



# LiveOps TransferData Service API Reference

Version: 2.0.3  
Revision Date: 2009-12-16

Copyright © 2009 LiveOps, Inc. This page contains confidential information. All materials on this page are the copyrighted property of LiveOps, Inc. No material from this page or any LiveOps web page may be copied, reproduced, republished, uploaded, posted, transmitted, or distributed in any way, except where permitted under a written agreement with LiveOps.

---

# LiveOps TransferData Service API Reference

Version: 2.0.3  
Revision Date: 2009-12-16

---

## Table of Contents

### 0: Preface

- Introduction
- About This Guide
- Intended Audience
- What You Should Know Before You Begin
- Typographical Conventions
- Contact Info
- Revision History

### 1: API Architecture

- URI Format
- Input Data
- Identifying a Session
- Query String Arguments

### 2: Requests

- edit Request
- get Request
- start Request
- stop Request

### 3: Responses

- Response XML format
- Data format
- Status codes

### 4: Example Sessions

- Starting a TDS Session
  - Adding Data to a TDS Session
  - Retrieving Data from a TDS Session
  - Stopping a TDS Session
-

## 0. Preface

### Introduction

The LiveOps TransferData Service (TDS) allows remote parties to associate data with a call that is being sent to the LiveOps On-Demand Contact Center Platform . A typical application for this service is sending data that has been previously collected by an IVR or agent of a different call center or platform to a LiveOps platform Agent. The data transferred to this service can be made available to a script or application via the normal LiveOps screen pop mechanism.

When a TDS session is initiated, LiveOps allocates a DNIS for the call to be transferred. When the call arrives to this DNIS in the LiveOps platform, all data belonging to that TDS session is associated with the call.

The LiveOps TransferData Service uses a REST style interface. REST requests are made using the HTTPS protocol by using GET and POST methods. Requests that modify data are made with a POST. GET requests are used to retrieve data from the service.

#### In this chapter...

- Introduction
- About This Guide
- Intended Audience
- What You Should Know Before You Begin
- Typographical Conventions
- Contact Info
- Revision History

### About This Guide

#### Organization

This *LiveOps TransferData Service API Reference* is organized into four chapters:

Table 0-1: LiveOps TransferData Service API Reference Organization

Chapter	Contents
<b>Chapter 1: API Architecture</b>	This chapter describes the key elements of the LiveOps TransferData Service, such as how to construct a request URI, the format of input data, how to identify a session and how to specify query string arguments.
<b>Chapter 2: Requests</b>	This chapter details the request types available in the LiveOps TransferData Service.
<b>Chapter 3: Responses</b>	This chapter describes the format and structure of the responses returned by LiveOps TransferData Service requests.
<b>Chapter 4: Example Sessions</b>	This chapter provides some real world examples of TDS requests and responses.

### Intended Audience

This *LiveOps TransferData Service API Reference* is intended for developers who wish to integrate with the **LiveOps TransferData Service**.

It is assumed that you already have been given credentials to access this service, and that contractual agreements governing the client's use of the API are in place. If you have questions about any of this, please contact your LiveOps Account Manager.

### What You Should Know Before You Begin

This *LiveOps TransferData Service API Reference* assumes that you are already familiar with the following concepts:

- using a web API,
- XML document formats,
- a basic understanding of the LiveOps On-Demand Contact Center Platform.

No specific language or platform skills are necessary. Any platform that can make customized HTTPS requests and process the subsequent response can be used with this API.

### Typographical Conventions

Throughout this guide, we use font color to denote the following:

- **Commands, button or menu names**

- **Terms and concepts** as they are introduced for the first time
- *Method or variable names*
- User input
- *Table or Figure caption*

In addition, we use a few icons to call out helpful information:

 **Tip**—advice or suggestions that will make your experience with the product more effective or enjoyable.

 **Note**—A possible catch or "gotcha" to watch out for or keep in mind.

 **See Also**—a reference to further reading material or resources.

If a term appears as a [green link](#), you can follow that to see the LiveOps On-Demand Contact Center Platform [Glossary](#) entry for that term.

Finally, you can easily navigate throughout the *LiveOps TransferData Service API Reference* using the breadcrumb links at the top of each page, or you can jump to the next or previous chapters using links at the top and bottom of each chapter.

---

## Contact Info

If you have comments or suggestions on this document, please send them via email to [techpubs@lists.liveops.com](mailto:techpubs@lists.liveops.com).

---

## Revision History

Version (Date)	Highlights
2.0.3 (2009-12-16)	Documentation cleanup
2.0.2 (2009-12-10)	Documentation cleanup
2.0.1 (2009-11-30)	Documentation cleanup
2.0 (2009-11-11)	Initial document. Version 1 of this document was internal LiveOps only.

---

## 1. API Architecture

This chapter describes the architecture of the LiveOps TransferData Service.

### In this chapter...

- URI Format
- Input Data
- Identifying a Session
- Query String Arguments

### URI Format

All of the TDS commands are used by formatting HTTP requests. The base URL for the requests is:

Table 1-1: TDS Request Base URL

```
https://api.liveops.com/louie/<client_id>/struct-xml/my-callcenter/campaigns/by-id/<campaign_id>/telephony/routing/transferdata/dnis
```

**i** *<client\_id>* must be replaced by the name of the client making the request. *<campaign\_id>* must be replaced by the campaign receiving the transfer.

Depending on the method and query type, a different path appended to the base URL will be used. The following example shows the URL to fetch data from an existing TDS session, using the GUID to identify the session:

Table 1-2: by-guid Example

```
campaigns/by-id/<campaign_id>/telephony/transferdata/dnis/by-guid/505a73bac591882c14c80bfe452cd757
```

Appended to the base URL, the complete URL for the above request would be:

Table 1-3: Complete TDS Example URL

```
https://api.liveops.com/louie/<client_id>/struct-xml/my-callcenter/campaigns/by-id/<campaign_id>/telephony/routing/transferdata/dnis/by-guid/505a73bac591882c14c80bfe452cd757
```

All requests must send an Authorization: header with the HTTP request for authentication. Please contact your LiveOps account manager if you do not know your credentials. LiveOps currently only supports "Basic" style authentication. Failure to include an Authorization: header will result in a 401 HTTP response.

### Input Data

Methods which create or modify a session most form an HTTP POST with an XML document naming the method and provide the method arguments. The Content-Type of the post must be "text/x-louie-xml". The following is an example XML document:

Table 1-4: Example XML Input Data

```
<request>
<method>start</method>
<ani>16505559999</ani>
<oemkey>1234567890</oemkey>
<routingdata>appname%3Dapp%26var1%3Dval1%26var2%3Dval2</routingdata>
<reportingdata>appname%3Dapp1%26var3%3Dval3%26var4%3Dval4</reportingdata>
<agentdata>var5%3Dval5</agentdata>
<command-version>1.0</command-version>
<service-name>transferdata</service-name>
</request>
```

The method tag is always required. Individual methods will require specific argument tags, some requiring no additional tags at all. An optional command-version tag may be specified to indicate the specific version of the method to call. If this tag is not specified, version 1.0 will always be used. Any tags outside of these and those defined for the method being called are ignored by the LiveOps TransferData Service.

### Identifying a Session

All methods which either retrieve or modify a TDS session must refer to a session in the URL. There are 3 ways to refer to a session

#### **by-guid/<guid>**

Every TDS session is assigned a unique GUID. All requests return the session GUID in a key named "guid". This is the ideal way to refer to a session.

Example URL to get a session by GUID:

*Table 1-5: by-guid Example*

```
https://api.liveops.com/louie/<client_id>/struct-xml/my-callcenter/campaigns/by-id/<campaign_id>/telephony/routing/transferdata/dnis/by-guid/505a73bac591882c14c80bfe452cd757
```

#### **by-dnis/<dnis> .<ani>**

This will look up the session by the allocated DNIS and the supplied ANI from the start request. If no ANI is available, use "0" for the ANI.

Example URL to get a session by DNIS/ANI:

*Table 1-6: by-dnis Example*

```
https://api.liveops.com/louie/<client_id>/struct-xml/my-callcenter/campaigns/by-id/<campaign_id>/telephony/routing/transferdata/dnis/by-dnis/18002223333.16504445555
```

#### **by-oemkey/<oemkey>**

As part of a start request, an optional <oemkey> tag can be supplied in the POST XML. This key can then be used to look up the session. The key must be unique across all the whole campaign.

Example URL to get a session by OEM key:

*Table 1-7: by-oemkey Example*

```
https://api.liveops.com/louie/<client_id>/struct-xml/my-callcenter/campaigns/by-id/<campaign_id>/telephony/routing/transferdata/dnis/by-oemkey/9123898123098123
```

---

## Query String Arguments

Optional query string arguments may be appended to the URLs in both the GET and POST methods. These arguments affect all methods, unless otherwise noted.

--echo-args

A value of '1' causes the request structure to be included with the response. A value of '0' (the default) omits the request structure from the response.

--echo-meta

A value of '1' causes a meta information structure about the available methods in this bundle to be included with the response. A value of '0' (the default) omits this meta information structure from the response.

---

## 2. Requests

There are four types of requests that can be performed as part of the LiveOps TransferData Service :

- **edit**—adds data to an existing TDS session.
- **get**—retrieves data from a TDS session.
- **start**—starts a new TDS session.
- **stop**—ends an existing TDS session.

Each of these request types are described below.

### In this chapter...

- edit Request
- get Request
- start Request
- stop Request

### edit Request

Table 2-1: edit Request Parameters

<b>Description</b>	Replaces data in an existing TDS session. All data in the existing session will be overwritten with the data you supply in the <i>edit</i> request. Be sure to pass any previous data sent that you want preserved in the <i>edit</i> request.
<b>HTTP Type</b>	POST
<b>URI</b>	campaigns/by-id/<campaign_id>/telephony/transferdata/dnis/by-<type>/<identifier>
<b>Arguments</b>	<ul style="list-style-type: none"> <li>• <b>method</b>—<i>edit</i></li> <li>• <b>routingdata</b>—data used to route the call appropriately</li> <li>• <b>reportingdata</b>—data that will be made available to the reporting system</li> <li>• <b>agentdata</b>—data passed to the agents screen pop</li> </ul>

Table 2-2: edit Request Return Values

<b>Environment</b>	zone—identifies the LiveOps data center that serviced the request
<b>Properties</b>	<ul style="list-style-type: none"> <li>• <b>status</b>—String containing the result of the operation</li> <li>• <b>status_code</b>—String containing the result of the operation (status_codes are described in <a href="#">Chapter 3: Responses</a>)</li> <li>• <b>guid</b>—Globally unique identifier of the session</li> <li>• <b>type</b>—currently TDS only supports a 'dnis' type of request</li> <li>• <b>dnis</b>—The allocated DNIS to where the call should be transferred</li> <li>• <b>campaign_id</b>—same as the campaign id to which the original request was sent</li> <li>• <b>ani</b>—optional, caller ID of caller, if a valid ANI was supplied in the original request</li> <li>• <b>oemkey</b>—optional, same as the oemkey provided in the original request, if any</li> </ul>

### get Request

Table 2-3: get Request Parameters

<b>Description</b>	Retrieves all previously associated data with a session.
<b>HTTP Type</b>	GET
<b>URI</b>	campaigns/by-id/<campaign_id>telephony/transferdata/dnis/by-<type>/<identifier>
<b>Query String Arguments</b>	<ul style="list-style-type: none"> <li>• <b>property</b>—Only return certain scalars from response. Repeat multiple times for multiple properties. Can specify "all" (the default) for all properties.</li> </ul>

Table 2-4: *get Request Return Values*

<b>Environment</b>	zone—identifies the LiveOps data center that serviced the request
<b>Properties</b>	<ul style="list-style-type: none"> <li>• status—String containing the result of the operation</li> <li>• status_code—String containing the result of the operation</li> <li>• guid—Globally unique identifier of the session</li> <li>• type—currently TDS only supports a 'dnis' type of request</li> <li>• dnis—The allocated DNIS for the session</li> <li>• routingdata—data used to route the call appropriately</li> <li>• reportingdata—data that will be made available to the reporting system</li> <li>• agentdata—data passed to the agents screen pop</li> <li>• campaign_id—same as the campaign id to which the original request was sent</li> <li>• ani—optional, caller ID of caller, if a valid ANI was supplied in the original request</li> <li>• oemkey—optional, same as the oemkey provided in the original request, if any</li> </ul>

## start Request

Table 2-5: *start Request Parameters*

<b>Description</b>	<p>Starts a new TDS session, reserves a DNIS and saves the data with the session. The DNIS reserved by this method is only kept available for a maximum of two minutes.</p> <p><b>!</b> This 2 minute limit is intended to help prevent exhausting the DNIS pool reserved for your Campaign. You should also take care that your application does not make too many <i>start</i> requests without quickly transferring the call, or you may use up your available DNIS.</p>
<b>HTTP Type</b>	POST
<b>URI</b>	campaigns/by-id/<campaign_id>/telephony/transferdata/dnis
<b>Arguments</b>	<ul style="list-style-type: none"> <li>• method—start</li> <li>• ani—Caller ID of caller, pass 0 if the ANI is unavailable</li> <li>• routingdata—data used to route the call appropriately</li> <li>• reportingdata—data that will be made available to the reporting system</li> <li>• agentdata—data passed to the agents screen pop</li> <li>• oemkey—An optional key to associate with the session</li> </ul>

Table 2-6: *start Request Return Values*

<b>Environment</b>	zone—identifies the LiveOps data center that serviced the request
<b>Properties</b>	<ul style="list-style-type: none"> <li>• status—String containing the result of the operation</li> <li>• status_code—String containing the result of the operation</li> <li>• guid—Globally unique identifier of the session</li> <li>• type—currently TDS only supports a 'dnis' type of request</li> <li>• dnis—The allocated DNIS to where the call should be transferred</li> <li>• campaign_id—same as the campaign id to which the original request was sent</li> <li>• ani—optional, caller ID of caller, if a valid ANI was supplied in the original request</li> <li>• oemkey—optional, same as the oemkey provided in the original request, if any</li> </ul>

## stop Request

Table 2-7: *stop Request Parameters*

<b>Description</b>	Frees the reserved DNIS and returns all the data previously associated with the session. After a stop request has been issued, you can still retrieve session data using by-guid and by-oemkey <i>get</i> requests.
--------------------	---

<b>HTTP Type</b>	POST
<b>URI</b>	campaigns/by-id/<campaign_id>/telephony/transferdata/dnis/by-<type>/
<b>Arguments</b>	method—stop

Table 2-8: stop Request Return Values

<b>Environment</b>	zone—identifies the LiveOps data center that serviced the request
<b>Properties</b>	<ul style="list-style-type: none"> <li>• status—String containing the result of the operation</li> <li>• status_code—String containing the result of the operation</li> <li>• guid—Globally unique identifier of the session</li> <li>• type—currently TDS only supports a 'dnis' type of request</li> <li>• dnis—The allocated DNIS for the session</li> <li>• routingdata—data used to route the call appropriately</li> <li>• reportingdata—data that will be made available to the reporting system</li> <li>• agentdata—data passed to the agents screen pop</li> <li>• campaign_id—same as the campaign id to which the original request was sent</li> <li>• ani—optional, caller ID of caller, if a valid ANI was supplied in the original request</li> <li>• oemkey—optional, same as the oemkey provided in the original request, if any</li> </ul>

## 3. Responses

This chapter describes the three kinds of responses you can expect from making LiveOps TransferData Service requests:

- Response XML format
- Data format
- Status codes

### In this chapter...

- Response XML format
- Data format
- Status codes

### Response XML Format

All requests return an XML document containing the response. The following is an example XML response:

*Table 3-1: Example XML Response*

```
<reply command='/<client_id>/struct-xml/mycallcenter/campaigns/byid/<campaign_id>/telephony/routing/transferdata
/dnis/start'>
<result>
<hash>
<hash name='environment'>
<scalar name='zone'>lv1</scalar>
</hash>
<hash name='properties'>
<scalar name='status'>success</scalar>
<scalar name='status_code'>1</scalar>
<scalar name='guid'>14a9f8c6f825091c7ca23da3bce1dfd8</scalar>
<scalar name='dnis'>18005551212</scalar>
<scalar name='ani'>16505559999</scalar>
</hash>
</hash>
</result>
</reply>
```

A TDS response will always be comprised of 2 sections as show above.

The "environment" section identifies where this request was serviced. In this example, the request was serviced in our Las Vegas data center.

The "properties" section will contain data specific to the request. Every properties section in a response will contain at least a "status", "status\_code" and "guid" tag. The other tags correspond to the additional keys returned by each method. The method description section details exactly what keys are found in each method response.

### Data Format

When specifying data to be saved to a session, or when retrieving a session, the data will be in URL encoded format. Key value pairs are specified with a URL encoded "=" (encoded to %3D) between the key and the value, and pairs are separated by a URL encoded "&" (encoded to %26). The values must also be URL encoded. For example, a key of "name" with a value of "John Smith" would be encoded as "name%3DJohn%20Smith".

*Table 3-2: Example Data*

```
age%3D5%26name%3DJohn%20Smith%26zip%20code%3D90210 is the encoded string for the following attributes:

age => 5

name => John Smith

zip code => 90210
```

The Requests section lists 3 different types of data that can be maintained using the TDS Service:

**routingdata**—This is data that will be used for making routing decisions within the Liveops platform. This data will be available to IVR scripts, and can be used to make routing decisions concerning campaign determination, pools, IVRs, queue multipliers (i.e. priority), etc. that can be used to process the call. This data is not passed to the agent application during a screen pop.

**reportingdata**—This is data that will be saved in the Liveops data store and be made available through the Liveops reporting system. Data supplied here must correspond to pre-defined program fields configured using the LiveOps configuration UI.

**agentdata**—This is data that will be passed to the agent application at screen pop time.

---

## Status Codes

Every response contains a status and a status\_code key. The following table describes the possible status types:

*Table 3-3: Status Codes*

Status	Status Code	Description
success	1	The operation completed normally.
error	100	An unknown error occurred.
pool_full	101	No free DNIS was available.
no_session	102*	The specified session could not be found or does not exist.
unknown_zone	103	The specified zone could not be found or does not exist.
campaign_not_found	104*	Campaign does not exist.
no_reps_logged_on	201	No reps logged in for this call type.
off_hours	202	Off hours for this call type.

\*When an invalid session is specified, an HTTP 404 is returned with body content describing the problem. A session can be invalid if its DNIS has expired or the session has received a stop request. Using an incorrect GUID or OEMKey will also return an invalid session response.

---

## 4. Example Sessions

This chapter provides "real world" examples of LiveOps TransferData Service requests and responses.

### In this chapter...

- Starting a TDS Session
- Adding Data to a TDS Session
- Retrieving Data from a TDS Session
- Stopping a TDS Session

### Starting a TDS Session

#### Request

Table 4-1: Starting a TDS Session: Request Example

```
POST /louie/<client_id>/struct-xml/my-callcenter/campaigns/by-id/<campaign_id>/telephony/routing/transferdata/dnis HTTP/1.1
TE: deflate, gzip;q=0.3
Connection: TE, close
Authorization: Basic ZmlkZWxpdk6dDNzdGNhbGw=
Host: api.liveops.com
User-Agent: libwww-perl/5.79
Content-Type: text/x-louie-xml
Content-Length: 136

<?xml version="1.0" encoding="UTF-8"?>
<request>
  <method>start</method>
  <routingdata>tsid%3D1257923146%26sessionkey%3D3188</routingdata>
  <agentdata>tsid%3D1257923146%26sessionkey%3D3188</agentdata>
  <reportingdata>tsid%3D1257923146%26sessionkey%3D3188</reportingdata>
  <ani>16505559999</ani>
</request>
```

#### Response

Table 4-2: Starting a TDS Session: Response Example

```
HTTP/1.1 200 OK
Date: Fri, 09 Feb 2007 22:06:06 GMT
Server: Apache
Connection: close
Transfer-Encoding: chunked
Content-Type: text/xml

<reply command='/'<client_id>/struct-xml/my-callcenter/campaigns/by-id/<campaign_id>/telephony/routing/transferdata/dnis/start'>
  <result>
    <hash>
      <hash name='environment'>
        <scalar name='zone'>lv</scalar>
      </hash>
      <hash name='properties'>
        <scalar name='ani'>16505559999</scalar>
        <scalar name='status_code'>1</scalar>
        <scalar name='dnis'>18005551212</scalar>
        <scalar name='status'>success</scalar>
        <scalar name='type'>dnis</scalar>
        <scalar name='guid'>33333fc9e2fe1c752245b2a3801a3c26</scalar>
        <scalar name='oemkey'>3188</scalar>
        <scalar name='campaign_id'><campaign_id></scalar>
      </hash>
    </hash>
  </result>
</reply>
```

### Adding Data to a TDS Session

#### Request

Table 4-3: Editing a TDS Session: Request Example

```

POST /louie/<client_id>/struct-xml/my-callcenter/campaigns/by-id/<campaign_id>/telephony/
routing/transferdata/dnis/by-dnis/16505559999.18005551212 HTTP/1.1
TE: deflate,gzip;q=0.3
Connection: TE, close
Authorization: Basic ZmlkZWxpdkhk6dDNzdGNhbGw=
Host: api.liveops.com
User-Agent: libwww-perl/5.79
Content-Type: text/x-louie-xml
Content-Length: 122

<?xml version="1.0" encoding="UTF-8"?>
<request>
  <method>edit</method>
  <routingdata>tsid%3D1257923146%26sessionkey%3D3188%26edited%3D1</routingdata>
  <agentdata>tsid%3D1257923146%26sessionkey%3D3188%26edited%3D1</agentdata>
  <reportingdata>tsid%3D1257923146%26sessionkey%3D3188%26edited%3D1</reportingdata>
</request>

```

## Response

Table 4-4: Editing a TDS Session: Response Example

```

HTTP/1.1 200 OK
Date: Fri, 09 Feb 2007 22:06:06 GMT
Server: Apache
Connection: close
Transfer-Encoding: chunked
Content-Type: text/xml

<reply command='<client_id>/struct-xml/my-callcenter/campaigns/by-id/<campaign_id>/telephony/
routing/transferdata/dnis/by-dnis/16505559999.18005551212/edit'>
  <result>
    <hash>
      <hash name='environment'>
        <scalar name='zone'>lv</scalar>
      </hash>
      <hash name='properties'>
        <scalar name='ani'>6505559999</scalar>
        <scalar name='status_code'>1</scalar>
        <scalar name='dnis'>18005551212</scalar>
        <scalar name='status'>success</scalar>
        <scalar name='type'>dnis</scalar>
        <scalar name='guid'>33333fc9e2fe1c752245b2a3801a3c26</scalar>
        <scalar name='oemkey'>3188</scalar>
        <scalar name='campaign_id'><campaign_id></scalar>
      </hash>
    </result>
  </reply>

```

## Retrieving Data from a TDS Session

### Request

Table 4-5: Getting TDS Session Data: Request Example

```

GET /louie/<client_id>/struct-xml/my-callcenter/campaigns/by-id/<campaign_id>/telephony/
routing/transferdata/dnis/by-dnis/16505559999.18005551212 HTTP/1.1
TE: deflate,gzip;q=0.3
Connection: TE, close
Authorization: Basic ZmlkZWxpdkhk6dDNzdGNhbGw=
Host: api.liveops.com
User-Agent: libwww-perl/5.79

```

### Response

Table 4-6: Getting TDS Session Data: Response Example

```

HTTP/1.1 200 OK
Date: Fri, 09 Feb 2007 22:06:06 GMT
Server: Apache
Connection: close
Transfer-Encoding: chunked
Content-Type: text/xml

<reply command='<client_id>/struct-xml/my-callcenter/campaigns/by-id/<campaign_id>/telephony/
routing/transferdata/dnis/by-dnis/16505559999.18005551212/serialize'>
  <result>
    <hash>
      <hash name='environment'>
        <scalar name='zone'>lv</scalar>
      </hash>
      <hash name='properties'>
        <scalar name='status_code'>1</scalar>
        <scalar name='status'>success</scalar>
        <scalar name='agentdata'>tsid%3D1257923146%26sessionkey%3D3188%26edited%3D1</scalar>
        <scalar name='routingdata'>tsid%3D1257923146%26sessionkey%3D3188%26edited%3D1</scalar>
        <scalar name='reportingdata'>tsid%3D1257923146%26sessionkey%3D3188%26edited%3D1</scalar>
        <scalar name='dnis'>18005551212</scalar>
      </hash>
    </result>
  </reply>

```

```

<scalar name='guid'>e1ab1:56d68992ff5a89fd94aaa940ff3a6701</scalar>
<scalar name='type'>dnis</scalar>
<scalar name='oemkey'>3188</scalar>
<scalar name='campaign_id'><campaign_id></scalar>
</hash>
</result>

```

## Stopping a TDS Session

### Request

Table 4-7: Stopping a TDS Session: Request Example

```

POST /louie/<client_id>/struct-xml/my-callcenter/campaigns/by-id/<campaign_id>/telephony/
routing/transferdata/dnis/by-dnis/16505559999.18005551212 HTTP/1.1
TE: deflate,gzip;q=0.3
Connection: TE, close
Authorization: Basic ZmlkZWxpdk6dDNzdGNhbGw=
Host: api.liveops.com
User-Agent: libwww-perl/5.79
Content-Type: text/x-louie-xml
Content-Length: 83

<?xml version="1.0" encoding="UTF-8"?>
<request>
  <method>stop</method>
</request>

```

### Response

Table 4-8: Stopping a TDS Session: Response Example

```

HTTP/1.1 200 OK
Date: Fri, 09 Feb 2007 22:06:07 GMT
Server: Apache
Connection: close
Transfer-Encoding: chunked
Content-Type: text/xml

<reply command='<client_id>/struct-xml/my-callcenter/campaigns/by-id/<campaign_id>/telephony/
routing/transferdata/dnis/by-dnis/16505559999.18005551212/stop'>
  <result>
    <hash>
      <hash name='environment'>
        <scalar name='zone'>lv</scalar>
      </hash>
      <hash name='properties'>
        <scalar name='status_code'>1</scalar>
        <scalar name='status'>success</scalar>
        <scalar name='agentdata'>tsid%3D1257923146%26sessionkey%3D3188%26edited%3D1</scalar>
        <scalar name='routingdata'>tsid%3D1257923146%26sessionkey%3D3188%26edited%3D1</scalar>
        <scalar name='reportingdata'>tsid%3D1257923146%26sessionkey%3D3188%26edited%3D1</scalar>
        <scalar name='dnis'>18005551212</scalar>
        <scalar name='guid'>e1ab1:56d68992ff5a89fd94aaa940ff3a6701</scalar>
        <scalar name='type'>dnis</scalar>
        <scalar name='oemkey'>3188</scalar>
      </hash>
    </result>
  </reply>

```