DAT® Connexion™ Software Development Kit (SDK) Developer's Guide — C# Version 2.3

DAT Solutions
Beaverton, OR

September 23, 2014

Contents

1 Welcome

The C# SDK for DAT® Connexion™ is a set of documentation, specification (XML schema and WSDL) and sample code demonstrating the use of Connexion, TransCore's SOAP-based web-services API that provides programmatic access to the following TransCore data and services:

- Post and search for loads and trucks in the DAT load board
- Create alarms to automatically send back information for future loads or trucks matching specified criteria
- Look up DAT Truckload Rate Index spot and contract rate information for a single lane or batch of lanes, plus 13 month lane history
- Retrieve DAT CarrierWatch carrier authority, safety, DOT insurance and private insurance certificate information Connexion calls send data to, and retrieve data from, DAT services in a standardized format that are easy to use.
- Receive data on carriers who have completed a profile and signed a contract using DAT Onboarding.

The C# SDK lets you develop impactful integrations of DAT data and services that provide a streamlined, high-productivity experience for your end users.

1.1 Requirements

To compile the sample C# code, use .NET version 4 or higher. The precompiled binaries have been tested on .NET 4.

1.2 Supported Tools

As a .NET developer, you will be using Visual Studio (the quick start guide refers to Visual Studio 2010) to create your project and `svcutil.exe` to create the proxy classes for the Connexion web service.

1.3 Development Process

Getting an application using Connexion up and running and in production — summary:

1. Acquire required information to access the Connexion test server. (See Section 4.)
2. Install SDK
3. (Optional) Configure and run the sample applications (See Section 5.3)

4. Develop code with Connexion calls for your application, test/debug. Please review the FAQ (Section 9) for some important Connexion guidelines. For support with any technical issues/questions, contact customer support at 800-547-5417 or Customer.Support@dat.com.
5. When development is complete, contact Customer Support to certify your application's integration.
    - Send a high level description of what your application does (with respect to integration with TransCore) and what Connexion calls it uses.
    - Run tests to exercise all your Connexion integration. For posting and searching, this should be at least 30 of each (but no more than 1000).
    - Inform Customer Support when your tests were run, and what percentage of projected load the tests were simulating
    - Incorporate any feedback provided by TransCore from analyzing your testing.
    - Briefly demo integration for Connexion Product Manager. When complete you'll receive an email documenting your certification.
6. Work with your account representative to acquire credentials for the Connexion production environment.

2 Connexion Services Summary

Connexion provides access to several types of services. This section provides summary information on each request, organized by category:

- Common — Requests related to all subsequent sections
- Asset Management — Posting and managing shipments and equipment.
- Search — Searching for shipments and equipment.
- Carrier Monitoring — Integration with TransCore's CarrierWatch carrier-monitoring offering.
- Rating — Integration with TransCore's Truckload Rate Index rating system.
- Alarm Match — Real time notification of current activity in the DAT system matching your criteria of interest.
- DAT Onboarding — Integration with automated carrier onboarding through DAT Onboarding.

Each request is linked to the detailed field-by-field documentation in Section 7.

2.1 Common

These requests are related to all of the other categories described below.

Login: Login to the Connexion system. A valid login ID and password must be provided before API calls will be serviced. The Login request returns a session token, which is conceptually analogous to a web session cookie, and must be returned with all subsequent 'real' operations.

Note: The client application should cache this session token and use it on all subsequent requests. The session token will remain valid for a 12 hours, a lifetime chosen with the intent of covering a full business day. Logging in separately for each 'real' request is not recommended or permitted. It adds considerable server load and latency, and can introduce a race condition in threaded or multi-process code

A user can have at most one active session at a time - a login will invalidate a user's existing session.

LookupCapabilities: Look up presence or absence of specific capabilities. Each user has the ability to perform certain operations within TransCore's system, depending on their active subscriptions. For example, only users with a premium CarrierWatch subscription are able to lookup insurance certificate information. A TMS system may want to alter its UI depending on the capabilities of the current user, so as to avoid presenting the user with functionality that will fail (e.g., removing a 'Search' tab for a user with a post-only subscription).

2.2 Asset Management

Requests related to posting, updating, and managing shipment and equipment postings.

NOTE: Clients must track their postings, and using the requests below, delete specific loads which are no longer available (e.g. the load was covered) or update the information for any loads which have changed. Implementations may not periodically simply delete all postings and re-post available trucks/loads. This will negatively impact system performance, and so is not allowed. As part of the certification process, DAT will check to ensure that all Freight Matching clients make use of the UpdateAsset request.

PostAsset: Post one or more assets (shipments or equipment) to TransCore's freight-matching system. Postings will be immediately searchable by all TransCore DAT users.

UpdateAsset: Update an existing shipment or equipment posting. Updated postings will be immediately searchable by all TransCore DAT users.

Note: Most elements in UpdateAssetRequest and its contained types are optional. If an element is omitted, no changes will be made to that portion of the asset (e.g., if the rate element is omitted, the asset's rate will be unchanged). If an element is included in the update (even if empty), the appropriate change will be attempted on the asset. For instance, if the rate element is included, but empty, the asset's rate will be deleted. Note that in some cases, business rules may preclude such updates (e.g., comments can be updated, but in most cases, attempting to remove a comment will result in an error).

DeleteAsset: Deletes one or more posted assets.

To "delete" means the asset is no longer available for freight matching. If the asset is displayed on the Truck Stop Network, it will be removed from there as well.

Clients are *strongly* encouraged to delete postings that are no longer available. This will prevent unwanted callbacks to the client, and help ensure that searchers see accurate match results.

LookupAsset: Lookup currently-booked assets. Note that access limits are placed on asset lookup. In general, a user can only lookup their own assets and those belonging to others within their sharing group (if any).

Lookup requests are generally useful when a user wants to view or update their assets. If you want to see the assets of other users outside your own company, see CreateSearchRequest and the types referenced therein.

2.3 Search

Requests related to freight matching searches. Searching allows the user to find shipments and equipment currently available in the DAT system.

As a reminder, all applications that support searching for loads or trucks via Connexion must also support posting loads and/or trucks using Connexion, as appropriate for the customer.

Note: we provide 2 sets of login credentials for the test system, and we *strongly* recommend that you develop controlled tests for your search application, posting as one user and validating that your posting appears in searches executed as the other user.

CreateSearch: Search the TransCore DAT freight-matching system for shipments or equipment that meet a set of caller-specified criteria.

LookupSearchMatches: Retrieves additional matches not included in the response to a prior CreateSearchRequest.

LookupSearch: Lookup searches recently performed by the requester. The results can be used to perform additional searches.

"Recently" may depend on the requester's configuration and TransCore policy, but in general usually means at least within the last 24 hours.

DeleteSearch: Delete a search.

It is not a requirement that searches ever be explicitly deleted. Deletion is a convenience for applications that make heavy use of multi-search lookups (e.g., `LookupSearchOperation.queryAllMySearches`) so they can easily be excluded from lookups.

CountAssets: The CountAssets request is functionally identical to the CreateSearch request, with two important differences:

1. The response contains only a count of matching assets, and does not include details on any of the matches.

2. The search record is not stored, so you cannot subsequently lookup or delete the search (i.e., there are no analogues to LookupSearchRequest or DeleteSearchRequest).

CountAssetsRequest is expected to be used for research and analytic activities, rather than active freight-matching. High search volumes are normal, but the total search volume will generally be limited on a daily, weekly, or monthly basis by subscription levels. Please contact your salesperson if you need to increase your subscription level to raise your search quotas.

2.4 Carrier Monitoring

Requests related to carrier monitoring, including access to the insurance information and certificates maintained by TransCore's CarrierWatch product.

LookupCarrier: Lookup insurance and DOT information about a carrier from TransCore's CarrierWatch system.

MonitorInsurance: Request that a carrier in the TransCore CarrierWatch system be monitored for insurance information.

CreateCarrier: Request that a carrier not yet in the TransCore CarrierWatch system be added for the purposes of monitoring DOT and insurance information. TransCore already monitors DOT information on the vast majority of interstate carriers, so this request is primarily useful for intra-state carriers. If the requester is a premium CarrierWatch customer (with the ability to lookup private insurance information), TransCore will additionally begin monitoring private insurance information on the specified carrier (see also MonitorInsuranceRequest).

2.5 Rating

Requests related to spot- and contract-market rates, provided by TransCore's Truckload Rate Index suite of products.

LookupRate: Request spot-market and/or contract rate information for a lane, including estimated rate, range, and a numeric representation of TransCore's confidence in that estimate.

Each rate requested will be credited against the pool of lookups associated with the user's Truckload Rate Index subscription.

Note that multiple LookupRateOperations can be submitted in a single LookupRateRequest, in batches of up to 50. If you need to request more than 50 lanes, please submit batches sequentially.

LookupHistoricSpotRates: Requests spot rate estimates for a lane over the past 13 months. Each rate returned will be credited against the pool of lookups associated with the user's Truckload Rate Index Subscription.

LookupHistoricContractRates: Requests contract rate estimates for a lane and national average rates over the past 13 months. Each rate returned will be credited against the pool of lookups associated with the user's Truckload Rate Index Subscription.

LookupDoeFuelPrices: Lookup diesel fuel prices over the past 13 months, as reported by the Department of Energy.

2.6 Alarm Match

Requests related to Alarm Match, providing real-time notification of incoming shipment and equipment postings. For further information on Alarm Match, see Section 6.

CreateAlarm: Create an alarm search for an existing truck or load posting. An alarm waits for assets (loads or trucks) that match the associated posting and when one is found, it notifies the user through a web service call or via one of TransCore's client applications (see the 'notifyAllApplications' option). Alarms become immediately active in the freight-matching system when created. An alarm expires when it receives the maximum number of matches or at 6:00 PM Pacific Time. Alarms created after 6 PM expire at midnight.

UpdateAlarm: Updates the search criteria for an existing alarm.

LookupAlarm: Lookup information on particular alarm(s). Only those alarms accessible by the requester can be successfully selected.

DeleteAlarm: Delete an alarm.

To "delete" means the alarm is no longer available for freight matching activities and is no longer retrieved on multi-alarm lookups.

If the alarm is currently active (i.e., not expired) then it will be automatically cancelled (and an alarm termination notice will be issued).

It is not a requirement that alarm ever be explicitly deleted. Deletion is a convenience for applications that make heavy use of multi-alarm lookups (e.g., `LookupAlarmOperation.LookupAllMyAlarms`).

UpdateAlarmUrl: Specifies the web service endpoint URL that alarm notifications will be delivered to. If a single endpoint is shared for an entire company or office, TransCore customer support can configure that URL. This request specifies a unique URL for the requester, and is appropriate when the endpoint is subject to frequent changes or differs between users.

LookupAlarmUrl: Lookup the web service endpoint URL that alarm notifications will be delivered to. This request can be used to verify the URL that TransCore has on file for the requester.

2.7 DAT Onboarding

Requests related to retrieval of data on carriers who have completed a profile and signed contracts with the customer using the DAT Onboarding product.

Clients should check for events using the LookupDobEvents request on a set interval, such as every 5 minutes, in order to retrieve data soon after it is available. Events will include the carriers for which data is available so that data can be retrieved by using LookupDobCarriers request.

Note: We encourage developers implementing DAT Onboarding to consult the detailed description of event handling test procedures in Section 9.23

LookupDobEvents: Lookup DAT Onboarding events received since a given date/time. Events include changes to carrier profile, carrier W9 status, carrier insurance agents, carrier attachments and contracts signed by a carrier.

This request is only supported through the secure (HTTPS) endpoint. Requests through the insecure endpoint will return an error.

LookupDobCarriers: Lookup information on a DAT Onboarding carrier. All data available for the carrier is returned.

This request is only supported through the secure (HTTPS) endpoint. Requests through the insecure endpoint will return an error.

LookupDobSignedCarriers: Lookup all carriers that have signed contracts through DAT Onboarding with the user. The intended usage is to synchronize the client's system with all relevant data available in the DAT Onboarding system. All carriers which have ever signed a contract with the customer and the contracts they have signed will be returned.

This request is only supported through the secure (HTTPS) endpoint. Requests through the insecure endpoint will return an error.

3 SDK Contents

The Connexion C# SDK is composed of source code, libraries, packages, schemas, help files, sample programs, and documentation. This document and other documentation are in the package root folder. The remainder of the SDK is organized as follows:

`bin`:

This folder contains the pre-built binaries for all sample applications and command-line scripts to run them from a Unix shell or Windows command-line.

`xml-schema`:

This folder contains the schema and WSDL files that describe the Connexion SOAP API which are used to build the SDK. Detailed documentation for the fields of each request is included in the file `TfmiFreightMatching.xsd`. `TcoreCarrierWatchTypes.xsd` contains similar descriptions of the fields returned from carrier lookups. Most of that documentation is also included in Section 7.

`source`:

This folder contains the source code for the sample application that demonstrates how to implement the various Connexion API calls and a listener that receives and processes alarm match notifications from Connexion.

4 Getting Started

Before you can begin making Connexion calls from your application you must be provided with the URL for the Connexion test system and a pair of test accounts (login ID and password). This should have been provided to you by a TransCore customer support specialist. If you do not have either of these please contact customer support at Customer.Support@dat.com and information will be provided to you.

4.1 Installing the SDK and Configuring your Environment

Unzip/Untar the SDK onto a local drive of your development workstation. Prior to running the sample applications, configure `login.properties` per the instructions in Section 5.3.

5 Sample Applications

The sample applications included in the SDK demonstrate common Connexion use cases, and are intended to provide a starting point for code integrating Connexion into your own application.

5.1 List of sample applications

5.1.1 Login

Demonstrates a simple user login request.

5.1.2 Post

Demonstrates posting a shipment.

5.1.3 Search

Demonstrates shipment search. To ensure that the search will return matches, the application first posts an asset to a known location which will then match the search it performs.

5.1.4 LookupCarrier

Demonstrates a simple carrier lookup by DOT number, MC number, and TransCore UserId.

Note that the TransCore UserId of the poster is included with freight matching search results, and the carrier information can be referenced by that UserId.

5.1.5 LookupRates

Demonstrates single-lane and batch rate lookups using TransCore's Truckload Rate Index service. Demonstrates spot and contract rate lookups.

5.1.6 Alarm

This application demonstrates a simple use of AlarmMatch. AlarmMatch is implemented by means of the .NET framework's System.Net.HttpListener class and System.Threading.Tasks facilities. The flow of control is:

1. Begin sessions for two distinct user accounts.
2. For the first account, call the Connexion service to update the URL at which the sample application will receive alarm match notifications.
3. Also with the first account, post a truck with an associated alarm.
4. On a background thread, establish a listener for alarm match notifications.
5. Block the main thread until the listener indicates that it is active, then (still on the main thread) use the second user account to post a load that will trigger an alarm match.
6. Block the main thread again until the listener receives an alarm. (The background thread will now be released by the application and reclaimed by the .NET runtime.)
7. Examine the SOAP packet received by the listener and write the body and certain header values into the application's output file.

Note that the above steps only describe the intended behavior; various timeout, fault, or login failures are possible and will be logged. For example, supplying two identical user accounts will log an error in step 1 and terminate the flow of control.

5.1.7 LookupDobCarriers

Demonstrates DAT on-boarding carrier lookup by carrier id.

5.1.8 LookupDobEvent

Demonstrates looking up the details of a DOB on-boarding event. An event represents a change to a carrier profile, update of a carrier's W9 status, contract status changes, etc. Please see Section 9.23 for details on using the TransCore test environment to develop an integration with DAT Onboarding.

5.1.9 LookupSignedCarriers

Demonstrates looking up all carriers that have signed contracts through DAT Onboarding with the user.

5.2 Implementation Notes

The main features are Login, Post, Search, LookupCarrier, LookupRates, and Alarm. These features are itemized in the enumeration listed in `Infrastructure/Feature.cs`. When the application runs, it parses its command-line input to find one of the values of this enum. This is how the program decides which feature to perform. Each feature is implemented by a subclass of the `Program` abstract class. For example, `Login` is implemented in `Login.cs` in the Source directory.

The typical flow of control is:

- The `Program` class's `Main()` method receives the command-line arguments as an array of strings.
- The `CommandLineInput` class is responsible for parsing and validating those arguments.
- The first argument specifies the feature to perform. It must correspond to a value of the `Feature` enum.
- The program redirects the console's output stream to a text file. By default, that output file is `C:\ConnexionTest.txt`, but an optional command-line argument prefixed with `-o:` will override the default.

- The `ConfiguredProperties` class is responsible for reading the configuration file. By default, the configuration file is the `login.properties` file in the root directory of the SDK, but an optional command-line argument prefixed with `-p:` will override the default.

- If the command-line input parses and the configuration file loads, the command-line input will specify a feature to perform. The program selects a class that corresponds to the feature. All such classes are subclasses of the `Program` abstract class. For example, if the command-line argument is 1, the program finds that this corresponds to the `Feature.Post` enumeration value, so the program constructs an instance of the type named `Post`.

- The program calls methods on the `Program` subclass to validate and execute the implementation of the feature.

We intend this code to demonstrate basic features of the Connexion web service. The code is intentionally not designed to demonstrate ideal coding practices. For example, the exception handling is designed to help the demonstrations run to completion while writing explanatory text to the output file, so that a developer unfamiliar with the behavior of the web service could understand what the program is attempting and whether it is succeeding or failing. Another noteworthy example is that the proxy objects generated from the Connexion web service are located at the root of the global namespace. This was done for simplicity, not as a recommended practice. Code designed to be deployed to production would have a much different set of design goals.

We intend the code to be readable with very few annotations or in-line comments, using mainstream conventions in object-oriented design. The names of objects were chosen to reflect their intended roles, and methods and properties were named with readability in mind. When it was feasible to encapsulate and hide the details of things that are not relevant to the Connexion web service (notably, `CommandLineInput` and `ConfiguredProperties`, and the entire `DisplayHelpers` namespace), objects were created to handle those concerns. We expect that the `Program` class, its subclasses for each demonstrated feature, and the `SessionFacade` will contain most of the interesting code.

5.3 Configuring the Sample Applications

Prior to running the sample programs, edit the login properties file (`login.properties`) with the login information for the test accounts provided to you. The file is found in the SDK root directory.

- Replace the login IDs and passwords with those provided by TransCore.

- Replace the server URL with the test system URL provided by TransCore (if the default does not match).

- (optional — only necessary if you wish to utilize AlarmMatch functionality). Replace the alarm properties with the HTTP URL path you intend to expose your alarm server on.

5.4 Running the Sample Applications from the Command-line

1. Change directory (`cd`) to `<sdk.home>`
2. Enter any the following command to run the sample programs:

```
bin\login.bat
bin\post.bat
bin\search.bat
bin\lookup-carrier.bat
bin\lookup-rate.bat
bin\alarm-demo.bat
bin\dob-lookup-carriers.bat
bin\dob-lookup-event.bat
bin\dob-lookup-signed-carriers.bat
```

Note regarding the alarm match: the alarm match sample application starts an HTTP listener on a local TCP port to receive alarm notifications from TransCore. If your development machine is directly visible from the Internet, you probably need to do no further configuration: the sample application will determine your IP address and set that using the `UpdateAlarmUrl` request.

However, if your machine is not directly visible from the Internet (e.g., if you are behind a NAT firewall), you will need to open an Internet-visible TCP port and direct that traffic to your machine. For example, assuming you choose port 8080, you can forward incoming traffic on port 8080 from your firewall to your development machine, again on port 8080. You must then modify the login.properties file to specify the URL TransCore should post notifications to when running the alarm demo. e.g.:

```
host=myhost.example.com
path=/AlarmMatch
port=8080
```

5.5 Editing the Sample Freight-Matching Application in Visual Studio

`File` → `Open` → `Project/Solution`.

The project file is:
`<sdk.home>\source\QuickStart\QuickStart.csproj`.

6 Alarm Match

This section describes and discusses TransCore's *Alarm Match* technology, as implemented in Connexion.

First, a definition or two: An 'Alarm', in TransCore / DAT terminology is a forward-looking search. That is, an alarm encodes all the normal criteria for a standard search (origin, destination, radius, equipment type, limits, etc.). Instead of returning currently available shipments or equipment, the alarm lives in TransCore's system for some time and provides an asynchronous alert (an 'alarm match') whenever a matching asset becomes available.

You can create an alarm by using the CreateAlarm request or be including the **alarm** element in a PostAsset request. The alarm will remain active until one of the following conditions occurs.

- The alarm is terminated by a DeleteAlarm request.
- The underlying ('basis') asset is deleted or expires.
- The alarm returns its maximum number of matches (the default is 25, but can be configured in the AlarmSearchCriteria).
- The end of the business day, defined as 6:00 PM Pacific Time (if the alarm is created after 6:00 PM, it will instead live until midnight Pacific).

Connexion returns alarm matches via a SOAP web service call. To receive those matches, you must expose a web service implementing the interface defined in `TfmiAlarmMatch.wsdl`. Alarms are inherently real-time asynchronous events and are not batched, so each match will be sent in a separate HTTP transaction. A single alarm covering a broad search area may return multiple matches per second, so please ensure your service is capable of handling the expected load.

For suggestions on implementing alarm match in a browser-based application, see Section 9.18.

7 Detailed Request Documentation

Required fields are in **bold**. In some cases, only one of a set of choices is required; these sets are prefaced by '**One of:**'. Fields marked as in bold in response documentation are required by the XML Schema and will always be present. Fields marked as optional (i.e., non-bold) may be omitted in some cases, as described in the accompanying field documentation.

7.1 Common

These requests are related to all of the other categories described below.

7.1.1 Login

Login to the Connexion system. A valid login ID and password must be provided before API calls will be serviced. The Login request returns a session token, which is conceptually analogous to a web session cookie, and must be returned with all subsequent 'real' operations.

Note: The client application should cache this session token and use it on all subsequent requests. The session token will remain valid for a 12 hours, a lifetime chosen with the intent of covering a full business day. Logging in separately for each 'real' request is not recommended or permitted. It adds considerable server load and latency, and can introduce a race condition in threaded or multi-process code

A user can have at most one active session at a time - a login will invalidate a user's existing session.

Request Fields

loginId (string)	Login ID assigned by TransCore. Length: 4 – 16
password (string)	Password (set by TransCore account representative or in the 3Sixty Power application). Length: 4 – 30
thirdPartyId (string)	This free-form identifier will be used to identify the application in which individual postings, searches, etc. originated, and will aid in debugging and user support issues. We recommend that each application accessing Connexion chooses its own identifier. e.g. "XYZLogisticsManagement", "ABCTMS", etc. Length: 0 – 32

apiVersion (int)	Specifies support for a particular Connexion API version. By default, API version 1 is assumed (equivalent to Connexion releases in 2012 and earlier). Currently supported versions: 1, 2 If apiVersion $\geq$ 2, Connexion will support new equipment types and search classes and new search by equipment type(s) functionality (released in other DAT applications in Nov 2012). Note that this may return postings, search results and alarm matches and equipment types and classes unknown to earlier client applications, so only applications which have tested and verified their support for new equipment types and classes should specify an apiVersion of 2 or higher.

Response Fields	
token	The session token for this session.
primary (base64Binary)	
secondary (base64Binary)	
expiration (dateTime)	
expiration (dateTime)	The date and time when the session token will expire. This is provided so that prior to making a CSB request, a client can easily check to see if its token has expired. Valid Values: 2003-01-01T00:00:00.000Z –

7.1.2 LookupCapabilities

Look up presence or absence of specific capabilities. Each user has the ability to perform certain operations within TransCore's system, depending on their active subscriptions. For example, only users with a premium CarrierWatch subscription are able to lookup insurance certificate information. A TMS system may want to alter its UI depending on the capabilities of the current user, so as to avoid presenting the user with functionality that will fail (e.g., removing a 'Search' tab for a user with

a post-only subscription).

Request Fields	
capability	Capability or capabilities of interest. A boolean result will be returned for each capability requested, in the same order as specified in the request. Valid Values: `Manage Postings`, `Search`, `Alarm Match`, `Lookup Spot Rates`, `Lookup Contract Rates`, `Lookup Basic Carrier Info`, `Lookup Premium Carrier Info`

Response Fields	
capabilityPresent (boolean)	Result of the capability query, in the same order as specified in the request.

7.2 Asset Management

Requests related to posting, updating, and managing shipment and equipment postings.

NOTE: Clients must track their postings, and using the requests below, delete specific loads which are no longer available (e.g. the load was covered) or update the information for any loads which have changed. Implementations may not periodically simply delete all postings and re-post available trucks/loads. This will negatively impact system performance, and so is not allowed. As part of the certification process, DAT will check to ensure that all Freight Matching clients make use of the UpdateAsset request.

7.2.1 PostAsset

Post one or more assets (shipments or equipment) to TransCore's freight-matching system. Postings will be immediately searchable by all TransCore DAT users.

Request Fields	
One of:	
shipment	
equipmentType	The type of equipment that is needed to haul this shipment. See EquipmentType

origin	The origination point of this shipment. See Place
destination	The destination point of this shipment. See Place
truckStops	All the information needed to advertise this shipment on the Truck Stop network. If omitted, this shipment will not be advertised on the Truck Stop Network.
One of:	
truckStopIds	
ids (int)	A list of TransCore/DAT identifiers for the truckstops the shipment should be displayed at. Valid Values: 0 – 9999
closest	Select the truckstops closest to the shipment origin. The number of selected locations is determined by the poster's subscription and contract.
alternateClosest	Select the truckstops closest to an alternate origin point. This option is typically used to display the shipment on truckstops in the desired lane. The number of selected locations is determined by the poster's subscription and contract.
alternateOrigin	Origin point. See Place
enhancements	Optional truck stop load monitor video enhancements for this shipment's advertisement. Valid Values: `Flash`, `Highlight`

posterDisplayName (string)	The short name of the poster's company, for display on the truck stop load monitors. If omitted, the short name of the poster's company will default to that contained in the poster's company record. Length: 0 – 8
rate	Payment rate in USD (flat-rate or per-mile).
baseRateDollars (float)	Rate in US Dollars (the rateBasedOn element defines whether this is a flat rate or per-mile). Valid Values: 0.0 – 99999.99
rateBasedOn	Per-mile or flat-rate. Valid Values: `Flat`, `PerMile`
rateMiles (int)	If the rate is based on per-mile, or flat-rate for a trip, this is the mileage used by the poster. It is supplied by the poster, and may not agree with HHG mileage or that computed by mileage vendors. Valid Values: 0 – 9999
equipment	
equipmentType	The type of this equipment. See EquipmentType
origin	The origination point of this equipment. See Place
destination	The desired destination (point, area, or open).
One of:	
place	A specific location. See Place
area	A geographic area, defined by a list of states or a region.
stateProvinces	A list of states and/or provinces in the desired region. See StateProvince

zones	A list of one or more zones in the desired region. See Zone
open	Represents "anywhere".
postersReferenceId (string)	Poster-supplied reference ID for this asset. Posters often use this to tag their assets with their own internal IDs. Length: 0 – 8
ltl (boolean)	"Less than truck load" For shipments, if set true, then this shipment will not occupy a full truck. For equipment, if set true, then this equipment does not have enough available capacity to carry a full shipment. The element `dimensions` should be used to specify cargo/capacity size. Note that in this usage, LTL is an attribute of a *posting* rather than of a carrier. Shippers or carriers who deal primarily in LTL logistics may post full loads and those who deal principally in full loads may occasionally post LTL assets, so this flag should not be used to infer anything regarding the owner's business model. Note: The Freight Matching Service never infers whether an asset is an LTL, based on physical dimensions, or any other criteria. It is entirely at the poster's discretion to specify whether an asset is an LTL. Default is false.
comments (string)	Optional comment(s) to be displayed for this asset on search results and the Truck Stop Network. Length: 0 – 70

count (int)	The number of assets represented by this description. This allows a requester to easily (and cheaply) post multiple identical assets. If omitted, defaults to 1. TransCore/DAT does *not* automatically resolve duplicate postings and combine them into a single posting (although the count field would make this possible). Some customers prefer to track duplicate postings separately, so we allow usage of this field, but do not enforce it. Valid Values: 1 – 99
dimensions	For shipments, this is the dimensions of the cargo. For equipment, this is the dimensions of the available cargo space.
lengthFeet (int)	Length of the shipment or truck deck in feet. Valid Values: 1 – 199
weightPounds (int)	Weight of the shipment or truck carrying capacity in pounds. Valid Values: 1 – 199999
heightInches (int)	Height of the shipment or truck box in inches. Valid Values: 1 – 299
volumeCubicFeet (int)	Volume of the shipment or truck box in cubic feet. Valid Values: 1 – 9999
stops (int)	Number of stops this asset must make. If omitted, defaults to 1 (i.e., the stop at the destination). Valid Values: 0 –

availability	For shipments, when available for pickup at its origination point. For equipment, when available to pickup shipments. If omitted, will be automatically defaulted to now (earliest), and now+24 hours (latest).
earliest (dateTime)	For shipments, the earliest the shipment can be picked up. For equipment, the earliest the equipment is available to pick up a shipment. Valid Values: 2003-01-01T00:00:00.000Z –
latest (dateTime)	For shipments, the latest the shipment can be picked up. For equipment, the latest the equipment is available to pick up a shipment. Valid Values: 2003-01-01T00:00:00.000Z –
alarm	If present, an alarm will automatically be generated for this new asset, using this criteria for the alarm. See AlarmSearchCriteria
includeAsset (boolean)	If set true, then the response will return a copy of the asset (and the alarm if one was specified) as they were actually processed and posted by the Freight Matching Service. This saves the requester from doing a subsequent LookupAsset request. This is usually unnecessary, since most asset fields will be identical to those submitted in the request, but geographic points will usually be resolved to points known to TransCore. This as-booked copy will include the actual location information recorded with the posting. If omitted, defaults to false.
postToExtendedNetwork (boolean)	For Shipments, post this asset to the DAT Extended Network. Defaults to false.

Response Fields

assetId (string)	The new asset's ID (assigned by the Freight Matching Service), by which the asset can be referenced for updates, deletions, lookups. Length: 0 – 8
alarmId (string)	If an alarm match was requested when the asset was posted, this is the ID of the alarm that was generated. If the alarm could not be created, this item will be omitted and a warning will be returned along with the success data. Length: 0 – 8
asset	If PostAssetOperation.includeAsset was set true, then this is an as-booked copy of the asset. See Asset
alarm	If the requester wants a copy of the asset after it was posted (see PostAssetOperation.includeAsset), and an alarm match was requested when the asset was posted, this is a copy of the alarm that was actually generated by the Freight Matching Service.
alarmId (string)	The ID of this alarm. Length: 0 – 8
status	
userId (int)	The ID of the owner of this freight matching entity. Valid Values: 0 –
startDate (dateTime)	When this entity becomes active. Valid Values: 2003-01-01T00:00:00.000Z –
endDate (dateTime)	When this entity is no longer active. Valid Values: 2003-01-01T00:00:00.000Z –

created	When this entity was created.
user (int)	
date (dateTime)	
updated	When this entity was last updated. Will initially be set to the same as 'created'.
user (int)	
date (dateTime)	
lastModified	When this entity was last modified. I.e., the latter of 'created' and 'updated'.
user (int)	
date (dateTime)	
expired (boolean)	True if this asset has expired and is no longer visible in searches.
basisAssetId (string)	The asset this alarm is based on. Length: 0 – 8
basisAssetPostersReferenceId (string)	The optional poster-supplied reference ID of the basis asset. If the asset did not have a poster-supplied reference ID, or this alarm was not based on an asset, then this element is omitted. Length: 0 – 8
matchesRemaining (int)	Number of matches remaining that can be returned for this alarm. Valid Values: 0 – 3000

maxMatches (int)	Maximum number of matches that will be returned over the lifetime of this alarm. Valid Values: 0 – 3000
alarmCriteria	The alarm criteria supplied when this alarm was created (or last updated). See AlarmSearchCriteria
matchingAssetIds (string)	List of the IDs of the assets that have been matched by this alarm. Length: 0 – 8

7.2.2 UpdateAsset

Update an existing shipment or equipment posting. Updated postings will be immediately searchable by all TransCore DAT users.

Note: Most elements in UpdateAssetRequest and its contained types are optional. If an element is omitted, no changes will be made to that portion of the asset (e.g., if the rate element is omitted, the asset's rate will be unchanged). If an element is included in the update (even if empty), the appropriate change will be attempted on the asset. For instance, if the rate element is included, but empty, the asset's rate will be deleted. Note that in some cases, business rules may preclude such updates (e.g., comments can be updated, but in most cases, attempting to remove a comment will result in an error).

Request Fields

One of:

assetId (string)	The Asset ID of the asset to be updated. Length: 0 – 8
postersReferenceId (string)	The Reference ID of the asset to be updated. Length: 0 – 8

One of:

shipmentUpdate	Shipment-specific items which can be updated.

ltl (boolean)	
comments (string)	The number of comments on a posting exposed to truckstops cannot be decreased. Omitting comments from an update is allowable (the previous comment(s) will be left unchanged), but including a single comment when the posting currently includes 2 will result in an error. Length: 0 – 70
count (int)	
dimensions	
lengthFeet (int)	Length of the shipment or truck deck in feet. Valid Values: 1 – 199
weightPounds (int)	Weight of the shipment or truck carrying capacity in pounds. Valid Values: 1 – 199999
heightInches (int)	Height of the shipment or truck box in inches. Valid Values: 1 – 299
volumeCubicFeet (int)	Volume of the shipment or truck box in cubic feet. Valid Values: 1 – 9999
stops (int)	
truckStops	Truckstop locations can be altered, but the number of truckstops cannot be decreased.
One of:	
truckStopIds	

ids (int)	A list of TransCore/DAT identifiers for the truckstops the shipment should be displayed at. Valid Values: 0 – 9999
closest	Select the truckstops closest to the shipment origin. The number of selected locations is determined by the poster's subscription and contract.
alternateClosest	Select the truckstops closest to an alternate origin point. This option is typically used to display the shipment on truckstops in the desired lane. The number of selected locations is determined by the poster's subscription and contract.
alternateOrigin	Origin point. See Place
enhancements	Optional truck stop load monitor video enhancements for this shipment's advertisement. Valid Values: `Flash`, `Highlight`
posterDisplayName (string)	The short name of the poster's company, for display on the truck stop load monitors. If omitted, the short name of the poster's company will default to that contained in the poster's company record. Length: 0 – 8
rate	Payment rate in USD (flat-rate or per-mile).
baseRateDollars (float)	Rate in US Dollars (the rateBasedOn element defines whether this is a flat rate or per-mile). Valid Values: 0.0 – 99999.99
rateBasedOn	Per-mile or flat-rate. Valid Values: `Flat`, `PerMile`

rateMiles (int)	If the rate is based on per-mile, or flat-rate for a trip, this is the mileage used by the poster. It is supplied by the poster, and may not agree with HHG mileage or that computed by mileage vendors. Valid Values: 0 – 9999
postToExtendedNetwork (boolean)	Post this shipment to the Extended Network. Defaults to false.
equipmentUpdate	There are currently no equipment-specific items which can be updated. This element is included for potential future expansion.
ltl (boolean)	
comments (string)	The number of comments on a posting exposed to truckstops cannot be decreased. Omitting comments from an update is allowable (the previous comment(s) will be left unchanged), but including a single comment when the posting currently includes 2 will result in an error. Length: 0 – 70
count (int)	
dimensions	
lengthFeet (int)	Length of the shipment or truck deck in feet. Valid Values: 1 – 199
weightPounds (int)	Weight of the shipment or truck carrying capacity in pounds. Valid Values: 1 – 199999
heightInches (int)	Height of the shipment or truck box in inches. Valid Values: 1 – 299

volumeCubicFeet (int)	Volume of the shipment or truck box in cubic feet. Valid Values: 1 – 9999
stops (int)	

Response Fields

No content required or provided

7.2.3 DeleteAsset

Deletes one or more posted assets.

To "delete" means the asset is no longer available for freight matching. If the asset is displayed on the Truck Stop Network, it will be removed from there as well.

Clients are *strongly* encouraged to delete postings that are no longer available. This will prevent unwanted callbacks to the client, and help ensure that searchers see accurate match results.

Request Fields

One of:

deleteAssetsByAssetIds	Delete one or more assets by AssetId.
deleteAssetByPostersReferenceId	Delete an asset by PostersReferenceId.
deleteAllMyAssets	Delete all assets owned by the requester.
deleteAllMyGroupsAssets	Delete all assets belonging by the requester or to other member's of the requester's sharing group.

Response Fields

No content required or provided

7.2.4 LookupAsset

Lookup currently-booked assets. Note that access limits are placed on asset lookup. In general, a user can only lookup their own assets and those belonging to others within their sharing group (if any).

Lookup requests are generally useful when a user wants to view or update their assets. If you want to see the assets of other users outside your own company, see CreateSearchRequest and the types referenced therein.

Request Fields

One of:

queryAssetsByAssetIds	Lookup one or more assets by AssetId.
queryAssetByPostersReferenceId	Lookup one or more assets by PostersReferenceId.
queryAllMyAssets	Lookup all assets belonging to the requester.
queryAllMyGroupsAssets	Lookup all assets belonging to the requester or to other members of the requester's sharing group.

Response Fields

assets	Details of the looked-up assets. See Asset
unreported (string)	In order to keep the payload size (and transmission time) reasonable, the number of assets returned is limited. For requesters with more assets, this element contains the IDs of those assets not reported in `assets`. Clients can then feed this element to a new LookupAssetRequest, using queryAssetsByAssetIds, repeating as necessary until `unreported` is no longer returned. Length: 0 – 8

7.3 Search

Requests related to freight matching searches. Searching allows the user to find shipments and equipment currently available in the DAT system.

As a reminder, all applications that support searching for loads or trucks via Connexion must also support posting loads and/or trucks using Connexion, as appropriate for the customer.

Note: we provide 2 sets of login credentials for the test system, and we *strongly* recommend that you develop controlled tests for your search application, posting as one user and validating that your posting appears in searches executed as the other user.

7.3.1 CreateSearch

Search the TransCore DAT freight-matching system for shipments or equipment that meet a set of caller-specified criteria.

Request Fields

criteria	The criteria to be used for this search. See SearchCriteria
sortOrder	The sort order to apply to the search results. The maximum number of matches returned depends on the requesting customer's subscriptions, configuration, and TransCore policies. If omitted, defaults to SortOrder.AGE (i.e., newest matching assets first). Valid Values: `None`, `Closest`, `Alpha`, `Age`
includeSearch (boolean)	If set true, then the response will return a copy of the search as it was actually processed by the Freight Matching Service. This saves the requester from doing a subsequent LookupSearch request. If omitted, defaults to false.

Response Fields

searchId (string)	ID of the new search (assigned by Freight Matching Service). Length: 0 – 8
matches	Detailed list of assets (if any) satisfying the search. Includes asset details and callback information. The matches are sorted according to the sort method specified in the search request. See MatchingAsset
remainingMatchingIds (string)	List of IDs of matching assets (if any) satisfying the search, less those already presented in detail in "matches". The matching asset IDs are sorted identically to "matches" (see above). This list is suitable (and intended) for submission in subsequent LookupSearchMatchesRequests. Length: 0 – 8
totalMatches (int)	The total number of matches found by this search. This number is always greater than or equal to the sum of the lengths of "matches" and "remainingMatchingIds". This number is not limited by system configuration, and depending on the search criteria, can be very large. Both the detailed matching assets and list of matching asset IDs are the "best" matches out of this total number of matches, based on specified sort and filter criteria. Valid Values: 0 –
search	If CreateSearch.includeSearch was set true, then this is an as-booked copy of the search. See Search

7.3.2 LookupSearchMatches

Retrieves additional matches not included in the response to a prior CreateSearchRequest.

Request Fields

searchId (string)	The ID of the search which provided the supplied list of matching asset IDs. Any IDs in the list which were not supplied (i.e., matched by) this search will be silently rejected. This search must be owned by the requesting user. Length: 0 – 8
matchingIds (string)	List of IDs of matching assets to lookup. This is typically a pass-around of CreateSearchSuccessData.remainingMatchingIds or an immediately-previous LookupSearchMatchesSuccessData.remainingMatchingIds. Length: 0 – 8

Response Fields

searchId (string)	ID of the search whose results were looked up (an echo-back of LookupSearchMatchesOperation.searchId). Length: 0 – 8

matches	Detailed information on matching assets specified in LookupSearchMatchesOperation.matchingIds. The number of detailed matches returned is limited to that specified in LookupSearchMatchesOperation.maxDetailedMatches. The detailed match information is supplied only for those matching asset IDs that are currently active at the time the LookupSearchMatchesRequest is processed. Assets from the matching ID list are checked until either enough are found to fulfill the specified maximum number of matches, or the list is exhausted. It is possible that no currently-active matching assets will be found. Therefore, if the client is always faithful to pass in an ID list resulting from CreateSearchSuccessData.matchingsIds or LookupSearchMatchesSucessData.remainingMatchingIds, it is guaranteed to always get back detailed match information on active assets in the sort order specified by the originating CreateSearchRequest. The client is, of course, free to tamper with the order of the IDs sent in on the LookupSearchMatchesRequest (and the resulting detailed match information will be in the same order as the IDs list). See MatchingAsset
remainingMatchingIds (string)	This is a proper subset of the list of IDs of matching assets supplied in the LookupSearchMatchesRequest, less those returned in element "detailedMatches" and less those that are now inactive or cannot be found or otherwise cannot be accessed by the requester. This provides the client with a pared-down list of matching IDs that can be submitted on subsequent LookupSearchMatchesRequest(s). It is possible that there will be no remaining matching assetIDs. Length: 0 – 8

7.3.3 LookupSearch

Lookup searches recently performed by the requester. The results can be used to perform additional searches.

"Recently" may depend on the requester's configuration and TransCore policy, but in general usually means at least within the last 24 hours.

Request Fields	
One of:	
querySearchesBySearchIds	Lookup one or more searches by SearchId.
queryAllMySearches	Lookup all searches belonging to the requester.
queryAllMyGroupsSearches	Lookup all searches belonging to the requester or to other members of the requester's sharing group.

Response Fields	
searches	Details of the looked-up searches. See Search
unreported (string)	In order to keep the payload size (and transmission time) reasonable, the number of searches returned is limited. For requesters with more searches, this element contains the IDs of those searches not reported in "searches". Clients can then feed this element to a new LookupSearchesRequest, using QuerySearchesBySearchIds, repeating as necessary until "unreported" is no longer returned. Length: 0 – 8

7.3.4 DeleteSearch

Delete a search.

It is not a requirement that searches ever be explicitly deleted. Deletion is a convenience for applica-

tions that make heavy use of multi-search lookups (e.g., `LookupSearchOperation.queryAllMySearches`) so they can easily be excluded from lookups.

Request Fields

One of:

deleteSearchesBySearchIds	Delete one or more searches by SearchId(s).
deleteAllMySearches	Delete all searches belonging to the requester.
deleteAllMyGroupsSearches	Delete all searches belonging to the requester or to other members of the requester's sharing group.

Response Fields

No content required or provided

7.3.5 CountAssets

The CountAssets request is functionally identical to the CreateSearch request, with two important differences:

1. The response contains only a count of matching assets, and does not include details on any of the matches.
2. The search record is not stored, so you cannot subsequently lookup or delete the search (i.e., there are no analogues to LookupSearchRequest or DeleteSearchRequest).

CountAssetsRequest is expected to be used for research and analytic activities, rather than active freight-matching. High search volumes are normal, but the total search volume will generally be limited on a daily, weekly, or monthly basis by subscription levels. Please contact your salesperson if you need to increase your subscription level to raise your search quotas.

Request Fields

criteria	The criteria to be used for this search. See SearchCriteria

Response Fields	
count (int)	The total number of matches found. Valid Values: 0–

7.4 Carrier Monitoring

Requests related to carrier monitoring, including access to the insurance information and certificates maintained by TransCore's CarrierWatch product.

7.4.1 LookupCarrier

Lookup insurance and DOT information about a carrier from TransCore's CarrierWatch system.

Request Fields	
version (decimal)	Deprecated and no longer used.
One of:	
userId (int)	The UserId of a user belonging to the requested carrier. This UserId will normally be obtained from a matching asset returned from a search or alarm match. Valid Values: 0–
docket	The docket prefix and number of the requested carrier (e.g. MC number, FF number).
prefix (string)	Docket Prefix. Not avaliable in the first release. Length: 0 – 2
number (int)	Docket Number of the company. Valid Values: 0–
intrastate	The instrastate code and state of the requested carrier.

intrastateState	The state the carrier is authorized to operate within. See StateProvince
intrastateCode (string)	The state's assigned operating code for this carrier. Length: 0 – 15
mcNumber (int)	The MC Number of the requested carrier. Note that MC number lookup is also available through the `docket` option. This option is retained for ease of use and backwards compatibility. Valid Values: 0 –
dotNumber (int)	The DOT Number of the requested carrier. Valid Values: 0 –
includeDotAuthority (boolean)	Specifies whether the response should include DOT Authority information.
includeDotInsurance (boolean)	Specifies whether the response should include DOT Active / Pending insurance information.
includeInsuranceCertificates (boolean)	Specifies whether the response should include the company's insurance certificate information maintained by TransCore. Note: Insurance certificate information is only available with premium CarrierWatch subscriptions. If you wish to upgrade, please contact your account representative.
includeFmcsaSafetyRating (boolean)	Specifies whether the response should include FMCSA Safety information.
includeFmcsaInspections (boolean)	Specifies whether the response should include FMCSA inspection information.
includeFmcsaCrashes (boolean)	Specifies whether the response should include FMCSA crash information.

includeFmcsaSafeStat (boolean)	Specifies whether the response should include FMCSA SafeStat information.
includeCsa2010SafetyFitness (boolean)	Specifies whether the response should include CSA 2010 safety fitness data.
includeCsa2010Basic (boolean)	Specifies whether the response should include CSA 2010 basic measurement data.
includeDotProfile (boolean)	Specifies whether the response should include the carrier's DOT Profile.
includeExtendedProfile (boolean)	Specifies whether the response should include the 'extended' profile information maintained by TransCore.

Response Fields

header	Basic information maintained by DOT about the carrier. Note that there are (hopefully rare) cases in which the DOT does not provide TransCore with even this basic information, in which case this element will be omitted.
tcoreId (long)	TransCore's own carrier identifier
legalName (string)	Best available legal business name. Length: 0 – 155
dbaName (string)	Best available commonly used business name. Length: 0 – 155
dotNumber (int)	DOT Identification Number of the carrier. Valid Values: 0 –
docket	
prefix (string)	Docket Prefix. Not avaliable in the first release. Length: 0 – 2

number (int)	Docket Number of the company. Valid Values: 0–
intrastate	
intrastateState	The state the carrier is authorized to operate within. See StateProvince
intrastateCode (string)	The state's assigned operating code for this carrier. Length: 0 – 15
duns (int)	DUNS Number, when available. Valid Values: 0–
scac (string)	When available, the Standard Carrier Alpha Code (SCAC) is a unique code used to identify transportation companies. Length: 0 – 4
physicalLocation	The best and last available physical address.
phone	Best available phone number corresponding to the address. See PhoneNumber
tollFreePhone	Best available toll free phone number corresponding to the address. See PhoneNumber
fax	Best available Fax number corresponding to the address. See PhoneNumber
street (string)	Street refers to a street name, which is an identifying name given to a street. The street name forms part of the address. It's a String 0-200 character. Length: 0 – 200

city (string)	City refers to a city name, which is a relatively large and permanent settlement. It's a String with 0-50 character. Length: 0 – 50
stateProvince	The DOT does not always supply a state, particularly for carriers located in Mexico. See StateProvince
postalCode (string)	North American Postal Codes • Canada: LNL NLN • Mexico: NNNNN • US ZIP: NNNNN(-NNNN) The DOT does not always supply a postal code, particularly for carriers located in Mexico. Length: 5 – 10
countryCode	Based on two-letter ISO 3166 country codes for countries supported by TransCore. This is a 2-character String. Valid Values: `US`, `CA`, `MX`
mailingLocation	The best and last available mailing address.
phone	Best available phone number corresponding to the address. See PhoneNumber
tollFreePhone	Best available toll free phone number corresponding to the address. See PhoneNumber
fax	Best available Fax number corresponding to the address. See PhoneNumber
street (string)	Street refers to a street name, which is an identifying name given to a street. The street name forms part of the address. It's a String 0-200 character. Length: 0 – 200

city (string)	City refers to a city name, which is a relatively large and permanent settlement. It's a String with 0-50 character. Length: 0 – 50
stateProvince	The DOT does not always supply a state, particularly for carriers located in Mexico. See StateProvince
postalCode (string)	North American Postal Codes • Canada: LNL NLN • Mexico: NNNNN • US ZIP: NNNNN(-NNNN) The DOT does not always supply a postal code, particularly for carriers located in Mexico. Length: 5 – 10
countryCode	Based on two-letter ISO 3166 country codes for countries supported by TransCore. This is a 2-character String. Valid Values: `US`, `CA`, `MX`
dotProfile	Provides the operation details of a carrier available at DOT.
lastUpdateOfDotProfile (dateTime)	The last date on which the DOT updated their profile for this carrier.
isActive (boolean)	Indicates whether or not the carrier is still operating or not. If not, no other data will be provided in this section.

entityType (string)	The DOT classification for the type of company this is. • Carrier • Shipper • Registrant • Carrier & Shipper Abnormal values may be: • Intermodal Equipment Provider • New Entrant Status Length: 0 – 30
operationType (string)	Identifies the carrier as being engaged in interstate, intrastate hazardous material, or intrastate non-hazardous material transport activities. • Interstate • Intrastate Only (HM) • Intrastate Only (Non-HM) Length: 0 – 25
outOfInterstateServiceDate (date)	The date the carrier went out of Interstate service. Only provided if the carrier is out of service.
powerUnits (int)	The number of power units operated by the carrier. Valid Values: 0 –
drivers (int)	The number of drivers employed and contracted by the carrier. Valid Values: 0 –
mcs150Mileage	

mileageYear (int)	Mileage year from the MCS-150 Registration Form. Valid Values: 0 –
formDate (date)	Date from the MCS-150 Registration Form.
mileage (long)	Mileage from the MCS-150 Registration Form.
mileageYear (int)	Mileage year from the MCS-150 Registration Form. Valid Values: 0 –
formDate (date)	Date from the MCS-150 Registration Form.
mileage (long)	Mileage from the MCS-150 Registration Form.
commodities	Commodities handled by the carrier. Valid Values: `General Freight`, `Household Goods`, `Metals: sheets, coils, rolls`, `Motor Vehicles`, `Fresh Produce`, `Liquids/Gases`, `Oilfield Equipment`, `Livestock`, `Chemicals`, `Dry Bulk`, `Refrigerated Food`, `Beverages`, `Logs, Poles, Beams, Lumber`, `Building Materials`, `Mobile Homes`, `Machinery, Large Objects`, `Grain, Feed Hay`, `Coal/Coke`, `Meat`, `Garbage/Refuse`, `Paper Products`, `Utilities`, `Agricultural/Farm Supplies`, `Construction`, `Drive/Tow away`, `Intermodal Cont.`, `Passengers`, `US Mail`, `Water Well`
specialCommodities (string)	Free form text containing a description of commodities not covered in the standard DOT commodity list. Length: 0 – 256

dotAuthority	Included only if the 'includeDotAuthority' flag is specified. This section describes the common, contract and broker authority status of the target carrier. If the 'includeDotAuthority' flag was specified and this element is *not* included in the response, then the carrier is *not* authorized.
lastUpdateOfDotAuthority (dateTime)	The last date on which the DOT updated their authority information for this carrier.
commonAuthority	Common authority for the carrier. Valid Values: `Active`, `Inactive`, `None`
pendingCommonAuthority (boolean)	Indicates whether or not common authority is pending.
revokedCommonAuthority (boolean)	Indicates whether or not common authority may be revoked in the near future.
contractAuthority	Contract authority for the carrier. Valid Values: `Active`, `Inactive`, `None`
pendingContractAuthority (boolean)	Indicates whether or not contract authority is pending.
revokedContractAuthority (boolean)	Indicates whether or not contract authority may be revoked in the near future.
brokerAuthority	Broker authority for the carrier. Valid Values: `Active`, `Inactive`, `None`
pendingBrokerAuthority (boolean)	Indicates whether or not broker authority is pending.
revokedBrokerAuthority (boolean)	Indicates whether or not broker authority may be revoked in the near future.
carriesFreight (boolean)	Indicates whether or not the carrier is authorized for freight.

carriesPassengers (boolean)	Indicates whether or not the carrier is authorized to carry passengers.
carriesHhg (boolean)	Indicates whether or not the carrier is authorized to carry household goods.
bipdRequired (int)	The amount of BIPD coverage this carrier is required to have. Valid Values: 0–
cargoRequired (boolean)	Indicates whether this carrier is required to file form BMC-34 showing evidence of cargo coverage.
bondSuretyRequired (boolean)	Indicates whether this carrier is required to file form BMC-84 or 85 showing evidence of a broker bond or equivalent trust fund.
bipdOnFile (int)	The amount of BIPD coverage on file. Coverage details will be available in the active/pending insurance section. Valid Values: 0–
cargoOnFile (boolean)	Indicates whether this carrier has filed form BMC-34 showing evidence of cargo coverage.
bondOnFile (boolean)	Indicates whether this carrier has also filed form BMC-84 or 85 showing evidence of a broker bond or trust fund. If true, coverage details will be in the active/pending insurance section.
dotInsurance	Describes a type of insurance carried by this carrier. An element will be included for each of the insurance types carried by the carrier. Included only if the 'includeDotInsurance' flag is specified.

lastUpdateOfDotInsurance (dateTime)	The last date on which the DOT's insurance records for this carrier were updated.
insuranceRecord	The DOT insurance records provided by the DOT.
coverageType	BIPD, Cargo, Trust, or Bond Valid Values: `BIPD`, `Cargo`, `Bond`, `Trust`
bipdClass	If insuranceType = BIPD this element indicates whether the following coverage details are for primary coverage or excess coverage. For other insurance types, this element will be omitted. Valid Values: `Primary`, `Cargo`, `Excess`
formCode (string)	34 = Cargo 91,91X = BIPD 82 = BIPD (Surety Bond) 83 = Cargo (Surety Bond) 84 = Property Brokers Surety Bond 85 = Property Brokers Trust Fund Agreement Length: 0 – 3
policyNumber (string)	The insurance policy number. Length: 0 – 40
coverageFrom (int)	Indicates the lower limit of the coverage. Usually zero except for Excess BIPD insurance. Valid Values: 0 –
coverageTo (int)	Indicates the upper limit of the coverage. Valid Values: 0 –
effectiveDate (date)	Date this insurance policy became effective.
canceledDate (date)	Date this insurance policy was canceled.
insuranceCarrier	The contact information for the insurance carrier.

id (int)	A unique identifier provided by Volpe. Valid Values: 0 –
companyName (string)	Insurance company name. Length: 0 – 55
contact (string)	Insurance company contact name. Length: 0 – 55
location	Insurance company address. Note that there are instances of incorrect registration data, in which the postal code is not of the proper format for the designated country code.
phone	Best available phone number corresponding to the address. See PhoneNumber
tollFreePhone	Best available toll free phone number corresponding to the address. See PhoneNumber
fax	Best available Fax number corresponding to the address. See PhoneNumber
street (string)	Street refers to a street name, which is an identifying name given to a street. The street name forms part of the address. It's a String 0-200 character. Length: 0 – 200
city (string)	City refers to a city name, which is a relatively large and permanent settlement. It's a String with 0-50 character. Length: 0 – 50
stateProvince	The DOT does not always supply a state, particularly for carriers located in Mexico. See StateProvince

postalCode (string)	North American Postal Codes • Canada: LNL NLN • Mexico: NNNNN • US ZIP: NNNNN(-NNNN) The DOT does not always supply a postal code, particularly for carriers located in Mexico. Length: 5 – 10
countryCode	Based on two-letter ISO 3166 country codes for countries supported by TransCore. This is a 2-character String. Valid Values: `US`, `CA`, `MX`
id (int)	A unique identifier provided by Volpe. Valid Values: 0 –
companyName (string)	Insurance company name. Length: 0 – 55
contact (string)	Insurance company contact name. Length: 0 – 55
location	Insurance company address. Note that there are instances of incorrect registration data, in which the postal code is not of the proper format for the designated country code.
phone	Best available phone number corresponding to the address. See PhoneNumber
tollFreePhone	Best available toll free phone number corresponding to the address. See PhoneNumber
fax	Best available Fax number corresponding to the address. See PhoneNumber

street (string)	Street refers to a street name, which is an identifying name given to a street. The street name forms part of the address. It's a String 0-200 character. Length: 0 – 200
city (string)	City refers to a city name, which is a relatively large and permanent settlement. It's a String with 0-50 character. Length: 0 – 50
stateProvince	The DOT does not always supply a state, particularly for carriers located in Mexico. See StateProvince
postalCode (string)	North American Postal Codes • Canada: LNL NLN • Mexico: NNNNN • US ZIP: NNNNN(-NNNN) The DOT does not always supply a postal code, particularly for carriers located in Mexico. Length: 5 – 10
countryCode	Based on two-letter ISO 3166 country codes for countries supported by TransCore. This is a 2-character String. Valid Values: `US`, `CA`, `MX`
One of:	
insuranceCertificates	Provides details on each insurance certificate carried by the carrier.
lastUpdateOfInsuranceCertificates (dateTime)	The last date on which the insurance certificate for this carrier was updated
maxCargoLimit	The maximum coverage limit in the cargo insurance.

amount (double)	
currency	
coverage	Describes cargo coverage carried by the carrier.
coverageCategory (string)	The type of category this certificate falls under: CARGO, AUTO, GENERAL, WORKERS_COMP, or OTHER. OTHER may occur multiple times, but the others will occur only once. Length: 0 – 20
coverageType (string)	The full name provided by the insurance company that describes the type of coverage. Length: 0 – 90
effectiveDate (date)	The date on which this policy went into or is going into effect.
expirationDate (date)	The date on which this policy will expire.
canceledDate (date)	The date on which this policy will be canceled.
confidence (string)	High or Low. The confidence level that the TransCore will be informed promptly of any changes to this policy by the insurance carrier. Length: 0 – 4
policyNumber (string)	The policy number of the policy. Length: 0 – 90
agentName (string)	Issuing insurance agent. Length: 0 – 90
agentPhone	Contact number of the insurance agent. See PhoneNumber

agentFax	Fax number of the insurance agent. See PhoneNumber
underwriter (string)	The company that is underwriting the policy. Length: 0 – 90
coverageId (long)	Deprecated. This field is no longer used.
certificateId (long)	Deprecated. This field is no longer used.
lastCertificateUpdate (dateTime)	The last date on which this certificate was updated.
isScheduledAuto (boolean)	Indicates if this certificate represents scheduled autos. This field is only provided for AUTO coverage categories.
coverageDescription (string)	A verbal description of the details on this certificate. Length: 0 – 100
insuranceCertificateImageUrl (anyURI)	A link to the image of the insurance certificate. Note: In some cases, TransCore may have insurance certificate information but *not* an image of the certificate.
coverageLimit	Describes the limits on the coverage
type (string)	Free-form description of a limit on the coverage Length: 0 – 90
limit	Coverage limit (in US or Canadian dollars).
amount (double)	
currency	
type (string)	Free-form description of a limit on the coverage Length: 0 – 90
limit	Coverage limit (in US or Canadian dollars).
amount (double)	

currency	
insuranceCertificateWarning	Returned if no insurance certificates were on file for the requested carrier. The lack of insurance certificates should not be interpreted as a negative mark on the carrier. To request that TransCore begin monitoring insurance on a carrier, see MonitorInsuranceRequest.
code (int)	
faultCode	
name (string)	
message (string)	Message provides a detailed description of the fault. It is intended to clarify the more general information indicated by the *code* Length: 0 – 1024
detailedMessage (string)	
safestat	Describes FMCSA-maintained safety data. Included only if the 'includeSafestat' flag is specified and if TransCore has current FMCSA SafeStat data on the specified company. Describes the carrier's most recent safety evaluations scores from the FMCSA SafeStat system. For more information regarding the source of this information go to http://ai.fmcsa.dot.gov/SafeStat/SafeStatMain.asp.
lastUpdateOfSafeStat (dateTime)	The date of the most recent SafeStat update. All numbers provided in this section are accurate as of this date. Safestat recalculates SEA Values every month.

seaDriver (float)	Most recent SEA_Driver score for this carrier provided by SafeStat. Ranges from 0 (best) to 100 (worst). Expect 2 decimal places. A SEA with a value from 75 to 100 is defined as deficient. This element will be omitted if the score is 0.
seaVehicle (float)	Most recent SEA_Vehicle score for this carrier provided by SafeStat. Ranges from 0 (best) to 100 (worst). Expect 2 decimal places. A SEA with a value from 75 to 100 is defined as deficient. This element will be omitted if the score is 0.
seaManagement (float)	Most current SEA_Management score for this carrier provided by SafeStat. SEA values range from 0 (best) to 100 (worst). Expect 2 decimal places. A SEA with a value from 75 to 100 is defined as deficient. This element will be omitted if the score is 0.
safetyRating	Describes FMCSA-maintained safety data. Included only if the 'includeSafetyRating' flag is specified and if TransCore has current FMCSA Safety data on the specified company. Describes the carrier's most recent safety rating from the FMCSA. For more information regarding the source of this information go to http://safer.fmcsa.dot.gov/saferhelp.aspx#SafetyRating
lastUpdateOfSafety (dateTime)	Date of the most recent FMCSA safety rating.

rating (string)	Most recent FMCSA Rating. If this element is not included, no safety record was found. Note: there are a very large number of carriers that do not have a safety rating. Lack of a safety rating should not be construed as a negative mark against the carrier. Length: 0 – 15
ratingDate (date)	The date the rating was assigned.
reviewType (string)	Indicates the type of review that resulted in the most recent rating. For more detailed information regarding the reviews go to http://safer.fmcsa.dot.gov/saferhelp.aspx#SafetyRating Length: 0 – 40
reviewDate (date)	The date on which the review occurred.
fmcsaInspections	Describes FMCSA-maintained crash and inspection data. Included only if the 'includeFmcsaCrashInspection' flag is specified and if TransCore has current FMCSA Crash/Inspection data on the specified company. Describes the carrier's crash and inspection record over the last 24 months. For more information regarding the source of this information go to http://safer.fmcsa.dot.gov/saferhelp.aspx#Inspection
lastUpdateOfInspections (dateTime)	Date of the most recent FMCSA inspection data.
totalInspections (int)	Total number of inspections for this carrier in the last 24 months. May be less than the sum of vehicle, driver, and hazmat inspections. Valid Values: 0 –

vehicleInspections	Vehicle inspections for this carrier in the last 24 months. Levels 1+2+5.
inspections (int)	Total number of inspections. Valid Values: 0–
outOfService (int)	Number of those inspections for which the driver or vehicle was pulled out of service. Valid Values: 0–
driverInspections	Driver inspections for this carrier in the last 24 months. Levels 1+2+3.
inspections (int)	Total number of inspections. Valid Values: 0–
outOfService (int)	Number of those inspections for which the driver or vehicle was pulled out of service. Valid Values: 0–
hazmatInspections	Hazmat inspections for this carrier in the last 24 months. Levels 1+2+3+4+5 when HazMat is present.
inspections (int)	Total number of inspections. Valid Values: 0–
outOfService (int)	Number of those inspections for which the driver or vehicle was pulled out of service. Valid Values: 0–

fmcsaCrashes	Describes FMCSA-maintained crash and inspection data. Included only if the 'includeFmcsaCrashInspection' flag is specified and if TransCore has current FMCSA Crash/Inspection data on the specified company. Describes the carrier's crash and inspection record over the last 24 months. For more information regarding the source of this information go to http://safer.fmcsa.dot.gov/saferhelp.aspx#Inspection
lastUpdateOfCrashes (dateTime)	Date of the most recent FMCSA inspection data.
crashesFatal (int)	Number of crashes for this carrier in the last 24 months that resulted in a fatality. May not be provided after CSA2010. May only provide `crashesFatalOrInjury`. Valid Values: 0–
crashesInjury (int)	Number of crashes for this carrier in the last 24 months that resulted in an injury. May not be provided after CSA2010. May only provide `crashesFatalOrInjury`. Valid Values: 0–
One of:	
crashesFatalOrInjury (int)	Number of crashes for this carrier in the last 24 months that resulted in an injury or a fatality. Valid Values: 0–
crashesTow (int)	Number of crashes for this carrier in the last 24 months that resulted in the vehicle being towed. Valid Values: 0–
crashesHazmat (int)	Number of crashes for this carrier in the last 24 months that involved hazardous materials. Valid Values: 0–

csa2010SafetyFitness	CSA 2010 Safety Fitness data. Included only if the `includeCsa2010SafetyFitness` flag is specified and if TransCore has current CSA 2010 data on the specified company.
lastUpdateOfCsa2010SafetyFitness (dateTime)	The last date on which the Csa 2010 Safety Fitness was updated.
interventionActivity (string)	Text describing CSA 2010 intervention activity on the carrier. Length: 0 – 256
safetyFitnessDetermination (string)	This will replace safety ratings and is calculated by FMCSA using the basic scores. Length: 0 – 20
csa2010Basics	CSA 2010 Basic Measurement data. Included only if the `includeCsa2010Basic` flag is specified and if TransCore has current CSA 2010 data on the specified company.
lastUpdateOfCsa2010Basics (dateTime)	The last date on which the Csa 2010 Basic Measurements was updated.
insuranceOrOtherViolation (boolean)	Insurance/Other BASIC Serious Violation Indicator (true = Serious Violation from investigation within previous 12 months)
insuranceOrOtherIndicator (boolean)	Insurance/Other BASIC Indicator (true = Serious Violation within previous 12 months)
measurement	The basic carrier safety measurements.

basic (string)	The types of FMCSA measurements: 'Unsafe Driving', 'Fatigued Driving', 'Driver Fitness', 'Controlled Substance and Alcohol', 'Vehicle Maintenance', or 'Cargo Related'. If no data is available for a type, that type will not be provided. Length: 0 – 40
measure (float)	The decimal measurement for the type, with 2 places after the decimal.
percentile (float)	The percentile for the type, with 1 place after the decimal.
numberOfInspections (int)	The number of inspections resulting in violation of the basic. Valid Values: 0 –
roadsideAlert (boolean)	BASIC Roadside Performance Over Threshold Indicator (Over Intervention Threshold)
seriousViolationIndicator (boolean)	BASIC Serious Violation Indicator (Serious Violation within previous 12 months)
basicIndicator (boolean)	BASIC Indicator (Roadside Performance Percentile over threshold and/or Serious Violation within previous 12 months)
basic (string)	The types of FMCSA measurements: 'Unsafe Driving', 'Fatigued Driving', 'Driver Fitness', 'Controlled Substance and Alcohol', 'Vehicle Maintenance', or 'Cargo Related'. If no data is available for a type, that type will not be provided. Length: 0 – 40
measure (float)	The decimal measurement for the type, with 2 places after the decimal.
percentile (float)	The percentile for the type, with 1 place after the decimal.

numberOfInspections (int)	The number of inspections resulting in violation of the basic. Valid Values: 0 –
roadsideAlert (boolean)	BASIC Roadside Performance Over Threshold Indicator (Over Intervention Threshold)
seriousViolationIndicator (boolean)	BASIC Serious Violation Indicator (Serious Violation within previous 12 months)
basicIndicator (boolean)	BASIC Indicator (Roadside Performance Percentile over threshold and/or Serious Violation within previous 12 months)
extendedProfile	Additional data collected from the carrier by TransCore. Included only if the `includeExtendedProfile` flag is specified.
lastUpdateOfExtendedProfile (dateTime)	The date of the most recent update to the extended profile.
companyBackground	Provides general information about the carrier. This element will be included when selected in the customer's options.
principle (string)	The name of the owner, CEO, or chief officer of the company. Length: 0 – 50
principalTitle (string)	Title of the principal of the company. Length: 0 – 30

primaryContact (string)	Name of the first person you should contact in the company. Length: 0 – 50
phone	Name of the first person you should contact in the company. See PhoneNumber
fax	Name of the first person you should contact in the company. See PhoneNumber
emailAddress (string)	Contact email address. Length: 0 – 256
website (anyURI)	Company website URL.
yearFounded (int)	Year the company was founded. Valid Values: 0 –
isWomanOwned (boolean)	Indicates that the company is owned by a woman. False if omitted.
isMinorityOwned (boolean)	Indicates that the company is owned by a minority. False if omitted.
principle (string)	The name of the owner, CEO, or chief officer of the company. Length: 0 – 50
principalTitle (string)	Title of the principal of the company. Length: 0 – 30
primaryContact (string)	Name of the first person you should contact in the company. Length: 0 – 50

phone	Name of the first person you should contact in the company. See PhoneNumber
fax	Name of the first person you should contact in the company. See PhoneNumber
emailAddress (string)	Contact email address. Length: 0 – 256
website (anyURI)	Company website URL.
yearFounded (int)	Year the company was founded. Valid Values: 0 –
isWomanOwned (boolean)	Indicates that the company is owned by a woman. False if omitted.
isMinorityOwned (boolean)	Indicates that the company is owned by a minority. False if omitted.
fleet	Describes the equipment and personnel of the carrier. This element will be included when selected in the customer's options
satelliteTracking (string)	The type of satellite tracking the carrier has. Length: 0 – 15
numberOfTeams (int)	The number of driver teams the company has. Valid Values: 0 –
hasXWideVans (boolean)	False if omitted.
hasXLongVans (boolean)	False if omitted.
hasXLongFlats (boolean)	False if omitted.
hasXLongReefers (boolean)	False if omitted.

hasFreezerReefers (boolean)	False if omitted.
hasRefrigeratedOnlyReefers (boolean)	False if omitted.
insuranceCoversAllDrivers (boolean)	False if omitted.
insuranceCoversAllTrucks (boolean)	False if omitted.
truck	Identifies how many of each type of truck a carrier has.
code (int)	The TransCore numeric identifier for this truck. Valid Values: 0–
type (string)	Textual description of the truck. Length: 0 – 30
numberOf (int)	The number of this type of truck. Valid Values: 0–
satelliteTracking (string)	The type of satellite tracking the carrier has. Length: 0 – 15
numberOfTeams (int)	The number of driver teams the company has. Valid Values: 0–
hasXWideVans (boolean)	False if omitted.
hasXLongVans (boolean)	False if omitted.
hasXLongFlats (boolean)	False if omitted.
hasXLongReefers (boolean)	False if omitted.
hasFreezerReefers (boolean)	False if omitted.
hasRefrigeratedOnlyReefers (boolean)	False if omitted.

insuranceCoversAllDrivers (boolean)	False if omitted.
insuranceCoversAllTrucks (boolean)	False if omitted.
truck	Identifies how many of each type of truck a carrier has.
code (int)	The TransCore numeric identifier for this truck. Valid Values: 0–
type (string)	Textual description of the truck. Length: 0 – 30
numberOf (int)	The number of this type of truck. Valid Values: 0–
code (int)	The TransCore numeric identifier for this truck. Valid Values: 0–
type (string)	Textual description of the truck. Length: 0 – 30
numberOf (int)	The number of this type of truck. Valid Values: 0–
services	Data on services provided by the carrier. Provided when Services is selected.
specialServices (string)	Free form text description of any special services provided. Length: 0 – 256
handlesOverweightLoads (boolean)	Indicates if carrier can handle overweight loads. False if ommitted.

handlesLongLoads (boolean)	Indicates if carrier can handle extra long loads. False if ommitted.
handlesWideLoads (boolean)	Indicates if carrier can handle extra wide loads. False if ommitted.
handlesEdi (boolean)	Indicates if carrier can handle EDI. False if ommitted.
isResponsibleCare (boolean)	Indicates if the carrier is a reponsible care carrier. False if omitted.
hasGarmentHanging (boolean)	Indicates if the carrier has garment hanging. False if omitted.
hasSpottedTrailers (boolean)	Indicates if the carrier has spotted trailers. False if omitted.
doesTrailerExchange (boolean)	Indicates if the carrier does trailer exchange. False if omitted.
doesPalletExchange (boolean)	Indicates if the carrier does pallet exchange. False if omitted.
doesDrayage (boolean)	Indicates if the carrier provides drayage service. False if omitted.
doesWarehousing (boolean)	Indicates if the carrier has warehousing. False if omitted.
doesLTL (boolean)	Indicates if the carrier has LTL. False if omitted.
doesMultiStops (boolean)	Indicates if the carrier will make multiple stops. False if omitted.
doesParcels (boolean)	Indicates if the carrier deals with Parcels. False if omitted.
carriesAlcohol (boolean)	Indicates if the carrier hauls Alcohol. False if omitted.
carriesCigarettes (boolean)	Indicates if the carrier hauls Cigarettes. False if omitted.

carriesElectronics (boolean)	Indicates if the carrier hauls Electronics. False if omitted.
carriesAmmunitionExplosives (boolean)	Indicates if the carrier hauls Ammunition or Explosives. False if omitted.
carriesCosmetics (boolean)	Indicates if the carrier hauls Cosmetics. False if omitted.
carriesGarments (boolean)	Indicates if the carrier hauls Cosmetics. False if omitted.
carriesHazmat (boolean)	Indicates if carrier can handle Hazardous Materials.
specialServices (string)	Free form text description of any special services provided. Length: 0 – 256
handlesOverweightLoads (boolean)	Indicates if carrier can handle overweight loads. False if ommitted.
handlesLongLoads (boolean)	Indicates if carrier can handle extra long loads. False if ommitted.
handlesWideLoads (boolean)	Indicates if carrier can handle extra wide loads. False if ommitted.
handlesEdi (boolean)	Indicates if carrier can handle EDI. False if ommitted.
isResponsibleCare (boolean)	Indicates if the carrier is a reponsible care carrier. False if omitted.
hasGarmentHanging (boolean)	Indicates if the carrier has garment hanging. False if omitted.
hasSpottedTrailers (boolean)	Indicates if the carrier has spotted trailers. False if omitted.
doesTrailerExchange (boolean)	Indicates if the carrier does trailer exchange. False if omitted.

doesPalletExchange (boolean)	Indicates if the carrier does pallet exchange. False if omitted.
doesDrayage (boolean)	Indicates if the carrier provides drayage service. False if omitted.
doesWarehousing (boolean)	Indicates if the carrier has warehousing. False if omitted.
doesLTL (boolean)	Indicates if the carrier has LTL. False if omitted.
doesMultiStops (boolean)	Indicates if the carrier will make multiple stops. False if omitted.
doesParcels (boolean)	Indicates if the carrier deals with Parcels. False if omitted.
carriesAlcohol (boolean)	Indicates if the carrier hauls Alcohol. False if omitted.
carriesCigarettes (boolean)	Indicates if the carrier hauls Cigarettes. False if omitted.
carriesElectronics (boolean)	Indicates if the carrier hauls Electronics. False if omitted.
carriesAmmunitionExplosives (boolean)	Indicates if the carrier hauls Ammunition or Explosives. False if omitted.
carriesCosmetics (boolean)	Indicates if the carrier hauls Cosmetics. False if omitted.
carriesGarments (boolean)	Indicates if the carrier hauls Cosmetics. False if omitted.
carriesHazmat (boolean)	Indicates if carrier can handle Hazardous Materials.
territory	Identifies areas that the carrier operates within.
isCsaApproved (boolean)	Indicates if the carrier is CSA approved. False if omitted.
isFastCertified (boolean)	Indicates if the carrier is FAST certified. False if omitted.
serviceToMexico (boolean)	Indicates if carrier provides service to Mexico. False if ommitted.

statesProvinces	The states and provinces that the carrier normally operates in. See StateProvince
lane	The lanes that a carrier normally operates within.
origin	The starting point of the lane.
stateProvince	The state or province of the end point. See StateProvince
metroArea	If one is specified, the metropolitan area of the end point.
code (int)	The numeric code reserved for this metropolitan area. Valid Values: 0–
name (string)	The name describing this metropolitan area. Length: 0 – 40
code (int)	The numeric code reserved for this metropolitan area. Valid Values: 0–
name (string)	The name describing this metropolitan area. Length: 0 – 40
destination	The ending point of the lane.
stateProvince	The state or province of the end point. See StateProvince
metroArea	If one is specified, the metropolitan area of the end point.
code (int)	The numeric code reserved for this metropolitan area. Valid Values: 0–

name (string)	The name describing this metropolitan area. Length: 0 – 40
code (int)	The numeric code reserved for this metropolitan area. Valid Values: 0 –
name (string)	The name describing this metropolitan area. Length: 0 – 40

7.4.2 MonitorInsurance

Request that a carrier in the TransCore CarrierWatch system be monitored for insurance information.

Request Fields

requesterName (string)	Requester's name. Length: 0 – 50
requesterPhoneNumber	Requester's phone number. See PhoneNumber
requesterVendorId (int)	The ID that the TMS uses for the carrier. Transcore's integrated customers have their own ID system for carriers and pass those IDs to Transcore to map Transcore's data to the desired entity in the TMS. The TMS might be able to grab this automatically based on context. However, if Transcore have not previously received the Vendor ID, it might not be of use on Transcore's side. Valid Values: 0 –

loadingToday (boolean)	Indicates a desire or plan to load the carrier today. If specified, this is considered a "Hot Request" with higher priority and shorter turn-around time (premium subscriptions only). If omitted or false, this request is treated as standard priority with approximately 48-hour turnaround. NOTE: This is designed only to be used for urgent situations, never as the default method for adding a carrier. If you plan to leverage this feature, you MUST discuss how you will utilize it in your TMS with TransCore FIRST before beginning your implementation.
carrierDotNumber (nonNegativeInteger)	DOT Number of the carrier.
carrierDocketNumber	Docket number of the carrier (e.g., MC, FF number).
prefix (string)	Docket Prefix. Not avaliable in the first release. Length: 0 – 2
number (int)	Docket Number of the company. Valid Values: 0 –
carrierIntrastateCode	Intrastate identifier of the carrier.
intrastateState	The state the carrier is authorized to operate within. See StateProvince
intrastateCode (string)	The state's assigned operating code for this carrier. Length: 0 – 15
comments (string)	Any additional pertinent information for the TransCore representative. Length: 0 – 512

Response Fields

No content required or provided

7.4.3 CreateCarrier

Request that a carrier not yet in the TransCore CarrierWatch system be added for the purposes of monitoring DOT and insurance information. TransCore already monitors DOT information on the vast majority of interstate carriers, so this request is primarily useful for intra-state carriers. If the requester is a premium CarrierWatch customer (with the ability to lookup private insurance information), TransCore will additionally begin monitoring private insurance information on the specified carrier (see also MonitorInsuranceRequest).

Request Fields

requesterName (string)	Requester's name. Length: 0 – 50
requesterPhoneNumber	Requester's phone number. See PhoneNumber
requesterVendorId (int)	The ID that the TMS uses for the carrier. Transcore's integrated customers have their own ID system for carriers and pass those IDs to Transcore to map Transcore's data to the right entity in the TMS. The TMS might be able to grab this automatically based on context. However, if Transcore have not previously received the Vendor ID, it might not be of use on Transcore's side. Valid Values: 0 –

loadingToday (boolean)	Indicates a desire or plan to load the carrier today. If specified, this is considered a "Hot Request" with higher priority and shorter turn-around time (premium subscriptions only). If omitted or false, this request is treated as standard priority with approximately 48-hour turnaround. NOTE: This is designed only to be used for urgent situations, never as the default method for adding a carrier. If you plan to leverage this feature, you MUST discuss how you will utilize it in your TMS with TransCore FIRST before beginning your implementation.
carrierCompanyName (string)	Carrier's name. Length: 0 – 50
carrierAddressLine1 (string)	Carrier's address. Length: 0 – 50
carrierAddressLine2 (string)	Carrier's address details. Length: 0 – 50
carrierCity (string)	Carrier's city. Length: 0 – 50
carrierState	Carrier's state. See StateProvince
carrierZipCode (string)	Carrier's zip/postal code. Length: 5 – 10
carrierDotNumber (nonNegativeInteger)	Carrier's DOT number.

carrierDocketNumber	Please enter at least one of the following, so TransCore can identify the carrier: • carrierMcNumber • carrierIntrastateCode • carrierDotNumber.
prefix (string)	Docket Prefix. Not avaliable in the first release. Length: 0 – 2
number (int)	Docket Number of the company. Valid Values: 0 –
carrierIntrastateCode	Carrier's intrastate identifier.
intrastateState	The state the carrier is authorized to operate within. See StateProvince
intrastateCode (string)	The state's assigned operating code for this carrier. Length: 0 – 15
carrierContactName (string)	Primary contact at the carrier. Length: 0 – 50
carrierPhoneNumber	Carrier's contact phone number. See PhoneNumber
carrierFaxNumber	Carrier's contact fax number. See PhoneNumber
comments (string)	Any additional pertinent information for the TransCore representative. Length: 0 – 512

Response Fields

resultMessage (string)	Brief acknowlegement message.

7.5 Rating

Requests related to spot- and contract-market rates, provided by TransCore's Truckload Rate Index suite of products.

7.5.1 LookupRate

Request spot-market and/or contract rate information for a lane, including estimated rate, range, and a numeric representation of TransCore's confidence in that estimate.

Each rate requested will be credited against the pool of lookups associated with the user's Truckload Rate Index subscription.

Note that multiple LookupRateOperations can be submitted in a single LookupRateRequest, in batches of up to 50. If you need to request more than 50 lanes, please submit batches sequentially.

Request Fields

equipment	Rates deal with broader groups of equipment than EquipmentTypes or EquipmentClasses used in Freight Matching, so the set of equipment categories is much smaller. Valid Values: `Vans`, `Flatbeds`, `Reefers`
origin	The origin of the lane, specified using any of the standard Connexion methods for describing locations. The choice of location specification method is generally less important for rating than for freight matching. We offer a variety of methods purely for convenience, but the method chosen will have no effect on rating accuracy. See Place
destination	The destination of the lane. See the description of 'origin' above for further details. See Place

includeSpotRate (boolean)	If true, the response will include TransCore's spot rate estimate for the specified lane. This allows the requester to compare his rate with the spot market rate. Normally defaults to false if omitted. However, if `includeContractRate` is false or omitted, this element defaults to true - i.e., if neither `includeSpotRate` or `includeContractRate` is specified, the request is assumed to be for spot rate information. Additionally, if `includeMyRate` is true, the response will include the market spot rate estimate. If the requester is not authorized to lookup spot rate information, an error will be returned.
includeContractRate (boolean)	If true, the response will include a contract rate estimate for the specified lane. Defaults to false. If the requester is not authorized to lookup contract rate information, an error will be returned.
includeMyRate (boolean)	If true, the response will include the requester's spot rate information for the specified lane. This allows the requester to compare their company's own rate with the overall spot market rate. Note that this element implies `includeSpotRate` as well. If `includeMyRate` is true, the spot rate estimate will be returned as well. If the requester is not a lane rate data contributor, this element will be ignored. Defaults to false.

Response Fields

spotRate	The spot market rate is returned if the `includeSpotRate` flag was specified on the request, or if no `includeContractRate` was false (i.e., by default, spot rate is returned, if no rate type was requested).

estimatedLinehaulRate (float)	The estimated per-mile rate.
lowLinehaulRate (float)	The low end of TransCore's estimated per-mile rate range. In general, this is the rate defining the lowest quartile of rates. 25% of rate contributions in the lane fall below this rate.
highLinehaulRate (float)	The high end of TransCore's estimated per-mile rate range. In general, this is the rate defining the upper quartile of rates. 25% of rate contributions fall above this rate.
estimatedLinehaulTotal (float)	The estimated linehaul total. Note in general this will not equal `estimatedLinehaulRate` x `mileage`. The estimated total is obtained by averaging total rates in the lane for trips with similar mileage. The LinehaulTotal is generally more appropriate to use for short trips where minimums and hourly costs are more important than mileage in determining rates. The per mile total is obtained by averaging per mile rates in the lane for trips with similar mileage. Per mile is more appropriate for long haul rates where per mile costs are most important in setting rates.
lowLinehaulTotal (float)	The low end of TransCore's estimated total rate range. In general, this can be taken as the lower end of an approximately 50% confidence interval (although it may not always reflect an exact statistical confidence interval).
highLinehaulTotal (float)	The high end of TransCore's estimated per-mile rate range. In general, this can be taken as the upper end of an approximately 50% confidence interval (although it may not always reflect an exact statistical confidence interval).

confidenceLevel (int)	A quantitative description of TransCore's confidence in the rate estimate provided. It ranges from 1 (highest) to 100 (lowest). This confidence is primarily derived from the amount of relevant data incorporated into the rate estimate. The confidence level incorporates factors such as: • The number of recent rate submissions on the requested lane or on closely related lanes. • The size of the geographic areas TransCore used to estimate the rate (e.g. postal prefix, market area, region). • The size of confidence intervals around the estimated rate (e.g., the size of a 95% confidence interval, or of the (approximate) 50% confidence range described above). Valid Values: 1 – 100
ratedLane	This is the lane that was actually used to compute the rate. It will denote the geographic region in which the requested origin and destination fall. See RatedLane
averageFuelSurchargeRate (float)	Average per-mile fuel surcharge for the lane, from the Department of Energy.
contributors (int)	The number of individual contributors to the lane data that were used to compute the rate. This element is provided as an additional means for a user to evaluate the significance of the rate estimate, but may be removed in the future as TransCore's rating algorithms evolve to utilize data from outside specific market areas and to incorporate long-term patterns. Valid Values: 0 –

moves (int)	The number of moves (i.e., contributed lane data) that were used to compute the rate. This element is provided as an additional means for a user to evaluate the significance of the rate estimate, but may be removed in the future as TransCore's rating algorithms evolve to utilize data from outside specific market areas and to incorporate long-term patterns. Valid Values: 0–
daysBack (int)	The time period over which the rate was calculated. This element is provided as an additional means for a user to evaluate the significance of the rate estimate, but may be removed in the future as TransCore's rating algorithms evolve to utilize data from outside specific market areas and to incorporate long-term patterns. Valid Values: 0–
yourRate	An estimate of the requester's own rate, derived from their historical rate contributions. The requester's rate is only returned if: 1) The requester asked for it (via LookupRateOperation.includeMyRate) and 2) The requester is a contributor (i.e., belongs to a company which contributes rates to TransCore) and 3) The requester's company has contributed enough data for the requested lane. If the requester is not a contributor, no error is returned (this element is simply omitted). The requester's rate will be based on the same lane as that used to generate the current rate, i.e., `spotRate.ratedLane`. Note that TransCore will may estimate a contributor's rate based on fewer rate contributions than the minimum required for a market-rate estimate. In some cases, the low, average, and high estimates may be identical.
estimatedLinehaulRate (float)	The estimated per-mile rate.

lowLinehaulRate (float)	The low end of TransCore's estimated per-mile rate range. In general, this is the rate defining the lowest quartile of rates. 25% of rate contributions in the lane fall below this rate.
highLinehaulRate (float)	The high end of TransCore's estimated per-mile rate range. In general, this is the rate defining the upper quartile of rates. 25% of rate contributions fall above this rate.
estimatedLinehaulTotal (float)	The estimated linehaul total. Note in general this will not equal `estimatedLinehaulRate` x `mileage`. The estimated total is obtained by averaging total rates in the lane for trips with similar mileage. The LinehaulTotal is generally more appropriate to use for short trips where minimums and hourly costs are more important than mileage in determining rates. The per mile total is obtained by averaging per mile rates in the lane for trips with similar mileage. Per mile is more appropriate for long haul rates where per mile costs are most important in setting rates.
lowLinehaulTotal (float)	The low end of TransCore's estimated total rate range. In general, this can be taken as the lower end of an approximately 50% confidence interval (although it may not always reflect an exact statistical confidence interval).
highLinehaulTotal (float)	The high end of TransCore's estimated per-mile rate range. In general, this can be taken as the upper end of an approximately 50% confidence interval (although it may not always reflect an exact statistical confidence interval).

confidenceLevel (int)	A quantitative description of TransCore's confidence in the rate estimate provided. It ranges from 1 (highest) to 100 (lowest). This confidence is primarily derived from the amount of relevant data incorporated into the rate estimate. The confidence level incorporates factors such as: • The number of recent rate submissions on the requested lane or on closely related lanes. • The size of the geographic areas TransCore used to estimate the rate (e.g. postal prefix, market area, region). • The size of confidence intervals around the estimated rate (e.g., the size of a 95% confidence interval, or of the (approximate) 50% confidence range described above). Valid Values: 1 – 100
ratedLane	This is the lane that was actually used to compute the rate. It will denote the geographic region in which the requested origin and destination fall. See RatedLane
moves (int)	The number of moves (i.e., contributed lane data) that were used to compute the rate. This element is provided as an additional means for a user to evaluate the significance of the rate estimate, but may be removed in the future as TransCore's rating algorithms evolve to utilize data from outside specific market areas and to incorporate long-term patterns. Valid Values: 0 –

daysBack (int)	The time period over which the rate was calculated. This element is provided as an additional means for a user to evaluate the significance of the rate estimate, but may be removed in the future as TransCore's rating algorithms evolve to utilize data from outside specific market areas and to incorporate long-term patterns. Valid Values: 0–
contractRate	The contract market rate is returned if the `includeContractRate` flag was specified on the request.
estimatedLinehaulRate (float)	The estimated linehaul rate.
lowLinehaulRate (float)	The low end of the contract-rate provider's estimated linehaul rate range.
highLinehaulRate (float)	The high end of the contract-rate provider's estimated linehaul rate range.
ratedLane	This is the lane that was actually used to compute the rate. It will denote the geographic region in which the requested origin and destination fall. See RatedLane
averageFuelSurchargeRate (float)	Average per-mile fuel surcharge for the lane.
averageAccessorialCharge (float)	Average accessorial charge for the lane.
contributors (int)	The number of individual contributors to the lane data that were used to compute the rate. This element is provided as an additional means for a user to evaluate the significance of the rate estimate. Valid Values: 0–
moves (int)	The number of moves (i.e., contributed lane data) that were used to compute the rate. This element is provided as an additional means for a user to evaluate the significance of the rate estimate. Valid Values: 0–

miles (int)	The road mileage between the origin and destination points. Valid Values: 0–

7.5.2 LookupHistoricSpotRates

Requests spot rate estimates for a lane over the past 13 months. Each rate returned will be credited against the pool of lookups associated with the user's Truckload Rate Index Subscription.

Request Fields

equipment	Rates deal with broader groups of equipment than EquipmentTypes or EquipmentClasses used in Freight Matching, so the set of equipment categories is much smaller. Valid Values: `Vans`, `Flatbeds`, `Reefers`
origin	The origin of the lane. See Place
destination	The destination of the lane. See Place

Response Fields

ratedLane	This is the lane that was actually used to compute the rate. It will denote the geographic region in which the requested origin and destination fall. See RatedLane

monthlyReport	Monthly rate reports for the requested lane. Returned in chronological ascending order. Only those months for which rate data is available will be returned, so there may be gaps. The most recent completed month is the last month returned (if sufficient data is available for that month). E.g., at any time in June, the monthly summary for May is the most recent summary available. If necessary, TransCore's rate estimation algorithms will use expanded geographic regions to obtain a statistically meaningful sample. The geographic region will be expanded until rates are available for 75% of the requested months. All monthly reports will reflect the same geographic expansion level.
estimatedLinehaulRate (float)	The estimated per-mile rate.
lowLinehaulRate (float)	The low end of TransCore's estimated per-mile rate range. In general, this is the rate defining the lowest quartile of rates. 25% of rate contributions in the lane fall below this rate.
highLinehaulRate (float)	The high end of TransCore's estimated per-mile rate range. In general, this is the rate defining the upper quartile of rates. 25% of rate contributions fall above this rate.
estimatedLinehaulTotal (float)	The estimated linehaul total. Note in general this will not equal `estimatedLinehaulRate` x `mileage`. The estimated total is obtained by averaging total rates in the lane for trips with similar mileage. The LinehaulTotal is generally more appropriate to use for short trips where minimums and hourly costs are more important than mileage in determining rates. The per mile total is obtained by averaging per mile rates in the lane for trips with similar mileage. Per mile is more appropriate for long haul rates where per mile costs are most important in setting rates.

lowLinehaulTotal (float)

The low end of TransCore's estimated total rate range. In general, this can be taken as the lower end of an approximately 50% confidence interval (although it may not always reflect an exact statistical confidence interval).

highLinehaulTotal (float)

The high end of TransCore's estimated per-mile rate range. In general, this can be taken as the upper end of an approximately 50% confidence interval (although it may not always reflect an exact statistical confidence interval).

confidenceLevel (int)

A quantitative description of TransCore's confidence in the rate estimate provided. It ranges from 1 (highest) to 100 (lowest).

This confidence is primarily derived from the amount of relevant data incorporated into the rate estimate. The confidence level incorporates factors such as:

- The number of recent rate submissions on the requested lane or on closely related lanes.
- The size of the geographic areas TransCore used to estimate the rate (e.g. postal prefix, market area, region).
- The size of confidence intervals around the estimated rate (e.g., the size of a 95% confidence interval, or of the (approximate) 50% confidence range described above).

Valid Values: 1 – 100

when

year (int)

4-digit year

Valid Values: 1900 – 2200

month (int)	2-digit month Valid Values: 1 – 12

7.5.3 LookupHistoricContractRates

Requests contract rate estimates for a lane and national average rates over the past 13 months. Each rate returned will be credited against the pool of lookups associated with the user's Truckload Rate Index Subscription.

Request Fields

equipment	Rates deal with broader groups of equipment than EquipmentTypes or EquipmentClasses used in Freight Matching, so the set of equipment categories is much smaller. Valid Values: `Vans`, `Flatbeds`, `Reefers`
origin	The origin of the lane. See Place
destination	The destination of the lane. See Place

Response Fields

ratedLane	This is the lane that was actually used to compute the rate. It will denote the geographic region in which the requested origin and destination fall. See RatedLane
monthlyReport	Monthly rate reports for the requested lane. Returned in chronological ascending order. Only those months for which rate data is available will be returned, so there may be gaps. The summary for the current month is first computed on the 22nd or 23rd of the month (depending on weekends and holidays). Prior to that recomputation, the most recent monthly summary available is for the previous month. On the 24th (and in some cases on the 23rd), the current month's summary will be included.

estimatedLinehaulRate (float)	The estimated linehaul rate.
lowLinehaulRate (float)	The low end of the contract-rate provider's estimated linehaul rate range.
highLinehaulRate (float)	The high end of the contract-rate provider's estimated linehaul rate range.
when	Date of this estimate.
year (int)	4-digit year Valid Values: 1900 – 2200
month (int)	2-digit month Valid Values: 1 – 12
nationalLinehaulRate (float)	The national average linehaul rate.

7.5.4 LookupDoeFuelPrices

Lookup diesel fuel prices over the past 13 months, as reported by the Department of Energy.

Request Fields

No content required or provided

Response Fields

monthlyReports	The DOE reports prices for the current month a few days into the month. If a price is available for the current month, the report will include the current month and the previous 12 months. If no price is available for the current month, the report will instead include the previous 13 months.
when	The month and year represented by this report.

year (int)	4-digit year Valid Values: 1900 – 2200
month (int)	2-digit month Valid Values: 1 – 12
pricePerGallon (float)	Average monthly diesel price in USD.

7.6 Alarm Match

Requests related to Alarm Match, providing real-time notification of incoming shipment and equipment postings. For further information on Alarm Match, see Section 6.

7.6.1 CreateAlarm

Create an alarm search for an existing truck or load posting. An alarm waits for assets (loads or trucks) that match the associated posting and when one is found, it notifies the user through a web service call or via one of TransCore's client applications (see the 'notifyAllApplications' option). Alarms become immediately active in the freight-matching system when created. An alarm expires when it receives the maximum number of matches or at 6:00 PM Pacific Time. Alarms created after 6 PM expire at midnight.

Request Fields

assetId (string)	The asset on which this alarm will be based. Length: 0 – 8
criteria	The criteria for this alarm. See AlarmSearchCriteria

Response Fields

alarmId (string)	The ID of this alarm. This ID must be used when cancelling, updating, or querying this alarm. Length: 0 – 8

alarm	A copy of the alarm that was created. This potentially saves the client a LookupAlarm request.
alarmId (string)	The ID of this alarm. Length: 0 – 8
status	
userId (int)	The ID of the owner of this freight matching entity. Valid Values: 0 –
startDate (dateTime)	When this entity becomes active. Valid Values: 2003-01-01T00:00:00.000Z –
endDate (dateTime)	When this entity is no longer active. Valid Values: 2003-01-01T00:00:00.000Z –
created	When this entity was created.
user (int)	
date (dateTime)	
updated	When this entity was last updated. Will initially be set to the same as 'created'.
user (int)	
date (dateTime)	
lastModified	When this entity was last modified. I.e., the latter of 'created' and 'updated'.
user (int)	
date (dateTime)	

expired (boolean)	True if this asset has expired and is no longer visible in searches.
basisAssetId (string)	The asset this alarm is based on. Length: 0 – 8
basisAssetPostersReferenceId (string)	The optional poster-supplied reference ID of the basis asset. If the asset did not have a poster-supplied reference ID, or this alarm was not based on an asset, then this element is omitted. Length: 0 – 8
matchesRemaining (int)	Number of matches remaining that can be returned for this alarm. Valid Values: 0 – 3000
maxMatches (int)	Maximum number of matches that will be returned over the lifetime of this alarm. Valid Values: 0 – 3000
alarmCriteria	The alarm criteria supplied when this alarm was created (or last updated). See AlarmSearchCriteria
matchingAssetIds (string)	List of the IDs of the assets that have been matched by this alarm. Length: 0 – 8
matches	If the alarm requested an initial lookback search, this is the list of matches from that search. If no matches were found, this list is empty. If the alarm did not request an initial lookback search, this element is omitted. See MatchingAsset

7.6.2 UpdateAlarm

Updates the search criteria for an existing alarm.

Request Fields

alarmId (string)	The ID of the alarm being updated. The requester must be authorized to update the alarm. Length: 0 – 8
originRadius	Radius around the asset's origin. If the asset has a non-Point origin (e.g., an Equipment with an Area/Empty origin), then this field will be ignored and a warning will be returned.
miles (int)	Mileage. Valid Values: 0 –
method	Mileage calculation method. Valid Values: `Air`, `Road`, `RoadProMiles`, `RoadAlk`
destinationRadius	Radius around the asset's destination. If the asset has a non-Point destination (e.g., an Equipment with an Area/Empty destination), then this field will be ignored and a warning will be returned.
miles (int)	Mileage. Valid Values: 0 –
method	Mileage calculation method. Valid Values: `Air`, `Road`, `RoadProMiles`, `RoadAlk`

Response Fields

No content required or provided

7.6.3 LookupAlarm

Lookup information on particular alarm(s). Only those alarms accessible by the requester can be successfully selected.

Request Fields

One of:

queryAlarmsByAlarmIds	Lookup specific alarms by AlarmId.
queryAllMyAlarms	Lookup all alarms belonging to the requester.
queryAllMyGroupsAlarms	Lookup all alarms belonging to the requester or to members of their sharing group.

Response Fields

alarms

alarmId (string)	The ID of this alarm. Length: 0 – 8
status	
userId (int)	The ID of the owner of this freight matching entity. Valid Values: 0 –
startDate (dateTime)	When this entity becomes active. Valid Values: 2003-01-01T00:00:00.000Z –
endDate (dateTime)	When this entity is no longer active. Valid Values: 2003-01-01T00:00:00.000Z –
created	When this entity was created.

user (int)	
date (dateTime)	
updated	When this entity was last updated. Will initially be set to the same as 'created'.
user (int)	
date (dateTime)	
lastModified	When this entity was last modified. I.e., the latter of 'created' and 'updated'.
user (int)	
date (dateTime)	
expired (boolean)	True if this asset has expired and is no longer visible in searches.
basisAssetId (string)	The asset this alarm is based on. Length: 0 – 8
basisAssetPostersReferenceId (string)	The optional poster-supplied reference ID of the basis asset. If the asset did not have a poster-supplied reference ID, or this alarm was not based on an asset, then this element is omitted. Length: 0 – 8
matchesRemaining (int)	Number of matches remaining that can be returned for this alarm. Valid Values: 0 – 3000
maxMatches (int)	Maximum number of matches that will be returned over the lifetime of this alarm. Valid Values: 0 – 3000

alarmCriteria	The alarm criteria supplied when this alarm was created (or last updated). See AlarmSearchCriteria
matchingAssetIds (string)	List of the IDs of the assets that have been matched by this alarm. Length: 0 – 8
unreported (string)	In order to keep the payload size (and transmission time) reasonable, the number of alarms returned is limited. For requesters with more alarms, this element contains the IDs of those alarms not reported in `alarms`. Clients can then feed this element to a new LookupAlarmsRequest, using QueryAlarmsByAlarmIds, repeating as necessary until `unreported` is no longer returned. Length: 0 – 8

7.6.4 DeleteAlarm

Delete an alarm.

To "delete" means the alarm is no longer available for freight matching activities and is no longer retrieved on multi-alarm lookups.

If the alarm is currently active (i.e., not expired) then it will be automatically cancelled (and an alarm termination notice will be issued).

It is not a requirement that alarm ever be explicitly deleted. Deletion is a convenience for applications that make heavy use of multi-alarm lookups (e.g., `LookupAlarmOperation.LookupAllMyAlarms`).

Request Fields

One of:

deleteAlarmsByAlarmIds	Delete one or more alarms by AlarmId.
deleteAllMyAlarms	Delete all alarms owned by the requester.

deleteAllMyGroupsAlarms	Delete all alarms belonging by the requester or to other member's of the requester's sharing group.

Response Fields

No content required or provided

7.6.5 UpdateAlarmUrl

Specifies the web service endpoint URL that alarm notifications will be delivered to. If a single endpoint is shared for an entire company or office, TransCore customer support can configure that URL. This request specifies a unique URL for the requester, and is appropriate when the endpoint is subject to frequent changes or differs between users.

Request Fields

alarmUrl (string)	The URL to send future alarm results to for this user. Note that a URL may be set by TransCore for an entire company or office, but that updating the URL overrides this default for a particular user only. It will not affect other users in their company or office. The URL specified is persisted between sessions. If a new URL is needed on a per-session basis (unlikely), the client should update it after login when beginning a new session. Length: 1 – 150

Response Fields

No content required or provided

7.6.6 LookupAlarmUrl

Lookup the web service endpoint URL that alarm notifications will be delivered to. This request can be used to verify the URL that TransCore has on file for the requester.

Request Fields

No content required or provided

Response Fields

alarmUrl (string)	The URL that alarm matches will be sent to for this user. Note that the URL may be set at the company, office, or user level and overridden at a lower level. The alarm URL element will be omitted if the user has not set a URL. Length: 0 – 150

7.7 DAT Onboarding

Requests related to retrieval of data on carriers who have completed a profile and signed contracts with the customer using the DAT Onboarding product.

Clients should check for events using the LookupDobEvents request on a set interval, such as every 5 minutes, in order to retrieve data soon after it is available. Events will include the carriers for which data is available so that data can be retrieved by using LookupDobCarriers request.

Note: We encourage developers implementing DAT Onboarding to consult the detailed description of event handling test procedures in Section 9.23

7.7.1 LookupDobEvents

Lookup DAT Onboarding events received since a given date/time. Events include changes to carrier profile, carrier W9 status, carrier insurance agents, carrier attachments and contracts signed by a carrier.

This request is only supported through the secure (HTTPS) endpoint. Requests through the insecure endpoint will return an error.

Request Fields

sinceDate (dateTime)	The date and time to look for events.
dobVersion (int)	The version of response to receive. The default is 1. Set the version to the version of the XML schema that you built to. The version of this schema is 2.

Response Fields

dobEvents	The DAT-On-Boarding events that were received since the given date.
carrierId (string)	The DAT-On-Boarding identifier for the carrier that signed contracts. Length: 0 – 40
eventTypes	The types of events that were received concerning this carrier. Valid Values: `Contract Signed`, `Profile Update`, `Insurance Agent Update`, `Tax Info Update`, `Document Update`
signedContracts	The contracts that this carrier signed since the date provided.
contract	The contract that was signed.
name (string)	The Name of the contract. Length: 0 – 100
description (string)	The description of the contract. Length: 0 – 100
version (int)	The version of the contract. Valid Values: 1 –

documentStatus	The status of the contract (Active or Inactive); Valid Values: `Active`, `Inactive`
signatory	The signature information for the contract.
name (string)	The Name of person signing a contract. Length: 0 – 100
title (string)	The title of the person signing a contract. Length: 0 – 100
signed (dateTime)	The date the contract was signed.
url (anyURI)	The URL from which this contract may be downloaded. Provided only when dobVersion >1 and for contracts signed through AssureSign. Max 256 characters.

7.7.2 LookupDobCarriers

Lookup information on a DAT Onboarding carrier. All data available for the carrier is returned.

This request is only supported through the secure (HTTPS) endpoint. Requests through the insecure endpoint will return an error.

Request Fields

dobVersion (int)	The version of response to receive. The default is 1. Set the version to the version of the XML schema that you built to. The version of this schema is 2.
carrierId (string)	The DAT-On-Boarding identifier for the carrier of interest. Length: 0 – 40

Response Fields

dobCarrier	The DAT-On-Boarding data available for the specified carrier.
lastUpdate (dateTime)	The last date and time anything for this carrier was updated.
carrierInfo	The carrier identifiers, names, addresses and phone numbers.
lastUpdate (dateTime)	The last date and time any of the data in this section was updated.
legalName (string)	The legal name that this carrier operates under. Length: 0 – 155
dbaName (string)	The Doing-Business-AS name of the carrier. Length: 0 – 155
dotNumber (int)	The FMCSA DOT number assigned to this carrier. Valid Values: 0 –
docket	The FMCSA MC/MX/FF number assigned to this carrier. DOT docket is made up of a prefix and a number. Prefix is a 2-character String. The number is a non-negative 32-bit integer.
prefix (string)	Docket Prefix. Not avaliable in the first release. Length: 0 – 2
number (int)	Docket Number of the company. Valid Values: 0 –

intrastate	The intrastate code given to this carrier. Only provided if the carrier is only a intrastate carrier. The intrastate type is made up of 1) The state the carrier is authorized to operate within. A 2-character string. 2) The state's assigned operating code for this carrier. A 15-digit number.
intrastateState	The state the carrier is authorized to operate within. See StateProvince
intrastateCode (string)	The state's assigned operating code for this carrier. Length: 0 – 15
duns (int)	The 9 digit duns number for this carrier. Valid Values: 1 – 999999999
scacCode (string)	The SCAC codes assigned to this carrier. Length: 0 – 4
website (anyURI)	Company website URL, max 256 characters.
physicalAddress	The physical address and phone numbers of this company.
phone	Best available phone number corresponding to the address. See PhoneNumber
tollFreePhone	Best available toll free phone number corresponding to the address. See PhoneNumber
fax	Best available Fax number corresponding to the address. See PhoneNumber
street (string)	Street refers to a street name, which is an identifying name given to a street. The street name forms part of the address. It's a String 0-200 character. Length: 0 – 200

city (string)	City refers to a city name, which is a relatively large and permanent settlement. It's a String with 0-50 character. Length: 0 – 50
stateProvince	The DOT does not always supply a state, particularly for carriers located in Mexico. See StateProvince
postalCode (string)	North American Postal Codes • Canada: LNL NLN • Mexico: NNNNN • US ZIP: NNNNN(-NNNN) The DOT does not always supply a postal code, particularly for carriers located in Mexico. Length: 5 – 10
countryCode	Based on two-letter ISO 3166 country codes for countries supported by TransCore. This is a 2-character String. Valid Values: `US`, `CA`, `MX`
mailingAddress	The mailing address and phone numbers of this company.
phone	Best available phone number corresponding to the address. See PhoneNumber
tollFreePhone	Best available toll free phone number corresponding to the address. See PhoneNumber
fax	Best available Fax number corresponding to the address. See PhoneNumber

street (string)	Street refers to a street name, which is an identifying name given to a street. The street name forms part of the address. It's a String 0-200 character. Length: 0 – 200
city (string)	City refers to a city name, which is a relatively large and permanent settlement. It's a String with 0-50 character. Length: 0 – 50
stateProvince	The DOT does not always supply a state, particularly for carriers located in Mexico. See StateProvince
postalCode (string)	North American Postal Codes • Canada: LNL NLN • Mexico: NNNNN • US ZIP: NNNNN(-NNNN) The DOT does not always supply a postal code, particularly for carriers located in Mexico. Length: 5 – 10
countryCode	Based on two-letter ISO 3166 country codes for countries supported by TransCore. This is a 2-character String. Valid Values: `US`, `CA`, `MX`
paymentInfo	The payment information for this carrier.
lastUpdate (dateTime)	The last date and time any of the data in this section was updated.
remitToName (string)	The name to make payments out to. Length: 0 – 155
remitToAddress	The address to send payments.

phone	Best available phone number corresponding to the address. See PhoneNumber
tollFreePhone	Best available toll free phone number corresponding to the address. See PhoneNumber
fax	Best available Fax number corresponding to the address. See PhoneNumber
street (string)	Street refers to a street name, which is an identifying name given to a street. The street name forms part of the address. It's a String 0-200 character. Length: 0 – 200
city (string)	City refers to a city name, which is a relatively large and permanent settlement. It's a String with 0-50 character. Length: 0 – 50
stateProvince	The DOT does not always supply a state, particularly for carriers located in Mexico. See StateProvince
postalCode (string)	North American Postal Codes • Canada: LNL NLN • Mexico: NNNNN • US ZIP: NNNNN(-NNNN) The DOT does not always supply a postal code, particularly for carriers located in Mexico. Length: 5 – 10
countryCode	Based on two-letter ISO 3166 country codes for countries supported by TransCore. This is a 2-character String. Valid Values: `US`, `CA`, `MX`
isFactoringCompany (boolean)	Indicates whether or not the remit-to name and address are for a factoring company.

interestInQuickPay (boolean)	Indicates interest in quick-pay plans.
references	Payment references that may be contacted.
companyName (string)	
contactName (string)	
phone	See PhoneNumber
contactInfo	The payment information for this carrier.
lastUpdate (dateTime)	The last date and time any of the contacts were updated.
contacts	List of personnel and their roles in the company.
role (string)	A carrier's contact's role. Length: 0 – 30
name (string)	A carrier's contact's name. Length: 0 – 155
title (string)	A carrier's contact's title. Length: 0 – 50
phone	See PhoneNumber
fax	See PhoneNumber
email (string)	A carrier's contact's email address. Length: 0 – 256
isPrimaryContact (boolean)	
services	The services that this carrier provides.

lastUpdate (dateTime)	The last date and time any of the data in this section was updated.
companyType	The type of company this is. Valid Values: `Corporation`, `Partnership`, `Sole Proprietorship`
yearFounded (int)	The year the company was founded. Valid Values: 1900 – 2200
womanOwned (boolean)	Indicates if the carrier is owned by a woman.
veteranOwned (boolean)	Indicates if the carrier is owned by a military veteran.
disabledVeteranOwned (boolean)	Indicates if the carrier is owned by a disabled military veteran. Only provided when dobVersion >1 is specified.
minorityOwnership	Indicates if the carrier is owned by a minority.
minorityOwned	
nmsdcCertified (boolean)	
certificateNumber (string)	
regionalCouncil (string)	
isSmartwayCarrier (boolean)	Indicates if carrier is a Smartway carrier.
isHazmatCertified (boolean)	Indicates if carrier is hazmat certified.
isRadioActiveCertified (boolean)	Indicates if carrier is certified to carry radioactive material.
isAceCertified (boolean)	Indicates if carrier is certified by ACE. Boolean type.
isCtpatMember (boolean)	Indicates if carrier is C-TPAT Member.

isTwicCardholder (boolean)	Indicates if carrier is a TWIC Cardholder.
isTsaRegistered (boolean)	Indicates if carrier is TSA Registered.
isResponsibleCareCarrier (boolean)	Indicates if carrier is a responsible care carrier.
carbCompliantTRUs (boolean)	Indicates if 100% of a carrier's Transport Refrigeration Units (TRU) are compliant with California's air quality regulations. Only provided if dobVersion >1 is specified.
fastCode (string)	The carrier's FAST Code. Length: 0 – 14
csaCode (string)	The carrier's CSA Code. Length: 0 – 4
iacCertification	The carrier's IAC information.
iacCode (string)	
expires (date)	
isUPSCarrier (boolean)	Indicates if carrier carries for UPS.
handlesRailDryage (boolean)	Indicates if carrier handles rail Dryage.
handlesPortDryage (boolean)	Indicates if carrier handles Port Dryage.
hasLTLCapabilities (boolean)	Indicates if carrier has LTL capabilities.
handlesExpeditedGround (boolean)	Indicates if carrier provides expedited ground services.
handlesAirFreightCartage (boolean)	Indicates if carrier handles air freight cartage.
handlesEdi (boolean)	Indicates if carrier can handle EDI.
handlesOversizeLoads (boolean)	Indicates if carrier can handle extra wide, long or heavy loads.

hasSpottedTrailers (boolean)	Indicates if carrier has spotted trailers.
hasCarbCompliantTrucks (boolean)	Indicates if carrier uses CARB compliant trucks.
doesTrailerInterchange (boolean)	Indicates if carrier does trailer interchanges.
doesPalletExchange (boolean)	Indicates if carrier does pallet exchanges.
doesMultiStops (boolean)	Indicates if carrier will make multiple stops.
doesParcels (boolean)	Indicates if the carrier deals with Parcels.
doesBrokerage (boolean)	Indicates if the carrier also has brokerage services.
brokerDocket	The broker docket number, if the carrier also has brokerage services.
prefix (string)	Docket Prefix. Not avaliable in the first release. Length: 0 – 2
number (int)	Docket Number of the company. Valid Values: 0 –
otherServices (string)	Other services that the carrier provides. Length: 0 – 256
airports (string)	The airports this carrier services. Length: 0 – 50
ports (string)	The ports this carrier services. Length: 0 – 50
equipment	The equipment that this carrier has.
lastUpdate (dateTime)	The last date and time any of the data in this section was updated.

numberOfPowerUnits (int)	The number of power units the carrier has. A signed 32-bit integer.
numberOfTeams (int)	The number of teams the carrier has. A signed 32-bit integer.
numberOfCompanyDrivers (int)	The number of company drivers the carrier has. A signed 32-bit integer.
numberOfOwnerOperators (int)	The number of owner operators that use their own authority, that the carrier works with. A signed 32-bit integer.
onBoardCommunications	The method used to communicate with the drivers. (Satelite, Cell, or Pager). Valid Values: `Not Specified`, `Cell`, `Satellite`, `Pager`
trailers	The types and number of trailers that a carrier has. Consisted of two parts: 1) The type of trailer Van, Reefer, or Flatbed. - A 6-15 character String. 2) The number of this type of trailer. - A signed 32-bit integer.
trailerType	The type of trailer Van, Reefer, or Flatbed. Valid Values: `Van 48ft`, `Reefer 48ft`, `Flatbed 48ft`, `Van 53ft`, `Reefer 53ft`, `Flatbed 53ft`, `Auto Carrier`, `Cargo Van`, `Container`, `Double Drop`, `Drop/Step`, `Dump Trailer`, `Hopper Bottom`, `Lowboy`, `Moving Van`, `Straight Truck`, `Stretch Trailer`, `Tanker`
numberOf (int)	The number of this type of trailer.

hasAirRide (boolean)	Indicates if the carrier has air ride trailers.
hasVented (boolean)	Indicates if the carrier has vented trailers.
hasHotshot (boolean)	Indicates if the carrier has hot shot trailers.
hasCurtains (boolean)	Indicates if the carrier has curtains.
hasInsulated (boolean)	Indicates if the carrier has insulated trailers.
hasConestoga (boolean)	Indicates if the carrier has conestoga trailers.
hasCoilRacks (boolean)	Indicates if the carrier has coil racks.
hasContainerLocks (boolean)	Indicates if the carrier has container locks.
hasTarps (boolean)	Indicates if the carrier has tarps.
hasEtrac (boolean)	Indicates if the carrier has ETRAC.
hasGarmentRacks (boolean)	Indicates if the carrier has garment racks.
hasPadWrap (boolean)	Indicates if the carrier has pad wrap.
hasStraps (boolean)	Indicates if the carrier has straps.
hasChains (boolean)	Indicates if the carrier has chains.
geographicCoverage	The geographic areas that this carrier services.
lastUpdate (dateTime)	The last date and time any of the data in this section was updated.
serviceInMexico (boolean)	Indicates service to the carrier provides service in Mexico.
operatingStateProvince	List of US states and Canadian provinces that the carrier operates within. See StateProvince
lanes	Lists the lanes that the carrier operates within.

origin	The starting point of the lane.
stateProvince	The state or province of the end point. See StateProvince
metroArea	If one is specified, the metropolitan area of the end point.
code (int)	The numeric code reserved for this metropolitan area. Valid Values: 0–
name (string)	The name describing this metropolitan area. Length: 0 – 40
code (int)	The numeric code reserved for this metropolitan area. Valid Values: 0–
name (string)	The name describing this metropolitan area. Length: 0 – 40
destination	The ending point of the lane.
stateProvince	The state or province of the end point. See StateProvince
metroArea	If one is specified, the metropolitan area of the end point.
code (int)	The numeric code reserved for this metropolitan area. Valid Values: 0–
name (string)	The name describing this metropolitan area. Length: 0 – 40
code (int)	The numeric code reserved for this metropolitan area. Valid Values: 0–

name (string)	The name describing this metropolitan area. Length: 0 – 40
taxInfo	The tax information for this carrier.
lastUpdate (dateTime)	The last date and time the tax info was updated.
One of:	
w9Info	The W9 tax form information provided by the carrier.
tin (int)	The federal tax id number. Valid Values: 10000000 – 999999999
type	The type of tax id provided, SSN or EIN. Valid Values: `EIN`, `SSN`
status	The status of the federal tax id: Pending, Valid, or Not-Valid. Valid Values: `Pending`, `Valid`, `NotValid`
name (string)	The carrier name as shown on their income tax return. Length: 0 – 155
businessName (string)	The carrier business name/disregard entity name, if different from above. Length: 0 – 155
street (string)	The street address. Length: 0 – 200
city (string)	The city address. Length: 0 – 50

stateProvince	The State or Province. See StateProvince
postalCode (string)	The Postal code. Length: 5 – 10
federalTaxClassification	The W9 Fedearal Tax Classification.
One of:	
classificationType	The simple tax classifications of the W-9 tax form. Valid Values: `Individual/sole proprietor`, `C Corporation`, `S Corporation`, `Partnership`, `Trust/estate`, `Exempt payee`, `Other`
limitedLiabilityType	The special limited liability company tax classification, which requires further describes the tax classification. Valid Values: `C Corporation`, `S Corporation`, `Partnership`
w8Info	The W8 tax form information provided by the carrier.
name (string)	The name of individual or organization that is the beneficial owner. Length: 0 – 155
typeOfBeneficialOwner	The type of beneficial owner. Valid Values: `Individual`, `Corporation`, `Disregarded`, `Partnership`, `Simple trust`, `Grantor Trust`, `Complex trust`, `Estate`, `Government`, `International organization`, `Central bank of issue`, `Tax-exempt organization`, `Private foundation`
permenantAddress	The permenant address.
phone	Best available phone number corresponding to the address. See PhoneNumber

tollFreePhone	Best available toll free phone number corresponding to the address. See PhoneNumber
fax	Best available Fax number corresponding to the address. See PhoneNumber
street (string)	Street refers to a street name, which is an identifying name given to a street. The street name forms part of the address. It's a String 0-200 character. Length: 0 – 200
city (string)	City refers to a city name, which is a relatively large and permanent settlement. It's a String with 0-50 character. Length: 0 – 50
stateProvince	The DOT does not always supply a state, particularly for carriers located in Mexico. See StateProvince
postalCode (string)	North American Postal Codes • Canada: LNL NLN • Mexico: NNNNN • US ZIP: NNNNN(-NNNN) The DOT does not always supply a postal code, particularly for carriers located in Mexico. Length: 5 – 10
countryCode	Based on two-letter ISO 3166 country codes for countries supported by TransCore. This is a 2-character String. Valid Values: `US`, `CA`, `MX`
mailingAddress	The mailing address, if different from permenantAddress.

phone	Best available phone number corresponding to the address. See PhoneNumber
tollFreePhone	Best available toll free phone number corresponding to the address. See PhoneNumber
fax	Best available Fax number corresponding to the address. See PhoneNumber
street (string)	Street refers to a street name, which is an identifying name given to a street. The street name forms part of the address. It's a String 0-200 character. Length: 0 – 200
city (string)	City refers to a city name, which is a relatively large and permanent settlement. It's a String with 0-50 character. Length: 0 – 50
stateProvince	The DOT does not always supply a state, particularly for carriers located in Mexico. See StateProvince
postalCode (string)	North American Postal Codes • Canada: LNL NLN • Mexico: NNNNN • US ZIP: NNNNN(-NNNN) The DOT does not always supply a postal code, particularly for carriers located in Mexico. Length: 5 – 10
countryCode	Based on two-letter ISO 3166 country codes for countries supported by TransCore. This is a 2-character String. Valid Values: `US`, `CA`, `MX`

foreignTaxId (string)	The foreign tax identifying number. Length: 0 – 50
foreignTaxIdType (string)	The type of foreign tax identifying number (BN or SIN). Only provided when dobVersion >1. Length: 0 – 3
tin (int)	The federal tax id number if they have one. Valid Values: 10000000 – 999999999
type	The type of tax id provided, SSN or EIN. Valid Values: `EIN`, `SSN`
insuranceAgentInfo	The information about the carrier's insurance agents.
lastUpdate (dateTime)	The last date and time the insurance agent information was updated.
agents	The types of coverage this agent provides.
coverageType	The types of coverage this agent provides. Valid Values: `CARGO`, `AUTO`, `GENERAL`, `WORKERS_COMP`
agent (string)	The name of the insurance agent's company. Length: 0 – 90
phone	The phone number of the insurance agent. See PhoneNumber
fax	The fax number for the insurance agent. See PhoneNumber
documents	The documents that this carrier provides.
created (dateTime)	The date and time the document was created.

documentType (string)	The type of document (Hazmat, Smartway or CTPAT) Length: 0 – 20
fileName (string)	The name of the file that the document is stored in. Length: 0 – 100
fileExtension (string)	The file extension that the document is stored in. Length: 0 – 4
documentName (string)	The name of the document. Length: 0 – 100
description (string)	A description of the document Length: 0 – 100
url (anyURI)	The URL from which this document may be downloaded. Not provided for inactive documents. max 256 characters.

7.7.3 LookupDobSignedCarriers

Lookup all carriers that have signed contracts through DAT Onboarding with the user. The intended usage is to synchronize the client's system with all relevant data available in the DAT Onboarding system. All carriers which have ever signed a contract with the customer and the contracts they have signed will be returned.

This request is only supported through the secure (HTTPS) endpoint. Requests through the insecure endpoint will return an error.

Request Fields	
dobVersion (int)	The version of response to receive. The default is 1. Set the version to the version of the XML schema that you built to. The version of this schema is 2.

Response Fields

signedCarriers	The list of carriers and the contracts that they have signed with the user.
carrierId (string)	The DAT-On-Boarding identifier for the carrier that signed contracts. Length: 0 – 40
signedContracts	The contracts that this carrier signed.
contract	The contract that was signed.
name (string)	The Name of the contract. Length: 0 – 100
description (string)	The description of the contract. Length: 0 – 100
version (int)	The version of the contract. Valid Values: 1 –
documentStatus	The status of the contract (Active or Inactive); Valid Values: `Active`, `Inactive`
signatory	The signature information for the contract.
name (string)	The Name of person signing a contract. Length: 0 – 100
title (string)	The title of the person signing a contract. Length: 0 – 100
signed (dateTime)	The date the contract was signed.

url (anyURI)	The URL from which this contract may be downloaded. Provided only when dobVersion >1 and for contracts signed through AssureSign. Max 256 characters.

7.8 Shared Types

The data structures described here are shared by multiple requests

7.8.1 Asset

assetId (string)	The historically unique ID assigned to this asset by the Freight Matching Service. Length: 0 – 8
status	The current status of this asset.
userId (int)	The ID of the owner of this freight matching entity. Valid Values: 0 –
startDate (dateTime)	When this entity becomes active. Valid Values: 2003-01-01T00:00:00.000Z –
endDate (dateTime)	When this entity is no longer active. Valid Values: 2003-01-01T00:00:00.000Z –
created	When this entity was created.
user (int)	
date (dateTime)	
updated	When this entity was last updated. Will initially be set to the same as 'created'.
user (int)	

date (dateTime)	
lastModified	When this entity was last modified. I.e., the latter of 'created' and 'updated'.
user (int)	
date (dateTime)	
expired (boolean)	True if this asset has expired and is no longer visible in searches.
One of:	
shipment	Shipment-specific information.
equipmentType	The type of equipment that is needed to haul this shipment. See EquipmentType
origin	The origination point of this shipment. See Place
destination	The destination point of this shipment. See Place
truckStops	All the information needed to advertise this shipment on the Truck Stop network. If omitted, this shipment will not be advertised on the Truck Stop Network.
One of:	
truckStopIds	
ids (int)	A list of TransCore/DAT identifiers for the truckstops the shipment should be displayed at. Valid Values: 0 – 9999
closest	Select the truckstops closest to the shipment origin. The number of selected locations is determined by the poster's subscription and contract.

alternateClosest	Select the truckstops closest to an alternate origin point. This option is typically used to display the shipment on truckstops in the desired lane. The number of selected locations is determined by the poster's subscription and contract.
alternateOrigin	Origin point. See Place
enhancements	Optional truck stop load monitor video enhancements for this shipment's advertisement. Valid Values: `Flash`, `Highlight`
posterDisplayName (string)	The short name of the poster's company, for display on the truck stop load monitors. If omitted, the short name of the poster's company will default to that contained in the poster's company record. Length: 0 – 8
rate	Payment rate in USD (flat-rate or per-mile).
baseRateDollars (float)	Rate in US Dollars (the rateBasedOn element defines whether this is a flat rate or per-mile). Valid Values: 0.0 – 99999.99
rateBasedOn	Per-mile or flat-rate. Valid Values: `Flat`, `PerMile`
rateMiles (int)	If the rate is based on per-mile, or flat-rate for a trip, this is the mileage used by the poster. It is supplied by the poster, and may not agree with HHG mileage or that computed by mileage vendors. Valid Values: 0 – 9999

equipment	Equipment-specific information.
equipmentType	The type of this equipment. See EquipmentType
origin	The origination point of this equipment. See Place
destination	The desired destination (point, area, or open).
One of:	
place	A specific location. See Place
area	A geographic area, defined by a list of states or a region.
stateProvinces	A list of states and/or provinces in the desired region. See StateProvince
zones	A list of one or more zones in the desired region. See Zone
open	Represents "anywhere".
postersReferenceId (string)	Poster-supplied reference ID for this asset. Length: 0 – 8
ltl (boolean)	For shipments, if set true, then this shipment will not occupy a full truck. For equipment, if set true, then this equipment does not have enough available capacity to carry a full shipment. The element "dimensions" should be consulted for cargo/capacity size.
comments (string)	Optional poster-supplied comment(s) for this asset, displayed on search match results and the Truck Stop Network. Length: 0 – 70

count (int)	The number of assets represented by this description. Will always be at least 1. Valid Values: 0 –
dimensions	For shipments, the dimensions of the cargo. For equipment, the dimensions of the available cargo capacity.
lengthFeet (int)	Length of the shipment or truck deck in feet. Valid Values: 1 – 199
weightPounds (int)	Weight of the shipment or truck carrying capacity in pounds. Valid Values: 1 – 199999
heightInches (int)	Height of the shipment or truck box in inches. Valid Values: 1 – 299
volumeCubicFeet (int)	Volume of the shipment or truck box in cubic feet. Valid Values: 1 – 9999
stops (int)	The number of stops this asset must make. Will always be at least 1. Valid Values: 0 –
availability	For shipments, when the cargo is available for pickup by a truck. For equipment, when the truck is available to pickup a shipment.
earliest (dateTime)	For shipments, the earliest the shipment can be picked up. For equipment, the earliest the equipment is available to pick up a shipment. Valid Values: 2003-01-01T00:00:00.000Z –

latest (dateTime)	For shipments, the latest the shipment can be picked up. For equipment, the latest the equipment is available to pick up a shipment. Valid Values: 2003-01-01T00:00:00.000Z –

7.8.2 EquipmentType

Valid Values

```
Auto Carrier
Container
Container, Insulated
Container, Refrigerated
Double Drop
Drop Deck, Landoll
Dump Trailer
Flatbed
Flatbed, Air-Ride
Flatbed, B-Train
Flatbed, Double
Flatbed, HazMat
Flatbed, Hotshot
Flatbed, Maxi
Flatbed w/Sides
Flatbed w/Tarps
Flatbed w/Team
Flatbed or Step Deck
Flatbed/Van/Reefer
Hopper Bottom
Lowboy
Moving Van
Pneumatic
Power Only
Reefer
Reefer, Air-Ride
Reefer, Double
Reefer, HazMat
Reefer, Intermodal
Reefer, Logistics
Reefer w/Team
Removable Gooseneck
```

Step Deck
Stretch Trailer
Tanker, Aluminum
Tanker, Intermodal
Tanker, Steel
Truck and Trailer
Van
Van, Air-Ride
Van, Conestoga
Van, Curtain
Van, Double
Van, HazMat
Van, Hotshot
Van, Insulated
Van, Intermodal
Van, Lift-Gate
Van, Logistics
Van, Open-Top
Van, Roller Bed
Van, Triple
Van, Vented
Van w/Team
Van or Flatbed w/Tarps
Van or Flatbed
Van or Reefer
Insulated Van or Reefer
Reefer or Vented Van
Flatbed, w/Chains
Reefer, w/Pallet Exchange
Van, w/Blanket Wrap
Lowboy or RGN
Van, w/Pallet Exchange
Step Deck or RGN
Conveyor
Flatbed, Over Dimension
Lowboy, Over Dimension
Conestoga
Flatbed Conestoga
Stepdeck Conestoga
Straight Box Truck

7.8.3 Place

One of:

postalCode	USPS or Canada Post postal code. The Freight Matching Service will use the city+state/province assigned to this postal code).
country	Country code. Valid Values: `US`, `CA`, `MX`
code (string)	ZIP or postal code. The hyphen in 9-digit US ZIP codes and the space in Canadian postal codes are optional. Length: 5 – 10
cityAndState	City + state/province. The Freight Matching Service will match the city + state/province against its atlas of known cities. If multiple cities of the same name exist in the state/province, it will use the optional county to disambiguate).
city (string)	City name. Length: 0 – 30
stateProvince	State or province. See StateProvince
county (string)	County. If present, the county is used to disambiguate between multiple similarly named cities in the state or province. Length: 0 – 30

namedPostalCode	Postal code + city + state/province. The Freight Matching Service will internally use the latitude/longitude coordinates of the city assigned to the postal code, but will list the specified city + state/province on assets/searches. This option is useful when the popularly named city is not what USPS/CP assigns to that postal code code (e.g., 97222 is officially Portland, Oregon, but residents of that ZIP refer to their locale as Milwaukie).
city (string)	City name. Length: 0 – 30
stateProvince	State or province. See StateProvince
county (string)	County. If present, the county is used to disambiguate between multiple similarly named cities in the state or province. Length: 0 – 30
postalCode	
country	Country code. Valid Values: `US`, `CA`, `MX`
code (string)	ZIP or postal code. The hyphen in 9-digit US ZIP codes and the space in Canadian postal codes are optional. Length: 5 – 10
coordinates	Latitude/longitude coordinates. The Freight Matching Service will internally assign the closest known city to those coordinates. Note that this could result in a city being selected that is not in the same state/province as the coordinates.
latitude (float)	Latitude of the desired point. Valid Values: 13.00 – 86.00

longitude (float)	Longitude of the desired point. Valid Values: -177.00 – -52.00
namedCoordinates	City + state/province + coordinates. DAT Connexion will use the city, state/province, and coordinates to resolve to a known place. If an exact match is not found, Connexion will use variations in the city spelling and proximity to the coordinates to resolve. If no resolution is found, Connexion will accept the specified city spelling as-is, and use the specified coordinates and state/province (although road mileages based on this city will not be possible).
latitude (float)	Latitude of the desired point. Valid Values: 13.00 – 86.00
longitude (float)	Longitude of the desired point. Valid Values: -177.00 – -52.00
city (string)	City name. Length: 0 – 30
stateProvince	State or province. See StateProvince

7.8.4 SearchCriteria

assetType	Type of asset to search for (shipment or equipment). Valid Values: `Shipment`, `Equipment`

equipmentClasses	Match on assets belonging to any equipment class in this list. Each equipment class represents a predefined set of equipment types. If you need finer-grained control of which types are returned, you can specify one or more EquipmentTypes using the 'equipmentTypes' element. Note that the 'Conestoga' class introduced in late 2012 is only supported for API versions ≥ 2 (see LoginRequest). See EquipmentClass
equipmentTypes	Match on assets belonging to any equipment type in this list. Search by equipment type(s) was introduced in late 2012, and is only supported for API versions ≥ 2 (see LoginRequest). See EquipmentType
ageLimitMinutes (int)	Search for assets that are no older than this number of minutes. The age of an asset is the time since it was created, or last updated, or last refreshed, whichever is latest. Note: An age of zero will result in no matches; this is usually not desireable. If omitted, defaults to 60 minutes. NOTE: if an alarm is being created as part of a PostAsset, then this field has no effect: an initial lookback search is not possible when posting an asset. Valid Values: $0-5940$
origin	Search origin. Note that the origin and destination cannot both be 'open'. I.e., they both cannot specify a GeoCriteria member that is of type 'SearchOpen'. All such search requests will be rejected with an error. See GeoCriteria
destination	Search destination. See GeoCriteria

availability	Search for assets that become available during specified limits. Assets available prior to "earliest" (i.e., assets whose 'latest' availability is prior to this criteria's 'earliest') will be excluded. Assets that are not available until after 'latest' (i.e., assets whose 'earliest' availability is after this criteria's 'latest') will be excluded. If this specification is omitted, then it will default to (now, now+24hrs). Note 1: "latest" (if specified) must be on or after 'earliest' (This cannot be enforced with schema, since schema doesn't have relational operators). A violation will result in defaulting 'latest' to 24 hours after 'earliest'. If 'earliest' is not specified, then it will be defaulted to 24 hours prior to 'latest'. Note 2: 'earliest' (if specified) must be on or prior to 'latest' (this cannot enforce with schema, since schema doesn't have relational operators). A violation will cause defaulting 'earliest' to 24 hours prior to 'latest'. If 'latest' is not specified, then it will be defaulted to 24 hours after 'earliest'.
earliest (dateTime)	For shipments, the earliest the shipment can be picked up. For equipment, the earliest the equipment is available to pick up a shipment. Valid Values: 2003-01-01T00:00:00.000Z –
latest (dateTime)	For shipments, the latest the shipment can be picked up. For equipment, the latest the equipment is available to pick up a shipment. Valid Values: 2003-01-01T00:00:00.000Z –

includeLtls (boolean)	If set true, when searching for shipments, include shipments that are marked by their poster as "LTL" (i.e., the shipment does not occupy all the capacity of a truck). If set true, when searching for equipment, include those that are marked by their poster as "LTL" (i.e., only part of the equipment's capacity is available). If omitted, defaults to true.
includeFulls (boolean)	If set true, when searching for loads, include shipments that are "full" loads (i.e., non-LTLs). If set true, when searching for trucks, include trucks that are not marked as LTL (i.e., they are available to carry a full shipment). If omitted, defaults to true.
limits	When searching for shipments, specifies the available capacity of the searcher's equipment (i.e., the maximum available cargo size). When searching for equipment, specifies the minimum cargo capacity needed to haul the searcher's shipment.
lengthFeet (int)	Length of the shipment or truck deck in feet. Valid Values: 1 – 199
weightPounds (int)	Weight of the shipment or truck carrying capacity in pounds. Valid Values: 1 – 199999
heightInches (int)	Height of the shipment or truck box in inches. Valid Values: 1 – 299
volumeCubicFeet (int)	Volume of the shipment or truck box in cubic feet. Valid Values: 1 – 9999

excludeOpenDestinationEquipment (boolean)	If set true, trucks posted with a destination of `Open` will be excluded. On shipment searches, this element is ignored. Note: This option is currently only supported for the CountAssetsRequest. It is ignored on the normal CreateSearchRequest and on alarms. If omitted, defaults to false.

7.8.5 EquipmentClass

Valid Values

```
Dry Bulk
Containers
Decks, Standard
Decks, Specialized
Flatbeds
Other Equipment
Reefers
Tankers
Vans, Standard
Vans, Specialized
Hazardous Materials
Conestoga
```

7.8.6 StateProvince

Valid Values

```
AB
AG
AK
AL
AS
AZ
AR
BC
BJ
BS
CA
```

CH
CI
CL
CO
CP
CT
CU
DC
DE
DF
DG
EM
FL
GA
GJ
GR
GU
HG
HI
IA
ID
IL
IN
JA
KS
KY
LA
MA
MB
MD
ME
MH
MI
MN
MO
MR
MS
MT
NA
NE
NL

NV
NB
NH
NJ
NM
NY
NF
NC
ND
NT
NS
NU
OA
OH
OK
ON
OR
PA
PE
PQ
PR
PU
QA
QR
RI
SC
SD
SI
SK
SL
SO
TA
TL
TM
TN
TX
UT
VA
VI
VL
VT

WA
WV
WI
WY
YC
YT
ZT

7.8.7 Zone

Valid Values
New England
North East
Mid-Atlantic
South East
Mid-West
North Central
Central
South
Mountain
West
Canada
Western Canada
Eastern Canada
Central Canada
Mexico
Northern Mexico
Contiguous USA

7.8.8 GeoCriteria

One of:

area	Search for postings matching a set of states and/or zones.
stateProvinces	A list of states and/or provinces in the desired region. See StateProvince

zones	A list of one or more zones in the desired region. See Zone
radius	Search for postings within a radius of a specific point. The radius measurement defaults to road mileage, but that default can be overridden to specify air mileage if desired. Connexion will enforce TransCore limitations on maximum radii.
place	Specifies the target location at the center of the search radius. See Place
radius	Radius around the search location.
miles (int)	Mileage. Valid Values: 0 –
method	Mileage calculation method. Valid Values: `Air`, `Road`, `RoadProMiles`, `RoadAlk`
open	Search for any postings, regardless of geography. Note that either an origin or destination must be provided. Open → open searches are not supported.

7.8.9 Search

searchId (string)	The historically unique ID assigned to this search by the Freight Matching Service. Length: 0 – 8
status	The current status of this search.
userId (int)	The ID of the owner of this freight matching entity. Valid Values: 0 –

startDate (dateTime)	When this entity becomes active. Valid Values: 2003-01-01T00:00:00.000Z –
endDate (dateTime)	When this entity is no longer active. Valid Values: 2003-01-01T00:00:00.000Z –
created	When this entity was created.
user (int)	
date (dateTime)	
updated	When this entity was last updated. Will initially be set to the same as 'created'.
user (int)	
date (dateTime)	
lastModified	When this entity was last modified. I.e., the latter of 'created' and 'updated'.
user (int)	
date (dateTime)	
expired (boolean)	True if this asset has expired and is no longer visible in searches.
specification	The search criteria used for this search. See SearchCriteria

7.8.10 MatchingAsset

asset	The matching asset. See Asset

tripMileage	The total mileage (and how it was computed: road or air miles) of the one-way origin → destination trip. Omitted if the origin and destination are not both points.
miles (int)	Mileage. Valid Values: 0 –
method	Mileage calculation method. Valid Values: `Air`, `Road`, `RoadProMiles`, `RoadAlk`
originDeadhead	Deadhead mileage from the origin of the searcher to the origin of the asset, i.e., from the current origin of a truck, to the origin of the shipment. If deadhead mileage cannot be computed (e.g., origin is not a point), then this item is omitted.
miles (int)	Mileage. Valid Values: 0 –
method	Mileage calculation method. Valid Values: `Air`, `Road`, `RoadProMiles`, `RoadAlk`
destinationDeadhead	Deadhead mileage from the destination of the searcher to the destination of the asset. i.e., from the current destination of a truck, to the destination of the shipment. If deadhead mileage cannot be computed (e.g., destination is not a point), then this item is omitted.
miles (int)	Mileage. Valid Values: 0 –

method	Mileage calculation method. Valid Values: `Air`, `Road`, `RoadProMiles`, `RoadAlk`
callback	Callback information on the poster of the matching asset, so the searcher can contact the poster.
userId (int)	The ID of the user that posted this asset. Valid Values: 0 –
One of:	
phone	
phone	See PhoneNumber
email	
email (string)	
name	The first and last name of the poster.
firstName (string)	
middleName (string)	
lastName (string)	
title (string)	
prefix (string)	
suffix (string)	
initials (string)	
companyName (string)	The full 50-char name of the company to which the poster belongs. Length: 0 – 50

displayCompany (string)	The short 8-char name of the company to which the poster belongs. Length: 0 – 8
postersStateProvince	The state/province in which the poster resides. See StateProvince
creditScore	Credit score of the poster's company
score (int)	
daysToPay (int)	
scoreTimeStamp (dateTime)	
thirdPartyInfo	Third party information on the poster's company.
rmisGreenLight (boolean)	
nmftaMember (boolean)	
ooidaMember (boolean)	
tiaP3Member (boolean)	
assurable (boolean)	
rivieraGreenLight (boolean)	
dotIds	DOT ID information (DOT #, Docket #s, etc.)
dotNumber (long)	
brokerMcNumber (int)	
carrierMcNumber (int)	
freightForwarderMcNumber (int)	

mexicoMcNumber (int)

7.8.11 PhoneNumber

countryCode (string)	Country code
number (string)	The primary number
extension (string)	Extension

7.8.12 AlarmSearchCriteria

originRadius	Radius around the asset's origin.
miles (int)	Mileage. Valid Values: 0 –
method	Mileage calculation method. Valid Values: `Air`, `Road`, `RoadProMiles`, `RoadAlk`
destinationRadius	Radius around the asset's destination. If the asset has a non-Point destination (e.g., an Equipment with an Area/Empty destination), then this field should be omitted; it will be ignored anyway.
miles (int)	Mileage. Valid Values: 0 –
method	Mileage calculation method. Valid Values: `Air`, `Road`, `RoadProMiles`, `RoadAlk`

ageLimitMinutes (int)	If this is non-zero, then the Alarm will do an initial live lookback-search, for assets that are no older than the specified number of minutes. If set to zero or omitted, then no initial lookback-search will be performed. NOTE: if an alarm is being created as part of a PostAsset, then this field has no effect: an initial lookback search is not possible when posting an asset. Valid Values: 0 – 5940
maxMatches (int)	Maximum number of matches to return over the lifetime of this alarm. Default is 25 matches. Valid Values: 0 – 3000
lifetimeMinutes (int)	Lifetime of this alarm. If omitted the lifetime will default to system policy. Actual lifetimes are subject to system policies, particularly for end-of-day termination. Valid Values: 1 – 1440
referenceId (string)	Client-supplied reference ID. This ID will be included in all alarm notifications generated by this alarm. If this alarm-specific referenceId is left unspecified, the referenceId of the associated asset will be used instead (if that referenceId is specified). Note that Connexion will *not* enforce any uniqueness constraints on `referenceId` (unlike `postersReferenceId` on a posting). Length: 0 – 8

notifyAllApplications (boolean)	If set to true, match, cancellation, and termination notifications produced by this alarm will be sent not only to the application creating the alarm, but also to any other applications into which the recipient (the alarm owner or members of the owner's sharing group) is currently logged in. This attribute should only be used in the rare scenarios in which sharing alarm matches across multiple applications is desirable. Defaults to false.

7.8.13 RatedLane

originGeography (string)	A human-readable description of the origin area used to compute the rate estimate. E.g., '982 Postal Code', 'Houston Market Area', 'Western Canada'. This element is provided as an additional means for a user to evaluate the significance of the rate estimate, but may be removed in the future as TransCore's rating algorithms evolve to utilize data from outside specific market areas and to incorporate long-term patterns. Note: Rate providers generally do not compute lane rates based on point-to-point rates, but rather on geographical areas that contain the requested points. This lane may differ from the requested point-to-point lane because of how the rate provider associates postal codes with cities (especially for suburban neighborhoods in larger urban areas) or due to city name abbreviation standards.
destinationGeography (string)	A human-readable description of the destination area used to compute the rate estimate. See `originGeography`.

8 Request / Response Layout and Error Handling

SOAP provides a `Fault` type, a standard mechanism for handling severe errors. The Web Service Interoperability specifications require that SOAP faults be returned in certain circumstances, including XML Schema validation failure (e.g., providing an alphanumeric string in an element defined to be floating-point). Connexion follows the WSI requirements and returns a SOAP fault in such cases. Most client-side libraries translate a SOAP Fault into an exception of some sort and report it to the calling code.

SOAP allows considerable flexibility for handling of less-severe errors. Many Connexion requests support multiple contained operations. For example, a **PostAsset** request can contain a batch of up to 100 separate postings. In this case, the SOAP Fault mechanism is inappropriate — some postings might succeed while others fail, and the client needs accurate error reporting for each posting in a batch.

To accommodate this case, Connexion requests follow a hierarchical structure and depend on a consistent naming convention. Each request/response pair is named according to the `XyzRequest` / `XyzResponse` pattern. A `Request` contains one or more `Operations`, and the resulting `Response` contains a `Result` for each `Operation`.

The `Result` consists of either a `SuccessData` (including the normal results of the request) or a `ServiceError` (specifying the reason for the error).

For example, see the following request/response pair:

```
<Envelope
    xmlns="http://schemas.xmlsoap.org/soap/envelope/"
    xmlns:tcor="http://www.tcore.com/TcoreHeaders.xsd"
    xmlns:tcor1="http://www.tcore.com/TcoreTypes.xsd"
    xmlns:tfm="http://www.tcore.com/TfmiFreightMatching.xsd">
  <Header>
  ...
  </Header>
  <Body>
    <tfm:postAssetRequest>
      <tfm:postAssetOperations>
        <tfm:shipment>
        ...
        </tfm:shipment>
      </tfm:postAssetOperations>
      <tfm:postAssetOperations>
        <tfm:equipment>
        ...
        </tfm:equipment>
      </tfm:postAssetOperations>
```

```
      <tfm:postAssetOperations>
        <tfm:shipment>
        ...
        </tfm:shipment>
      </tfm:postAssetOperations>
    </tfm:postAssetRequest>
  </Body>
</Envelope>

<Envelope xmlns="http://schemas.xmlsoap.org/soap/envelope/">
  <Header/>
  <Body>
    <tfm:postAssetResponse xmlns:tfm="http://www.tcore.com/TfmiFreightMatching.xsd">
      <tfm:postAssetResults>
        <tfm:postAssetSuccessData>
        ...
        </tfm:postAssetSuccessData>
      </tfm:postAssetResults>
      <tfm:postAssetResults>
        <tcor:serviceError xmlns:tcor="http://www.tcore.com/TcoreTypes.xsd">
        ...
        </tcor:serviceError>
      </tfm:postAssetResults>
      <tfm:postAssetResults>
        <tfm:postAssetSuccessData>
        ...
        </tfm:postAssetSuccessData>
      </tfm:postAssetResults>
    </tfm:postAssetResponse>
  </Body>
</Envelope>
```

In this example, the first and third postings succeeded, but the second failed.

Error Handling

As a Connexion client, it is important that you handle errors smoothly and report them appropriately to the user. Your code should check each `Result` for the corresponding `SuccessData`. If a `ServiceError` is included instead, you should log the contained error code and message and (if appropriate) report the detailed error message to the user.

Common error codes and messages are described in Appendix A

9 FAQ

9.1 Where is the WSDL?

The production WSDL is at:
http://transcoreservices.com:8000/wsdl/TfmiFreightMatching.wsdl

The test system WSDL is available at:
http://cnx.test.dat.com:9280/wsdl/TfmiFreightMatching.wsdl.

9.2 How should the `SessionToken` be used?

The intent of the `LoginRequest` is that you cache the resulting `SessionToken` and use it for multiple subsequent 'real' requests, saving the authentication and authorization time on subsequent requests. A Session Token is analogous to a session cookie in a web application, and you can generally treat it in the same way. A user can have at most one active session at a time—a login will invalidate a user's existing session.

Logging in separately for each 'real' request is not recommended. It can introduce a race condition in threaded or multi-process code that might cause you errors under load. e.g., consider these two scenarios:

This ordering is fine:

1. Thread 1 - `Login` (returns a session)
2. Thread 1 - `PostAsset`/`CreateSearch`/`LookupCarrier`/etc. (works as expected)
3. Thread 2 `Login` (returns a session)
4. Thread 2 `PostAsset`/`CreateSearch`/`LookupCarrier`/etc. (works as expected)

However, if the independent threads are scheduled differently:

1. Thread 1 - `Login` (returns a session)
2. Thread 2 `Login` (returns a session, but invalidates Thread 1's session)
3. Thread 2 `PostAsset`/`CreateSearch`/`LookupCarrier`/etc. (works as expected)
4. Thread 1 - `PostAsset`/`CreateSearch`/`LookupCarrier`/etc. (**Fails with an invalid session error**)

Even if your code is not threaded (or is synchronized to avoid the race condition above), the extra SOAP requests will add latency and slow down your response times, since each post or search now requires two HTTP requests over the Internet instead of one.

We recommend that you login once per day and reuse the same session token for all your subsequent requests. The session lifetime is 12 hours, so once per business day is usually sufficient. If you ever receive an invalid session error, it's appropriate (and expected) that you'll re-login to obtain a new token.

9.3 How do I specify a location?

Locations can be specified in a variety of ways. These choices are roughly in order of decreasing ambiguity:

1. Postal code only: Connexion will use the city+state/province assigned to this postal code.

2. City + state/province: Connexion will match the city + state/province against its atlas of known cities. If multiple cities of the same name exist in the state/province, it will use the optional county to disambiguate.

3. Postal code + city + state/province: Connexion will internally use the latitude/longitude coordinates of the city assigned to the postal code, but will list the specified city + state/province on assets/searches. This option is useful when the popularly named city is not what USPS or Canada Post assigns to that postal code (e.g., 97222 is officially Portland, Oregon, but residents of that ZIP refer to their locale as Milwaukie).

4. Latitude/longitude coordinates (Connexion will internally assign the closest known city to those coordinates. Note that this could result in a city being selected that is not in the same state/province as the coordinates.

5. City + state/province + coordinates (The Freight Matching Service will use the city, state/province, and coordinates to resolve to a known place. If an exact match is not found, FMS will use variations in the city spelling and proximity to the coordinates to resolve. If no resolution is found, FMS will accept the specified city spelling as-is, and use the specified coordinates and state/province (although road mileages based on this city will not be possible)).

   Note: places will always be returned from the system as `namedCoordinates`, saving the client from having to deal with the variety of choices. This applies to lookups, search results, etc.

9.4 How do you determine distance between city/state pairs and/or postal codes?

We resolve a postal code to the primary city for that code (using the USPS or Canada Post 'primary' label). Once we have a pair of known points, we compute road mileage distance between those two points.

9.5 How are city names abbreviated?

The following abbreviation rules are applied sequentially, until the abbreviated name fits in 14 characters.

1) First Words

EAST	→	E
FORT	→	FT
GR	→	GRAND
MOUNT	→	MT
MOUNTAIN	→	MTN
NORTH	→	N
SAINT	→	ST
SAINTE	→	ST
SNTA	→	SANTA
SOUTH	→	S
STE	→	ST
WEST	→	W

2) Main words (not first, not last)

CHRSTI	→	CHRISTI
ISLAND	→	IS
PNT	→	PT
POINT	→	PT
POINTE	→	PT
PRAIRIEDUCHEIN	→	PRAIRIEDUCHIEN

3) Final words (last word in a multi-word city name)

BEA	→	BCH	ISLE	→	IS
BEAC	→	BCH	JUNCTION	→	JCT
BEACH	→	BCH	KENSINGTON	→	KENSINGTN
BLUFFS	→	BLFS	MILLS	→	MLS
CENT	→	CTR	MOUNT	→	MT
CENTER	→	CTR	MOUNTAIN	→	MTN
CENTRE	→	CTR	PLAIN	→	PLNS
CIT	→	CY	PLAINS	→	PLNS
COMMER	→	CMMRCE	PRUSSI	→	PRUSS
CREK	→	CK	SPRIN	→	SPGS
CREEK	→	CK	SPRING	→	SPG
CTH	→	CH	SPRINGS	→	SPGS
CUCAMON	→	CUCMNGA	SPRS	→	SPGS
FAL	→	FLS	SPS	→	SPGS
FALL	→	FLS	STATIO	→	STA
FALLS	→	FLS	STATION	→	STA
FURNA	→	FURNC	TOWNSHI	→	TWP
GATEW	→	GTWY	TOWNSHIP	→	TWP
HARBO	→	HBR	VAL	→	VLY
HEIGH	→	HTS	VALLEY	→	VLY
HEIGHT	→	HTS	VILG	→	VLG
HEIGHTS	→	HTS	VILL	→	VL
INDUST	→	INDUS			

9.6 Why do we see a discrepancy between search results returned through Connexion and through the Power client?

The power client specifies radii in air miles instead of road miles. Most other discrepancies are the result of mismatches in search criteria.

9.7 Why can't I update attribute x on a posting?

Only a few fields are updateable (see UpdateAsset). Some limitations are in place for technical reasons or to maintain consistent and interpretable pricing for each posting, but most limitations are to reduce confusion about posting attributes. For instance, Connexion does not allow changing the origin or destination of a posting — the system considers a posting with altered origin/destination to be a new posting, so in this case it is appropriate (and expected) that you will delete the original posting and re-post to the correct locations.

9.8 What is a 'sharing group'?

A sharing group allows users to share their postings and work on them together. This includes sharing the alarm matches that are returned on those postings. If one user in a

sharing group enters a posting, others in the group can 1) see and manipulate the posting; and 2) see and act on any alarm matches returned for alarms based on the posting.

9.9 How do I search for carriers willing and able to carry HazMat?

This is currently not supported. It is possible to post a truck using one of our Hazmat equipment types (Flatbed, HazMat; Reefer, HazMat; and Van, HazMat), but carriers rarely do so. The carrier is generally willing to haul anything and doesn't want to preclude non-HazMat loads. TransCore does not (and in fact cannot) force HazMat-capable carriers to post their trucks as HazMat types.

However, although searching for HazMat-capable carriers is difficult, posting your HazMat load is generally **very** effective. Carriers who are qualified to haul HazMat will seek out loads labeled as such, since they're likely to come with a premium price tag. So, in general, a HazMat load posting on a popular lane will get prompt response, often from a number of carriers.

9.10 Are locations on search results always specified using the 'namedCoordinates' option?

Yes.

9.11 Can I refer to another user's postings by Poster's ReferenceId?

No. `UserReferenceId` is enforced to be unique on a per-user basis (a single user cannot have two postings with the same `UserReferenceId`, but two separate users can each have a posting with identical `UserReferenceIds`). This prevents ReferenceId conflicts between separate logistics systems or user naming conventions (since two users might each have a posting they refer to as '1001', for example).

That being the case, it doesn't make sense to operate on another user's postings by `ReferenceId`. If you want to operate on another user's posting, you must refer to it by `AssetId`, since `AssetIds` are the only identifiers we guarantee to be globally unique (and thus the only way we can determine with certainty which posting you want to operate on).

9.12 Why can't I validate a schema saved from a browser?

Most browsers are XML-savvy enough to expand entity references (e.g. `&` → `&`), so the version displayed in the browser is not the 'real' schema. Either use the 'Download this link' option in a browser or use a tool like `wget` or `curl` to download the file without allowing the browser to corrupt it.

9.13 Why do my carrier queries differ in production and test?

Our production environment is updated nightly with government data and throughout the day with insurance data from our TransCore Insurance Monitoring team. The test environment is only updated occasionally, so queries to the two environments will often return different responses.

9.14 Do I need to send all the SOAP headers defined in `TcoreHeaders.xsd`?

No. The `SessionHeader` is required on all requests following the `LoginRequest`. All other headers are optional.

9.15 When are expired loads purged from your DB? Why do I sometimes get a 'duplicate reference ID' error when trying to re-post a load?

All Connexion postings will expire at 23:59:59 Pacific time on the next business day following their creation. They are not purged automatically from our database until at least 48 hours after their expiration. That allows a user to 'resurrect' an expired asset in the event they decide to expand the availability window.

Expired assets which still exist in our database are visible in Connexion and in other TransCore applications. Uniqueness of the `UserReferenceId` field is enforced on expired assets as well as on active ones. That is, if a user already has an asset in the system with a particular `referenceId`, he will not be able to post a second asset with the same `referenceId` before deleting the first asset. In that case, he/she should either:

1. Update the original asset to expand the availability window (which makes sense if it's truly the same asset) Note: This option isn't possible in Connexion. The thick client simulates it using #2 (below).

2. Delete the original asset and post a new one (which makes sense if you want to re-use a referenceId on a new asset)

   Once an expired asset is removed from our system (48+ hours after expiration), attempts to update it will fail. In that case, re-posting the asset is appropriate.

   Purging of expired assets is a nightly batch job, so you'll see assets which expired 48+ hours ago until the end of the day and then they will disappear by the next morning.

9.16 Can we tell whether a search result is from the 'live' or 'daily' database?

No.

9.17 Should I delete my searches?

In general, the answer is no. This request is provided specifically for those users who lookup their own historical search activity and want to exclude certain searches from those lookups.

9.18 How should we deliver alarm matches to web-based clients?

Goal: receive and display Alarm Matches (results of a forward-looking search) promptly in a web application. For background, see Section 6.

For a desktop thick-client interface, the delivery mechanism is fairly simple: the application maintains a persistent TCP connection to a TransCore server, which notifies the desktop application in real-time of relevant matches.

Alternatively, a client-server TMS might implement a SOAP web service which TransCore can use to push out alarm matches. Note that this requires an open firewall port to the TMS server.

Historically, it's been more difficult for a web application to use similar architectures - browser-based apps are generally stateless and do not maintain persistent connections; and even if the browser-based app is able to open a network server socket, it's impractical to expect firewall support for that route.

Our recommended architecture for a hosted web-based TMS looks something like this: On the server, implement the aforementioned SOAP web service, so TransCore can notify the server of matches for all users in real-time. The server then has several options for passing that notification on to the browser. In decreasing preference order:

- **WebSockets** (http://en.wikipedia.org/wiki/WebSocket) This is a fairly new technology, not yet implemented in all browsers, which allows a persistent connection from the browser to the server. The server can use this connection to push matches to the browser, which would then update user screens using JavaScript code. The current WebSockets standard is supported in Firefox 6+, Chrome 14+, and Safari 5+. IE 10 is expected to include support as well, but at the time of this writing, its release date is unknown.

- **Polling** The browser can poll in the background using whatever AJAX library and infrastructure you prefer. This obviously isn't quite as real-time as a persistent connection, and increases the load on your servers. We've generally stayed away from it for the latter reason - we're tasked with supporting 10-20k simultaneous users and delivering upwards of 1000 alarm matches per second during peak hours. But if you don't need to scale quite to that level, a polling solution should be practical.

- **Browser Plugin Integration** Flash and Java both support persistent connections and both are supported by a wide array of older browsers. There are JavaScript libraries which allow JS/plugin integration with both. So you could in theory:

1. Create a persistent connection using a Flash or Java applet
2. Notify JS from that applet whenever a match arrives
3. Use JS to update the UI

We recommend WebSockets as the preferred approach (with fallback to polling for browsers that do not support WebSockets), but the alternative architectures are also effective.

9.19 How do we specify the Alarm URL for TransCore to post alarm matches back to?

There are two ways to specify this URL:

1. If your endpoint URL generally remains consistent, and particularly if it is shared by all users in your company or office, we recommend having a representative from our Technical Support department set it for you. You can email it to techsupportteam-leads@dat.com.

2. Call the `UpdateAlarmUrl` request to set the endpoint URL for a user. If your endpoint URL changes regularly, or if you want to use separate URLs for each user, we recommend implementing the functionality to call this request, allowing you to automate the process of changing or adding URLs without requiring a phone call or email. However, most customers find that a single static URL is sufficient, saving the effort of implementing this call. So we recommend you consider option 1 first.

9.20 If a user changes their TransCore password in the administration pages of another application, will their Connexion login fail?

Yes. If the user changes their password, that new information does need to be reflected in the next Connexion login call. We recommend that in the event of a login failure, your application request the updated password from the user (e.g., via a dialog box).

If that technical solution is not possible, this problem can be addressed by training your users to alert the application administrators when they update their password so they can update the user/password mapping manually.

9.21 Can I periodically delete all postings and re-post available loads or trucks?

No. This will negatively impact system performance, and so is not allowed. Clients must track their postings and delete specific loads which are no longer available (e.g. the load was covered) or update the information for any loads which have changed.

9.22 Can I implement searching without also posting loads or trucks?

No. All applications that support searching for loads or trucks via Connexion must also support posting loads and/or trucks using Connexion, as appropriate for the customer.

9.23 How do I test asynchronous events in DAT Onboarding?

Events in the DAT Onboarding application, including creation of new carriers, profile updates, contract updates, etc., are created outside of Connexion. To enable testing of your event integration, we have set up a program to create events at specified intervals. You can retrieve these events by polling with the LookupDobEvents request and retrieve data for the referenced carriers using the LookupDobCarriers request. The carriers listed below can also be retrieved using the LookupDobSignedCarriers request.

We will populate events for the following carriers:

Legal Name	**CarrierId**
F. B. T. INC	d235409d-9f48-4d99-a208-db4e03582902
ABOVE ALL TRANSPORT LLC	6140d5de-4f23-4611-a1e3-06c57bd496d1
A & A EXPRESS LLC	2f69a23b-592c-47f6-a33c-966c4be46dac
ACE DORAN HAULING & RIGGING CO	e1402f1d-a4e6-4e78-b53a-4989489a4d5d
OLD DOMINION FREIGHT LINE INC	6d1fc081-3212-4e5d-bad1-2ec7bcf1b723
C J LOGISTICS LLC	8ef74472-bdd6-4ec9-977e-ce3f35234f1c

Events will be populated each day at 8 AM, 10 AM, 12 PM and 2 PM EST. The events at each interval will be different. However, the events populated at each interval will be the same each day. So you should expect events at 10 AM distinct from those retrieved at 8 AM, the events retrieved at 8 AM Tuesday will be identical to those retrieved at 8 AM on Monday.

The carriers and events available will be:

8 AM

CarrierId	**Event**
d235409d-9f48-4d99-a208-db4e03582902	Contract Signed
6140d5de-4f23-4611-a1e3-06c57bd496d1	Profile Update
2f69a23b-592c-47f6-a33c-966c4be46dac	Insurance Agent Update
e1402f1d-a4e6-4e78-b53a-4989489a4d5d	Tax Info Update
6d1fc081-3212-4e5d-bad1-2ec7bcf1b723	Document Update
8ef74472-bdd6-4ec9-977e-ce3f35234f1c	Contract Signed

10 AM

CarrierId	Event
d235409d-9f48-4d99-a208-db4e03582902	Profile Update
6140d5de-4f23-4611-a1e3-06c57bd496d1	Contract Signed
2f69a23b-592c-47f6-a33c-966c4be46dac	Tax Info Update
e1402f1d-a4e6-4e78-b53a-4989489a4d5d	Insurance Agent Update
6d1fc081-3212-4e5d-bad1-2ec7bcf1b723	Contract Signed
8ef74472-bdd6-4ec9-977e-ce3f35234f1c	Document Update

12 PM

CarrierId	Event
d235409d-9f48-4d99-a208-db4e03582902	Document Update
6140d5de-4f23-4611-a1e3-06c57bd496d1	Profile Update
2f69a23b-592c-47f6-a33c-966c4be46dac	Insurance Agent Update
e1402f1d-a4e6-4e78-b53a-4989489a4d5d	Contract Signed
6d1fc081-3212-4e5d-bad1-2ec7bcf1b723	Tax Info Update
8ef74472-bdd6-4ec9-977e-ce3f35234f1c	Profile Update

2 PM

CarrierId	Event
d235409d-9f48-4d99-a208-db4e03582902	Contract Signed
d235409d-9f48-4d99-a208-db4e03582902	Profile Update
2f69a23b-592c-47f6-a33c-966c4be46dac	Insurance Agent Update
2f69a23b-592c-47f6-a33c-966c4be46dac	Contract Signed
6d1fc081-3212-4e5d-bad1-2ec7bcf1b723	Document Update
8ef74472-bdd6-4ec9-977e-ce3f35234f1c	Profile Update

Alphabetical Index of Connexion Requests

A Error codes and messages

The table below lists some of the possible error codes and messages returned by Connexion. This list is not exhaustive, but encompasses most of the likely errors. In most cases, in addition to the default message listed here, the error will include a detailed message providing more information about the specific cause of the error. For instance, error 104000760 might be accompanied by a detailed message of the form "An asset with reference ID 'DS012h34' already exists".

Note: If the supplied SessionToken is expired or invalid, a SOAP Fault is returned instead of a ServiceError structure, so this list does not include an error code for an invalid session. Most SOAP client stacks will throw an exception of some sort upon receiving a SOAP Fault. Your client-side code should handle that event properly, and offer the user an opportunity to re-authenticate if appropriate.

Code	**Message**
101000010	Authentication failure
104000010	Requestor can not access requested asset/alarm/search *
104000030	You do not have the capability to perform the requested action *
104000090	Search ID not specified
104000100	Unable to find matches.
104000310	No search ID specified
104000320	Invalid search criteria *
104000330	System Error. The user session is invalid.
104000340	Cannot create an active search.
104000350	Cannot associate active search with an alarm.
104000360	Invalid live search definition *
104000400	You do not have the capability to create searches
104000700	No asset-based alarm search criteria specified
104000710	Cannot create alarm on asset you or your group does not own *
104000760	Reference ID already exists. Please check the number and try again. Reference IDs must be unique.
104000770	Invalid availability criteria specified. Check start/end dates.
104000780	Posting not submitted. Make sure that all required fields are completed and try again.
104000860	Cannot change this item on updates *
104000870	The post/search has been cancelled or has expired and cannot be updated.
104000920	Invalid reference ID.
105000021	Unknown location.
105000125	Truckstop lookup failed.
105000130	Truckstop lookup failed. Please make sure the city name is spelled correctly.

105000140	Truckstop lookup failed. Please make sure the city name or zip/postal code is valid.
108000010	You do not have the capability to create shipment searches
108000020	You do not have the capability to create equipment searches
108000040	You do not have the capability to create alarms
108000060	You do not have the capability to lookup matches
110000010	You do not have the capability to lookup carrier info
110000020	No CarrierWatch profile found for the specified company.
110000030	CarrierWatch is temporarily unavailable
110000040	Multiple MC numbers found for the specified company.
110000050	You do not have the capability to lookup carrier info
110000060	You do not have the capability to lookup private insurance information
110000070	TransCore does not have private insurance information for the requested carrier
111000010	You do not have the capability to lookup current spot market rates
111000020	You do not have the capability to lookup historic spot market rates
111000030	You are not a rate contributor, so cannot lookup your own spot market rate information
111000100	Unsupported lane type specified
111000200	No rate information available for the specified lane