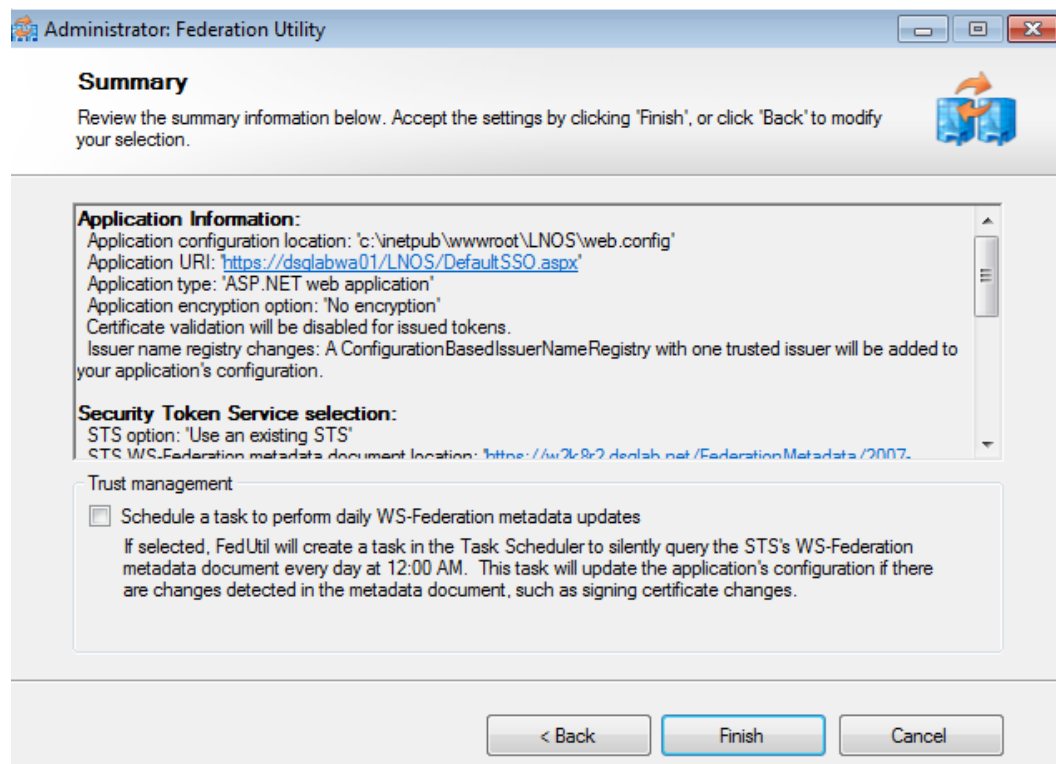
Enable Encryption would be recommended on a production environment.

13 Click **Next**.



The Claims supported by LNOS FedSSO application need to be added manually. This utility is not support or import additional claims.

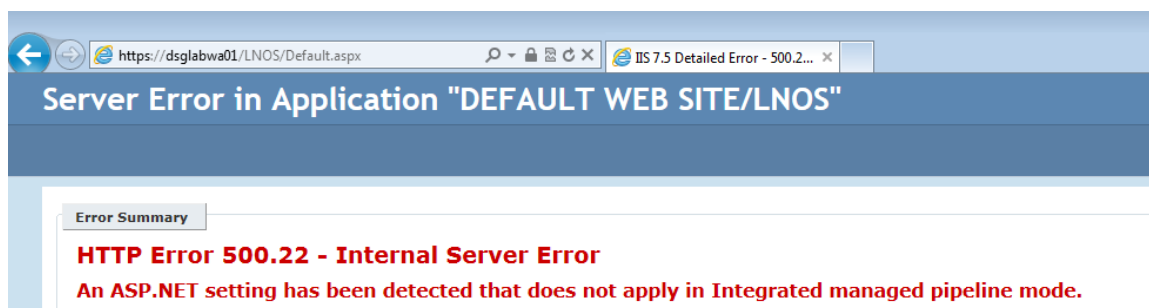
14 Summary, Click **Finished**.



The Schedule a task option would be helpful on a production environment.

15 At this point the URL <https://dsglabwa01/LNOS/Default.aspx> is no longer working.

The following message may appear:



Resolution:

Run the appcmd update from the command line.

- 1** Open command window.
- 2** Run the following command:

```
%SystemRoot%\system32\inetsrv\appcmd migrate config "Default Web Site/LNOS/"
```

```
C:\>%SystemRoot%\system32\inetsrv\appcmd migrate config "Default Web Site/LNOS/"
Successfully migrated section "system.web/httpModules".
Successfully migrated section "system.web/httpHandlers".
C:\>_
```

<https://dsqglabwa01/LNOS/Default.aspx> will return an error

<https://dsqglabwa01/LNOS/DefaultSSO.aspx> will actually redirect to the STS server but will fail.

STS does not "trust" this URL yet.

Setup LNOS Specific Claims list

Next we will have to update the web.config and FederationMetadata.xml with the LNOS specific claims.

The web.config modification:

- 1 Open Windows® Explorer.
- 2 Navigate to c:\inetpub\wwwroot\LNOS.
- 3 Open web.config with notepad.
- 4 From the root find the section:

```
microsoft.identityModel/service/applicationService/claimTypeRequired
```

- 5 Add the following four claims:

```
<claimType type="http://schemas.xmlsoap.org/ws/2005/05/identity/claims/name" optional="false" />
<claimType type="http://descartes.com/claims/2011/01/organization" optional="false" />
<claimType type="http://descartes.com/claims/2011/01/usergroup" optional="false" />
<claimType type="http://descartes.com/claims/2011/01/role" optional="false" />
<claimType type="http://descartes.com/claims/2011/01/redirectURL" optional="true" />
```

The WIF logic will ensure that these claims are being provided before handing them over to the application.

FederationMetadata.xml modifications:

- 1 Open Windows® Explorer.
- 2 Navigate to C:\inetpub\wwwroot\LNOS\FederationMetadata\2007-06.
- 3 Open FederationMetadata.xml file. (if you have VS2010 use and click CTRL+E, D).
- 4 Within:

```
<fed:ClaimTypesRequested>
```

Past the following xml (remove original)

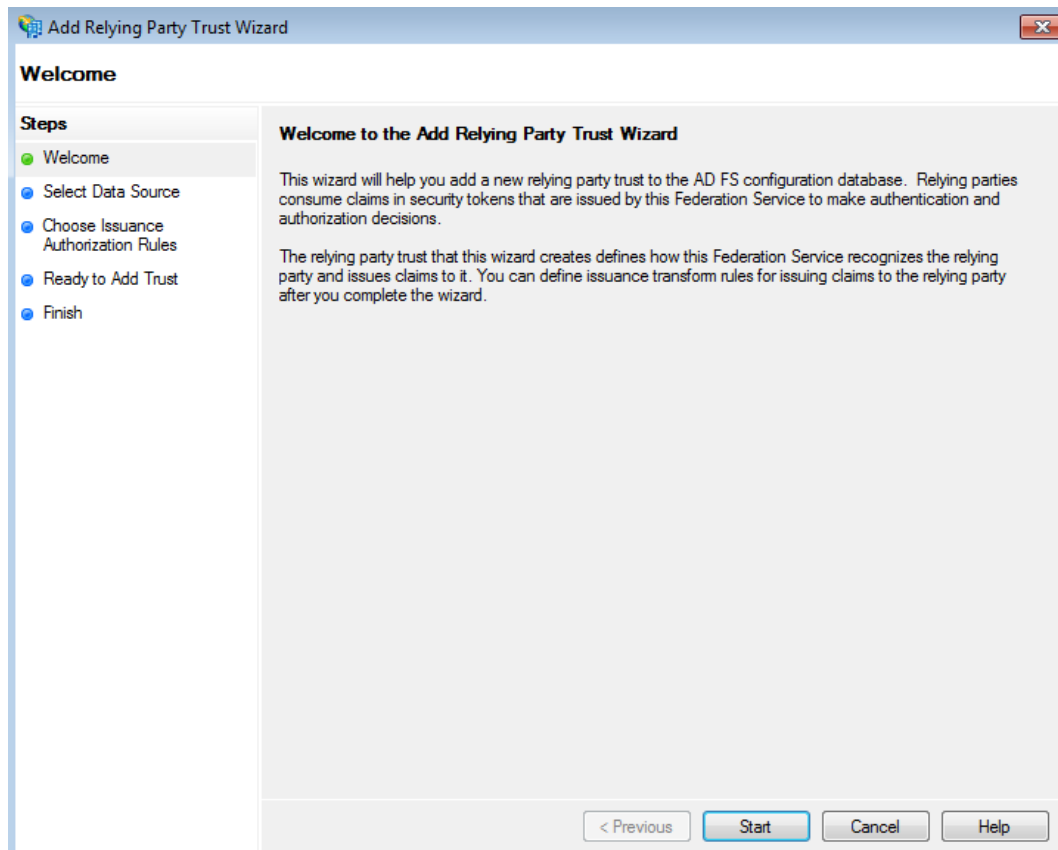
```
<auth:ClaimTypeUri="http://schemas.xmlsoap.org/ws/2005/05/identity/claims/
name" Optional="false" xmlns:auth="http://docs.oasis-
open.org/wsfed/authorization/200706" /><auth:ClaimTypeUri="http://descartes
.com/claims/2011/01/organization" Optional="false" xmlns:auth="http://docs.o
asis-
open.org/wsfed/authorization/200706" /><auth:ClaimTypeUri="http://descartes
.com/claims/2011/01/usergroup" Optional="false" xmlns:auth="http://docs.oasi
s-
open.org/wsfed/authorization/200706" /><auth:ClaimTypeUri="http://descartes
.com/claims/2011/01/role" Optional="false" xmlns:auth="http://docs.oasis-
open.org/wsfed/authorization/200706" /><auth:ClaimTypeUri="http://descartes
.com/claims/2011/01/redirectURL" Optional="true" xmlns:auth="http://docs.oas
is-open.org/wsfed/authorization/200706" />
```

This section will provide the STS with the information of the claims this application consumes.

Setup STS Trust with Application

Setting up STS trust for the application is the next step. To setup, the full URL is necessary to DefaultSSO.aspx and the FederationMetadata.xml.

- 1 Logon to the server running ADFS 2.0 (STS).
- 2 Open ADFS 2.0 Manager.
- 3 Select **Start > All Programs > Administrative Tools > AD FS 2.0 Manager**.
- 4 On the Action panel on the left select **Add Relying Party Trust**.
- 5 Click **Start**.



- 6 Browse directly to the location of the FederationMetadata.xml within the application. In this case
[\\dsq1abwa01\c\\$\inetpub\wwwroot\LNOS\FederationMetadata\2007-06\FederationMetadata.xml](\\dsq1abwa01\c$\inetpub\wwwroot\LNOS\FederationMetadata\2007-06\FederationMetadata.xml)
- 7 Click **Next**.

The screenshot shows the 'Add Relying Party Trust Wizard' window. The title bar reads 'Add Relying Party Trust Wizard'. The main heading is 'Select Data Source'. On the left, a 'Steps' pane lists: 'Welcome', 'Select Data Source' (highlighted), 'Choose Issuance Authorization Rules', 'Ready to Add Trust', and 'Finish'. The main area contains three radio button options:

- ☐ Import data about the relying party published online or on a local network
Use this option to import the necessary data and certificates from a relying party organization that publishes its federation metadata online or on a local network.
Federation metadata address (host name or URL):

Example: fs.contoso.com or https://www.contoso.com/app
- ☒ Import data about the relying party from a file
Use this option to import the necessary data and certificates from a relying party organization that has exported its federation metadata to a file. Ensure that this file is from a trusted source. This wizard will not validate the source of the file.
Federation metadata file location:
- ☐ Enter data about the relying party manually
Use this option to manually input the necessary data about this relying party organization.

At the bottom are four buttons: '< Previous', 'Next >' (highlighted), 'Cancel', and 'Help'.

8 Click **Next**.

The screenshot shows the 'Add Relying Party Trust Wizard' window. The title bar reads 'Add Relying Party Trust Wizard'. The main heading is 'Specify Display Name'. On the left, a 'Steps' pane lists: Welcome, Select Data Source, Specify Display Name (current step), Choose Issuance Authorization Rules, Ready to Add Trust, and Finish. The main area contains the instruction 'Type the display name and any optional notes for this relying party.' Below this, the 'Display name:' field contains 'DSGLABWA01 - LNOS'. The 'Notes:' text area contains 'This is the LNOS Federated SSO url.'. At the bottom are buttons for '< Previous', 'Next >', 'Cancel', and 'Help'.

9 Check **Permit all users**.

10 Click **Next**.

The screenshot shows a Windows-style wizard window titled "Add Relying Party Trust Wizard". The current step is "Choose Issuance Authorization Rules". On the left, a "Steps" pane lists the following steps: Welcome, Select Data Source, Specify Display Name, Choose Issuance Authorization Rules (which is highlighted), Ready to Add Trust, and Finish. The main content area explains that issuance authorization rules determine whether a user is permitted to receive claims for the relying party. It asks the user to choose one of two options for the initial behavior of this relying party's issuance authorization rules:

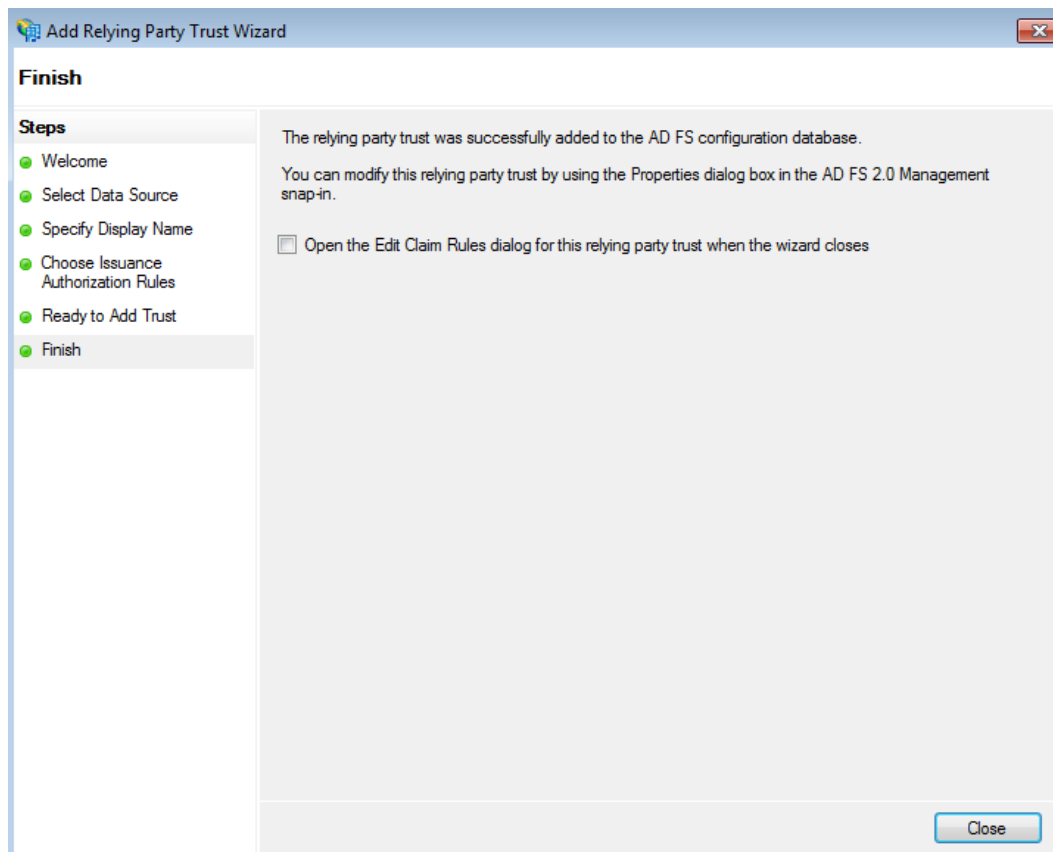
- ☒ **Permit all users to access this relying party**
The issuance authorization rules will be configured to permit all users to access this relying party. The relying party service or application may still deny the user access.
- ☐ **Deny all users access to this relying party**
The issuance authorization rules will be configured to deny all users access to this relying party. You must later add issuance authorization rules to enable any users to access this relying party.

Below these options, a note states: "You can change the issuance authorization rules for this relying party trust by selecting the relying party trust and clicking Edit Claim Rules in the Actions pane." At the bottom right, there are four buttons: "< Previous", "Next >" (which is highlighted with a blue border), "Cancel", and "Help".

11 Review the Relying Trust party. Click **Next**.

The screenshot shows the 'Add Relying Party Trust Wizard' window. The title bar reads 'Add Relying Party Trust Wizard'. The main window has a left sidebar with a 'Steps' list: Welcome, Select Data Source, Specify Display Name, Choose Issuance Authorization Rules, Ready to Add Trust (highlighted), and Finish. The main area is titled 'Ready to Add Trust' and contains the text: 'The relying party trust has been configured. Review the following settings, and then click Next to add the relying party trust to the AD FS configuration database.' Below this text is a tabbed interface with tabs: Monitoring, Identifiers, Encryption, Signature, Accepted Claims, Organization, Endpoints, and N. The 'Monitoring' tab is active. It contains the text 'Specify the monitoring settings for this relying party trust.' followed by a text box for 'Relying party's federation metadata URL:'. Below this are two checkboxes: 'Monitor relying party' and 'Automatically update relying party', both of which are unchecked. Further down, it says 'This relying party's federation metadata data was last checked on: < never >' and 'This relying party was last updated from federation metadata on: < never >'. At the bottom right are four buttons: '< Previous', 'Next >', 'Cancel', and 'Help'.

12 Click **Close**.



Step 4: Add Claims Mapping Rules

Claims Mapping Rules

Once an application is setup as a trusted party, the claims required by the application have to be populated from the user currently logged into the domain.

ADFS 2.0 has many different ways to populate the claims: from Active Directory®, LDAP, SQL and Custom Rules.

LNOS Claims

The LNOS WIF implementation consumes five claims:

- <http://schemas.xmlsoap.org/ws/2005/05/identity/claims/name>
Mandatory. This claim represents the login-name of the user who wants to login.
- <http://descartes.com/claims/2011/01/organization>
Mandatory. This claim represents the organization name as known within LNOS.
- <http://descartes.com/claims/2011/01/usergroup>

Mandatory. This claim is only used when the user (login-name) does not exist within LNOS.

Multiple usergroup claims can be provided, one usergroup per claim. From the list of provided usergrp claims "atleast" one usergrp has to exist with LNOS. If user gets created the "first" existing usergrp will be set as the "default"

- <http://descartes.com/claims/2011/01/role>

Mandatory. This claim is only used when the user (login-name) does not exist within LNOS.

Multiple role claims can be provided, one role per claim. From the list of provided role claims "atleast" one role has to exist with LNOS. All matching roles will be assigned.

- <http://descartes.com/claims/2011/01/redirectURL>

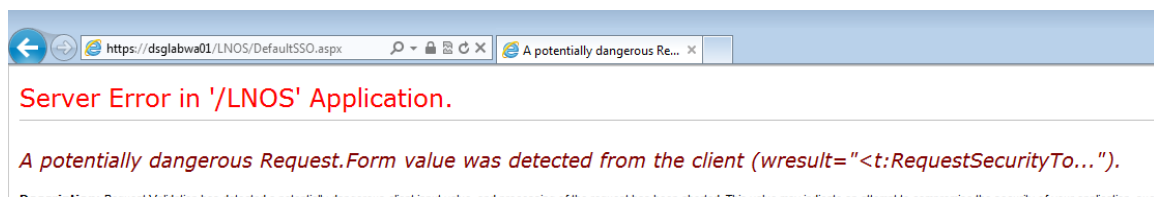
Optional. Defines the LNOS URL of the ASP application. After successful login the user will be redirected to this URL.

Claim setup

Next is setup of the ADFS 2.0 trusted application in such a way that information from LDAP/Active Directory is translated in the LNOS Claims. Open ADFS 2.0 Management on the ADFS server.

Complete Round Trip Test

Within test environments one could receive the following error:



Caused by XML being sent to the IIS.

Resolution:

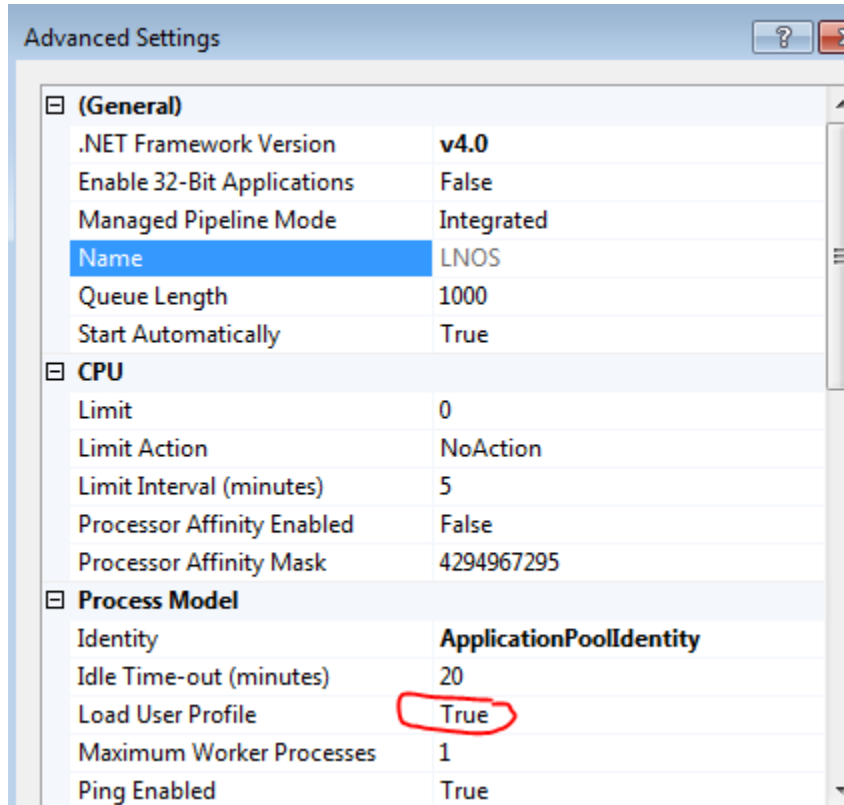
- 1 Open web.config, and find httpRuntime node.
- 2 Add the attribute requestValidationMode="2.0"

Another potential issue:



Resolution:

Ensure that the user App Pool has "Load User Profile" set to True



Another potential issue: Screen appears blank and nothing happens. When one looks at the source of the page it will show some scripting. Due to browser security setting it might not allow JavaScript running.

Appendix G: Support and Handling of Miss Cases

The following sections describe all miss cases that Descartes Route Planner supports, describing how the system handles each combination of schedule setting and command XML attributes in UpdateStatus and UnassignRescheduleOrder.

API: UnassignRescheduleOrder

- If the order's stop (the delivery stop in the case of a complex order) was already completed, hooked, or spotted, the system returns an exception with the message "A successfully completed order can be neither unassigned nor rescheduled."
- If all stops of the order are in Pending, Enroute or Skipped status, the system unassigns the order. Otherwise, Descartes Route Planner creates an unassigned clone of the order. However, if the schedule setting does not allow the rescheduling of a partially completed order (FWSchedule.ReschedPartialCmplOrd=0), the clone is only created if the stop's last status prior to departed was Exception, Missed, NoFreight, Refused or Returned.

API: UpdateStatus

Use Case	Order Type	Stop Type	Command XML		Schedule Setting	Expected Outcome
			Attempt	Reschedule	CreateReturnOnMiss	
A	Simple	Pickup or Dropoff	0	0	(n/a)	System marks the stop as missed and changes the stop's location to the lat/long reported by the miss message.
B	Simple	Pickup or Dropoff	0	1	(n/a)	System unassigns the order.
C	Simple	Pickup or Dropoff	1	0	(n/a)	System marks the stop as missed.
D	Simple	Pickup or Dropoff	1	1	(n/a)	System marks the stop as missed and creates an unassigned clone of the order.

Use Case	Order Type	Stop Type	Command XML		Schedule Setting	Expected Outcome
			Attempt	Reschedule	CreateReturnOnMiss	
E	Complex	Pickup	0	0	0	System marks the stop as missed and changes the stop's location to the lat/long reported by the miss message.
F	Complex	Pickup	0	0	1	System marks the stop as missed and changes the stop's location to the lat/long reported by the miss message. If delivery stop was already fulfilled, the system does nothing. Otherwise, if there are sibling pickups that have already been completed, the system creates a return order for each such pickup. A return order is a double-ended order consisting of a pickup at the reported lat/long and a dropoff at a completed pickup stop. All the remaining stops of the order in the pending/skipped/enroute status are also missed.
G	Complex	Pickup	0	1	0	If none of the sibling pickups has been completed, none of the missed pickups were ever attempted, and the delivery wasn't fulfilled yet, the system unassigns the order. Otherwise, the current stop is marked as missed. If delivery stop was already fulfilled, the system does nothing further. Otherwise, the system marks all stops in Pending, Arrived, Enroute, Skipped and Begun status as missed. If FWSchedule.ReschePartialCmplOrd=1, the system creates an unassigned clone of the order.
H	Complex	Pickup	0	1	1	See Use Case G. The system also creates a return order for each completed pickup stop. A return order is a double-ended order consisting of a pickup at the reported lat/long and a dropoff at a completed pickup stop.

Use Case	Order Type	Stop Type	Command XML		Schedule Setting	Expected Outcome
			Attempt	Reschedule	CreateReturnOnMiss	
I	Complex	Pickup	1	0	0	System marks the stop as missed.
J	Complex	Pickup	1	0	1	System marks the stop as missed. If delivery stop was already fulfilled, the system does nothing else. Otherwise, if there are sibling pickups that have already been completed, the system creates a return order for each such pickup. A return order is a double-ended order consisting of a pickup at the current pickup location (not necessarily the same as the reported lat/long) and a dropoff at a completed pickup stop. All the remaining stops of the order in Pending, Skipped, Enroute status are also missed.
K	Complex	Pickup	1	1	0	System marks the current stop as missed. If delivery stop was already fulfilled, the system does nothing else. Otherwise, the system marks all stops in Pending, Arrived, Enroute, Skipped, Begun as missed. If FWSchedule.ReschePartialCmplOrd=1, the system creates an unassigned clone of the order.
L	Complex	Pickup	1	1	1	See Use Case K. The system also creates a return order for each completed pickup stop. A return order is a double-ended order consisting of a pickup at the current pickup location (not necessarily the same as the reported lat/long) and a dropoff at a completed sibling stop.

Use Case	Order Type	Stop Type	Command XML		Schedule Setting	Expected Outcome
			Attempt	Reschedule	CreateReturnOnMiss	
M	Complex	Dropoff	0	0	0	System marks the delivery stop as missed and changes the stop's location to the lat/long reported by the miss message. All Pending, Arrived, Enroute, Skipped and Begun pickups are marked as 'exception'.
N	Complex	Dropoff	0	0	1	See UseCase M. If there are sibling pickups that have already been completed, the system also creates a return order for each such pickup. A return order is a double-ended order consisting of a pickup at the reported lat/long and a dropoff at a completed pickup stop.
O	Complex	Dropoff	0	1	0	If none of the sibling pickups has been completed yet and none of the missed pickups were ever attempted, the system unassigns the order. Otherwise, the system marks the delivery stop as missed and all pickups in Pending, Arrived, Enroute, Skipped, Begun as exception. It also creates an unassigned clone of the order.
P	Complex	Dropoff	1	1	1	See Use Case O. The system also creates a return order for each completed pickup stop. A return order is a double-ended order consisting of a pickup at the reported lat/long and a dropoff at a completed pickup stop.

Use Case	Order Type	Stop Type	Command XML		Schedule Setting	Expected Outcome
			Attempt	Reschedule	CreateReturnOnMiss	
Q	Complex	Dropoff	1	0	0	The system marks the delivery stop as missed and marks all pickups in Pending, Arrived, Enroute, Skipped and Begun as 'exception'.
R	Complex	Dropoff	1	0	1	See Use Case Q. If there are sibling pickups that have already been completed, the system creates a return order for each such pickup. A return order is a double-ended order consisting of a pickup at the current delivery location (not necessarily the same as the reported lat/long) and a dropoff at a completed pickup stop.
S	Complex	Dropoff	1	1	0	RP marks the delivery stop as missed and all pickups in Pending, Arrived/, Enroute, Skipped and Begun as 'exception'. It also creates an unassigned clone of the order.
T	Complex	Dropoff	1	1	1	See Use Case S. The system also creates a return order for each completed pickup stop. A return order is a double-ended order consisting of a pickup at the current delivery location (not necessarily the same as the reported lat/long) and a dropoff at a completed pickup stop.

Appendix H: Mutual Client Authentication between Descartes Route Planner and Descartes wGLN

Descartes wGLN Authentication Extensions

The Descartes wGLN network services and Descartes Route Planner application support data exchange authentication methods selected by the client.

Mutual SSL Authentication to Descartes wGLN

Server functions

The server web service offers an access point requiring X509 client certificate authentication.

- The web server supports SHA-2 X509 chains.
- The web server proposes TLS v1.2 as the preferred protocol in the negotiation.
- The web server publishes a list of accepted public CA certificates including the public CA certificate specified by the client.
- The web server component validates the client certificate by comparing the hash fingerprint of the presented certificate to a configured value.
- Server component converts the client certificate into SSO requestor identity for data access control and auditing purposes.
- The mutual authentication requirement is limited to the web service interfaces used by Descartes Route Planner and is not available to the Provisioning and/or Tracking functions. For these, server-based SSL Authentication in combination with Basic Authentication credentials are required.

API Key Implementation Descartes wGLN Outbound

With Descartes wGLN support for the client API key authentication method, the user is responsible for configuring and renewing the application key in unison with the change of the application key on the Layer 7 appliance.

Management

Descartes wGLN Provisioning allows designated client staff members with the wGLNAdministrator role to enter the API Key identifier (username) and the API Key itself (password) in the communication profile for the client's Descartes Route Planner instance.

- The API key is stored in the Descartes wGLN configuration database after being encrypted using a symmetrical encryption mechanism.
- On the provisioning screen, the application ID is visible, but the Application Key is not; the application key is handled the same way as a remote password.
- Updates to the communication profile containing the Application Key results in an update of the Modified and Modified By fields in the record.

- A communication profile option 'Use [Client] Authorization Key' has been added as an optional configuration item to the communication profile.

Execution

- The API Key is only used for the URL specified in the communication profile.
- The same coding policies that apply to remote passwords are applied to the client API key. API keys will never be logged or added to diagnostics messages.
- The addition of the Authorization header does not require a HTTP 401. The Authorization header is added proactively based on the 'Use [Client] Authorization Key' configuration option.
- Similar to Basic Authentication, the API key is added as an Authorization member to the HTTP request header.
- Instead of a base64 encoding of "Basic username:password" as used for Basic Authentication, the value of this member is set to "APPID:APIKEY" (i.e. without the Basic prefix or the base-64 encoding).
- The regular HTTP return code semantics will be expected by the Layer 7 proxy. This includes HTTP 401 for authentication failures.

Descartes Route Planner Authentication Extensions

Previously, Descartes Route Planner authenticated inbound web-service requests based on credentials stored in the XML request. To enhance security, unencrypted credentials will no longer be stored. The existing content authentication has been replaced with a transport authentication mechanism.

Descartes Route Planner Inbound Authentication

For the authentication between the client inbound L7 device and the IIS instances that contain the Descartes Route Planner HTTP listeners, the following methods exist:

Microsoft® Corporation's Directory® authentication: If the L7 device and the Descartes Route Planner IIS instances both support Kerberos authentication against the same active domain, the client may elect to use this authentication mechanism between L7 and IIS.

SSL client authentication: The L7 device initiates a mutual authenticated SSL session with IIS. IIS can be configured to accept this session and convert to an active directory identity to which the L7 key is mapped.

In either of these alternatives, Descartes Route Planner accepts the identity as validated by the IIS (i.e. the local operating system). Descartes Route Planner associates the operating system identity with an internal Descartes Route Planner identity that is used to authorize processing of the content of the message.

Server Functions

The Descartes Route Planner message handler has been extended with an Active Directory® authentication module.

- The Active Directory® authentication module is activated based on a configuration setting that applies to all back-end web-service interface.
- The identity mapping between the Active Directory® user and the Descartes Route Planner identity is stored in the Descartes Route Planner database.
- The internal token associated with the Descartes Route Planner identity is stored encrypted.

Descartes Route Planner Outbound Authentication

For the authentication from Descartes Route Planner to either the L7 device or Descartes wGLN directly, Descartes recommends the use of SSL client authentication. The X509 private keys can be managed and protected by the operating system, and no proprietary mechanisms are needed.

Client functions

Details on the Descartes Route Planner HTTPs communication client is provided below:

- An additional URL is configured to provide outbound HTTPs POST interaction using a SSL client certificate.
- The SHA-1 fingerprint of the key pair stored in the operating system Machine key store is configured in the Descartes Route Planner configuration database.
- The private key is accessible by the operating system identity running the Descartes Route Planner application pool containing the communication client.
- The key pair should be signed by a well-known public CA.
- The Descartes Route Planner communication client assumes that the CN on the certificate matches FQDN in the destination URL.
- The Descartes Route Planner communication client relies on the operating system to validate Certificate Revocation status of the server certificate.
- The Descartes Route Planner HTTPs Communication client supports SHA-2 X509 chains.
- The team operationally responsible for Descartes Route Planner will ensure that the certificate is valid and will notify the recipient (either L7 or Descartes wGLN) if the key is replaced.

Appendix I: Retrieving LNOS Pathing Service Version Information

Using a web interface, users can return LNOS Pathing Service version information in a format specified in the Accept header by the client. LNOS Pathing Service will also set the Content-Type header in the response consistent with the content sent.

Use the following URL to retrieve version information, where <server> is the name or IP of the server where LNOS Pathing Service is running.

`http://<server>/pathingservice/lnospathingservice.aspx?version`

The following scenarios are supported by this feature when the specified URL is posted in a browser:

- If Accept header is set to "text/xml", LNOS Pathing Service will return version information in XML format, e.g.:

```
<Response>
  <Result>Success</Result>
  <Version>16.9.1.838</Version>
</Response>
```
- If Accept header is set to "application/json", version information will be returned in json format, e.g.:

```
{
  "Response": {
    "Result": "Success",
    "Version": "16.9.1.838"
  }
}
```
- If Accept header is set to "text/plain", version information will be returned in a string, e.g. "16.9.1.838" (no quotes).
- If Accept header is not set (*/*), version information is returned in XML format as detailed above.

Additionally, the **Test** button used to verify the connection to LNOS Pathing Service will also use the supported header and format the version information as specified above.