



DOTS Telephone Verification

Developer's Guide

Version 1.1
September 28, 2010
Alex Parsons

Contents

- DOTS Telephone Verification..... 1
 - Developer’s Guide 1
- Contents..... 2
- Operations..... 4
 - PlaceCall..... 4
 - PlaceCall Inputs 4
 - PlaceCall Outputs..... 4
 - SendSMS 6
 - SendSMS Inputs..... 6
 - SendSMS Outputs 6
 - GetPhoneInfo 7
 - GetPhoneInfo Inputs..... 7
 - GetPhoneInfo Outputs 7
 - GetExchangeInfo 8
 - GetExchangeInfo Inputs 8
 - GetExchangeInfo Outputs..... 8
 - GetTransactionDetail..... 9
 - GetTransactionDetail Inputs 9
 - GetExchangeInfo Outputs..... 9
- Accessing the Service..... 10
 - SOAP 1.2 10
 - HTTP GET..... 11
 - HTTP POST..... 12
- Error Codes..... 13
 - User Input Errors..... 13
 - Service Objects Fatal Errors 14
- More Information..... 15
- Appendix A – Supported SMS Characters..... 16
- Appendix B - Tokens: An Advanced Feature 17

About DOTS Telephone Verification

DOTS Telephone Verification (referred to as “Telephone Verification” or “TV”) is a publicly available XML web service with several methods to verify a telephone number. The verification options range from looking at the phone number’s exchange information to dialing the phone number in real time.

Telephone Verification is accessible over three protocols: HTTP GET, HTTP POST, and SOAP over HTTP POST. Most programming languages have built-in classes to access web services over SOAP, but you may access the service via any of these three methods.

Operations

PlaceCall

The PlaceCall operation dials the phone number and reads a verification code aloud. Applications should require an end user to input the same code to verify he is accessible via this phone number.

PlaceCall Inputs

The inputs to this operation are all string objects, and are listed in the table below.

Input Parameter	Type	Description
CountryCode	string	CountryCode for the destination phone number
PhoneNumber	string	PhoneNumber to be verified
Extension	string (optional)	Extension to be dialed after initial pickup
ExtensionPauseTime	string (optional)	Seconds to wait before dialing the Extension
VerificationCode	string (optional)	Code that will be read aloud to the end user
CallerID	string	Caller ID that will display to the end user
Language	string	Language code, e.g. "en" for English
LicenseKey	string	Your license key

Notes:

- As of May 1, 2010, the only CountryCode supported is 1 (USA and Canada). As more CountryCodes are supported, the cost for the PlaceCall operation may be determined by destination country or region.
- If the end user is at an extension, you may specify the extension and the amount of time to wait after the initial pick up to dial the extension digits. We recommend a value of 5 for ExtensionPauseTime (5 seconds).
- If the VerificationCode is not specified, TV will generate a random 3-digit code to read aloud and return this code in the output fields.
- The CallerID field must be a valid contact phone number for your organization.
- As of May 1, 2010, the only Language supported is "en" (English). In the future, other languages may be supported.
- VerificationCodes should be stored in server-side session variables or in a database—they should remain unknown to the end user except through the phone line. For example, do not use a form variable to post the verification code from page to page.

PlaceCall Outputs

The outputs of this operation are grouped under the TelephoneInfo object, containing the member objects listed in the table below.

Output Parameter	Type	Description
Description	string	A brief description of the CountryCode + PhoneNumber prefix
TokensUsed	int	Number of tokens consumed in this request (see Appendix B)
VerificationCode	string	The code read aloud to the user
TransactionID	string	Unique identifier for this request
Debug	string	Utility field; safe to ignore

Note: VerificationCode will return the code read aloud to the end user. If you did not specify this code in the input, a code will be randomly generated for you, and you will need this output value to verify the end user's phone number.

SendSMS

The SendSMS operation sends a text message containing a verification code to the phone number specified. Applications should require the end user to enter the same code to verify he is accessible via this phone number.

SendSMS Inputs

Input Parameter	Type	Description
CountryCode	string	CountryCode of the destination phone number
PhoneNumber	string	Phone number to be verified
Message	string	Your verification message, containing a random code
LicenseKey	string	Your license key

Input Notes:

- As of May 1, 2010, the only CountryCode supported is 1 (USA and Canada). As more CountryCodes are supported, the cost for the SendSMS operation may be determined by destination countries or regions.
- The contents of the Message input must be no more than 140 characters. Most commonly-used characters are supported, such as alphanumeric, punctuation, and most characters you can type on a cell phone. Unsupported characters will be removed from the Message. For a list of supported characters, see [Appendix A – Supported SMS Characters](#).
- We recommend a simple Message string, such as “Your verification code is: XXX” to verify phone numbers. You should generate your codes randomly (do **not** generate them from a system’s internal clock, for example). The verification codes should be stored in server-side session variables or in a database—they should remain unknown to the end user except through the text message. For example, do not use a form variable to post the verification code from page to page.

SendSMS Outputs

The outputs of this operation are grouped under the TelephoneInfo object, containing the member objects listed in the table below.

Output Parameter	Type	Description
Description	string	A brief description of the CountryCode + PhoneNumber prefix
TokensUsed	int	Number of tokens consumed in this request (see Appendix B)
TransactionID	string	Unique identifier for this request
Message	string	The same message input to the operation; for your reference
Debug	string	Utility field; safe to ignore.

GetPhoneInfo

The GetPhoneInfo operation returns contact information associated with a U.S. or Canadian phone number such as address, city, state, zip and exchange information such as carrier, line type, city, and state. Its cost is variable depending on how much information is returned.

GetPhoneInfo Inputs

The inputs to this operation are all string objects, and are listed in the table below.

Input Parameter	Type	Description
PhoneNumber	string	U.S. or Canadian phone number to get info
LicenseKey	string	Your license key

Input Notes:

- Only U.S. and Canadian phone numbers are supported in this operation.

GetPhoneInfo Outputs

The outputs of this operation are grouped under the TelephoneInfo object, containing the member objects listed in the table below.

Output Parameter	Type	Description
TokensUsed	int	Number of tokens consumed in this request (see Appendix B)
TransactionID	string	Unique identifier for this request
Provider	-	(xml parent node)
-Name	string	Name of the exchange carrier
-City	string	City of the exchange
-State	string	State of the exchange
-LineType	string	Type of the exchange: LANDLINE, WIRELESS, VOIP, UNKNOWN
-Latitude	string	Latitude of the city, state
-Longitude	string	Longitude of the city, state
Contact	-	(xml parent node)
-Name	string	Name of the registered contact
-Address	string	Address of the registered contact
-City	string	City of the registered contact
-State	string	State of the registered contact
-Zip	string	Zip of the registered contact
-Type	string	Type of contact: BUSINESS, RESIDENTIAL, UNKNOWN
-Quality	string	Quality of the data source
Debug	string	Utility field; safe to ignore.

GetExchangeInfo

The GetExchangeInfo operation returns exchange information such as carrier, line type, city, and state for any valid U.S. or Canadian phone number.

GetExchangeInfo Inputs

Input Parameter	Type	Description
PhoneNumber	string	Phone number whose exchange is to be verified
LicenseKey	string	Your license key

Input Notes:

- Only U.S. and Canadian phone numbers are supported in this operation.

GetExchangeInfo Outputs

The outputs of this operation are grouped under the TelephoneInfo object, containing the member objects listed in the table below.

Output Parameter	Type	Description
TokensUsed	int	Number of tokens consumed in this request (see Appendix B)
TransactionID	string	Unique identifier for this request
Provider	-	(xml parent node)
-Name	string	Name of the exchange carrier
-City	string	City of the exchange
-State	string	State of the exchange
-LineType	string	Type of the exchange: LANDLINE, WIRELESS, VOIP, UNKNOWN
-Latitude	string	Latitude of the city, state
-Longitude	string	Longitude of the city, state
Debug	string	Utility field; safe to ignore.

Note: Due to line number portability, it is possible that certain phone numbers have been moved to different carriers, cities, or even line types. Exchange information will not reflect ported phone numbers.

GetTransactionDetail

The GetTransactionDetail operation returns detailed information about a previously-submitted transaction. You may only view information about transactions submitted by your license key. No one else can view transactions submitted by your license key.

GetTransactionDetail Inputs

Input Parameter	Type	Description
TransactionID	string	Unique identifier of the transaction
LicenseKey	string	Your license key

GetExchangeInfo Outputs

The outputs of this operation are grouped under the TelephoneInfo object, containing the member objects listed in the table below.

Output Parameter	Type	Description
TransactionID	string	Unique identifier for this request
TransactionDateTime	string	Timestamp of the initial request
Operation	string	Operation requested for this transaction
PhoneNumber	string	Phone number submitted
Status	string	Status: Dialing, Answered, Success, or Error
StatusCode	int	A code representing one of the above, or a negative number indicating a more detailed type of error
Data1	string	Varies by operation
Data2	string	Varies by operation
Data3	string	Varies by operation
TokensUsed	int	Tokens used by this transaction (see Appendix B)
Debug	string	Utility field; safe to ignore.

Accessing the Service

Below are examples of how to request the PlaceCall operation over SOAP 1.2, HTTP GET, and HTTP POST. You may view the same examples at the operation URL. If you require sample code, please contact us.

Operation URL:

<http://trial.serviceobjects.com/TV/TelephoneVerification.asmx?op=PlaceCall>

SOAP 1.2

The following is a sample SOAP 1.2 request and response. The **placeholders** shown need to be replaced with actual values.

```
POST /tv/TelephoneVerification.asmx HTTP/1.1
Host: trial.serviceobjects.com
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap12="http://www.w3.org/2003/05/soap-
envelope">
  <soap12:Body>
    <PlaceCall xmlns="http://www.serviceobjects.com/">
      <CountryCode>string</CountryCode>
      <PhoneNumber>string</PhoneNumber>
      <Extension>string</Extension>
      <ExtensionPauseTime>string</ExtensionPauseTime>
      <VerificationCode>string</VerificationCode>
      <CallerID>string</CallerID>
      <Language>string</Language>
      <LicenseKey>string</LicenseKey>
    </PlaceCall>
  </soap12:Body>
</soap12:Envelope>

HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap12="http://www.w3.org/2003/05/soap-
envelope">
  <soap12:Body>
    <PlaceCallResponse xmlns="http://www.serviceobjects.com/">
      <PlaceCallResult>
        <Country>string</Country>
        <Description>string</Description>
        <TokensUsed>int</TokensUsed>
        <VerificationCode>string</VerificationCode>
        <TransactionID>int</TransactionID>
        <Message>string</Message>
        <CallerID>string</CallerID>
        <CallerIDName>string</CallerIDName>
        <VerificationDate>string</VerificationDate>
        <Provider>
          <Name>string</Name>
          <City>string</City>
          <State>string</State>
        </Provider>
      </PlaceCallResult>
    </PlaceCallResponse>
  </soap12:Body>
</soap12:Envelope>
```

```

    <LineType>string</LineType>
    <Latitude>string</Latitude>
    <Longitude>string</Longitude>
  </Provider>
  <Contact>
    <Name>string</Name>
    <Address>string</Address>
    <City>string</City>
    <State>string</State>
    <Zip>string</Zip>
    <Type>string</Type>
    <Quality>string</Quality>
  </Contact>
  <Error>
    <Type>string</Type>
    <TypeCode>string</TypeCode>
    <Desc>string</Desc>
    <DescCode>string</DescCode>
  </Error>
  <Debug>string</Debug>
</PlaceCallResult>
</PlaceCallResponse>
</soap12:Body>
</soap12:Envelope>

```

HTTP GET

The following is a sample HTTP GET request and response. The **placeholders** shown need to be replaced with actual values.

```

GET
/tv/TelephoneVerification.asmx/PlaceCall?CountryCode=string&PhoneNumber=string&Extension=string&E
xtensionPauseTime=string&VerificationCode=string&CallerID=string&Language=string&LicenseKey=string
 HTTP/1.1
Host: trial.serviceobjects.com

HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<TelephoneInfo xmlns="http://www.serviceobjects.com/">
  <Country>string</Country>
  <Description>string</Description>
  <TokensUsed>int</TokensUsed>
  <VerificationCode>string</VerificationCode>
  <TransactionID>int</TransactionID>
  <Message>string</Message>
  <CallerID>string</CallerID>
  <CallerIDName>string</CallerIDName>
  <VerificationDate>string</VerificationDate>
  <Provider>
    <Name>string</Name>
    <City>string</City>
    <State>string</State>
    <LineType>string</LineType>
    <Latitude>string</Latitude>
    <Longitude>string</Longitude>
  </Provider>
  <Contact>
    <Name>string</Name>
    <Address>string</Address>
    <City>string</City>
    <State>string</State>
    <Zip>string</Zip>
    <Type>string</Type>
    <Quality>string</Quality>

```

```

</Contact>
<Error>
  <Type>string</Type>
  <TypeCode>string</TypeCode>
  <Desc>string</Desc>
  <DescCode>string</DescCode>
</Error>
<Debug>string</Debug>
</TelephoneInfo>

```

HTTP POST

The following is a sample HTTP POST request and response. The **placeholders** shown need to be replaced with actual values.

```

POST /tv/TelephoneVerification.asmx/PlaceCall HTTP/1.1
Host: trial.serviceobjects.com
Content-Type: application/x-www-form-urlencoded
Content-Length: length

```

```

CountryCode=string&PhoneNumber=string&Extension=string&ExtensionPauseTime=string&VerificationCode
=string&CallerID=string&Language=string&LicenseKey=string

```

```

HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

```

```

<?xml version="1.0" encoding="utf-8"?>
<TelephoneInfo xmlns="http://www.serviceobjects.com/">
  <Country>string</Country>
  <Description>string</Description>
  <TokensUsed>int</TokensUsed>
  <VerificationCode>string</VerificationCode>
  <TransactionID>int</TransactionID>
  <Message>string</Message>
  <CallerID>string</CallerID>
  <CallerIDName>string</CallerIDName>
  <VerificationDate>string</VerificationDate>
  <Provider>
    <Name>string</Name>
    <City>string</City>
    <State>string</State>
    <LineType>string</LineType>
    <Latitude>string</Latitude>
    <Longitude>string</Longitude>
  </Provider>
  <Contact>
    <Name>string</Name>
    <Address>string</Address>
    <City>string</City>
    <State>string</State>
    <Zip>string</Zip>
    <Type>string</Type>
    <Quality>string</Quality>
  </Contact>
  <Error>
    <Type>string</Type>
    <TypeCode>string</TypeCode>
    <Desc>string</Desc>
    <DescCode>string</DescCode>
  </Error>
  <Debug>string</Debug>
</TelephoneInfo>

```

Error Codes

Generally, an error is anything that happens during execution of DOTS Telephone Verification that causes the service to fail. The source of these errors may be user input, license key transaction limits, timeouts, and unexpected circumstances. **Applications should expect and handle any and all errors appropriately.** Occasionally, new error types or descriptions may be introduced to the web service as new features and data sources are implemented. Description strings may change slightly, so use the description code to determine which error has occurred. Errors are organized into types, listed in the table below.

Type	Type Code	Billable	Standard Across Gen2
Authorization	1	No	Yes
User Input	2	Yes	No
Service Objects Fatal	3	No	Yes
Domain Specific	4	Yes	No

The Error object appears under the main return object. Below is an example of the error returned when the PlaceCall operation is accessed without a valid license key.

```
<TelephoneInfo>
  <Error>
    <Type>Authorization</Type>
    <TypeCode>1</TypeCode>
    <Desc>Please provide a valid license key for this web service.</Desc>
    <DescCode>1</DescCode>
  </Error>
</TelephoneInfo>
```

User Input Errors

Code	Description
1	Please enter a valid CountryCode plus PhoneNumber combination.
2	PlaceCall is not enabled for this CountryCode and PhoneNumber combination.
3	Please enter a valid 10-digit phone number for the CallerID field.
4	Please enter a valid extension consisting of digits and the 'w' character.
5	Please enter a valid extension pause time in whole seconds.
6	Please use a supported language code. Currently supported languages: <i>(varies)</i>
7	VerificationCode must be an empty string or a string of 3 <i>(varies)</i> digits.
8	VerificationCode must be a string of 3-6 <i>(varies)</i> digits.
9	SendSMS is not enabled for this CountryCode and PhoneNumber combination.
10	Please enter a valid message between 1 and 140 characters long.
11	Please enter a valid 10-digit phone number for the PhoneNumber field.
12	TransactionID does not exist.

Service Objects Fatal Errors

Errors of type code 3, Service Objects Fatal, are unexpected errors. With complex web services, unexpected errors will occur, and we will work to correct them as quickly and prudently as possible. If you observe an error of this type, please record the specific input values passed to the web service, as well as the date and time, and report them to Service Objects.

Code	Description
1	Database error retrieving country code information.
2	Error placing call in primary queue.
3	Error placing call in secondary queue.
4	Error sending message to primary SMS provider.
5	Error sending message to secondary SMS provider.
?	Other error occurred. Contact Service Objects.

More Information

Service Objects is proud to offer you a free trial of DOTS Telephone Verification. Please contact sales@serviceobjects.com.

Technical questions should be directed to support@serviceobjects.com or your sales rep.

We welcome your feedback! Please let us know what you think of our services, documentation, or customer support.

Contact URL:

<http://www.serviceobjects.com/about-us/contact-us>

Appendix A – Supported SMS Characters

The following is a list of characters supported in the SendSMS operation. Characters not in the table below are unsupported and will be removed from the Message.

Unicode	IA5	Unicode	IA5	Unicode	IA5	Unicode	IA5	Unicode	IA5	Unicode	IA5
64	0	928	22	44	44	66	66	88	88	110	110
163	1	968	23	45	45	67	67	89	89	111	111
36	2	931	24	46	46	68	68	90	90	112	112
165	3	952	25	47	47	69	69	196	91	113	113
232	4	926	26	48	48	70	70	214	92	114	114
233	5	SP	27	49	49	71	71	209	93	115	115
249	6	198	28	50	50	72	72	220	94	116	116
236	7	230	29	51	51	73	73	167	95	117	117
242	8	223	30	52	52	74	74	191	96	118	118
199	9	201	31	53	53	75	75	97	97	119	119
10	10	SP	32	54	54	76	76	98	98	120	120
216	11	33	33	55	55	77	77	99	99	121	121
248	12	34	34	56	56	78	78	100	100	122	122
13	13	35	35	57	57	79	79	101	101	228	123
197	14	164	36	58	58	80	80	102	102	246	124
229	15	37	37	59	59	81	81	103	103	241	125
916	16	38	38	60	60	82	82	104	104	252	126
95	17	39	39	61	61	83	83	105	105	224	127
934	18	40	40	62	62	84	84	106	106		
915	19	41	41	63	63	85	85	107	107		
923	20	42	42	161	64	86	86	108	108		
937	21	43	43	65	65	87	87	109	109		

Appendix B - Tokens: An Advanced Feature

The standard usage and pricing plan for use of DOTS Telephone Verification is based on Transactions as the aggregated and charged for unit. Transactions in this context are defined as: any DOTS Telephone Verification procedure call made to the Service Objects web service, regardless of returned information or use of returned information.

There is an advanced feature of DOTS Telephone Verification for more sophisticated users called Tokens. A Token is a unit of cost where different DOTS Telephone Verification Procedure calls are weighted differently resulting in a variable cost per different transaction type. Transactions pricing is therefore variable, reflecting differing Token costs for less complex Transactions, and therefore the plan is based on purchasing an amount of Tokens per month, not Transactions.

If a user desires closer accounting and measurement of specific Procedures used by their application and a payment plan that reflects this through variable pricing then the use of Tokens may provide added value. The Token cost structure is listed in the following table.

Per Procedure Token Costs	Tokens
PlaceCall	70
SendSMS	85
GetExchangeInfo	15
GetPhoneInfo	65
GetTransactionalDetail	0

If you decide that use of the Token based structure and pricing plan may be an advantage for you please contact Service Objects Sales Team for assistance: sales@serviceobjects.com or 1-805-963-1700.