DESCARTES™

EXTERNAL PATHER ADMINISTRATOR GUIDE

Descartes® Route Planner™

17.05
May 2017

The Descartes Systems Group Inc. | TSX: DSG | NASDAQ: DSGX | 120 Randall Drive, Waterloo, Ontario, N2V 1C6, Canada
Toll Free 800.419.8495 | Int'l 519.746.8110 | info@descartes.com | www.descartes.com
CONFIDENTIAL AND PROPRIETARY TO THE DESCARTES SYSTEMS GROUP INC. AND ITS AFFILIATES

© 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 AltaMap, Descartes wGLN, Descartes Global Logistics Network (GLN), Descartes, the phrase "Helping Customers Deliver," and the Descartes logo are either registered trademarks or trademarks of Descartes and/or its affiliates.

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

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

TomTom is a registered trademark of TomTom International BV.

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

Descartes® Route Planner™
16.06

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

Document Conventions 5
Overview 6
WebService Components 6
1. LNOS Pathing Service 6
2. Route Matrix Web Service (RMWS) 7
3. LNOS Map Editor 8
4. GeoServerWS 9
5. TileServer 9
Prerequisites for External Pather 11
Cache Invalidation and Upgrade Planning (Descartes AltaMap, Map Data, LNOS Pathing Service) 12
External Pathing Descartes Route Planner Configuration 13
Enabling the UI/Interfaces to Use External Pather (Descartes AltaMap) 13
Enabling BGO to Use External Pather (Descartes AltaMap) 15
Purging existing cache 15
Handling Commercial Pathing Restrictions 16
Commercial Pathing Configuration (UI) 16
Hazmat Restrictions (UI) 19
Schedule Group Configuration 20
Localizing the Map Data 21
Minimum/Maximum Vehicle Profiles 22
External Pathing Violation Radius 23
Configuring Routing Parameters 23
Settings Used by External Pather 24
Configuring Speed Categories 24
Allowable Road Types 25
Optimization Costs 26
Traffic Patterns 26
Ignoring Commercial Restrictions near the Origin/Destination 26
Additional RMPI Configuration Changes 27
Traffic Patterns with Partial Day of Week Configuration 27
Traffic Pattern Interpolation 28
External Pathing LNOS Pathing Service Web Service Configuration 30
Basic Configuration 30
Maintenance/Performance 30
Communication 31
Pathing Methods 32
External Pathing Advanced Configuration - Performance 33

Optimization Terminology 33
Understanding the External Pather Computation Time 34
Understanding the External Pather Caching Model 34
Importance of BGO in Caching with the External Pather 35
Speed and Vehicle Profile(s) 36
Maximum and Minimum Vehicle / Number of Slots Explained 36
Number of Vehicle Slots Explained 37
Great Circle Mile (GCM) Filter 37
LNOS Pathing Service Recycling 39
External Pather Timeouts 39
Recommended Timeout Configurations: 41
Using Map Edits with the LNOS Pathing Service 42
Overview 42
Deployment Stages (the "Status") 42
Delayed Activation of Map Edits to Preserve Cache 43
Appendix 44
Appendix A: RMPI configuration settings for External Pather 44
Debugging settings 46
Appendix B: BGO Configuration Settings for External Pather 48
Appendix C: Example RMPI/BGO Configuration File 50
UI/BIF/BGO Rmpi.ini 50
BGO Local.properties 50
Appendix D: DMDebugLog parsing 51
Basic Parsing using Microsoft® Corporation's Windows® PowerShell 51
Appendix E: Windows® PowerShell Script to Shutdown Pathing Service and Replace Cache 53
Appendix F: External Pather in Larger Deployments Best Practices 56
BGO Dataslice Design 56
Caching Under Different Business Models 56
Dividing the LNOS Pathing Service by Region 56
Appendix G: External Router vs. LMB summary 58
Appendix H: Incremental Migration /Priming LNOS Pathing Service in Parallel 60
Create Organization for External (Descartes AltaMap) pathing 60
Create Org Specific Configuration settings 64
Verify Configuration 65
Enable BGO Priming 65
Migrating a Branch to the "-GM" Ext Pather Org 66
Removing the "-GM" Ext Pather Org when finished 67
Appendix I: Summary of Settings for External Pathing 69
Appendix J: Conversion Factors 72
Appendix K: Terminology 73
Appendix L: LPS and RMWS Performance Numbers 75

One-way pathing, NO Traffic 75
One-way pathing, WITH traffic (compared to without traffic) 76
One-way Traffic Pathing Time, Traffic Performance 77
One-way with Map Edits 79
Appendix M: Unrestricted Cost Limit 79
Appendix N: Debugging Requests 81
Common Requests from Descartes Route Planner 81
Deconstructing the request: 84
Debugging No Paths 86
Appendix O: Dividing up LPS servers 88
Appendix P: Cache Save Scheduling 90
Fields and Special Characters 90
Appendix Q: Cache Management During Upgrades 92

Document Conventions

This document uses the following conventions:

- Names of windows, frames, dialogs, menus, list boxes, and lists begin with uppercase and are bolded. (Tools menu, Save button)
- Key combinations that you press 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).
- Text that you type 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 and unexpected results. Disregarding information presented as a warning may result in damage to your software or data.

Overview

This guide provides step-by-step instructions to configuring Descartes® Route Planner™ to efficiently use the External Pather with Descartes AltaMap™. The External Pather is currently slower than LMB on a per-calculation basis, however, with appropriate tuning can be configured to handle a wide range of optimization problems.

WebService Components

This engine has three primary components:

1. **LNOS Pathing Service**
   - Used to receive requests from one or more Descartes Route Planner servers
   - Uses an in-memory cache to response to requests, cache is written to disk with each IIS recycle event or on a schedule defined by the ScheduledSaveCacheTime property.
   - Cache data on disk is compressed
   - **Core Files**
     - Web.config – determines configuration of the LNOS Pathing Service, which traffic patterns to compute, etc.
     - GeneratedFiles\MapEdits – local cache of the LNOS Map Editor map edits.
     - GeneratedFiles\Vehicle_all.dat – enumeration of vehicle profiles previously used
     - GeneratedFiles\p2pcache_default.lps – collection of all paths previously computed and paths not associated with a map edit.
     - GeneratedFiles\p2pcache_[partition]_[ScheduleGroupKey].lps – collection of all paths previously computed associated with a map edit for the specific org/schedulegroupkey

       The [partition] is composed of the following elements:

       - DCF database name
       - Descartes GLN™ Org ID
       - Descartes Route Planner Organization Name
       - SQL Server® name
       - SQL instance name
   - **Primary Logs:**
     - \PathingService\Logs\service.YYYYMMDD[.#].log – captures high level and debug information related to pathing calls; includes point-to-point calls, version, status calls, etc.
     - \PathingService\Logs\DMDebug.YYYYMMDD[.#].log – captures key and debug information related to certain calls only, in INFO mode (default)

it records performance-related information that is easily parseable into a CSV file format using a "|" delimiter.
    - The messages to be included is configurable in the web.config.
- **Redundancy/Max Instances**
    - At this time there can be a primary pathing service and an alternate/failover instance. In a failover scenario previously determined paths may need to be re-computed.
    - If the primary cannot be reached at the start of optimization, the optimizer (RMPI) will automatically try the Alternate instead. If neither is available, RMPI will use cached values and straight line distances for the remainder of the optimization calls.

    ➲ **Note—** The primary pathing service can be different on a ScheduleGroup basis.
- **RM Assembly**
    - Is a sub-component of the LNOS Pathing Service
    - Is used to handle communication with the Descartes AltaMap pathing engine.
    - All Descartes AltaMap pathing engine requests are executed remotely by the Route Matrix Web Service (RMWS) servers configured.
    - It is also responsible for splitting these large matrix calculations into sub-matrices to be computed by individual RMWS nodes.
    - Serializes (queues) the requests to the RMWS nodes. Internally there are three queues:
        - Matrix (large)
        - Matrix (small/priority)
        - Point to point / Path / Directions
    - Has its own log file
        - \PathingService\Logs\RMLog\Log*YYYYMMDD*RouteMatrix[.##].log
- **Dependencies**
    - As of 16.06 the LNOS Pathing Service is a standalone web-application and does not require a local copy of Descartes AltaMap.
    - Requires access (http/https) to at least one RMWS server
    - Requires access (http/https) to a Map Editor
    - Requires access (http/https) to GeoServer (defaults to http://altamap.gln.com, hosted by Descartes) for geocoding functions.

    ➲ **Note—** Descartes Route Planner's usage of LNOS Pathing Service does not require this feature.

2. **Route Matrix Web Service (RMWS)**
    - Used to process large matrix calculations or a sub-set of them.

- Used to process other Descartes AltaMap requests like point-to-point (p2p). See RM Assembly for more details of what is computed via RMWS vs RM.
- **Core Files**
  - Web.config – determines configuration of the RMWS.
- **Primary Logs**
  - \RouteMatrixWS\Logs\RouteMatrixWS[.##].log
- **Dependencies**
  - Requires a local copy of HERE®/TomTom® map data and the Descartes AltaMap LBS platform to be installed.

➲ **Note—** HERE® map data is owned by HERE North America, LLC.

➲ **Note—** TomTom® map data is owned by TomTom North America, Inc.

3. **LNOS Map Editor**

- Used to make changes to the map data used for pathing.
- Can link to multiple Descartes Route Planner environments through the use of a radio button on the login screen.
- Edits are partitioned by ScheduleGroupKey
- Uses Descartes Route Planner security to determine access to the system.
  - **Role**:
  - Read-only or Edit (ReadWrite) based on the following function group:
    - Fleetwise - MapEdit ReadOnly
    - Fleetwise - MapEdit ReadWrite
  - **ScheduleGroups**:
  - All schedules a user has access to, regardless of active group.

➲ **Note—** If a user belongs to two groups, one which has read-only access for all schedules and one that has full access to two schedules the Map Editor is not currently able to distinguish this and would give map edit access to all schedule groups.

- **Core Files**
  - **Web.config** – determines configuration of the LNOS Map Editor.
  - GeneratedFiles\MapEdits\ClientUrls.json – contains the list of Descartes Route Planner sites LNOS Map Editor can use for authentication.
  - GeneratedFiles\MapEdits\[partition].[ScheduleGroupKey].pme

    The [partition] is composed of the following elements:
    - DCF database name
    - Descartes GLN Organization ID
    - Descartes Route Planner Organization Name

    - SQL Server® name
    - SQL instance name
  - GeneratedFiles\Exports – map edits that have been "Published" for use, only these edits will be utilized by the PathingService (Map Editor 16.06 and later).
- **Primary Log**
  - \MapEditor\Logs\service.log[YYYYMMDD.log] – captures all information related to the execution of the website
- **Redundancy/Max Instances**
  - At this time there can only be one map editor instance "up" at a time; the application serves the map edits stored in the local file system. It is recommended to use a DNS or an active/passive load balancer configuration to handle the case of a failover.

  ➲ **Note—** The LNOS Pathing Service caches the results from the LNOS Map Editor so if unavailable pathing is not interrupted.
- **Dependencies**
  - As of 16.10 the Map Editor Service is a standalone web-application and does not require a local copy of Descartes AltaMap.
  - As of 16.10 the Map Editor Service will use Route Planner for geocoding functions.
  - Requires access (http/https) to at least 1 RMWS server
  - Requires access (http/https) to GeoServer (defaults to http://altamap.gln.com, hosted by Descartes) for geocoding functions if not using Route Planner.

4. **GeoServerWS**
   - Used to provide geocoding services through Descartes AltaMap.
   - **Core Files**
     - **Web.config** – determines configuration of the GeoServerWS.
   - **Primary Log**
     - \GeoServerWS\Logs\service[.##].log – captures all information related to the execution of the website
   - **Dependencies**
     - Requires a local copy of HERE®/TomTom® map data and the Descartes AltaMap LBS platform to be installed.

     ➲ **Note—** HERE® map data is owned by HERE North America, LLC.

     ➲ **Note—** TomTom® map data is owned by TomTom North America, Inc.
5. **TileServer**
   - Used to provide map tiling (Display) services through Descartes AltaMap.

- **Core Files**
  - **Web.config** – determines configuration of the TileServer.
- **Primary Log**
  - \TileServer\Logs\service[.##].log – captures all information related to the execution of the website
- **Dependencies**
  - Requires a local copy of HERE®/TomTom® map data and the Descartes AltaMap LBS platform to be installed.

  ➲ **Note—** HERE® map data is owned by HERE North America, LLC.

  ➲ **Note—** TomTom® map data is owned by TomTom North America, Inc.

Prerequisites for External Pather

In order for Descartes Route Planner to use Descartes AltaMap via External Pather, the first step is to deploy Descartes AltaMap and External LNOS Pathing Web Service.

The installation and basic configuration instructions for this can be found in the existing *LNOS Pathing Services Installation Guide* packaged with the installer.

Please see *Appendix Q: Cache Management During Upgrades* for steps to ensure the existing cache is not lost.

It is recommended that the RMWS is installed on a separate server than the LNOS Pathing Service, this is especially important if the requests are large to avoid out of memory situations.

Warning— Once installed, check the \PathingService\Logs\RMLog\Log and make sure there are no errors being reported in connecting to any of the RMWS nodes, if the map configuration does not match the LNOS Pathing Service will not be able to use the RMWS node.

Cache Invalidation and Upgrade Planning (Descartes AltaMap, Map Data, LNOS Pathing Service)

Whenever upgrading the pathing service it is important to properly prepare the environment to avoid degradation in performance following the upgrade. Cache files generated in one release are usually not compatible with the new release, this means the entire cache will be invalidated.

There are four primary causes for cache invalidation:

- Enable/Disable Traffic Patterns
  - The cache schema is different when traffic is enabled/disabled
  - Note: priming should be done prior to making this switch offline.
- Vendor change (e.g. TomTom® vs. HERE®)
- RM assembly or RMWS version change
- LNOS Pathing Service Schema change (associated with an LNOS Pathing Service upgrade)

There are three main steps to transition the cache:

1. During installation the user may choose to "keep previous cache".
   - This option is enabled by default during installs.
   - It allows a "Hybrid" cache whereby old values are used in the new LNOS Pathing Service until such time as the cache can be re-primed.
2. Using an offline BGO the cache needs to be re-built.
   - This can be done using a script from the Descartes engineering/services team
   - Or; Using an offline copy of the database and other BGO's (e.g. preprod environment)

   Note— It is very important that the Vehicle_All.data and any map edits be copied from production and that the versions of map data, RMWS and LNOS Pathing Service are identical.
3. The cache files need to be swapped out.
   - Stop PathingService application pool. Wait until process has terminated in task manager and service.log.
   - Make a backup copy of the existing .lps in the PathingService\GeneratedFiles folder.
   - Copy/replace the existing files with the newly primed files.
   - Start the PathingService.
   - Send at least one request using the test page; this will cause it to "load" the new cache data.

- If the process works it should write out the number of entries loaded. If the new cache is invalid (vendor, config, version) it will instead rename the cache file to _old and will load zero entries from the cache file.

Example failure to load new cache file.

```
2017-01-30 19:06:14,613 INFO  Loaded 10 entries from
H:\inetpub\wwwroot\PathingService\GeneratedFiles\Vehicle_All.dat,
time = 3 (ms)
2017-01-30 19:06:15,691 INFO  Copying exisitng cache file :
H:\inetpub\wwwroot\PathingService\GeneratedFiles\P2pCache_default
.lps to backup cache file :
H:\inetpub\wwwroot\PathingService\GeneratedFiles\P2pCache_default
_old.lps
2017-01-30 19:06:15,691 INFO  Delete cache file named:
H:\inetpub\wwwroot\PathingService\GeneratedFiles\P2pCache_default
.lps
2017-01-30 19:06:15,894 INFO  Diagnostics,4656,LoadCache-
begin,113,161,(MB)
2017-01-30 19:06:15,894 INFO  Loading compressed P2PCost results
used since 1/1/0001
2017-01-30 19:06:15,894 INFO  Trimmed 0 entries
2017-01-30 19:06:15,894 INFO  Loaded 0 entries from
H:\inetpub\wwwroot\PathingService\GeneratedFiles\P2pCache_default
.lps, time = 6 (ms)
```

External Pathing Descartes Route Planner Configuration

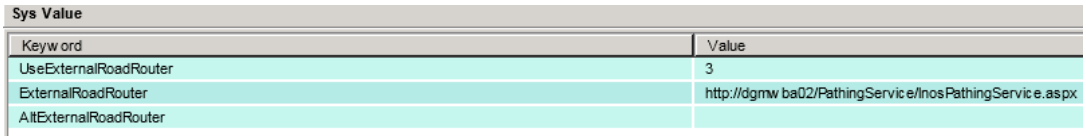
Enabling the UI/Interfaces to Use External Pather (Descartes AltaMap)

By default, Descartes Route Planner does not use external pathing. The steps below will enable the user interface/interface calls to use External Pathing. The procedure to enable map edits in Descartes Route Planner is described below.

- Select **App Setup > System Values** to open the **Sys Value** page.
- Enter **External** into the Find search text box and click **Find** to retrieve the relevant records.
- Change their values as follows:
- Set **UseExternalRoadRouter** to "3"

  ➲ **Note—** Changing this back to "0" will turn off the external router.

- Set **ExternalRoadRouter** to "http://<server>/**PathingService**/**LNOSPathingService**.aspx" where <server> is the name of the server where LNOS Pathing Service is installed/configured.



Sys Value

Keyword	Value
UseExternalRoadRouter	3
ExternalRoadRouter	http://dgmwba02/PathingService/lnosPathingService.aspx
AltExternalRoadRouter	

➲ **Note—** The ExternalRoadRouter setting determines the default External Pather to use if no overrides are provided. This Sys Value will be explained in more detail later.

- Optionally, configure **AltExternalRoadRouter** as well. This setting will be used in the case that the primary ExternalRoadRouter url is unavailable. If neither is available, it will use straight line for the duration of the session. This should be configured in all production environments.

➲ **Note—** These setting are cached and will not take effect until the next restart of the COM+ / services.

Enabling BGO to Use External Pather (Descartes AltaMap)

External Pathing needs to be separately enabled for the Background Optimizer (BGO). A BGO cannot be used for both LMB and External Pather.

To enable external pathing for the BGO, the following RMPI entries need to be added to the local.properties in the Routing section:

```
Routing.UseExternalRoadRouter=3
Routing.ExtPathingServer
=http://<server>/PathingService/LNOSPathingService.aspx
Routing.ExtPathingServerAlt
=http://<server2>/PathingService/LNOSPathingService.aspx
Routing.ExtPathingCacheFile = cache.le1
Routing.AsyncPollingSleepSeconds=1
Routing.MinStopsForAsync=100
```

Warning— If **ExtPathingServerAlt** is not defined and the ExtPathingServer is unavailable, the BGO will use straight line until the next restart of the scheduler. It is therefore critical that both the "primary" and "alt" are configured in the local.properties.

Note— If ExtPathingCacheFile is omitted, RMPI will base the path/file on the OriginalDistanceCacheFile.

Note— For a complete listing of RMPI settings related to External Pather see *Appendix A: RMPI configuration settings for External Pather.*

Additionally, the following BGO properties should be configured:

```
com.descartes.escheduler.build_cache=2
com.descartes.escheduler.build_cache_territory=0
com.descartes.escheduler.build_cache_timeout=30
```

For a complete list and explanation of these setting see *Appendix B: BGO Configuration Settings for External Pather.*

Purging existing cache

Once the configuration is completed, the existing cache files should be removed to ensure they do not interfere with future pathing requests.

- Clear the RMPI Cache:
  - Stop the DCF$/LNOS$ services and COM+ Application
    - This can be done using RestartCOM

- Delete cache.ldr, cache.lde, cache.ldc, and cache.ld1 in the C:\LNOS\fleetwise\cache (sub)folder(s).
  - `Cd %DCF_ROOT%\lnos\fleetwise\cache`
  - `Del *.ld* /s/q`
  - `Del *.le* /s/q`
- Start the DCF$/LNOS$ services
  - This can be done using RestartCOM by choosing the Restart DCF System option.

Handling Commercial Pathing Restrictions

One of the primary purposes of using external pathing is for one or more commercial pathing restrictions that are not available through LMB pathing. The External Pather supports configuration settings on both the Schedule and Resource levels.

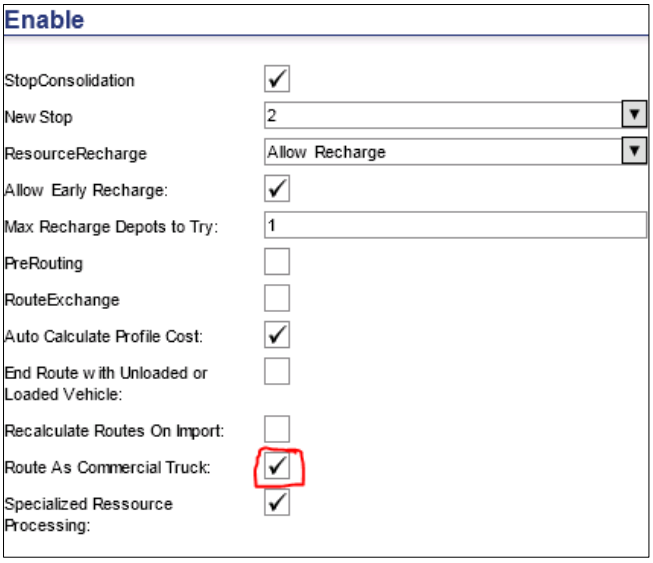
In order to use commercial pathing, a premium data license is required. If configured and not licensed these settings will be ignored during optimization and pathing.

Commercial Pathing Configuration (UI)

Commercial routing is now controlled at the Schedule and Resource levels. It does NOT need to be enabled at the schedule level first. **These features are only available if the customer has a premium Data license.** If this license is not enabled, the features will be ignored.

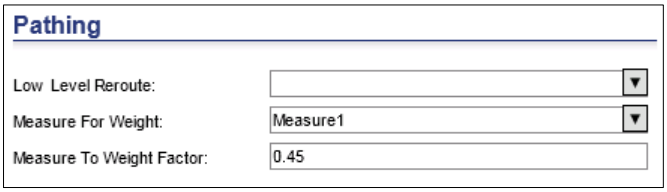
There are two optional settings at the schedule level.

Optionally, in the **Enable** section, click the checkbox for **Route as Straight Truck**. All passenger vehicles will be treated as Straight Trucks as long as the ScheduleGroup maximum is at least straight truck.



In the **Pathing** Section:

- Choose the measure from the **Measure For Weight** dropdown to use for onboard weight calculations.
- Provide a conversion ratio in **Measure For Weight Factor**. The conversion needs to result in a weight in **Kilograms (kg)**.

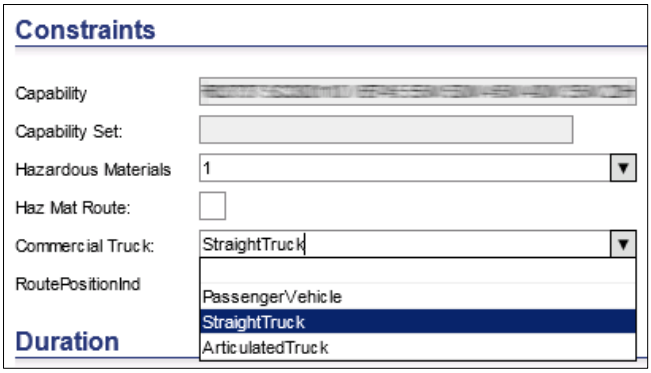


➲ **Note—** This setting should be avoided as it can slow down optimization significantly. It is impossible to quantify how much it can slow down optimization and is highly variable.

➲ **Note—** If the weight measure is pounds (lbs), it is recommended to use the same conversion factor as Descartes Route Planner uses, see *Appendix J: Conversion Factors* for more details.

To control the per truck behavior, edit the **Resource** or **Route Template** and configure as follows:

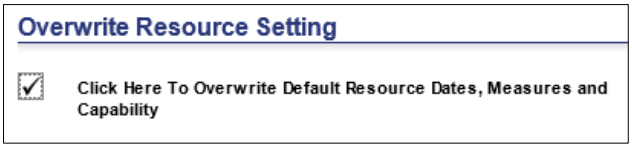
In the **Constraints** section, click the dropdown for **Commercial Truck** and choose the appropriate vehicle type.



➲ **Note—** If provided at the truck/trailer level it will override the resource settings with priority given to the truck.

⚠ **Warning—** The schedule group configuration limits which options will be used by the system. E.g. if minimum is "Straight truck" and the resource is PassengerVehicle it will be routed as a straight truck. Likewise if a resource is configured as a straight truck and the group is configured as an Articulated Truck it will always route as a straight truck.

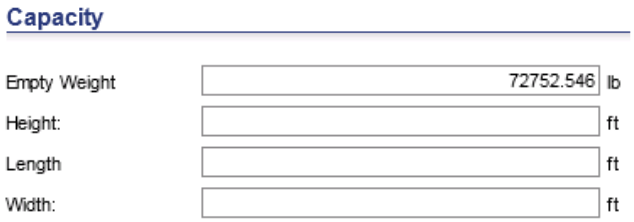
To enable vehicle dimensions, first enable **Overwrite Resource Settings**.



In the **Capacity** section:

- Specify **Empty Weight**
  - If measurement system is **US** it is in pounds (lbs)
  - If measurement system is **Metric** it is in kilograms (kg)

  ➲ **Note—** If the Measure for Weight schedule property is not used or is 0, a typical loaded weight should be specified.
- Specify Height, Length, Width
  - If measurement system is **US** it is in **feet**
  - If measurement system is **Metric** it is in **meters** (100cm).



➲ **Note—** These settings are used only if Resource Commercial Truck is Straight Truck or Articulated truck or if Schedule Route As Straight Truck is enabled.

Integration fields are:

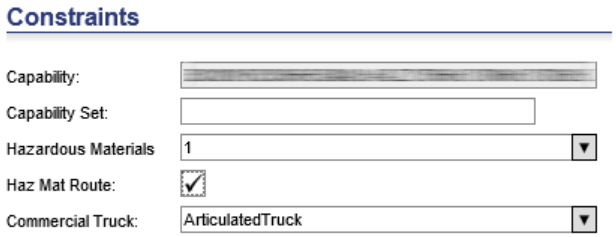
- DocResource/@CommercialTruck
- DocResource/@Weight
- DocResource/@Height
- DocResource/@Length

➲ **Note—** These dimensions are always metric when sent by an external interface (DocResource)

Hazmat Restrictions (UI)

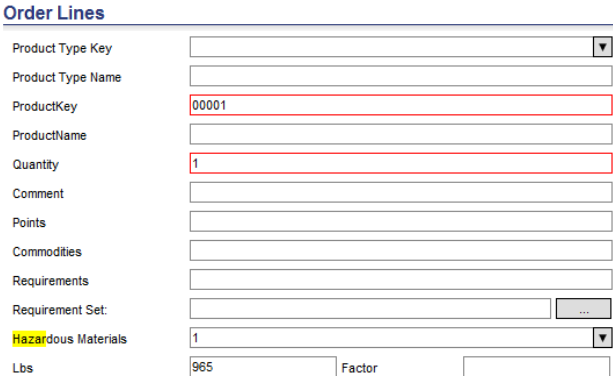
External Pather can be configured to use hazmat road restrictions. For this feature to work Hazmat must be configured at the resource and will be respected when passed with an order.

To control the behavior per route, pass a value of "1" in the **/DocResource/@HazMatType** or through the UI edit the **Resource** or **Route Template** and configure as follows:



In the **Constraints** section, Select **1** from the **Hazardous Materials** dropdown.

To control the behavior **per order**, pass a value of **1** in the **/DocBOL/BOLLine/UDF/@HazMatType** or through the UI as shown below:



In the **Order Lines** section, select **1** from the **Hazardous Materials** dropdown.

Alternatively, enable **Haz Mat Route**:

- The per order hazmat configuration is ignored
- *Hazmat is used for all pathing calls* regardless of the order details.

➲ **Note—** If using HazMatType then ScheduleGroup "min" should be hazmat = false, the "max" should be hazmat = true. If using HazMatRoute the min and max should both be hazmat = true unless only some routes have this option enabled.

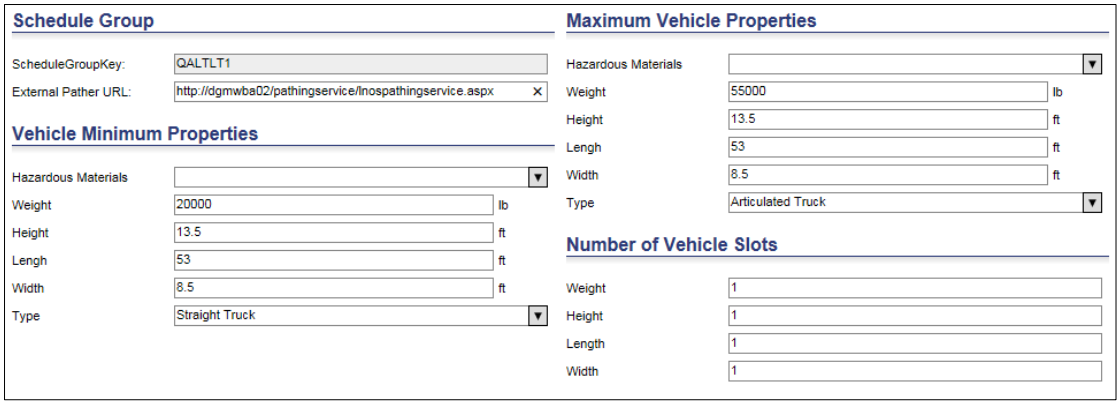
ⓘ **Tip—** This setting can be used to avoid integration changes for existing customers.

Schedule Group Configuration

In addition to configuring the resources with the appropriate vehicle sizes and types, the largest and smallest vehicles in the (local) fleet must be defined. Configuring this appropriately can significantly improve the performance of External Pathing. For more details on the performance aspect of this feature see *External Pathing Advanced Configuration - Performance*.

The minimum and maximum sizes of the vehicles are defined at the Schedule Group level (introduced in 15.2.1).

The Schedule Group list can be accessed via **Data > Schedules > Schedule Group**.



Localizing the Map Data

The Schedule Groups may contain one or more schedules and usually represent a branch or region for optimization and all their associated schedules.

For example:

- Schedule Group **PA**
  - PA_Planning
  - PA_Review
  - PA_Dispatch
- Schedule Group **NY**
  - NY_Planning
  - NY_Review
  - NY_Dispatch

A schedule group could contain schedules for more than one branch as well.

For example:

- Schedule Group **NorthEast**
  - PA_Planning,PA_Review,PA_Dispatch
  - NY_Planning,NY_Review,NY_Dispatch
- Schedule Group **SouthEast**
  - FL_Planning,FL_Review,FL_Dispatch
  - GA_Planning,GA_Review,GA_Dispatch

This separation of branches/regions into schedule groups provides several features to the administrator:

- **Cache is localized to the ScheduleGroup**
  - Map edits only affect branches in this group.
  - When map edits change, the cache is only purged for the group.
- **Map edits are secured by ScheduleGroup**

- LNOS Map Editor users can only make changes to edits in their group and cannot see other group's edits.

- **Different LNOS Pathing Services can be used**
  - Different schedule groups can use different LNOS Pathing Service servers.
  - This provides a physical separation of the cache (separate resources).

See *Appendix F: External Pather in Larger Deployments Best Practices* and *Appendix O: Dividing up LPS servers* for more details on using multiple LNOS Pathing Service servers.

Minimum/Maximum Vehicle Profiles

The **Vehicle Minimum properties** determines the smallest/least restrictive type of truck that will be used for pathing while the **Vehicle Maximum properties** determines the largest/most restrictive type of truck that will be used for pathing. Vehicles outside of these dimensions will be constrained to these limits (including truck type as of Descartes Route Planner 16.06 patch C / 16.10).

It is recommended that these limits be well defined to match the fleet of vehicles for either the branch or the company as a whole. The limits should also take into account any commercial restrictions. The closer these limits are to each other, the less pathing the system may need to do.

The **Number of Vehicle Slots** value limits the number of vehicle combinations that the system has to consider for pathing. This setting also improves the speed of pathing because the system does not need to account for every unit of weight change of the vehicle (e.g. 12,000, 12,100, 12,200, 12,300 could all be treated as 13,000 for pathing).

Each combination of Weight, Height, Length, Width requires additional pathing calls. With a reasonable Minimum/Maximum configuration, this consideration usually results in additional sizes only requiring about one to ten percent of the computing resources of the Minimum/Maximum vehicle calculations. However, this may vary between implementations or even Schedule Groups.

> ⓘ **Tip**— Although the application can distinguish Straight Truck from Articulated Truck in practice, there are almost no roads that distinguish eligible roads by this criteria exclusively. Usually the dimensions of the truck cause the restriction. However, there is a performance cost for having different vehicle types, so better performance can be gained by specifying all trucks as Straight Truck instead of Articulated Truck as long as vehicle dimensions are used.

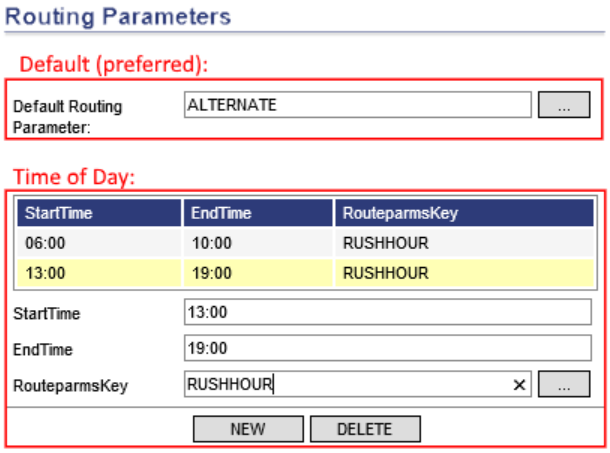
External Pathing Violation Radius

See ***Ignoring Commercial Restrictions near the Origin/Destination*** for an explanation of this setting.

Configuring Routing Parameters

Due to the simplified configuration of road speeds and the performance impact on pathing it is recommended that at most two routing parameter speed profiles exist: normal and rush hour. **If traffic patterns are used, then it should not be necessary to have more than one routing parameter per schedule.**

If using a routing parameter other than "DEFAULTS" it is important that this be configured as the "Default routing parameter" on the resource and NOT the "time of day" section. The "DEFAULTS" routing parameter will still be used for pathing, even if the timeframe is 00:00 – 23:59. It is important that ALL resources be configured with this routing parameter, if even one is missed, the system will automatically path using "DEFAULTS" as well during the Min/Max calculation which can significantly affect performance.



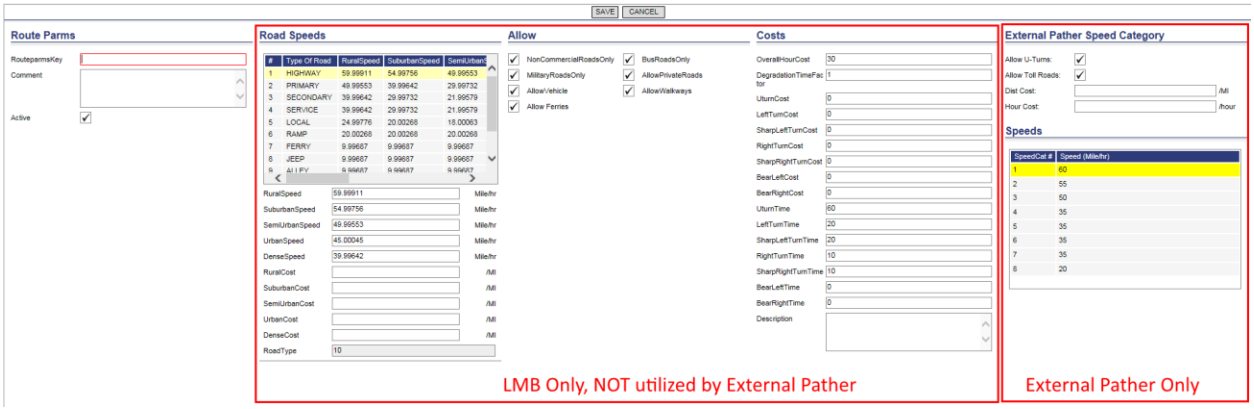
Tip— Additional routing parameters can be setup with different costs. However, the speeds should either match one configuration or the other for the same Schedule Group to avoid unnecessary pathing. Each change to the pathing rules requires additional pathing calls and can dramatically reduce optimization speeds as the cache is rebuilt.

Warning— It is recommended that edits to routing parameters be done during off-hours in production to

prevent performance problems. This is especially important for "shared" parameters like DEFAULTS. Changes to routing parameters require rebuilding the cache for all schedules utilizing the routing parameter.

Settings Used by External Pather

When external pathing is enabled, only the **External Pather Speed Category** subset of the routing parameter settings are used. All other sections are ignored.



Configuring Speed Categories

There are eight speed categories. These categories do not necessarily equate to the functional road classification as some limited access highways could have reduced speed limits in urban locations which could lower the speed class.

➲ **Note—** Not all speed categories are used in every country (e.g. Speed Category 1 in the US)

The road speeds are classified / categorized as follows:

Speed Class	Typical KM/H	Typical MP/H
1	> 130	> 80
2	101-130	65-80
3	91-100	55-64
4	71-90	41-54
5	51-70	31-40

Speeds

SpeedCat #	Speed (Mile/hr)
1	75
2	65
3	55
4	45
5	35
6	25
7	10
8	5

6	31-50	21-30
7	11-30	6-20
8	< 11	< 6

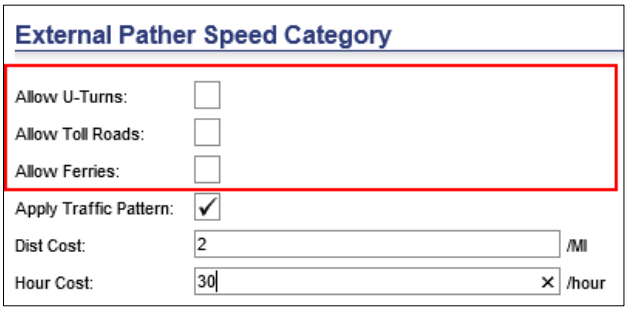
When routing from a depot that uses a combination of rural and urban locations, it is recommended that Speed Category 1 and 2 are either not modified or are modified less than the slower categories unless the routes are constrained to only urban locations via territories.

Warning— It is very important that the speeds are decreasing as the category number increases, otherwise strange behavior and performance problems could occur. For example, if Speed Category 3 is reduced to 40 then category 4 should be 39 or less.

Allowable Road Types

At this time, external pathing supports the ability to turn off the following road types:

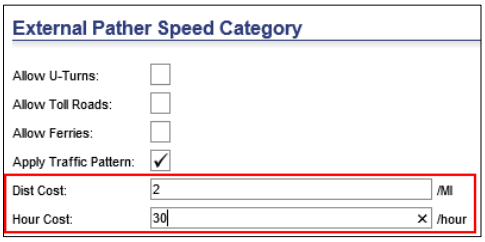
- Ferries (Requires Descartes AltaMap 16.04)
- U-Turns (Also enables "side of street" pathing).
- Toll Roads



Note— When these road types are allowed, it is not possible to apply extra costs to them.

Optimization Costs

The costs for driving (Time and Distance) are used to determine optimization decisions exclusively. Unlike LMB, this does not have an impact on pathing decisions. The costs are applied **after** pathing when evaluating which orders should be assigned to which routes. In most LMB solutions the cost/mile is the same for all road types and densities so this change will usually not impact the solution.

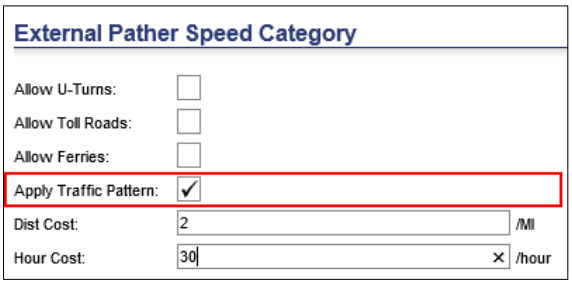


Warning— Distance and hour cost should be greater than zero. If set to ”0”, unexpected optimization results may occur.

Traffic Patterns

In order to use traffic patterns for routing, the routing parameter must have the **Apply Traffic Patterns** property enabled for each routing parameter. *This requires a premium license to use.*

Additionally, the LNOS Pathing Service web service must be configured to calculate traffic patterns. If the **Apply Traffic Patterns** traffic patterns property is not enabled or a license is not obtained, traffic patterns are still calculated, but they are not applied. This can significantly affect performance.

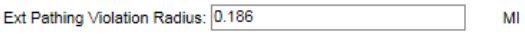


Ignoring Commercial Restrictions near the Origin/Destination

In some cases, the origin or destination location (e.g. customer) is in a commercially restricted area. In LMB pathing this would cause RMPI to use straight line pathing from origin to destination. With the External Pather (Descartes AltaMap), there is now a setting called **Ext Pathing Violation**

Radius. This radius determines how far the system should look to find a commercially viable path. Once this path is found, a non-commercially viable path will be taken to the origin/destination itself.

This setting can be configured at the Schedule Group level, **Data > Schedule > Schedule Group**.



- A typical range for this setting is 2.9 miles (~5 km) to 10 miles (~15 km). See *Appendix M: Unrestricted Cost Limit* for a visualization of this process.

  Warning – Excessively large values can significantly impact performance.

Additional RMPI Configuration Changes

As of version 16.06, the installer should correctly configure the RMPI.ini by default. If using separate INI files for org option, it may however be necessary to make these changes manually.

In addition to enabling External Pathing, additional changes are required to the caching configuration. Ensure references to Routing.DistanceCacheFile are removed or commented out. This file name and path are determined automatically by the OriginalDistanceCacheFile property.

Remove all references to Routing.TelogisPathingCaching.

The following lines may need to be added:

```
[Routing]
ClearCacheCopy=1
ExtPathingCacheFile = [drive]:\lnos\fleetwise\cache\cache.le1
AsyncPollingSleepSeconds=1
MinStopsForAsync=200
```

Note— For a complete listing of RMPI settings related to External Pather see *Appendix A: RMPI configuration settings for External Pather*.

Traffic Patterns with Partial Day of Week Configuration

It is possible to configure traffic patterns on the LNOS Pathing Service with only a subset of the days of week. This configuration is typically done to improve performance. If only partial days of week are computed (e.g. Sunday, Tuesday, Wednesday), the computed values can be used for other days of the week to provide full week coverage.

To configure LNOS Pathing Service with a subset of days, the rmpi.ini and local.properties files need to be manually edited on each server.

In **rmpi.ini**, **[Routing]** section, add the following line:

```
TrafficDOWMapping=R(T),M(W),F(W),S(U)
```

In **local.properties** (anywhere) add the following line:

```
Routing.TrafficDOWMapping=R(T),M(W),F(W),S(U)
```

The format of this setting is [DOW to Modify]([DOW to Use]) and is a comma separated list.

Breaking down the example above:

- Thursday will use Tuesday R(T)
- Monday will use Wednesday M(W)
- Friday will use Wednesday F(W)
- Saturday will use Sunday S(W)

As an example for a matrix size of 630 stops (locations) this can equate to a 20 percent reduction in calculation time.

Traffic Pattern Interpolation

When traffic patterns are enabled not all hours are usually enabled. To account for this, RMPI will determine the traffic effect based on the proximity of the departure time to the available time windows.

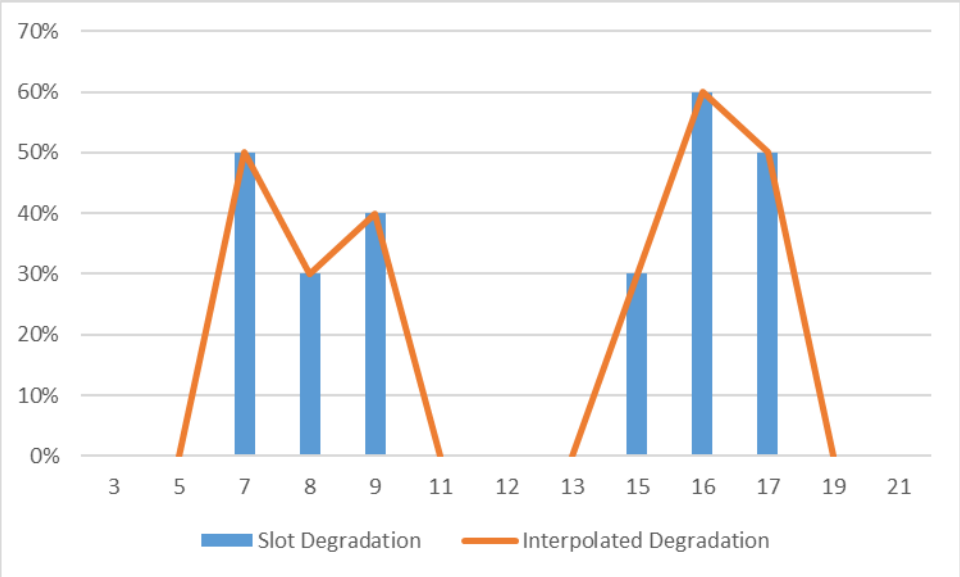
With an interpolation interval of four hours, if no timeslot is found within four hours of the current timeslot the speed factor will be computed using 50 percent of the interval on each side of the time slot using a linear progression towards 0.

For example, if the time slots are 7:00 am, 8:00 am, 9:00 am, 3:00 pm, 4:00 pm and 5:00 pm

The 7:00 am timeslot does not have a timeslot between 3:00 am and 7:00 am so RMPI will interpolate between 5:00 am and 7:00 am. At 5:30am it will be 25 percent of degradation, at 6:00 am it will be 50 percent, at 6:30 it will be 75 percent.

The same behavior would occur again between 9:00 am to 11:00 am, 1:00 pm to 3:00 pm and 5:00 pm to 7:00 pm.

➲ **Note—** In 17.05 the default interpolation interval was changed to be four hours (recommended value). Prior to 17.05 the default value was 16 minutes; this made it so that traffic patterns were largely ignored.


70%
60%
50%
40%
30%
20%
10%
0%
3
5
7
8
9
11
12
13
15
16
17
19
21
Slot Degradation
Interpolated Degradation

External Pathing LNOS Pathing Service Web Service Configuration

Basic Configuration

During the installation process, the user is prompted with certain key information. These configuration changes are saved in the web.config. There are additional configurations that are usually required after installation to get the LNOS Pathing Service web service to work as required.

The configuration file, web.config, can be found in the installation directory (usually C:\inetpub\wwwroot\PathingService)

> **Warning**— The installer will **replace** the web.config with each upgrade, the previous configuration will be available at web.old.config, this means any environment specific configurations like the RMWS service url's will need to be manually re-configured. *For customers upgrading from a version prior to 15.2 the file must be manually backed up prior to upgrading.*

There are three groups of configuration settings that are commonly configured, for a complete listing of all settings please referring to the *Descartes LNOS Pathing Service Installation and Configuration Guide.*

Maintenance/Performance

- **ForceGCInMinutes**
  - This setting forces the LNOS Pathing Service to perform garbage collection periodically.
  - Recommended (default) value for this is 30.
  - An additional property (GCLevel) can be configured which determines which level of garbage collection to use.
- **MatrixPool.SmallQueueThreshold**
  - Maximum number of *cells* to compute that can qualify using a small queue. Otherwise, standard queue will be used.
  - This setting is intended to allow smaller requests (e.g. BGO sees one new customer) that compute quickly to avoid waiting in line behind large requests (e.g. BGO sees a new branch or a branch whose cache has been purged).
- **MatrixPool.UISmallQueueThreshold**
  - Applies to "UI" requests only (e.g. suggest, drag/drop, etc). The maximum number of *cells* to compute that can qualify using a small queue. Otherwise, standard queue will be used. This threshold should be higher than the MatrixPool.SmallQueueThreshold.

- This setting is intended to allow small user interface requests (e.g. suggest, resequencing, calculate distance, etc) to bypass the main queue used by BGO. Ensuring faster UI response time.

- **CacheRetentionPeriod**
  - This value should be the smallest number of days required for the operations. Each time the LNOS Pathing Service restarts, it will remove the records that have been used less recently than this number of days.
  - For businesses with repeating customers, this number should be at least the typical number of days between repeat business:

Repeat/Static Customers	Dynamic Customers	Recommended Retention	Example Business
<5% per 7 days	95%+	2-3 days	Home Delivery
70%+ per 7 days	<5%	11 days	DC to Store
Multi-week rotation	< 10%	7*n+2 days	Door to door sales n week rotation+ ad-hoc customers

➲ **Note—** A few days extra are given to handle holidays and analysis of recently executed routes.

- **ScheduledSaveCacheTime**
  - In addition to application pool startup/shutdown time, the cache can be saved on a periodic schedule.
  - The default setting will schedule the task to run at 03:01AM, Monday through Saturday.
  - `<add key="ScheduledSaveCacheTime" value="0 01 03 ? * MON-SAT" />`
  - A tutorial on how to customize schedules can be found in the appendix or by performing an internet search for "cron trigger".
  - See *Appendix P: Cache Save Scheduling*
- **UsePreviousVersionCache**
  - Enabled by default during install, allows the LPS to continue running using the previous versions cache until the cache can be reprimed.
  - See *Appendix Q: Cache Management During Upgrades* for more details.

Communication

- **MatrixPool.QueueSize**
  - Sets the size of the queue for matrix requests that are handled remotely through RMWS
  - This value must be large enough to handle all requests for Descartes Route Planner. There should be one per LNOS or BGO service and COM+ package.

- As a general rule this can be calculated using the following formula: **[**(# UI) * 6 + (# BIF) * 6 + (# BGO) * 4 + (# Other) * 6**]** * #RMWS

  Where:

  # UI = Number of UI servers

  # BIF = Number of BIF servers

  # BGO = Number of BGO servers

  # Other = Number of other LNOS servers (e.g. DFL, BE, batch processor, etc).

  # RMWS = Number of RMWS servers, there is one queue entry per RMWS node.

  ➲ **Note—** This setting must be integer (no decimal places)

- **MatrixPool.ServiceUrl.#**
  - This setting determines the RMWS nodes that LNOS Pathing Service should use for RM calculations. For each node in the pool, increment the number and the end of the key.
  - For example:
    - MatrixPool.ServiceUrl.1: http://server1/RouteMatrixWS/service1.asmx
    - MatrixPool.ServiceUrl.2: http://server2/RouteMatrixWS/service1.asmx
    - MatrixPool.ServiceUrl.3: http://server3/RouteMatrixWS/service1.asmx
  - Optionally, servers can be scheduled. In that case the format is different.
  - schedule=[key];url=http://…
  - For example:
  - <add key="MatrixPool.ServiceUrl.2" value="schedule=Cloud;url=http://awsip/routematrixws/service1.asmx" />
- **P2P.MaxConcurrent**
  - Sets maximum number of P2P concurrent requests to RMWS, total Simultaneous across all RMWS servers.
- **P2P.BypassQueueThreshold**
  - Sets maximum of number of p2p paths in a request to go direct or via small queue, inclusive.

Pathing Methods

- **MatrixPool.UseOneWayRouting**
  - Default value is true
  - New pathing algorithm, significantly faster, better paths

- **QueryForTraffic**
  - Default value is false
  - Enables traffic patterns, requires premium license.

  ➲ **Note—** Should only be enabled if licensed and enabled as the data will not be used unless a valid license is installed in Descartes Route Planner and traffic patterns significantly increases time to path.

- **StoredTrafficSlots**
  - Default value is 40 patterns over six days with one hour increments during typical rush hour periods
  - Maximum time resolution is 15 minutes
  - Determines which timeslots will be computed for each path (cell)
  - Must always start with at least one Sunday slot between 00:00 and 04:00
  - Format is **[Dow] Time** and is a comma separated list of slots
  - Where **[DoW]** is one or more of U, M, T, W, R, F, S
  - For example:
    - U 00:00, UTW 06:00, UTW 07:00, UTW 08:00
    - Will compute Sunday at midnight
    - And 6am, 7am ,8am on Sunday, Tuesday and Wednesday

  ➲ **Note**— Using a high number of slots will significantly slow down optimization and pathing. It is recommended to not use more than approximately 40 traffic slots *per week*. Users must choose between time of day resolution and day of week resolution.

External Pathing Advanced Configuration - Performance

In order to maximize the performance of External Pather, additional configuration details are required. These settings constrain the pathing system to improve performance. These settings also require a thorough understanding of the customer's pathing requirements.

Optimization Terminology

Before discussing External Pather performance it is important to understand the new terminology associated with the External Pather. See *Appendix K: Terminology* for details.

CONFIDENTIAL AND PROPRIETARY TO THE DESCARTES SYSTEMS GROUP INC. AND ITS AFFILIATES

Understanding the External Pather Computation Time

The first step in understanding the overall user experience with External Pather is considering the time required to compute paths. Compared to LMB, each path computation takes longer. The following graph provides a simple benchmark of expected computation performance by problem size *assuming we calculate all the cells* using a single RMWS server.

Performance without Traffic

For a more details, see *Appendix L: LPS and RMWS Performance Numbers*

As the size of the request becomes larger, the time to compute can increment significantly. As such, users will want to ensure that a path is only computed once and this path is cached for later re-use.

It is also important to note that the computation time per cell decreases at the matrix size increases. For this reason, it is better to have fewer large matrices than many small matrices.

Understanding the External Pather Caching Model

The second step to understanding the External Pather performance is the caching model. As previously described, the time to compute paths in large problems becomes significant so it is advantageous to perform these calculations as few times as possible. To solve this problem, a two-level caching model is used.

Level 1 Cache – RMPI Distance Cache File (_[pid].ldc, _[pid].ld1) & External Pather Cache File (_[pid].le1)

As with LMB, each RMPI process has its own cache file. Once a calculation is completed, it is stored in the local cache for later re-use. Storing values in this cache prevents the optimizer connecting to the WebService and/or Level 2 Cache. If created from an Original Distance Cache file, the Distance Cache file is the same filename with the process ID suffixed to prevent collisions with other processes.

Level 1.5 Cache – RMPI Original Distance Cache File (xxx.ldc, xxx.ld1) & External Pather Cache File (xxx.le1)

As with LMB, each RMPI process can write its updates to a master cache file. This cache file can be shared between multiple services/processes. The update of the master cache file occurs on completion of an optimization. If the ODC has not changed since the start of the optimization, it is replaced; if it has changed it is discarded. A similar process exists for all the other RMPI cache files (le1, ld1, ldc, etc) depending on if the ODC has changed or the pathing configuration has changed the updates to these files may also be skipped.

Level 2 Cache – LNOS Pathing Service (Web Service)

In addition to computing paths, the LNOS Pathing Service stores the already calculated values in an in-memory cache of all origin/destination pairs and their D/T values. This cache can be quite large as it is shared by multiple services/servers. It is written to disk (\PathingService\GeneratedFiles\p2pCache_[orgname].dat) when the Web Service/Application Pool is shut down and at specific configured intervals.

When a pathing request is received, it first checks the cache to see if some/all cells are known and only the cells missing are requested for further computation by the RM/RMWS. The entire set of points is returned to the calling process regardless of source (cache/computation).

Importance of BGO in Caching with the External Pather

Because of the time required to perform the calculations and the importance of caching, the BGO becomes a crucial part of implementations. The BGO provides a secondary benefit to Optimization in external pathing, it also is responsible for "priming" the Level 2 cache. When the BGO begins work on a dataslice, the first step it performs is to build the cache for the entire dataslice (unconstrained/constrained).

As the normal optimization progresses, the intermediate paths, if required, will also be pathed and cached. By pre-building this cache during the BGO optimization cycle by the time the routes are ready to be reviewed, the Level 2 cache will already exist. Initial requests from the user interface may be slightly slower than existing LMB optimization however as more user interface requests occur, the user interface servers will build the Level 1 cache using the Level 2 cache. Only when a new order is added will additional pathing calls be required.

It is very important for the BGO to have access to the data as soon as possible even if it is not building routes so that the BGO can start priming the cache. In the case of large dataslices, the initial computation could take minutes or even tens of minutes to complete; the earlier it starts pathing, the less time it will take when optimization is truly required.

Tip— If BGO optimization is not being used, the BGO can still be used to "prime" the cache exclusively by turning off all the phases (inter, intra, assign, batch_assign, etc.). The BGO can also be used to prime the cache based on historical data by creating dataslices that look at the archive schedule and place them into an areakey and property group that disables the normal optimization phases.

Speed and Vehicle Profile(s)

Resources in Descartes Route Planner can have one or more routing parameters and each resource can have different dimensions. For pathing, these are grouped in the LNOS Pathing Service into "profiles". The profile consists of the routing parameter speeds as well as the resource pathing requirements (vehicle type, dimensions, hazmat, etc).

Maximum and Minimum Vehicle / Number of Slots Explained

When RMPI begins an optimization involving External Pather, the first step is to perform a series of **Matrix** calculations to determine whether an origin/destination pair or cell is affected by commercial restrictions.

For each profile, a pair of matrix calls are sent by RMPI/External Pather to the LNOS Pathing Service.

- The first calculation is called the **Unconstrained** profile matrix
  - Represents the *least* restrictive path available.
  - Uses the **Minimum Vehicle Properties** configured.
- The second calculation is called the **Constrained** profile matrix
  - Represents the *most* restrictive path available.
  - Uses the **Maximum Vehicle Properties** configured.

➲ **Note—** If not configured, the internal defaults are extremely large so it is highly recommended that this always be configured.

In addition to caching the paths for these pre-calculated matrices in the Level 1 and 2 caches, they are also used to determine if future matrix or point-to-point calculations will be required for future calls.

It is crucial to keep the number of vehicle profiles to a minimum as each variation requires additional computation time and, depending on the size of the problem, this computation time can be significant.

If a path is required for a resource that does not match either the unconstrained or constrained profiles, the External Pather will check if the computed paths differ (time and/or distance). If they do, then a point-to-point or matrix call will be issued using the new profile (based on the **Number of Vehicle Slots** configured).

See *Schedule Group Configuration* for details for instructions on how to configure these settings.

➲ **Note—** Vehicle Dimensions are truncated at 2 decimals places (cm), they are not rounded so 1.3852 would be used as 1.38 not 1.39. Vehicle weights are truncated to 0 decimals (kg), they are not rounded so 9105.9 would be 9105 not 9106.

Number of Vehicle Slots Explained

The profiles between the Minimum and Maximum are grouped into slots according to the number of vehicle slots defined at the schedule group.

For example:

If the Minimum Weight is 9,000 kilograms and the Maximum Weight is 25,000 kilograms and the number of slots is set to two, then the pathing bands will be:

- 0 to 9,000 kg = 9,000 kg (Minimum)
- 9,000 to 17,000 kg = 17,000kg (Slot 1)
- 17,000 kg to 25,000kg = 25,000kg (Slot 2/Maximum)

With the configuration above, a truck with a weight of 13,000 kg will be pathed as if it were a 17,000 kg truck, i.e. the weight is rounded up.

Any resources with a weight greater than 25,000 kg would be treated as if it is 25,000kg as well. The same principle applies to the length, height and width dimensions.

Great Circle Mile (GCM) Filter

Starting with Descartes Route Planner 15.1 the system will pre-filter the stops being sent for calculation to only those within proximity of another customer or the depot. In large problem sets or regional-based routing with multiple depots, this significantly reduces the number of calculations required. In a 1200 point matrix with appropriate business filter, the time required to calculate was reduced by nearly three-quarters or ~45 min. These thresholds can be configured.

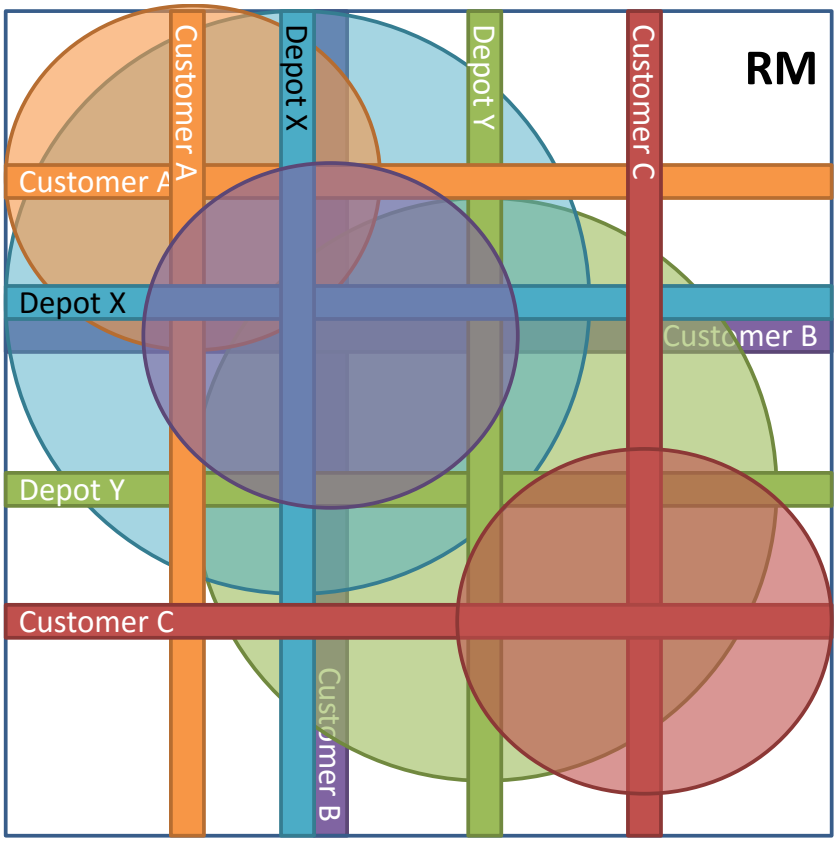
There are two GCM filters:

- **Depot to customer**
  - Uses the larger of the Schedule and Resource Max Distance From Depot settings
  - This setting is always active
  - If Max Distance From Depot is not configured, it will default to 1000km
- **Customer to customer**
  - Uses the Schedule Max Distance From Customer settings
  - This setting is always active
  - If Max Distance From Depot is not configured, it will default to 1000km

The following picture depicts how these filters work together. In the Route Matrix (RM) we will visualize two depots and a customer. Each location represents both a row and a column in the matrix. The GCM filters are then applied around each Depot and each location. When computing a row in the

matrix, only those columns which have a matching GCM Filter will be computed.

➲ **Note** – The GCM has significantly less benefit with one-way pathing compared to two-way pathing unless the distances involved are very large.



When computing the row for Customer "B" we will compute the cell for:

- Customer "A"
- Depot "X"
- Depot "Y"
- NOT customer "C" (GCM filter does not overlap)

When computing the row for Customer "C" we will compute the cell for:

- Depot "Y" *only* (Only GCM filter to overlap)

LNOS Pathing Service Recycling

The LNOS Pathing Service should be configured to recycle between once per day and once per week. Use the task scheduler to reschedule by day of week. The time required to restart is approximately 1 minute per million cells cached.

This is important because staged map edits take effect on recycle and old or expired cache entries are only trimmed out on recycle.

It is advisable to backup this cache to the "alternate" External Pather server ~15-30 min after the shutdown has occurred. See *Appendix E: Windows® PowerShell Script to Shutdown Pathing Service and* Replace Cache for an example script to shut down the alternate LNOS Pathing Service and replace the cache from a staged copy before starting again. This ensures in the case of service failure the alternate server has an reasonably up-to-date copy of the cache.

Restarting the LNOS Pathing Service specific days of the week can be achieved through a scheduled task executing the following command:

```
%systemroot%\system32\inetsrv\appcmd.exe recycle apppool
"PathingService"
```

External Pather Timeouts

The External Pather, being a web-based solution, is subject to timeouts. In general, as you move up the chain of communication the timeouts need to increase.

Consequences of timing out: Depending on where the timeout occurs, it can have significant consequences to the overall system performance.

- **RMWS**
  - Entire calculation is discarded. If the caller does not realize the calculation is discarded, the system could keep re-requesting the same calculation consuming all available resources.
  - Usually occurs when environment is under-sized.
- **LNOS Pathing Service**
  - Will attempt to abort the RMWS calculation. May result in partial cache and may not.
  - Cells not computed will be pathed as straight line until the next optimization request (which may also timeout).

➲ **Note—** If LNOS Pathing Service timeout is smaller than RMWS timeout, the abort will not be requested.

- **External Pather**
  - Will attempt to abort the RMWS calculation.
  - Will result in straight line pathing until next optimization request for un-computed paths.
  - If this occurs in the initial Unconstrained/Constrained calls of the BGO, it could result in ad-hoc RMWS calculations which are very high in number and very inefficient.

  ➲ **Note—** If External Pather timeout is smaller than the LNOS Pathing Service timeout, the abort will not be requested.

- **BGO Build Cache**
  - Will attempt to abort the RMWS calculation.
  - Will result in straight line pathing until the next optimization request for un-computed paths.
  - If this occurs in the initial Unconstrained/Constrained calls of the BGO, it will result in ad-hoc RMWS calculations which are very high in number and inefficient.

  ➲ **Note—** Calculations already completed by LNOS Pathing Service will be cached, data is not lost.

CONFIDENTIAL AND PROPRIETARY TO THE DESCARTES SYSTEMS GROUP INC. AND ITS AFFILIATES

Recommended Timeout Configurations:

Communication Tier	Value	Setting
Route Matrix Web Service (RMWS) Web.config	900	**Matrix.TimeoutSecs** Base timeout for a single matrix request
	930	**system.web/httpRunTime/@ExecutionTimeout** HTTP Request timeout, *+X* seconds to account for any non-computation activity
LNOS Pathing Service (LPS) Web.Config	1400	**MatrixPool.TimeeoutSec** Timeout that RM assembly should wait for the RMWS to perform a computation, *+X* seconds to account for any communication overhead to RMWS. **Note—** Does not include queue time
	1600	**system.web/httpRunTime/@ExecutionTimeout** HTTP Request timeout, *+X* minutes to account for LNOS Pathing Service processing and time in RM queue.
RMPI External Pather Rmpi.ini Or; Local.Properties	1800	**RMHTTPTimeout** HTTP request timeout for External Pather, *+X* minutes for communication time. Needs to be more than LPS httpRunTime.
Background Optimizer (BGO) Local.properties	3600	**Build_Cache_Timeout** Maximum time to build cache entries, needs to be at least 2x Ext Pather Timeout. Multiply by number of routing parameters.

Using Map Edits with the LNOS Pathing Service

Overview

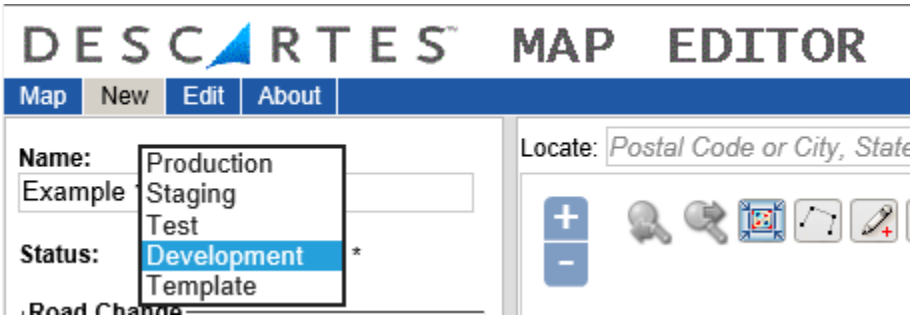
Map edits are created using the LNOS Map Editor which is a standalone web application. It requires a local instance of Descartes AltaMap to analyze the edit being requested. This instance is usually deployed on the Tile Server or LNOS Pathing Service Server. Only one LNOS Map Editor can be live at a time. The LNOS Pathing Service must be configured to communicate with a specific LNOS Map Editor server, however an LNOS Map Editor server can communicate with one or more LNOS Pathing Service servers.

Load balancing is not currently supported. In addition, the LNOS Pathing Service registers itself with the LNOS Map Editor so that changes to existing "Production" edits will be synchronized from LNOS Map Editor to LNOS Pathing Service. When a map edit changes, the entire LNOS Pathing Service cache for the Schedule Group is invalidated.

Edits should generally be made in a "Staging" status. "Production" edits should be avoided at all costs as they will significantly slow down the solution while the cache is rebuilt.

Deployment Stages (the "Status")

A new map edit can be created with one of several statuses. The status determines when the edit will be used by the system..



The only edits considered by pathing are those in the "Production" or "Staging" status.

➲ **Note—** When an edit is introduced any the Schedule Group LNOS Pathing Service cache is invalidated and needs to be recalculated. Depending on the size of group this could take a long time.

"Staging" edits are promoted to production on the next recycle of the LNOS Map Editor. "Production" edits are implemented immediately and will also immediately affect the LNOS Pathing Service cache.

The LNOS Map Editor recycle should be scheduled for a period of time in which the BGO will have plenty of time to rebuild the cache. Usually this will be in the evening of a weekend.

Example 1: Batch optimization

If a customer operates from 8 am to 6 pm and the BGO runs from 12am to 7 am then the best time to recycle would be between 6 pm to 12 am.

Example 2: Incremental Optimization

If a customer operates from 8 am to 6 pm and routes are due by 3 pm for review:

- BGO runs 24 hours a day for today+2 … today+n
- BGO runs from 2 am to 3 pm for today+1, but since most orders are created between 10am to 2pm the best time to recycle would be between 6pm and 2am.

Delayed Activation of Map Edits to Preserve Cache

In some cases, it is desirable to have the map edits created throughout the week but then be implemented on a day when few or no users are expected. In this case, the activation should be "delayed" between the time they are marked as "Staging" to "Production".

Usually the LNOS Map Editor recycle on the weekend is sufficient but if there is maintenance or a problem with the server, it is possible that the map edits will be promoted early. The best way to avoid this problem would be to use the following process:

- Throughout the week, new edits are created in a "Development" status (ignored by pathing and promotion)
- On Friday, an administrator or the user updates the desired edits from "Development" to "Staging"

Appendix

Appendix A: RMPI configuration settings for External Pather

The following table contains all configurable properties, including debug properties. When configured in the rmpi.ini file, the property is entered under the [section]. When configured in BGO local.properties the format is [section].[property].

Main Configuration Settings

Section	Property	Default Value	Usage/Purpose/Explanation
Routing	UseExternalRoadRouter	0	Enables/Disables External Road Router. 3 = On 0 = Off **Note—** required for BGO local.properties and must match SysValue.
Routing	ExtPathingServer	null	URL to pathing server, is used as the default when not overridden by schedule, schedule group or SysValue. Example: http://server/PathingService/InosPathingService.asp x **Note—** Required for BGO local.properties.
Routing	ExtPathingServerAlt	null	URL to "alternate" pathing server, if the primary server is unavailable during initialization this server is used instead. If not configured it will revert to straight line. **Note—** Required for BGO local.properties.
Routing	ConvertP2PtoRM	1	Whenever a point-to-point call would be made, RMPI will instead perform a matrix request that includes the new stop and any stop previously computed. **Recommended Value: 0** This could result in much slower optimization if a cache is not well established as it will do a matrix of 1 stop, then 2 then 3, etc.

			1 = On 0 = Off
Routing	ClearCacheCopy	1	Once an optimization is completed, determines if the process specific cache files will be deleted. **Recommended value: 1** 1 = On 0 = Off
Routing	AsyncPollingSleepSeconds	5	If performing an asynch pathing call, polling interval to check for LNOS Pathing Service calculation results. **Recommended Value: 1 (UI/BIF), 5 (BGO)**
Routing	MinStopsForAsync	200	When performing a matrix call >= N stops, the request will be done asynchronously allowing RMPI to abort the request. This adds an overhead to call time due to async polling.
Routing	P2PHttpTimeout	30	Maximum time to wait for a point-to-point call to complete (seconds). Typical response time is <5 ms if cached and 15-300ms if not with some taking 1-1.5 seconds. The larger the geography the longer it will take.
Routing	RMHttpTimeout	1800	Maximum time to wait for an RM call to complete (seconds). This includes LNOS Pathing Service processing time, RM queue and RMWS processing time. See *External Pather Timeouts* for more details around recommended values.
Routing	TrafficDOWMapping	DOW(DOW)	Configures RMPI to copy traffic patterns from one day of week to another, thereby eliminating the need to calculate for certain days of the week. This is useful when patterns are similar for certain days (e.g. Saturday and Sunday). The property is a comma separate list of pairs in the format **To(From)**. For Example: `TrafficDOWMapping=R(T),M(W),F(W),S(U)` Would achieve the following: Thursday will use Tuesday R(T) Monday will use Wednesday M(W)

			Friday will use Wednesday F(W) Saturday will use Sunday S(W) The example above would reduce computation time by approximately 20% for a 600 location matrix.
Routing	InterpolationBoundary LimitMinutes	240	Determines how big the interpolation window is around a timeslot, see *Traffic Pattern Interpolation* for further details on how this setting works.

Debugging settings

In addition to the required flags, there are several optional flags that can be enabled for debugging.

Section	Property	Default Value	Usage/Purpose/Explanation
Routing	HttpPostDebug	0	Creates a log file containing the input sent by the External Pather library to the lnos pathing webservice (LPWS) and response received back. This file can grow quite large quickly, there are n + *3 rows per point* for each request and m + *square of number points* in response. So a 10 point request would contain n+3*10+m+10*10 or n+m+130. "n" and "m" are a static number of lines per request/response.
Routing	ExtPathingDebug	path/filename	Contains details on what RMPI is requesting from the External Pather. RMPI may request a matrix from the External Pather but the External Pather may not actually send the request or may modify it based on data already known. This log grows extremely quickly as well, caution should be taken when enabling it to disable it again later.
Routing	ExtPathingDebugLevel	0→5	Determines how verbose the ExtPathingDebugLog is (e.g. at level 4/5 traffic adjustment factors are logged)
Routing	GenerateHttpLogs	1 (on) 0 (off)	Must be enabled if HTTP logs are to be created at specified path.
Routing	ExtRunTimes	1 (on) 0 (off)	Should be enabled if others are enabled
Routing	CacheCoverage	1.0	Threshold to determine whether External Pather will request pathing of LNOS Pathing Service.

			Should not be modified unless directed by Descartes engineering.

Appendix B: BGO Configuration Settings for External Pather

The complete list of BGO External Pather configuration properties are detailed in the following table.

Key (com.descartes.escheduler)	**Default Value**	**Description**
build_cache	1	When set to 1, will build the cache each time a new dataslice is downloaded from Descartes Route Planner. When set to 2, will build the cache each time a new dataslice is received, *and on every refresh*. For this reason it is important that the optimizer setting **refresh_lnosfw_data** be increased to a higher limit (e.g. 15 minutes) When set to 0, will not pre-build the cache, this is NOT recommended. **Recommended Value: 2** **Note—** The cache build will only occur if there is at least 1 resource selected by the dataslice.
build_cache_territory	0	The default value for this setting is 0 (Off) If configured, when performing a re-build of the cache, it will evaluate the territories for the schedule. If they exist, the system will split the request into multiple smaller requests grouped by territory. All the depots will be included in each request. Swaps are evaluated. By reducing the size of each call, this setting reduces the time other requests will remain in the queue. **Recommended Value: 0** **Note—** Due to a known RMPI issue in 16.06 and earlier, this setting does not work as intended and must be disabled.

build_cache_timeout		The default value for this setting is 60 minutes. This setting determines the maximum time to wait for the build cache call to complete, including the time in queue that a cache is waiting to be pathed. **Recommended Value:** See the *External Pather Timeouts* section for more details on best practices regarding timeouts. **Note—** Optimizer setting **new_lnosfw_data** should always be larger than the configured value. It is important that *route_optimization_count* and *unassigned_optimization_count* should both be configured to ensure an individual dataslice is not optimized for too long a period.

Appendix C: Example RMPI/BGO Configuration File

UI/BIF/BGO Rmpi.ini

```
[Routing]
…snip…
CacheCoverage                   =  1.0
ConvertP2PtoRM                  =  0
ClearCacheCopy                  =  1
; allow halting during larger matrix sizes
AsyncPollingSleepSeconds=1
MinStopsForAsync=200
GenerateHttpLogs=0
P2PHttpTimeout=30
RMHttpTimeout=1800
;InterpolationBoundaryLimitMinutes=240
;TrafficDOWMapping=S(U)
…snip…
```

BGO Local.properties

```
Routing.UseExternalRoadRouter = 3
Routing.ExtPathingServer = http://LPS01/PathingService/PathingService.aspx
Routing.ExtPathingServerAlt = http://LPSAlt01/PathingService/PathingService.aspx

Routing.CacheCoverage                   =  1.0
Routing.ConvertP2PtoRM                  =  0
Routing.ClearCacheCopy                  =  1
; InterpolationBoundaryLimitMinutes=240
; Routing.TrafficDOWMapping=S(U)

; allow halting during larger matrix sizes
Routing.AsyncPollingSleepSeconds=5
Routing.MinStopsForAsync=200
Routing.GenerateHttpLogs=0
Routing.P2PHttpTimeout=30
Routing.RMHttpTimeout=1800


com.descartes.escheduler.build_cache =2
com.descartes.escheduler.build_cache_timeout = 30
com.descartes.escheduler.build_cache_territory = 0
```


CONFIDENTIAL AND PROPRIETARY TO THE DESCARTES SYSTEMS GROUP INC. AND ITS AFFILIATES

Appendix D: DMDebugLog parsing

Example DMDebugLog (info mode):

```
2015-06-09 14:26:07,982 INFO  RoutingParamterSetupTime|68ms|NumStops:20|DTMS3WBA01_3444_JMTEST|||||||||||||
2015-06-09 14:26:07,982 INFO  RouteMatrixProblemSetupTime|3ms|NumStops:20|DTMS3WBA01_3444_JMTEST|||||||||||||
2015-06-09 14:26:10,962 INFO  RouteMatrixCalculateTime for
          (CalculateAll)|2981ms|NumStops:20|DTMS3WBA01_3444_JMTEST|StartDate:0001-01-01T00:00:00|111110 301.09
          26.822,24.587,22.352,17.881,13.411,8.940,6.705,3.129 StraightTruck 0 381,548,-1| 2
          18739|MissingCells=210|TotalCells=400|47.00%|20|20|10|Succeeded
2015-06-09 14:26:11,009 INFO  MatrixExtractionTime and cache insertion
          time|39ms|NumStops:20|DTMS3WBA01_3444_JMTEST|||||||||||||
```

Basic Parsing using Microsoft® Corporation's Windows® PowerShell

The contents of the DMDebug log can be parsed with a single line with Windows® PowerShell:

```
$gmperflog = get-content .\DMDebug.log | ConvertFrom-Csv -Delimiter '|' -
Header @('Date Time','LogLevel','PathingServiceTask','Duration
(ms)','NumStops','ClientInfo','Start Date','Routing Profile','Truck Profile
#','Missing Cells','Total Cells','% Cached','Missing Full Columns','Missing
Full Rows','Est New Stops')
```

Once he object array is saved it can be output easily:

```
$gmperflog | format-list
```

```
Date Time            : 2015-05-12 21:27:14,423
LogLevel             : INFO
PathingServiceTask   : RouteMatrixCalculateTime for (CalculateAll)
Duration (ms)        : 53964ms
NumStops             : NumStops=1553
ClientInfo           : DefaultSchedule_DTMS3WBA01_2148
Start Date           : StartDate=0001-01-01T00:00:00
Routing Profile      : 100110 19.11 44.704,29.057,24.587,18.328,13.858,9.387,2.682,1.341
StraightTruck 0 -1,-1,-1
Truck Profile #      : 1 33001
Missing Cells        : MissingCells=0
Total Cells          : TotalCells=2411809
% Cached             : 100.00%
Missing Full Columns : 0
Missing Full Rows    : 0
Est New Stops        : 0

Date Time            : 2015-05-12 21:27:14,438
LogLevel             : INFO
PathingServiceTask   : MatrixExtractionTime and cache insertion time
Duration (ms)        : 16ms
NumStops             : NumStops=1553
ClientInfo           : 300110_Fill_BULK_GM_TR_DTMS3WBA01_2148
Start Date           :
Routing Profile      :
Truck Profile #      :
```

```
Missing Cells        :
Total Cells          :
% Cached             :
Missing Full Columns :
Missing Full Rows    :
Est New Stops        :
```

To filter for just the RM calculates the query can be modified as follows:

```
$gmperflog | where({$_.PathingServiceTask -match 'CalculateAll'}) | format-
list
```

To return specific columns only the query can be modified as follows:

```
$gmperflog | where({$_.PathingServiceTask -match 'CalculateAll'}) | select
'date time', '% cached', 'NumStops' | format-list
```

For more details on Windows® PowerShell scripting see:
https://technet.microsoft.com/en-ca/library/hh857337.aspx
https://technet.microsoft.com/en-us/scriptcenter/dd742419.aspx

Appendix E: Windows® PowerShell Script to Shutdown Pathing Service and Replace Cache

```
#
# Invoke_RestartPathingAppAndReplaceCache.ps1
#
# these 2 lines enable functions and import the core functions required to administer IIS

Set-ExecutionPolicy unrestricted -Force
import-module webadministration

# this function does the work to restart and replace cache
function Invoke-RestartPathingAppAndReplaceCache ([string]$AppName, [int] $TimeOutSec,
[string]$wwwroot, [string]$CacheStagingFolder)
{
	if ([string]::IsNullorEmpty($wwwroot))
	{ $wwwroot = 'c:\inetpub\wwwroot' }
	if ([string]::IsNullorEmpty($CacheStagingFolder))
	{ $CacheStagingFolder = 'Staging' }

	$cacheFolder = "$wwwroot\$AppName\GeneratedFiles"
	$backupFolder = "$cacheFolder\${CacheStagingFolder}_old"
	$newFolder = "$cacheFolder\$CacheStagingFolder"

	if (! (Test-Path $backupFolder)) { md $backupFolder }
	if (! (Test-Path $newFOlder))    { md $newFOlder }
	Write-Host ("Stopping App Pool [$AppName], current state is [{0}]" -f (Get-
WebAppPoolState -name $AppName).Value)
	$stop = [DateTime]::NOW
	if ((Get-WebAppPoolState -name $AppName).Value -eq "Started")
	{
		stop-webapppool -name $AppName
	}
	$w3wplist = Get-WmiObject -Class win32_process | where({$_.name -eq 'w3wp.exe'}) |
			select Name,@{n="UserName";e={$_.GetOwner().user}}
	Write-Host "Waiting for process to shutdown"
	do
	{
		[Threading.Thread]::sleep(5000) # wait a second
		$ProcessesRunning = ($w3wplist | where({$_.UserName -eq $AppName})).Count
		$WaitTime=((New-TimeSpan -Start $stop -End ([DateTime]::NOW)).TotalSeconds)
		Write-Verbose ("Waited: {0}/{1}" -f $WaitTime,$TimeOutSec)
		$state = (Get-WebAppPoolState -name $AppName).Value
	} while (($ProcessRunning -gt 0) -and ($WaitTime -lt $TimeOutSec) -and
		 ($state -ne "Stopped"))

	if ($WaitTime -lt $TimeOutSec)
	{
		Write-Host "Process is stopped"
```

```
            try
            {
                  if ((get-childitem
"c:\inetpub\wwwroot\$AppName\GeneratedFiles\Staging\*.dat").Count -gt 0)
                  {
                        Write-Host "Migrating Cache"
                        copy-item "$cacheFolder\p2p*.lps" $backupFolder -Force
                        copy-item "$newFOlder\p2p*.lps" $cacheFolder -Force
                        copy-item "$cacheFolder\Vehicle*.dat" $backupFolder -Force
                        copy-item "$newFOlder\Vehicle*.dat" $cacheFolder -Force
                  }
                  else
                  {
                        Write-Host "No Cache to migrate, skipping!"
                  }
            }
            catch {
                  write-error "Failed to migrate cache from staging folder
[$newFolder] or to backup to [$backupFolder]"
                  throw $_
            }
            finally {
                  Write-Host ("Starting App Pool [$AppName], current state is [{0}]" -
f
                              (Get-WebAppPoolState -name $AppName).Value)
                  $start = [DateTime]::NOW
                  do
                  {
                        [Threading.Thread]::sleep(5000) # wait a second
                        $ProcessesRunning = ($w3wplist |
                                    where({$_.UserName -eq
$AppName})).Count
                        $WaitTime=((New-TimeSpan -Start $start `
                                          -End
([DateTime]::NOW)).TotalSeconds)
                        Write-Verbose ("Waited: {0}/{1}" -f $WaitTime,$TimeOutSec)
                        $state = (Get-WebAppPoolState -name $AppName).Value
                  } while (($WaitTime -lt $TimeOutSec) -and ($state -eq "Stopping"))

                  if ((Get-WebAppPoolState -name $AppName).Value -eq "Stopped")
                  {  Start-WebAppPool -name $AppName }
            }
      }
      else
      {
            throw "Process took too long to close [$WaitTime]"
      }
}
```

```
# this invokes the function do the restart and replace and is all that is needed once the
function and import has occurred.
Invoke-RestartPathingAppAndReplaceCache -AppName 'PathingService' -TimeOutSec 120
```

Appendix F: External Pather in Larger Deployments Best Practices

BGO Dataslice Design

When the BGO starts up, it will initiate a pathing call for all orders in the dataslice. The BGO does not know whether the orders are eligible to swap or not; for this reason it is important to avoid putting orders into a slice that are never eligible for swapping. The most common problem is creating a dataslice that spans multiple days when all orders and resources are in a single day. In this scenario, the dataslice would path between Monday and Wednesday customers even though these customers will never be eligible for pathing. This issue requires excessive working on the LNOS Pathing Service and the RMWS engine.

Caching Under Different Business Models

Different business models have different caching requirements. In forecast based solutions, there is usually a pool of orders with a date range per order. On any given day there may be 20 to 30 percent new orders and five to 10 percent deletes but approximately sixty to seventy percent of the orders do not change. In this scenario, a longer cache is useful. The cache should be maintained for approximately seven to 14 days, depending on customer repeat rates.

In a home delivery business, there is a nearly new set of customers (approximately 90 to 95 percent) on a daily basis. In this scenario, longer caches are less useful and the cache grows significantly day by day. In this business model, the cache should only be maintained for three to seven days depending on customer repeat rates. A calculator is available to help predict the size of the cache.

Dividing the LNOS Pathing Service by Region

In very large deployments, it might be necessary to use multiple LNOS Pathing Service instances and sectioning by region or geography to avoid overwhelming a single server due to cache size.

- 5 million cells will use approximately 5gb of memory
- 15 million cells will use approximately 15gb of memory
- 30 million cells will use approximately 30gb of memory

When planning an LNOS Pathing Service server, there should always be at least a 8gb buffer to avoid the Operating System paging out the cache from memory to disk. This can cause a significant slowdown in the application. This buffer includes memory for the operating system, working memory and free memory.

The LNOS Pathing Service server is configurable per Schedule Group.

See *Appendix O: Dividing up LPS servers* for a query to help with managing which schedule groups should belong to which LPS server.

Appendix G: External Router vs. LMB summary

Pathing	**LMB**	Path between points is calculated using a cost weighted system defined by the routing parameters (**Data > Routing Parameters**), this provided control of fastest vs. shortest vs. simplest
	Descartes AltaMap	Descartes AltaMap only supports two pathing algorithms, Fastest and Shortest
Speeds	**LMB**	The driving time between two points was calculated using the speed definitions defined in the routing parameters, regardless of posted speed limits. These speeds varied by population density.
	Descartes AltaMap	The driving time between two points is calculated using the posted speed limits (if known). If unknown, reverts to a set of internal speeds. These can be adjusted using Speed Categories (**Data > Speed Categories**). If speed categories are defined they override the posted speed limits. Traffic pattern data can be used to manage rush hour and time-of-day differences. This traffic pattern only affects the time to get between customers and does not affect the path being chosen at this time. **Note—** This service must be licensed from the map vendor (Historical Traffic Patterns)
Commercial Routing	**LMB**	Roads were either flagged as Commercially Restricted or not; if commercially restricted, no trucks would be allowed to use them if unchecked in the routing parameters.
	Descartes AltaMap	Roads flagged as Commercially Restricted will be used by default. To enable commercial routing, the Resource Truck Type attribute must be configured. Descartes AltaMap can also use HazMat routing. To enable hazmat routing the order must have the **HazMat** option enabled. Likewise the resource must have the **HazMat** option enabled. When enabled, Descartes AltaMap will not use roads classified as **HazMat** restricted. Descartes AltaMap can also use vehicle dimension routing. To enable vehicle dimension routing, the **Weight** (GVW), **Height**, **Width**, **Length** need to be specified on the resource. These restrictions will only be used *if defined in the map data* and will only be used for the vehicles in violation of these restrictions. In addition, the onboard vehicle weight will be considered when selecting roads. This requires configuration of the schedule settings for weight measure and weight measure factor (to kilograms) to be configured properly. This onboard weight is added to the **Empty Weight** property of the resource. It is highly recommended not to use this feature due to the potential significant performance impact.

		Descartes AltaMap can ignore the violations near the origin/destination based on the **Ext Pathing Violation Radius**. **Note—** This must be licensed from the map vendor (Truck Attributes)
Costing	**LMB**	Routing parameter costs were used to determine the actual paths taken. This provided significantly flexibility to the costing for driving.
	Descartes AltaMap	Routing parameter costs are no longer used for pathing. Instead, costs are calculated after pathing but are considered during optimization.
Map Editor	**LMB**	Maps were edited using the RIMMS software. These files would then need to be copied and deployed to each server into the cache folders and the cache files would also need to be cleared.
	Descartes AltaMap	Maps are now edited using the web-based LNOS Map Editor. Once map edits are completed, the cache file will be automatically be purged both on the Map Server and the Descartes Route Planner servers. The type of road edit determines when this purge occurs. A single map edit will cause the entire cache to be removed.

Appendix H: Incremental Migration /Priming LNOS Pathing Service in Parallel

Prior to switching over to the External Pather/Descartes AltaMap, it will likely be necessary to perform parallel optimization with LMB and External Pather. This optimization may be performed to compare results or to allow the BGO to prime the cache prior to a cutover.

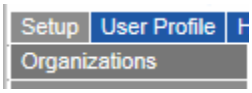
To perform a partial migration requires changes on the interface and user end. Normally, normally the prime test process uses the alternate organization and then switches over the branches at once. However, for large organizations, this is very risky and it is worth taking the time to do a partial migration.

The following process outlines a framework of how to setup an alternate organization that will use the External Pather parallel to existing LMB optimizations.

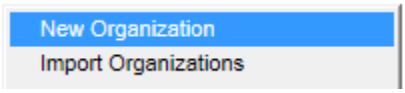
> **Warning:** Using two organizations is not currently supported by the Descartes Route Planner licensing engine which means the license will enter a grace period as soon as this alternate organization is setup. Prior to making this change, it should be communicated to Descartes Customer Support and the customer and the time period should be kept as short as possible.

Create Organization for External (Descartes AltaMap) pathing

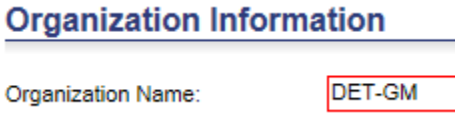
1. Create a new organization with "GM" suffix in the Security database.
   a. Login as [org]/[org]-**SecAdmin** using any user interface.
   b. Navigate to **Setup > Organizations**.



   c. Right-click the work space and select **New** Organization from the right-click menu.

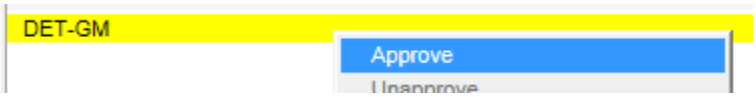


   d. Specify the new "GM" organization.



e Click **Save**.

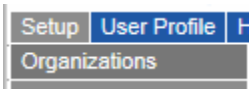
f Right-click the new organization and select **Approve** from the right-click menu.



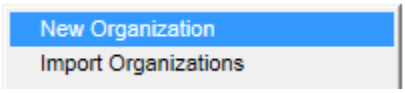
2 Create a new organization "GM" suffix in the DCF database.

a Login as Descartes Systems Group/secadmin using the DCF user interface.

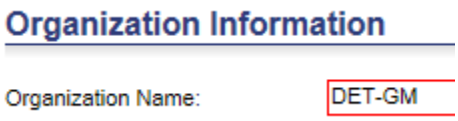
b Navigate to **Setup > Organizations**.



c Right-click on work space and select **New Organization** from the right-click menu.

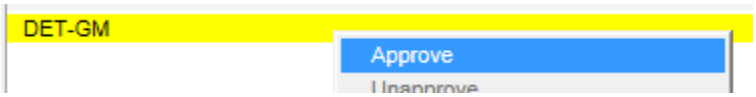


d Specify the new "GM" organization.



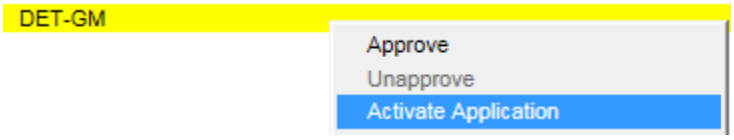
e Click **Save**.

f Right-click the new organization and select **Approve** from the right-click menu.



3 Activate required applications.

a Right-click the new organization and select **Activate Application** from the right-click menu.

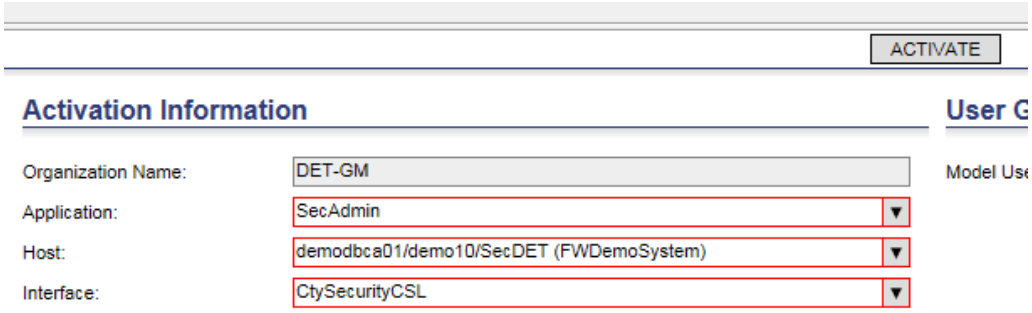


b Choose the application to **Activate**.

c Typically LNOSFW, ShipperTad, SecAdmin

d Choose the host database that matches the "parent" organization.

e Choose the interface CtySecurityCSL.

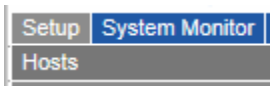


f Click **Activate**.

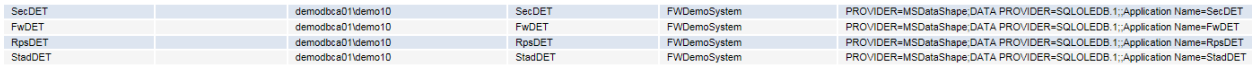
g Repeat for each application.

4 Duplicate host connections for GM organizations.

a Navigate to **Setup > Hosts**

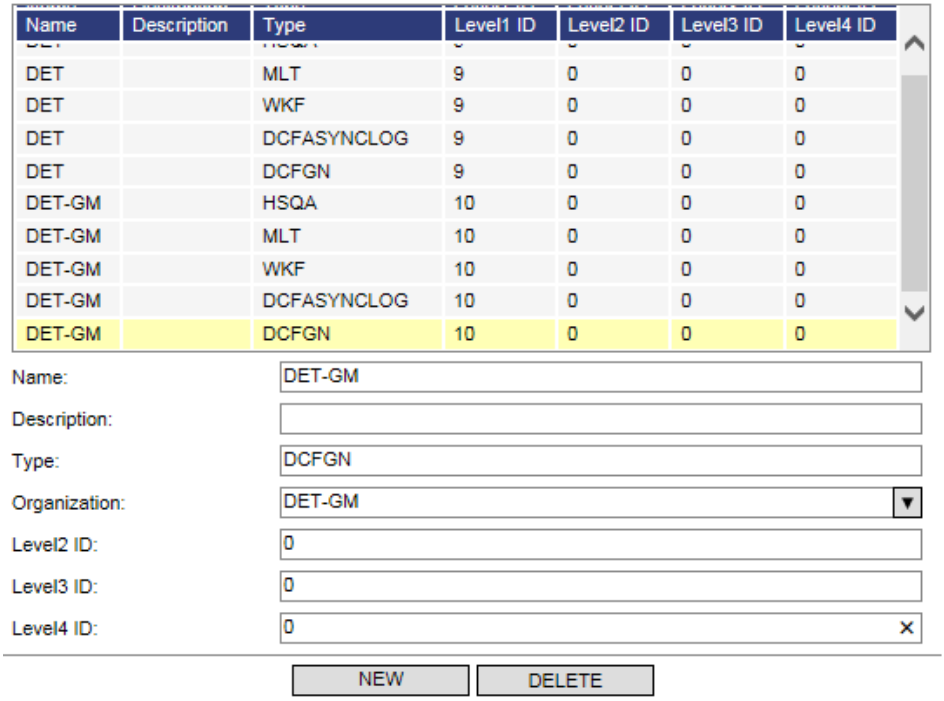


b Edit the host record for each application activated for the "parent" organization.



SecDET		demodbca01\demo10	SecDET	FWDemoSystem	PROVIDER=MSDataShape;DATA PROVIDER=SQLOLEDB.1;;Application Name=SecDET
FwDET		demodbca01\demo10	FwDET	FWDemoSystem	PROVIDER=MSDataShape;DATA PROVIDER=SQLOLEDB.1;;Application Name=FwDET
RpsDET		demodbca01\demo10	RpsDET	FWDemoSystem	PROVIDER=MSDataShape;DATA PROVIDER=SQLOLEDB.1;;Application Name=RpsDET
StadDET		demodbca01\demo10	StadDET	FWDemoSystem	PROVIDER=MSDataShape;DATA PROVIDER=SQLOLEDB.1;;Application Name=StadDET

c In the connections section, click **New** and recreate each "type" choosing the "GM" organization from the dropdown.



Name	Description	Type	Level1 ID	Level2 ID	Level3 ID	Level4 ID
DET		MLT	9	0	0	0
DET		WKF	9	0	0	0
DET		DCFASYNCLOG	9	0	0	0
DET		DCFGN	9	0	0	0
DET-GM		HSQA	10	0	0	0
DET-GM		MLT	10	0	0	0
DET-GM		WKF	10	0	0	0
DET-GM		DCFASYNCLOG	10	0	0	0
DET-GM		DCFGN	10	0	0	0

d Click **Save**.

e Repeat for each host record.

5 Update the DCFOrg table with the glnorgid value for a new organization in both the DCF and Sec databases.

a Connect to DCF and Sec[xxx] databases.

b Run the following query:

- Update DCFOrg
- Set GLNOrgId=yyyyyy
- Where OrgName='xxx-GM'

6 Create a user group for the "GM" organization.

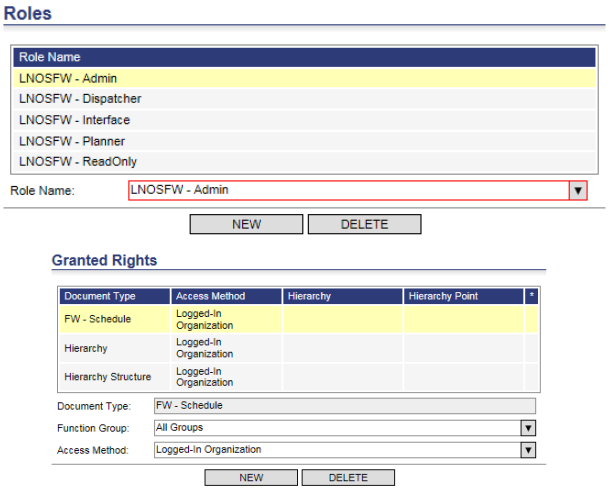
a Login as xxx/xxx-secadmin.

➲ **Note—** Do NOT use the "GM" suffix.

b Navigate to **Setup > Organizations**.

c Right-click the new "GM" organization and choose **User Groups** from the right-click menu.

d Create a new group with the following roles and rights at a minimum:



➲ **Note—** The access method may be different. In this case, "logged in org" refers to GM schedules only.

➲ **Note—** Full access provides access to both "GM" and non-"GM" schedules, which is required if priming against LMB archive schedules.

e Click **Resecure** in the **Rights** tab.

f Right-click the user group and choose **Activate** from the right-click menu.

Create Org Specific Configuration settings

7 Run the following query to create the org-specific external pathing settings:

```
insert into ctysysvalue
select (select max(sysvalueid) from ctysysvalue)+
            (rank() over (order by sysvalueid)),
            accountid=6,groupid,value,modifiedby,
            modifydate,keyword,keygroup
from ctysysvalue
where keyword like '%ext%'
```

➲ **Note—** AccountID=6 is the orgid in the SecXXXX..dcforg table for the "GM" org.

8 Configure the org-specific external pathing settings.

a Through the database, edit these settings and configure appropriately. AccountID=6 is the key to identifying them.

- 5 = original org
- 6 = new org
- 0 = all orgs

b The following query can be used for configuration:

```
update CtySysValue
set Value=case
        when Keyword='UseExternalRoadRouter' then '3'
        when Keyword='ExternalRoadRouter' then
'http://server/lnospathingservice/lnospathingservice.aspx'
        when Keyword='AltExternalRoadRouter' then
'http://server2/lnospathingservice/lnospathingservice.aspx'
        END
where AccountID=6
```

➲ **Note—** If an Alternate External Road Router does not exist, set this to null.

9 Shutdown COM+ services to reload settings.

Verify Configuration

10 Login as xxx/xxx-usradmin

a Right-click the "GM" org and choose **Users** from the right-click menu.

b Create a new user in the user group and assign at least one of the LNOSFW roles that allows optimization.

11 Verify the "GM" organization.

a Login with the "GM" organization.

b Specify the user created in the previous step.

c Perform an optimization operation.

12 Monitor the pathing service log on the GM server and verify that pathing requests are received and processed.

Enable BGO Priming

1 Create new dataslices in a new areakey for priming.

a Navigate to **App Setup > Groups**. Create "Group" and enter a new areakey (e.g. ExtPatherPriming). Set the type to DEDICATED_BGO.

b Navigate to **App Setup > Values**. Create "Values" and use the archive scheduled with previous "n" days as the date range.

c Navigate to **App Setup > Properties**. Create a new BGO "Properties" set with all BGO phases set to disabled (0).

d Create a Schedule Set, associate properties to the schedule and provide a time period in which it should run.

e Navigate to **App Setup > Groups**. Select priming groups and assign to the new schedule set.

f Create one more group against a "test" schedule. Set the type to BGO.

2 Configure one (or more) BGOs to prime.

a Stop or Disable the optimizer/scheduler.

b Configure local.properties with new a area_key and the "-GM" organization and username.

c Configure local.properties with the External Pather settings.

3 Run the BGO against the test dataslice.

a Enable the optimizer/scheduler then start them.

b Verify that the BGO is running against the "test" schedule only. Verify that pathing is occurring but no optimization.

4 Run the BGO against all dataslices.

a Set the "test" dataslice to DEDICATED_BGO type. Set the remaining dataslices to Type BGO.

b Verify that the BGO is running against the new dataslices and verify that pathing is occurring but no optimization.

➲ **Note**– If using a mobile device, routes in archive will have a status of "End of Trip". This status currently will prevent BGO from being able to prime the routes. At this time the only workaround is to change the LastReportedStatus of the FWRoute table from (17) to something else (3). Once primed, the final status can be restored.

Migrating a Branch to the "-GM" Ext Pather Org

13 To migrate a "branch":

a Migrate the user group:

- Run the following query to migrate the user group:
- `Update ctyusergrp set orgid=6 where usergrpname in ('xxx')`

b Migrate the user:

- Run the following query to migrate the user:
- `Update ctyuser set orgid=6 where loginname in ('xxx','yyy','zzz')`

- Login as xxx/xxx-usradmin.
- Navigate to **Setup > Organizations**.
- Right-click the "GM" organization and choose **Users** from the right-click menu.

c Migrate the schedule:

- Run the following query to migrate the schedule:
- `Update fwschedule set level1id=6 where schedulekey like 'xxx[_]%'`

➲ **Note—** "xxx" is the branchcode using the convention of xxx_planning, xxx_review, etc.

- Resecure the usergroup to complete the migration.
- Login as xxx/xxxx-secadmin.
- Navigate to **Setup > Organizations**.
- Right-click the "GM" organization, choose **User Groups** from the right-click menu.
- Double-click the migrated usergroup.
- Click **Resecure**.

d Login as the migrated user(s) and test.

Removing the "-GM" Ext Pather Org when finished

14 Deactivate the Organization

a Deactivate in **Security** database:

- Navigate to http://[server]/DCF
- Login as [Org]/[Org]-SecAdmin
- Right click [Org]-GM org, **Unapprove**
- Right click [Org]-GM, choose **User Groups**
- Right click each user group, **Deactivate**
- Right click [Org]-GM, choose **Users**
- Right click each user group, **Deactivate**
- Logout

b Deactivate in **DCF** database:

- Navigate to http://[server]/DCF
- Login as Descartes Systems Group/secadmin
- Navigate to **Setup > Organizations**
- Right click [Org]-GM org, **Unapprove**
- Navigate to Setup>Hosts
- Edit each [Org] entry for the parent organization, e.g. FW[Org], Sec[Org], etc.

- For each [Org] entry, delete connections reference [Org]-GM. They should show as duplicate connections but with a different LevelID in the connections list.

14 Removal the Organization entirely:

Note— Removing all traces of the org requires performing SQL queries directly in the database, before any such query is executed a backup of the database should be taken with a full rollback plan.

a Cleaning up the **Security** database:

- Delete entries from various tables that reference the org/users/groups deactivated.
- The sequence of delete would be:

Table	Comment
CtyRight	Rights for the group (link by UserGrpID)
CtyUserGrpRole	Roles for the group (link by UserGrpID)
CtyUserUserGrp	User to user group links (link by UserGrpID)
CtyUserGrp	User Group (link by OrgID)
CtyUserRole	Rights for the user (link by UserID)
CtyUser	User (link by OrgID)
DCFOrg	Organization

b Cleaning up the **DCF** database

- Delete the DCFOrgApp entries from DCFRTG12 for LPG-GM
- Delete the DCFOrg entry from DCFRTG12 & SecLPG orgs

Appendix I: Summary of Settings for External Pathing

This section provides a quick summary of the available settings for external pathing. The values in parentheses represent the possible and/or typical range of values or their units. The first in the list within the parentheses is the default/recommended.

Taking UseExternalRoadRouter as an example, (0,3/0) means that the recommended value is "3", the default value is "0" and the possible values are "0" or "3".

- **System Values**
  - UseExternalRoadRouter (0,3/0)
  - ExtPatherURL (null,url)
  - AltExtPatherURL (null,url)
- **Schedule Group**
- External Pather URL (null,url)
- Ext Pathing Violation Radius (2.9mi,1→15)
  - *Minimum*
    - Hazardous (null,null/1)

    ➲ **Note—** if HazMatRoute is enabled for all routes, this setting should be set to "1".

    - Weight (null,kg/lb)
    - Height (null,m/ft)
    - Length (null,m/ft)
    - Width (null,m/ft)
    - Vehicle Type (Passenger/Straight/Articulated)

    ➲ **Note—** "Articulated" provides minimal benefit with significant extra calculations.

  - *Maximum*
    - Hazardous (null,null/1)

    ➲ **Note—** If HazMatRoute or HazMatType is enabled for any routes, this setting should be set to "1".

    - Weight (null,kg/lb)
    - Height (null,m/ft)
    - Length (null,m/ft)
    - Width (null,m/ft)
    - Vehicle Type (Passenger/Straight/Articulated)

    ➲ **Note—** Articulated provides minimal benefit with significant extra calculations

- *Vehicle Slots*
  - Weight (1,1→4)
  - Height (1,1→4)
  - Length (1,1→4)
  - Width (1,1→4)

➲ **Note—** Fewer slots = faster pathing

- **Schedule**
  - Measure for Weight (null,measure#)
  - Measure for Weight Conversion (null,null/#, kg **only**)
  - Route As Straight Truck (off,on/off)

➲ **Note—** This was previously labelled "Route As Commercial Truck"

  - Max Distance From Depot (1000km,GCM Filter)
  - Max Distance From Customer (100km,GCM Filter)
- **Resource**
  - *UI settings:*
    - HazMatType (null,null/1)
    - HazMatRoute (0,0/1)
    - Weight (0,kb/lb)
  - *Integration Fields:*
    - DocResource/@HazMatType
    - DocResource/@HazMatRoute
    - DocResource/@CommercialTruck

➲ **Note—** This setting can violate the schedule group. Configure with caution.

    - DocResource/@Weight
    - DocResource/@Height
    - DocResource/@Length
- **Order**
  - *UI settings:*
    - Order Line > Hazardous Material(null,null/1)
  - *Integration Fields*
    - DocBOL/BOLLine/UDF/@HazMatType (null,null/1)
- **Optimizer.Properties (Dataslice Settings)**
  - com.descartes.escheduler.optimizer.refresh_lnosfw_data (10→30+)

➲ **Note—** Needs to be longer if build_cache=2 in local.properties

- **Local.properties**
  - Routing.UseExternalRouting (0,3/0)

- Routing.ExtPatherURL (url, null/url)
- Routing.ExtPathingCacheFile (cache.le1)
- Routing.ClearCacheCopy (1,1/0)
- com.descartes.escheduler.build_cache (1,2/1/0)
- com.descartes.escheduler.build_cache_territory (0,1/0)
- com.descartes.escheduler.build_cache_timeout (60,15→120)

- **Rmpi.ini**
  - ExtPathingCacheFile ([folder]\cache.le1)
  - ClearCacheCopy (1,1/0)
- **LPS Web.Config**
  - *Maintenance/Performance*
    - Matrix.LocalSizeThreshold (0,0→50)
    - ForceGCInMinutes (30,30→14400)
    - MatrixPool.SmallQueueThreshold (500,1→10000)
    - MatrixPool.UISmallQueueThreshold (1000,1→10000)
    - CacheRetentionPeriod (7, 1→30)
    - ScheduledSaveCacheTime (1-2x per day, 1 per week→every hour)
  - *Communication*
    - MatrixPool.QueueSize (32,1→200)

  ➲ **Note—** # BGO's + 6*(# UI/BIF servers)

    - MatrixPool.ServiceUrl.# (url)
    - MatrixPoolSmall.ServiceUrl.# (url)
  - *Pathing Methods*
    - QueryForTraffic (false,false/true)
    - StoredTrafficSlots (40 patterns/6days, [DoW] Time, comma separated, always start with U 00:00)
    - MatrixPool.UseOneWayRouting (true,true/false)

Appendix J: Conversion Factors

The following values are used by the application when converting from the user interface to the database. When interfacing certain fields (e.g. Resource.Weight), the same conversion should be applied first to ensure proper presentation in the user interface and to prevent slight mismatches to the unconstrained/constrained values.

If information in host system is stored in Imperial the following conversions should be used:

- Ft→m, Dimension / 3.2808399
- Lb→kg, Lbs / 2.20462262

If information in host system is stored in Metric the following conversions should be used:

- Cm→m , Dimension * 0.001
- M→m, Dimension
- Kg→kg, Weight in kg

Appendix K: Terminology

The use of the External Pather introduces a new set of terminology which is described below.

- **Background Optimizer (BGO)** – a sub-component of Descartes Route Planner that performs optimization automatically.
- **Cell** – A point in a matrix (equivalent to a point-to-point call). A cell has an origin latitude/longitude, destination latitude/longitude, the type of vehicle used. A cell contains the time and distance values along with other computed values like traffic patterns.
- **Destination –** Ending point in a pathing call.
- **External Pather** – Sub-component of RMPI used to communicate with the LNOS Pathing Service and make decisions regarding how best to respond to pathing requests requiring external pathing.
- **GCM Filter** – A way of reducing the size of a matrix call by filtering out cells to compute based on a radius around each customer and each depot.
- **Great Circle Mile (GCM)** – a mathematical calculation used to compute the distance between any 2 stops (as the crow flies) on the earth.
- **Layer 1 Cache** – see RMPI Cache
- **Layer 2 Cache** – see LNOS Pathing Service Cache
- **LME** – LNOS Map Editor, see *WebService Components* for more details.
- **LPS** – LNOS Pathing Service, see *WebService Components* for more details.
- **LPS Cache** – Pathing cache
- **Matrix (RM)** – A pathing call with multiple origin origin/destination points. Variable depending on number of points. The matrix is **n** x **n** where **n** is the number of distinct latitude/longitude (to six decimal places). The rows represent origins, columns represent destinations

  Example Matrix:

Destination / Origin	O1	O2	O3
O1		D=49mi, T = 52 min	D=5mi, T=5 min
O2	D=50mi, T = 45min		D=120 mi, T = 120 min
O3	D = 7 mi, T = 8 min	D = 120 mi, T = 120 min	

  - In a matrix calculation the diagonal points are not computed as they are always 0. The cells can be referred to by the OD pair e.g. O1O2 is equivalent to a point-to-point call from O1→O2 and O2O1 is equivalent to a point-to-point call from O2→O1 and the Distance/Time (D/T) values may be different.
  - As the matrix grows, the number of cells to compute increases exponentially (n^2-n)

➲ **Note—** In reality, the definition of a row/column/cell including details about the vehicle being pathed, traffic patterns, etc.

- **Origin –** Starting point in a pathing call.
- **Origin/Destination Pair (OD) –** Pair of points starting at the origin and ending at the destination.
- **Point-to-point (P2P)** – A pathing call with a single origin and destination between any two stops: Origin=A, Destination=B.
- **Point2Point** – See *Point to Point*
- **Pathing Cache** – An in-memory or on-disk collection of pathing data previously computed.
- **RMPI** – Optimization Engine used by Descartes Route Planner
- **RMPI Cache** – The on-disk cache used by Descartes Route Planner when making optimization decisions. It is populated from the Layer 2 (LNOS Pathing Service) Cache as required.
- **Schedule Group** – Represents a collection of Schedules and their associated external pathing configuration.
- **RMWS** – Route Matrix Web Service, see *WebService Components* for more details.

Appendix L: LPS and RMWS Performance Numbers

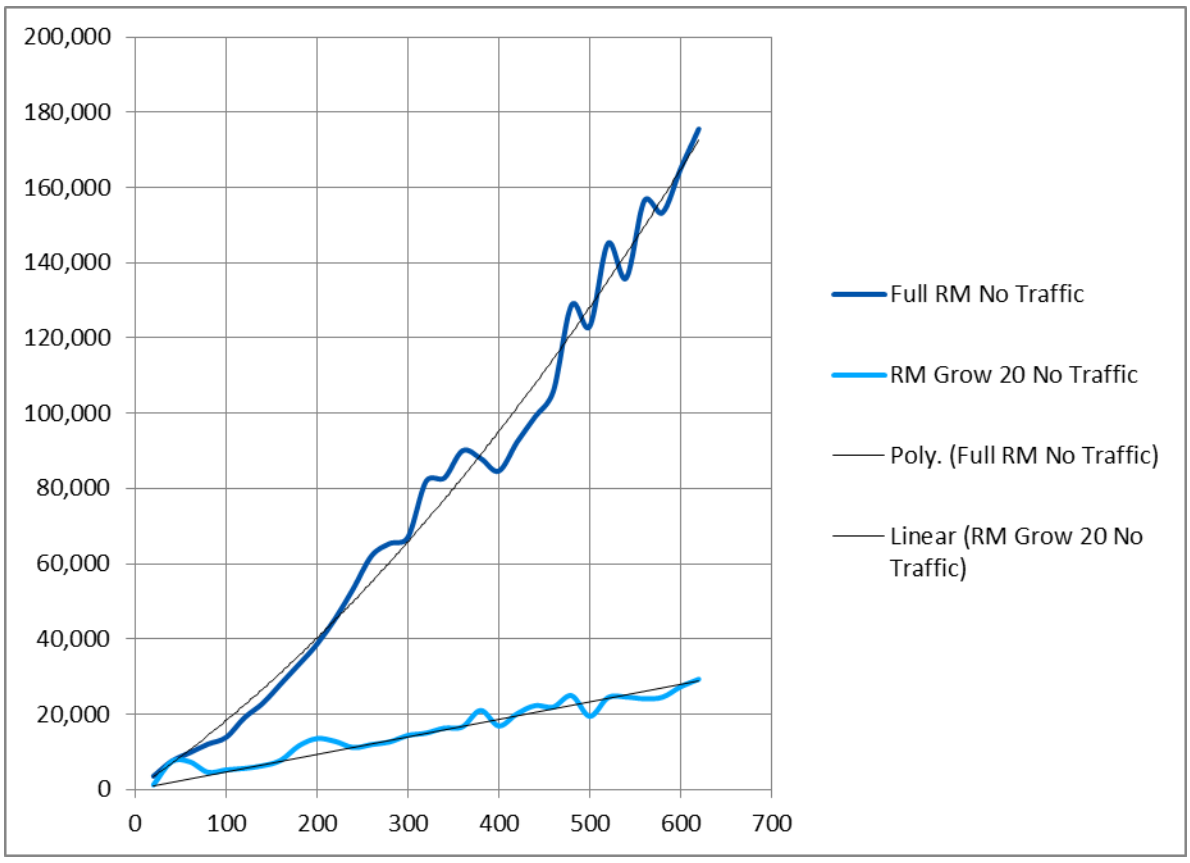
In the graphs below, "RM" refers to a single pathing call Matrix for a given set of vehicle/speed profiles.

The "Grow 20" scenario is when 20 new stops are being calculated against a given dataset. This is the typical growth rate of BGO each time it refreshes or acquires a new dataslice.

Each time an optimization is started, there are two "full" RM calculations: one for the "Minimum" and one for the "Maximum". Intermediate calculations are possible but infrequent, approximately one to five percent of the "full" calculations.

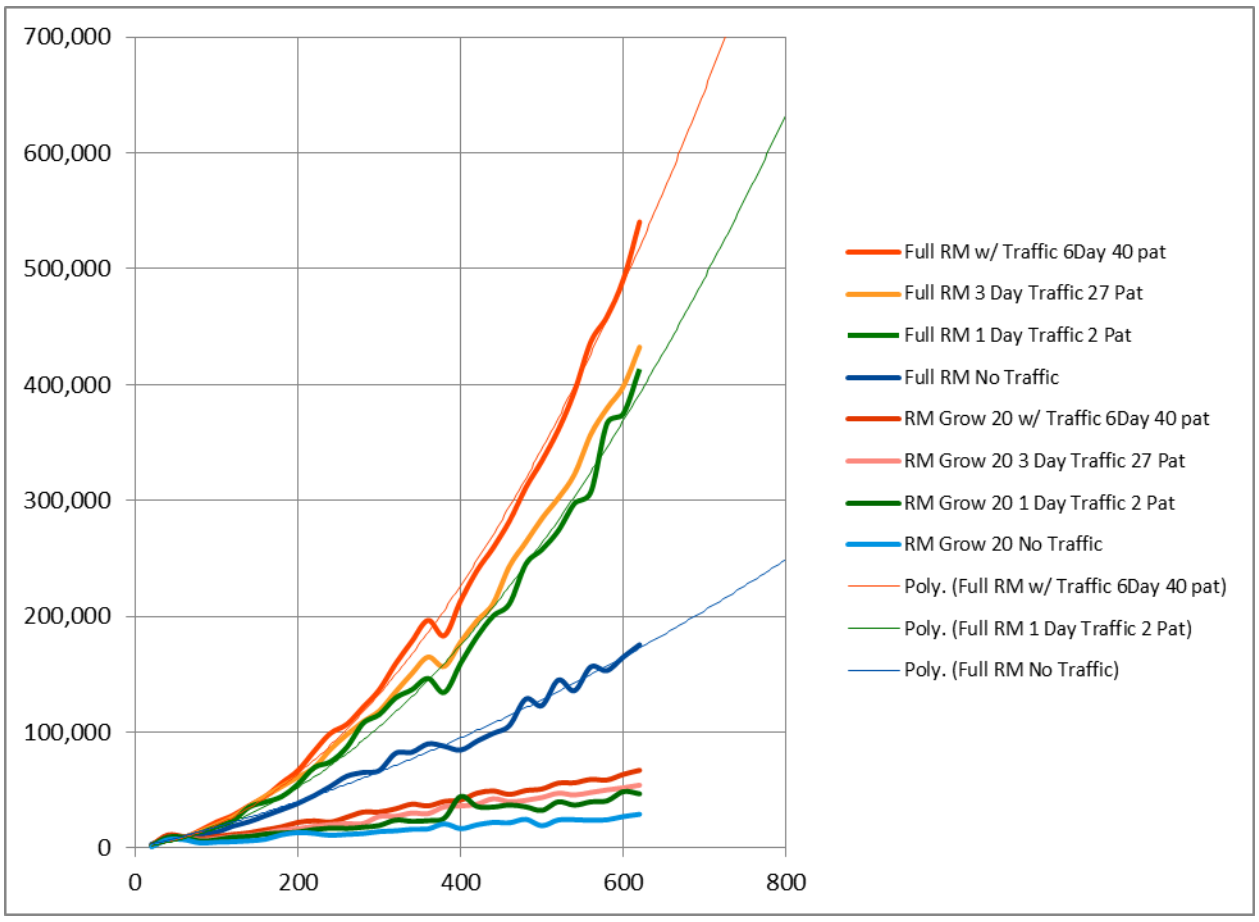
The X-axis represents the number of stops. The Y-axis represents the response times in milliseconds.

One-way pathing, NO Traffic



➲ **Note—** Grow 20 represents a matrix growing by 20 new stops with each request.

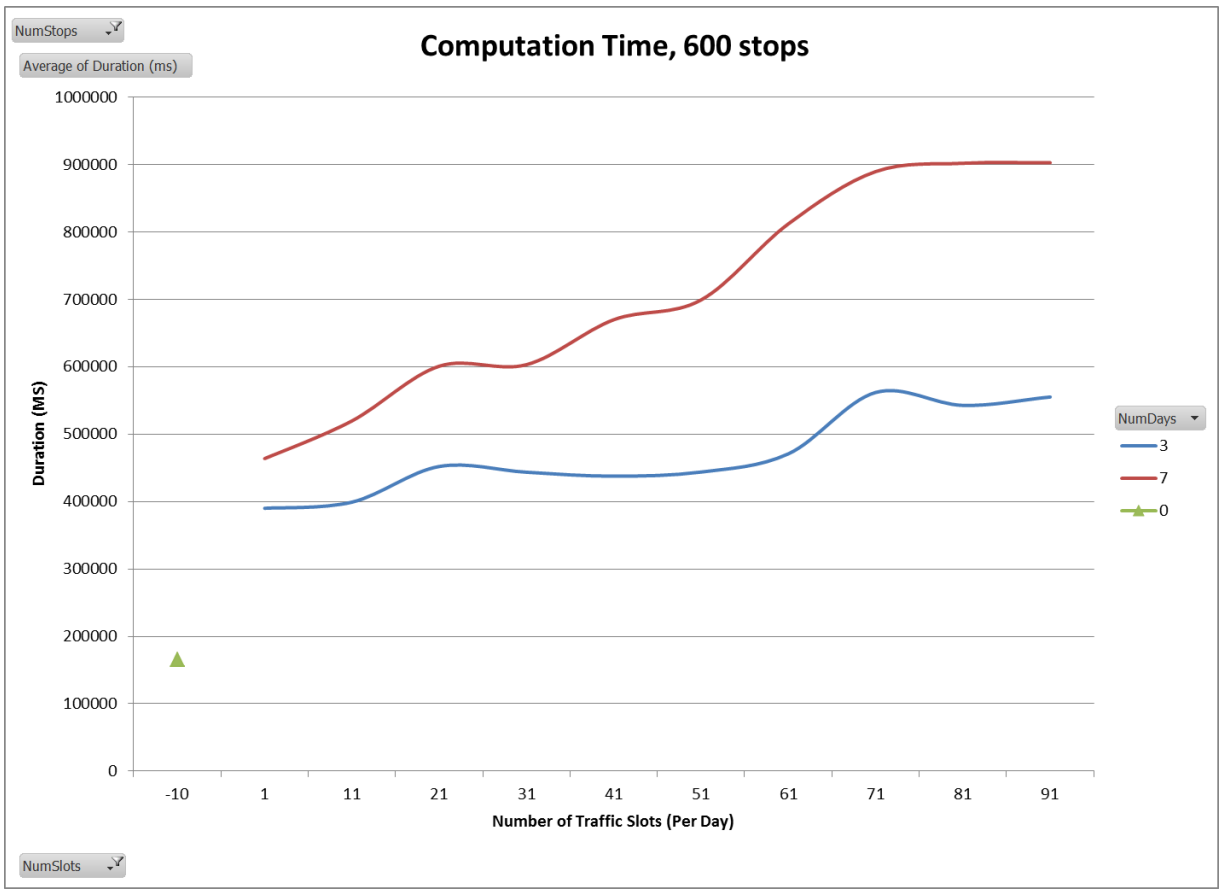
One-way pathing, WITH traffic (compared to without traffic)



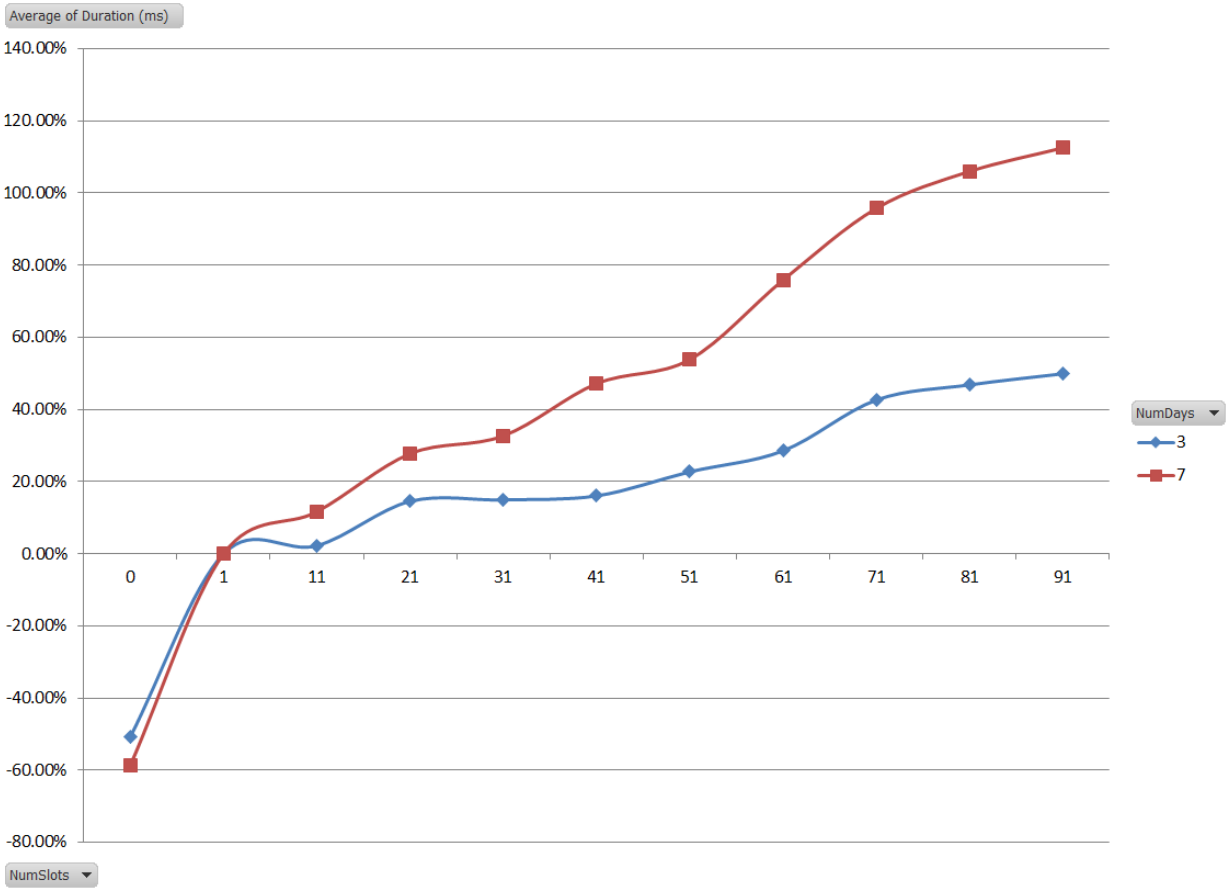
As this graph shows, there is a significant cost to access traffic pattern data and there is then an incremental cost for each day requested. The cost of this traffic data is offset by needing fewer routing parameters to account for traffic by time of day. The larger the number of stops, the more significant this cost becomes as a total percentage of computation time.

One-way Traffic Pathing Time, Traffic Performance

The following graph shows the same pathing request without traffic (0 days, -10 slots).

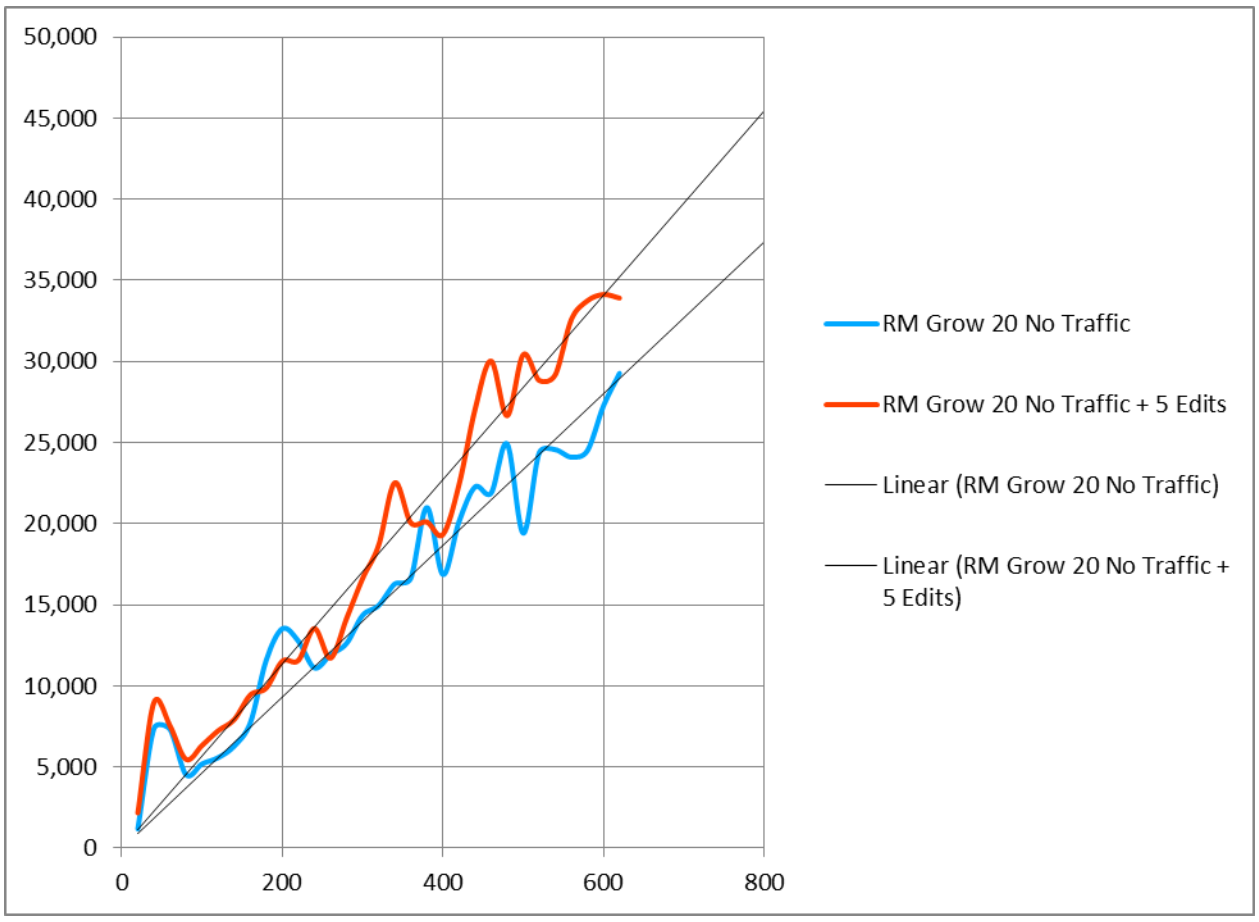


Presented differently here are the numbers above as a percentage of one slot averaged across the number of stops.



➲ **Note—** NumSlots = 0 represents traffic is turned off.

One-way with Map Edits



As is shown the use of map edits does impact the performance of the matrix. The impact will vary somewhat depending on the location and number of edits created.

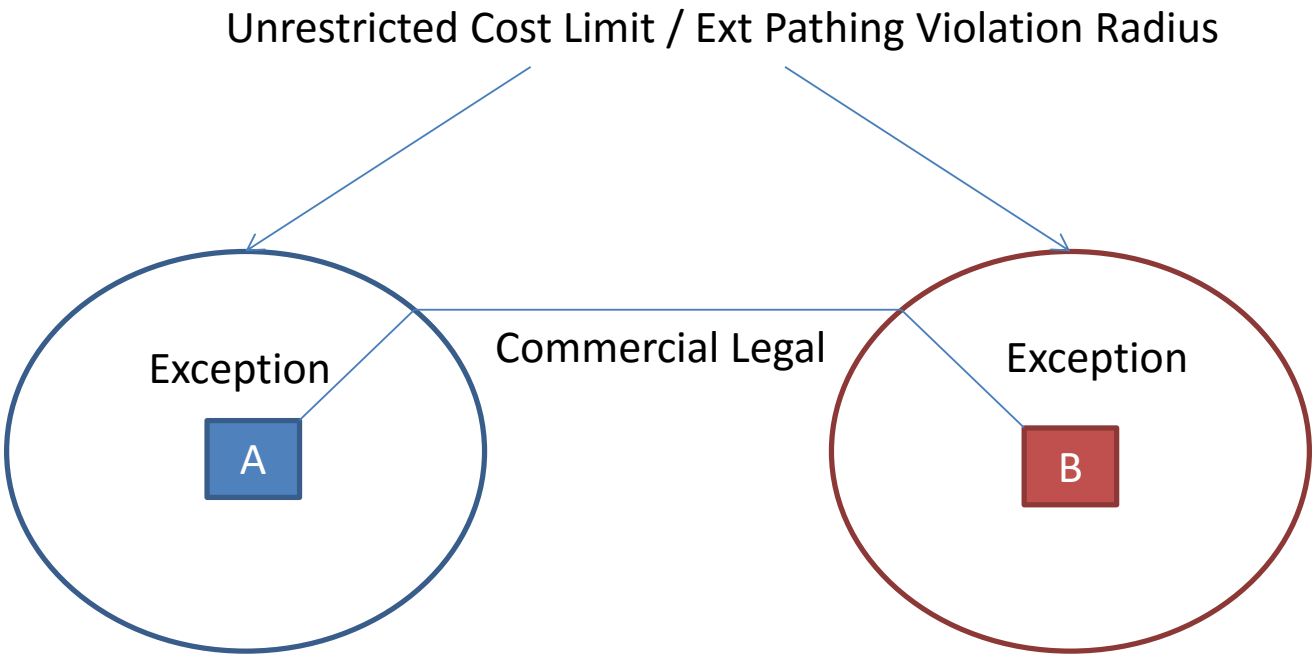
Appendix M: Unrestricted Cost Limit

Although the user interface presents the configuration as "per Mile" or "per KM", this setting is actually in seconds. The value is converted using approximately 35 miles per hour by dividing the distance (in meters) by 15.6464 to return the duration in seconds. Only the integer portion is used for calculation.

The image below details how this works. Use the following list as a legend.

- Origin is A
- Destination is B

- Blue circle represents the approximate allowed "legal" violation radius *near* the **origin**
- Blue circle represents the approximate allowed "legal" violation radius *near* the **destination**
- The system will always follow a commercially "legal" path between the blue and red circles
- The system **may** following a commercially "illegal" path between A and the blue circle (origin)
- The system **may** following a commercially "illegal" path between B and the red circle (destination)



Appendix N: Debugging Requests

Common Requests from Descartes Route Planner

Descartes Route Planner utilizes several different "request" types. The LNOS Pathing Service request is a simple text command with a [command] directive at the start. The requests are generated by RMPI via the External pather, for details on how to capture these messages, see *Appendix A: RMPI configuration settings for External Pather, Debugging Settings*.

The following [command] directives are currently utilized by Descartes Route Planner:

- **[GetP2PCost]**
  - Used to calculate the distance/time/traffic details for a single origin/destination. (e.g. suggest, optimize all, etc)
  - Inputs include:
    - Origin: Lat1/Lon1
    - Destination: Lat2/Lon2
    - Routing parameter: e.g. Speeds[1], AllowTollRoads, etc
    - Truck details: e.g. HasTruck, Weight,Height,Length, etc
  - Output:
    - Distance (meters), Time (Sec), and other details
  - Test Page:
    - http://localhost/pathingservice/tests/testgetp2pcost.aspx
  - Example:
    - [Success]
    - MapEditTimeStamp=0001-01-01T00:00:00.0000000
    - 6010.00,856,0.00,3
    - Where:
      - 6010.00 = Distance (Meters)
      - 856 = Time (Seconds)
- **[GetDistanceMatrix]**
  - Used to calculate the distance/time/traffic details for multiple origin/destination. (e.g. suggest, optimize all, etc)
  - Inputs include:
    - Number of locations: NumStops
    - Location details: Lat#/Lon#, where # = 1 … NumStops
    - Routing parameter: e.g. Speeds[1], AllowTollRoads, etc
    - Truck details: e.g. HasTruck, Weight,Height,Length, etc

- Output:
  - Distance (meters), Time, and other details
  - Each line represents a cell in the matrix. The results are provided 1 row at a time.
    - Result 1 = row 0, col 0,
    - Result 2 = row 0, col 1, etc.
    - Result n = row 1, col 0,
    - Result n+1 = row 1, col 1, etc.
- Text Page:
  - http://localhost/PathingService/tests/testGetDistanceMatrix##.aspx
  - Where ## is 10,20,50,100, etc.
- Example matrix:

DM	A	B	C	D
A	AA	AB	AC	AD
B	BA	BB	BC	BD
C	CA	CB	CC	CD
D	DA	DB	DC	DD

  - A = Lat1/Lon1
  - B = Lat2/Lon2
  - C = Lat3/Lon3
  - D = Lat4/Lon4
  - Input:
    - [GetDistanceMatrix]
    - CheckCachedValuesFirst=false
    - NumStops=4
    - StartDate=2012-03-01T08:30:00
    - Lat1=41.805008757413
    - Lon1=-71.5139541932317
    - Lat2=41.8305433887567
    - Lon2=-71.8787775563465
    - Lat3=41.8175950239655
    - Lon3=-71.8858183554303
    - Lat4=41.8163933942552
    - Lon4=-71.8838388850593
  - Output:
    - [Success]

CONFIDENTIAL AND PROPRIETARY TO THE DESCARTES SYSTEMS GROUP INC. AND ITS AFFILIATES

- Asynchronous=False
- MapEditTimeStamp=0001-01-01T00:00:00.0000000
- StopCount=4
- 0.00,0,0.00,3,0 ← *AA* , ***first row, O=D, initial cell***
- 41967.00,2187,0.00,3,0 ← *AB*
- 38466.00,2205,0.00,3,0 ← *AC*
- 38234.00,2176,0.00,3,0 ← *AD*
- 48185.00,2613,0.00,3,0 ← *BA*, ***second row, != AB***
- 0.00,0,0.00,3,0 ← *BB* ***O=D***
- 45342.00,2147,0.00,3,0 ← *BC*
- …
- 33666.00,1030,0.00,3,0 ← DC, **fourth row**
- 0.00,0,0.00,3,0 ← DD **O=D, final cell**
-
- If traffic is enabled, each entry will have one or more traffic details as well:
  - 41967.00,2187,0.00,3,0 **A924B32/B941057/…**
  - Each traffic block is separated by a / and represents 4 separate speed factors which are compressed.

- **[Path]**
  - Used to retrieve the road segments and details for one or more destinations. (e.g. show along the road)
  - Inputs include:
    - Number of locations: NumStops
    - Location details: Lat#/Lon#, where # = 1 … NumStops
    - Routing parameter: e.g. Speeds[1], AllowTollRoads, etc
    - Truck details: e.g. HasTruck, Weight,Height,Length, etc
  - Output:
    - Lat, Long, Distance (meters), Time (sec)
  - Text Page:
    - http://localhost/PathingService/tests/testPath0.aspx
  - Example Output:
    - [Success]
    - PointCount=191 ← segments in path
    - 44.8875604,-93.3653342,0,0

CONFIDENTIAL AND PROPRIETARY TO THE DESCARTES SYSTEMS GROUP INC. AND ITS AFFILIATES

        - 44.88751,-93.36532,5.6,0.2
        - 44.88756,-93.36502,23.9,0.9
        - …
- **[Directions]**
    - Used to retrieve road segments and instructions (english).
    - Inputs include:
        - Number of locations: NumStops
        - Location details: Lat#/Lon#, where # = 1 … NumStops
        - Routing parameter: e.g. Speeds[1], AllowTollRoads, etc
        - Truck details: e.g. HasTruck, Weight,Height,Length, etc
    - Output:
        - Lat, Long, Distance (meters), Time, description
    - Text Page:
        - http://localhost/PathingService/tests/testDirections1.aspx
    - Example Output:
        - [Success]
        - DirCount=6
        - 44.98447,-93.34448,275.00,39,Go west on Olson Memorial Hwy
        - 44.9846,-93.34801,507.00,54,Take the Mn-100 Ramp to the right
        - 44.98895,-93.34935,3061.00,329,Merge to the left on Mn-100 N
        - 45.01679,-93.34926,477.00,60,Exit to the right on 36th Ave N Ramp
        - 45.02098,-93.34794,459.00,54,Bear Left on Mn-100 N Ramp
        - 45.02512,-93.34746,1962.00,211,Merge to the left on Mn-100 N
        - …

Deconstructing the request:

- The following table details a typical LNOS Pathing Service command request.
- It is composed of a command followed by a series of Key/Value elements in the form Key=Value.

Element	Explanation
[GetDistanceMatrix]	Command to execute
CheckCachedValuesFirst=false	whether to use existing cache, when true only new cells are calculated

NumStops=10	max lat/long to use, lat#/lon# must be >= to this number
StartDate=2012-03-01T08:30:00	timestamp for pathing, determines which edits will be used
Lat1=41.805008757413	Latitude of stop #, if > NumStops ignored, more than 5 decimals are ignored
Lon1=-71.5139541932317	Longitude of stop #, if > NumStops ignored, more than 5 decimals are ignored
Lat2=41.8305433887567	Latitude of stop #, if > NumStops ignored, more than 5 decimals are ignored
Lon2=-71.8787775563465	Longitude of stop #, if > NumStops ignored, more than 5 decimals are ignored
Lat3=41.8175950239655	Latitude of stop #, if > NumStops ignored, more than 5 decimals are ignored
…	etc
AllowUTurns=false	enable/disable u turns
UseTollRoads=false	enable/disable toll roads
StrategyType=fastest	fastest or slowest (always fastest)
Speeds[1]=35	routing parameter speed category (mps)
Speeds[2]=29	routing parameter speed category (mps)
Speeds[3]=24	routing parameter speed category (mps)
Speeds[4]=18	routing parameter speed category (mps)
Speeds[5]=13	routing parameter speed category (mps)
Speeds[6]=9	routing parameter speed category (mps)
Speeds[7]=2	routing parameter speed category (mps)
Speeds[8]=1	routing parameter speed category (mps)
SpeedUnits=MetersPerSecond	Speed unit (RP always uses MPS)
HasVehicle=true	if true commercial routing fields will be evaluated, truck must also be StraightTruck or ArticulatedTruck
AllHazmat=false	Hazmat
Hazardous=false	Hazmat
Hazmat_1=false	Hazmat, not currently implemented
Hazmat_2=false	Hazmat, not currently implemented
Hazmat_3=false	Hazmat, not currently implemented
Hazmat_4=false	Hazmat, not currently implemented
Hazmat_5=false	Hazmat, not currently implemented
Hazmat_6=false	Hazmat, not currently implemented

Hazmat_7=false	Hazmat, not currently implemented
Hazmat_8=false	Hazmat, not currently implemented
Hazmat_9=false	Hazmat, not currently implemented
Hazmat_I=false	Hazmat, not currently implemented
GrossWeight=30500	Weight in KG, includes onboard if enabled
Height=411	Height in CM
Length=1615	Length in CM
Width=250	Width in CM
TruckType=PassengerVehicle	Truck Type: PassengerVehicle,StraightTruck or ArticulatedTruck (Tractor/Trailer)

Debugging No Paths

When a no path situation is identified, this can result from several probable causes:

- **Cause: Origin/Destination is invalid**
  - In some cases, the origin or destination may be invalid – for example, in the middle of a lake, in the middle of the ocean or in a disconnected section of the map (e.g. US to UK).
  - **Investigation: View on map**
    - Usually this is most easily investigated by placing the two stops on a map and visually comparing them to the road network.
  - **Resolution: Re-geocode origin and/or destination**
    - The only option is to ensure both the origin and destination is valid by modifying the latitude and longitude of one or both locations. Users can modify the latitude and longitude values by right-clicking the stop, selecting **View Geocode** and making the necessary adjustments in the **Location Geocode Result** window.
- **Cause: Routing Parameter restrictions (No Toll, No U-Turn, No Ferries, etc)**
  - Routing parameters can restrict which roads are eligible for use during pathing. I In some cases, these roads are required to reach the destination.
  - **Investigation: Reduce restrictions / alternate routing parameter**
    - Try using an alternative routing parameter on the resource that does not have these restrictions. These restrictions are not currently visible on the map
    - Calculate Distance may not work because it always uses DEFAULTS.

- When doing calculating distance, keep in mind that the origin and destination may not be exact.

➲ **Note—** When U-turns are disabled the side of the road matters as it enforces side of street pathing.

- **Resolution: Use alternate routing parameter when required or update original routing parameter**
  - If this is the cause, the only option is to temporarily assign a "NORESTRICTIONS" routing parameter for the day or to make the change to the routing parameter permanently.

➲ **Note—** Any changes to routing parameters immediately invalidates the cache built using that routing parameter.

- **Cause: Commercial Restrictions**
  - In some cases, the origin or destination is in a commercially restricted area or the roads connecting them are blocked by commercial restrictions.
  - **Investigation: Reduce restrictions / calculate distance**
    - As commercial road restrictions are not currently visible on the map, the only option is to try pathing without restrictions. One way to do this is using the calculate distance tool, this uses the "DEFAULTS" routing parameter (or internal defaults it parameter does not exist) **without commercial restrictions**. This approach can cause confusion with routing parameter restrictions if the resource is not using DEFAULTS.
    - Another method is to edit the resource and set the Weight/Height/Length/Width parameters to 0.

➲ **Note—** If MeasureToWeight schedule settings are configured, this weight will not be removed.

- If this is identified to be related to commercial restriction, it must be determined if the issue can be resolved using the violation radius:
  - Copy the schedule (with commercial restrictions)
  - Create a new schedule group
  - Assign the new schedule group to the copy
  - Edit the schedule group and increase the radius to 10 miles (~16 km)
  - Refresh route
  - If problem is not resolved, increase radius to 15 miles (~23 km)
  - Refresh route

- If problem is not resolved, review restrictions, it may not be possible to visit the customer.
- If problem is resolved, reduce the radius to the minimum amount required to service the customer + a small buffer. The default radius, if not configured, is ~3 miles (~4.5km). Start at 3 miles and increase in 2 to 3 mile increments until optimal route is identified.

➲ **Note—** Large values for this radius can **significantly** slow down optimization.

- **Resolution: Update Schedule Group Violation Radius / use different asset**
  - If the problem can be resolved using the violation radius, use the value identified in configuration and modify the original schedule group.
  - If it cannot, there is likely a commercial restriction that is significant and usually implies it is not safe to visit the customer. Use an alternate route or accept the no path solution.

- **Cause: Map Edits**
  - Sometimes a map edit can cause a path to no longer be viable.
  - **Investigation: Test for impact of map edits**
    - Create a copy of the schedule and assign to a new/different schedule group (without edits).
    - Perform show along the road, review the path, if path is available, compare it to the map edits defined for the original schedule group.

    ➲ **Note—** Map edits are directional, so it is possible that the edit works from A→B but not going from B→A

  - **Resolution: Update Map Edits**
    - If map edits are the problem, they should either be edited or deleted if they are not valid. Otherwise it may be necessary to use a different route or to accept the no path scenario.

Appendix O: Dividing up LPS servers

In the case of more than one LNOS Pathing Service, it is recommended to divide the schedule groups according to the relative amount of cache to avoid overwhelming an individual server. The following SQL query can be used to allocate the Schedule Groups to LNOS Pathing Service Servers based given "n" pathing servers.

```
SELECT
```

```
--UPDATE SG SET -- run from this line forward to update
ExternalPatherURL = 'http://' + CASE WHEN Details2.Rank%2=0 THEN 'LPS0' ELSE
'LPS1' END
                    + '/PathingService/lnospathingservice.aspx',
[Rank], SG.ScheduleGroupKey, -- comment this line out to update!
[TotalLocations], LPSServer  -- comment this line out to update!
FROM   FWScheduleGroup SG
JOIN (SELECT *,
             Dense_rank()
               OVER(
                 ORDER BY TotalLocations DESC) [Rank]
      FROM   (SELECT SG.ScheduleGroupId,
                     SG.ScheduleGroupKey,
                     Upper(Substring(SG.ExternalPatherURL, 8,
                         Patindex('%/P%', SG.ExternalPatherURL) - 8)) [LPSServer],
                     Count(DISTINCT S1.LocationId)  [TotalLocations]
              FROM   FWSchedule S
                     LEFT JOIN FWScheduleGroup SG
                            ON SG.ScheduleGroupID = S.ScheduleGroupId
                     JOIN FWStop S1
                       ON S.ScheduleID = S1.ScheduleID
                     JOIN FWActivityWIndow AW
                       ON S1.ActivityID = AW.ActivityID
              WHERE  ( S.ScheduleKey LIKE '____[_]Planning'
                        OR S.ScheduleKey LIKE '____[_]Review'
                        OR S.ScheduleKey LIKE '____[_]Dispatch'
                        OR S.ScheduleKey LIKE '____[_]Archive' )
                     AND AW.EarliestDate >= Getdate() - 14
              GROUP  BY SG.ScheduleGroupId,
                        SG.ScheduleGroupKey,
                        SG.ExternalPatherURL)Details) Details2
  ON SG.ScheduleGroupID = Details2.ScheduleGroupID
```

➲ **Note—** It may be necessary to adjust the ScheduleKey / Date Filters depending on the convention used and data available.

Appendix P: Cache Save Scheduling

The mapeditor schedule is defined using the cron trigger format.

```
┌──────────── seconds (0 - 59)
│ ┌────────── min (0 - 59)
│ │ ┌──────── hour (0 - 23)
│ │ │ ┌────── day of month (1 - 31)
│ │ │ │ ┌──── month (1 - 12)
│ │ │ │ │ ┌── day of week (0 - 6) (0 to 6 are Sunday to
│ │ │ │ │ │                Saturday, or use names; 7 is also Sunday)
* * * * * *
```

Examples:

Key	Description
0 1 03 ? * MON-SAT	Will schedule the job to run from Monday-Saturday at 3:01:00 AM
0 1 03,11,15 ? * *	The job to run from every day at 3:01:00 AM, 11:01:00 AM and 3:01:00 PM.

Fields and Special Characters

Field	Required	Allowed values	Allowed special characters
Seconds	Yes	0-59	* , -
Minutes	Yes	0-59	* , -
Hours	Yes	0-23	* , -
Day of month	Yes	1-31	* , - ?
Month	Yes	1-12 or JAN-DEC	* , -
Day of week	Yes	0-6 or SUN-SAT	* , - ?

The special characters are defined as below:

Star (*)

Stars are used to indicate the entire range of allowed values.

Comma (,)

Commas are used to separate items of a list. For example, using "MON,WED,FRI" in the 6th field (day of week) means Mondays, Wednesdays and Fridays.

Hyphen (-)

Hyphens define ranges. For example, 2000-2010 indicates every year between 2000 and 2010, inclusive.

Question Mark (?)

Used to mark the day of month or day of week as "not applicable" and will be ignored by the scheduler, generally equivalent to star.

See https://en.wikipedia.org/wiki/Cron for further examples of how this can be configured.

Appendix Q: Cache Management During Upgrades

If upgrading the LNOS Pathing Service or RMWS the cache that has been previously built will be invalidated. For most customers the there is insufficient time to rebuild the cache during live operations. To help with this the LNOS Pathing Service has been enhanced to use the previous cache until such time a separately primed cache is ready to be replaced.

It is important to understand that the cache itself does not know which entries are new/old so the results may be a blend of the two RMWS datasets.

The steps to use the old cache are:

1. If upgrading, get the *QueryForTraffic* and *StoredTrafficSlots* property values from the production LNOS Pathing Service configuration file.
2. If using traffic, use the following steps:
   - **a** Stop the old LNOS Pathing Service before installing the new LNOS Pathing Service.
   - **b** Move the .lps & .dat files to a temporary folder, one that would not be touched by the LNOS Pathing Service installer.
   - **c** Install the new LNOS Pathing Service.
   - **d** Stop the LNOS Pathing Service web service in IIS by stopping the application pool (e.g. PathingService).
   - **e** Make certain that *QueryForTraffic* and *StoredTrafficSlots* information matches with previous version (from step 1 above).
   - **f** Move the .lps & .dat files from the temporary folder to the GeneratedFiles folder.
   - **g** Start the upgraded LNOS Pathing Service.
3. If NOT using traffic use the following steps:
   - **a** If upgrading, upgrade the production LNOS Pathing Service first. (It is important that **this is done before RMWS** servers are upgraded. Not doing this will invalidate the server caches).
   - **b** Stop the LNOS Pathing Service web service in IIS by stopping the application pool (e.g. PathingService).
4. With 16.10.1 release, there is a new property in the LNOS Pathing Service config file - *UsePreviousVersionCache* with the default value of **true** and this **MUST** be set in the LNOS Pathing Service before upgrading the RMWS servers. If not upgrading, this value must be manually configured.
5. And; Update the *QueryForTraffic* and *StoredTrafficSlots* values from step 1 to the upgraded LNOS Pathing Service configuration file. It is critical that

the *QueryForTraffic* and *StoredTrafficSlots* information is the same after upgrade.

6 Now, go ahead and upgrade the RMWS servers that production LNOS Pathing Service uses, and start the production LNOS Pathing Service. The upgraded production LNOS Pathing Service will continue to use existing server caches, add any new cache values to its cache in the production environment. This is true if the stored schema information has not changed (which has not since LNOS Pathing Service started supporting traffic patterns over an year back) and the traffic information after upgrade matches the pre-upgrade settings.

7 If upgrading, upgrade the "priming" LNOS Pathing Service with the new LNOS Pathing Service version.

8 Set the *UsePreviousVersionCache* to 'false' on this LNOS Pathing Service.

9 Set the desired *QueryForTraffic* and *StoredTrafficSlots* values.

10 Upgrade its RMWS servers (which may be the same ones, if primed during off-peak).

11 Start Microsoft® PowerShell script to prime the caches on this priming LNOS Pathing Service (available from engineering team).

12 When there is a reasonable coverage of the caches in the priming LNOS Pathing Service for ALL organizations and schedule groups catered by the production LNOS Pathing Service, stop this service and the production LNOS Pathing Service. Copy all the server cache information from priming LNOS Pathing Service to the production LNOS Pathing Service. These are the .lps and .dat files in the GeneratedFiles folder.

13 If the QueryForTraffic and StoredTrafficSlots information in step 4a was set, then make certain that these values are copied over to the production LNOS Pathing Service configuration.

14 Once these files are copied over, set the *UsePreviousVersionCache* to **false** on the production LNOS Pathing Service, and start the service. This will trigger an invalidation of extPather caches (because the LNOS Pathing Service is now passing the correct version information from RMWS) and they will get the all new version values.