**Department of Technology**

# Truck Posting Web Service API v6

**March 2011**

# Table of Contents

## *Truck Posting Web Service URL*

### *Production*

http://webservices.truckstop.com/v4/Posting/TruckPosting.svc

### *Test*

http://testws.truckstop.com:8080/v4/Posting/TruckPosting.svc

## *Truck Posting Web Service Requests*

### PostTrucks: Posts and updates a collection of trucks.

#### Input:

TruckPostingRequest

#### Requirements

Authentication requires a valid user name, password, and integration id that has Truck Posting web service enabled

A truck posting requires and Origin (city, state, country), Equipment type (see supported equipment types)

Updating a truck has the same requirements as posting with the addition of the trucks ID

#### Output:

TruckPostingReturn

#### Sample Code - .NET 3.5 using C#

```csharp
…
    TruckPostingClient client = new TruckPostingClient();

    TruckPostingRequest request = new TruckPostingRequest();
    request.IntegrationId = xx;
    request.UserName = "xxxxxxxx";
    request.Password = "xxxxxxxx";

    Truck newTruck = new Truck
    {
        DateAvailable = DateTime.Now,
```

```
         DestinationCity = "Boise",
         DestinationState = "ID",
         DestinationCountry = "USA",
         OriginCity = "Denver",
         OriginState = "Co",
         OriginCountry = "USA",
         EquipmentType = "V"
    };

    request.Trucks = new Truck[] { newTruck };

    TruckPostingReturn result = client.PostTrucks(request);

    foreach (Error error in result.Errors)
    {
        Console.WriteLine(error.ErrorMessage);
    }
    if (result.TruckIds!=null)
    {
        Console.WriteLine(result.TruckIds.Length);
    }
...
```

## GetTruckDetailResults: Gets the details for a specified truck.

### Input:

TruckDetailRequest

### Requirements

Authentication requires a valid user name, password, and integration id that has Truck Details service enabled

Getting the truck details requires a valid truck Id

### Output:

TruckDetailReturn

### Sample Code - .NET 3.5 using C#

```
…
    TruckPostingClient client = new TruckPostingClient();

    TruckDetailRequest request = new TruckDetailRequest();
    request.IntegrationId = X;
    request.UserName = "xxxxxxx";
```

```
   request.Password = "xxxxxxx";
   request.TruckId = xxxxxxx;

   TruckDetailReturn result = client.GetTruckDetailResults(request);

   foreach (Error error in result.Errors)
   {
      Console.WriteLine(error.ErrorMessage);
   }

   Console.WriteLine(result.TruckDetail.TruckCompanyName);
   Console.WriteLine(result.TruckDetail.TruckCompanyPhone);
...
```

## DeleteTrucks: Deletes the specified truck posting.

### Input:

TruckDeleteRequest

### Requirements

Authentication requires a valid user name, password, and integration id that has Truck Posting web service enabled

The truck id of the posting being Deleted

### Output:

TruckDeleteReturn

### Sample Code - .NET 3.5 using C#

```
…
   TruckPostingClient client = new TruckPostingClient();
   TruckDeleteRequest request = new TruckDeleteRequest();
   request.IntegrationId = X;
   request.UserName = "xxxxxxx";
   request.Password = "xxxxxxxx";
   request.Trucks = new int[]{xxxxxxx};

   TruckDeleteReturn result = client.DeleteTrucks(request);
   foreach (Error error in result.Errors)
   {
      Console.WriteLine(error.ErrorMessage);
   }
...
```

### Get Trucks: Gets list of trucks posted by user

#### Input:

TruckListRequest

#### Requirements

Authentication requires a valid user name, password, and integration id that has Truck Posting web service enabled

#### Output:

TruckListReturn

#### Sample Code - .NET 4 using C#

```csharp
...

protected void gridTestResults_NeedDataSource(object source, GridNeedDataSourceEventArgs e)
    {
        using (TruckPosting truckPosting = new TruckPosting())
        {
            TruckListRequest truckListRequest = new TruckListRequest
                {
                    IntegrationId = truckPostingRequest.IntegrationId,
                    Password = truckPostingRequest.Password,
                    UserName = truckPostingRequest.UserName
                };

            var TruckSearchResults = truckPosting.GetTrucks(truckListRequest);
            gridTestResults.DataSource = TruckSearchResults.PostedTrucks;
        }
    }
...
```

# Appendix 1 SOAP Object Descriptions

## Error

1. string ErrorMessage

## TrailerOptionType(enum)

1. Tarps
2. Hazardous

3. PaletteExchange
   4. Team

## Truck

   1. DateTime DateAvailable
   2. string DestinationCity
   3. string DestinationState
   4. string DestinationCountry
   5. TrailerOptionType[] EquipmentOptions
   6. string EquipmentType
   7. bool IsLoadFull
   8. string Length (must be numeric or blank)
   9. string MinDistance (must be numeric or blank)
   10. string OriginCity
   11. string OriginState
   12. string OriginCountry
   13. int OriginRadius (optional, 25 to 300 miles)
   14. int Quantity
   15. string RatePerMile
   16. string SpecInfo
   17. string TimeAvalilable
   18. int TruckID
   19. string TruckMultiDest
   20. string Weight (must be numeric or blank)
   21. string Width (must be numeric or blank)

### TruckDeleteRequest

   1. int IntegrationId
   2. string Password
   3. int[] Trucks

4. string UserName

### TruckDeleteReturn

1. Error[] Errors

### TruckDetailRequest

1. int IntegrationId
2. string Password
3. int TruckId
4. string  UserName

### TruckDetailResult

1. string CarrierRating
2. string DateTruckAvailable
3. string DestinationCity
4. string DestinationState
5. string DispatchName
6. string DispatchPhone
7. string DOTNumber
8. DateTime Entered
9. string Equipment
10. TrailerOptionType EquipmentOptions
11. bool HasFileCheck
12. int ID
13. bool IsFullTruck
14. string Length
15. string MCNumber
16. string MinMiles
17. string OriginCity
18. string OriginState
19. string PercentFull

20. double PerMile

21. string PointOfContact

22. string PointOfContactPhone

23. string Quantity

24. string SpecInfo

25. bool TCAMember

26.  string TimeTruckAvailable

27. string TMCNumber (carrier MC Number)

28. string TruckCompanyCity

29. string TruckCompanyEmail

30. string TruckCompanyFax

31. string TruckCompanyName

32. string TruckCompanyPhone

33. string TruckCompanyState

34. string TruckDesiredDirection

35. string Weight

36. string Width

### TruckDetailReturn

1. Error[] Errors
2. TruckDetailResult TruckDetail

## TruckPostingRequest

1. int IntegrationId
2. string Password
3. Truck[] Trucks
4. string UserName

## TruckPostingReturn

1. Error[] Errors
2. int[]  TruckIds

**TruckListRequest**

1. int IntegrationId
2. string Password
3. string UserName


**TruckListReturn**

1. Error[] Errors
2. TruckPost[] PostedTrucks


**TruckPost**

1. int Id
2. string Handle
3. string Equipment
4. bool IsFull
5. DateTime DateAvailable
6. bool IsDaily
7. string OriginCity
8. string OriginState
9. string OriginCountry
10. string DestinationCity
11. string DestinationState
12. string DestinationCountry

# Appendix 2 Supported Equipment Types

**Code      Definition**

2F       - Two 24 or 28 foot flats

ANIM  - Animal Carrier

AUTO  - Auto carrier

B-TR   - B-Train/Supertrain (Canada only)

BELT  - Conveyor Belt

BOAT  - Boat Hauling Trailer

CH      - Convertible Hopper

CONT  - Container Trailer

DD      - Double Drop

DUMP - Dump Trucks

ENDP  - End Dump

F        - Flatbed

FEXT  - Stretch Trailers or Extendable Flatbed

FINT   - Flatbed Intermodal

FO      - Flatbed over-dimension loads

FSD    - Flat or Step Deck

FVR    - Flatbed, Van, or Reefer

FWS   - Flatbed With Sides

HOPP  - Hopper Bottom

HS      - Hot Shot

HTU    - Haul and Tow Unit

LAF    - Landoll Flatbed

LB      - Lowboy

LBO    - Lowboy over-dimension load

LDOT  - Load-Out are empty trailers you load and haul

LIVE   - Live Bottom Trailer

MAXI  - Maxi or Double Flat Traillers

MBHM - Mobile Home

PNEU  - Pneumatic

PO      - Power Only (tow-away)

R        - Refrigerated Carrier (Reefer)

RGN    - Removable Goose Neck & Multi-Axle Heavy Haulers

RGNE  - RGN Extendable

RINT   - Refrigerated Intermodal

ROLL  - ROLL Top Conestoga

RPD   - Refrigerated Carrier W/Plant Decking

SD     - Step Deck

SDL    - Step Deck w/Loading Ramps

SDO   - Step Deck over-dimension load

SPEC   - Unspecified Specialized Trailers

SV     - Straight Van

TANK  - Tanker (Food grade, liquid, bulk, etc.)

V      - Van

V-OT   - Open Top Van

VB     - Blanket Wrap Van

VC     - Curtain Van

VCAR  - Cargo Vans (1 Ton capacity)

VF     - Flatbed or Van

VINT   - Van Intermodal

VIV    - Vented Insulated Van

VLG    - Van w/Liftgate

VM     - Moving Van

VR     - Van or Reefer

VV     - Vented Van

WALK  - Walking Floor

VVR    - Vented Van or Refrigerated

VIVR   - Vented Insulated Van or Refrigerated

## Table of Contents

# Truck Searching Web Service URL

## Production

http://webservices.truckstop.com/v3/Searching/TruckSearch.svc

## Test

http://testws.truckstop.com:8080/v3/Searching/TruckSearch.svc

## *Truck Searching Web Service Requests*

### *GetTruckSearchResults: Searches for trucks based on search criteria.*

Input:

TruckSearchRequest

Requirements

Authentication requires a valid user name, password, and integration id that has truck search web service enabled

Searching Requires an origin, as a state, City and State with a radius.  For an anywhere search, leave city and state blank and pass country.  For multi-state pass comma delimited list of states to origin state ("CA,TX,NY,ID").

For destination pass in blank city and state with a valid country for an anywhere search.  For multi-state pass in comma delimited list of states to destination state.

Valid countries are USA, CAN, MEX.

Pass in HoursOld for searching on age.  Set HoursOld to zero if no filtering on hours old is desired.

You Must Specify the Truck Type (All, Part, or Full)

Output:

TruckSearchReturn

Sample Code - .NET 3.5 using C#

```
…
   TruckSearch client = new TruckSearch();

   TruckSearchRequest request = new TruckSearchRequest();
```

```csharp
request.IntegrationId = X;
request.UserName = "xxxxxxx";
request.Password = "xxxxxxx";

TruckSearchCriteria criteria = new TruckSearchCriteria();
 criteria.OriginCountry = "USA";
 criteria.OriginCity = "Boise";
 criteria.OriginState = "ID";
 criteria.OriginRange = 200;
 criteria.EquipmentType = "V";
 criteria.LoadType = LoadType.All;
 ...
TruckSearchReturn result = client.GetTruckSearchResults(request);
foreach (Error error in result.Errors)
{
    Console.WriteLine(error.ErrorMessage);
}
foreach (TruckSearchItem item in result.SearchResults)
{
    Console.WriteLine(string.Format("Origin {0}, {1}",item.OriginCity, item.OriginState));
    Console.WriteLine(string.Format("Destination {0}, {1}", item.DestinationCity,
item.DestinationState));
}
...
```

## *GetTruckSearchDetailResults: Get the details for a specified truck.*

## **Input**:

TruckDetailRequest

### *Requirements*

Authentication requires a valid user name, password, and integration id that has truck search web service enabled

Getting the truck detail requires a valid truck id.

## **Output**:

TruckDetailReturn

Sample Code - .NET 3.5 using C#

```csharp
...
   TruckSearch client = new TruckSearch();
   TruckDetailRequest request = new TruckDetailRequest();
```

```
 request.IntegrationId = 4;
 request.UserName = "TestUser";
 request.Password = "tupass123";
 request.TruckId = 304515275;

 TruckDetailReturn result = client.GetTruckSearchDetailResults(request);
 foreach (Error error in result.Errors)
  {
     Console.WriteLine(error.ErrorMessage);
  }
 Console.WriteLine(result.TruckDetail.TruckCompanyName);
 Console.WriteLine(result.TruckDetail.TruckCompanyPhone);
...
```

### *DeleteTrucks: Deletes the specified truck posting.*

## Input:

TruckDeleteRequest

### *Requirements*

Authentication requires a valid user name, password, and integration id that has
Truck Posting web service enabled

The truck id of the posting being Deleted

## Output:

TruckDeleteReturn

Sample Code - .NET 3.5 using C#

```
…
  TruckPosting client = new TruckPosting();
  TruckDeleteRequest request = new TruckDeleteRequest();

  request.IntegrationId = X;
  request.UserName = "xxxxxxx";
  request.Password = "xxxxxxxx";
  List<int> trucks = new List<int>();
  trucks.add(TruckID);
  request.Trucks = trucks.ToArray();
  TruckDeleteReturn result = client.DeleteTrucks(request);
  foreach (Error error in result.Errors)
  {
     Console.WriteLine(error.ErrorMessage);
  }
```

# Appendix 1 SOAP Object Descriptions

### *Error*

1. string ErrorMessage

### *TrailerOptionType(enum)*

1. Tarps
2. Hazardous
3. PalletExchange
4. Team

### *TruckSearchCriteria*

1. string DestinationCity
2. int DestinationRange
3. string DestinationState
4. string DestinationCountry
5. string EquipmentType
6. LoadType (enum) LoadType
7. string OriginCity
8. int OriginRange
9. string OriginState
10. string OriginCountry
11. DateTime AvailableDate
12. int HoursOld

### *TruckDetailRequest*

1. int IntegrationId
2. int TruckId
3. string Password
4. string UserName

### *TruckDetailResult*

1. string CarrierRating
2. string DOTNumber
3. string DateTruckAvailable
4. string DestinationCity
5. string DestinationState
6. string DispatchName
7. string DispatchPhone
8. string Entered
9. boolean EnteredSpecified
10. string Equipment
11. boolean HasFileCheck
12. boolean HasFileCheckSpecified
13. int ID
14. boolean IDSpecified
15. boolean IsFullTruck
16. boolean IsFullTruckSpecified
17. string Length
18. string MCNumber
19. int MinMiles
20. string OriginCity
21. string OriginState
22. string PerMile
23. boolean PerMileSpecified
24. string PercentFull
25. string PointOfContact
26. string PointOfContactPhone
27. int Quantity
28. string SpecInfo
29. boolean TCAMember

30. boolean TCAMemberSpecified

31. string TMCNumber

32. string TimeTruckAvailable

33. string TruckCompanyCity

34. string TruckCompanyEmail

35. string TruckCompanyFax

36. string TruckCompanyName

37. string TruckCompanyPhone

38. string TruckCompanyState

39. string TruckDesiredDirection

40. int Weight

41. int Width

### *Truck DetailReturn*

1. Error[] Errors

2. TruckDetailResult TruckDetail

### *Truck SearchResults*

1. string Age  (number of hours old)

2. string CompanyName

3. string DateAvailable

4. string DestinationCity

5. string DestinationCountry

6. int DestinationDistance

7. string DestinationState

8. string Equipment

9. boolean FileCheck

10. boolean FileCheckSpecified

11. string HomeBaseState

12. int Id

13. int Length

14. LoadType LoadType

15. string Miles

16. string OriginCity

17. string OriginCountry

18. string OriginDistance (air miles from the search Origin)

19. string OriginState

20. string PointOfContactPhone

21. string RatePerMile

22. string Weight

23. bool IsFriend

### TruckSearchRequest

1. LoadSearchCriteria Criteria

2. int IntegrationId

3. string Password

4. string UserName

### TruckSearchReturn

1. Error[] Errors

2. TruckSearchReturn.SearchResults

### LoadType (enum)

1. All

2. Full

3. Partial

## Appendix 2 Supported Equipment Types

**Code      Definition**

2F       - Two 24 or 28 foot flats

ANIM   - Animal Carrier

AUTO   - Auto Carrier

B-TR    - B-Train/Supertrain (Canada only)

BELT   - Conveyor Belt

BOAT   - Boat Hauling Trailer

CH       - Convertible Hopper

CONT  - Container Trailer

DD       - Double Drop

DUMP - Dump Trucks

ENDP  - End Dump

F          - Flatbed

FEXT   - Stretch Trailers or Extendable Flatbed

FINT    - Flatbed Intermodal

FO        - Flatbed over-dimension loads

FSD      - Flat or Step Deck

FVR      - Flatbed, Van, or Reefer

FWS      - Flatbed With Sides

HOPP   - Hopper Bottom

HS        - Hot Shot

HTU     - Haul and Tow Unit

LAF      - Landoll Flatbed

LB        - Lowboy

LBO      - Lowboy over-dimension load

LDOT   - Load-Out are empty trailers you load and haul

LIVE    - Live Bottom Trailer

MAXI  - Maxi or Double Flat Traillers

MBHM - Mobile Home

PNEU  - Pneumatic

PO        - Power Only (tow-away)

R          - Refrigerated Carrier(Reffer)

RGN     - Removable Goose Neck & Multi-Axle Heavy Haulers

RGNE   - RGN Extendable

RINT    - Refrigerated Intermodal

ROLL  - ROLL Top Conestoga

RPD    - Refrigerated Carrier W/Plant Decking

SD       - Step Deck

SDL    - Step Deck w/Loading Ramps

SDO    - Step Deck over-dimension load

SPEC  - Unspecified Specialized Trailers

SV       - Straight Van

TANK  - Tanker (Food grade, liquid, bulk, etc.)

V         - Van

V-OT   - Open Top Van

VB       - Blanket Wrap Van

VC       - Curtain Van

VCAR  - Cargo Vans (1 Ton capacity)

VF       - Flatbed or Van

VINT   - Van Intermodal

VIV    - Vented Insulated Van

VLG    - Van w/Liftgate

VM     - Moving Van

VR       - Van or Reefer

VV       - Vented Van

WALK  - Walking Floor

VVR    - Vented Van or Refrigerated

VIVR   - Vented Insulated Van or Refrigerated