Forward Phone Append API Developer's Guide

Document Version 1.1.1 – December 11th, 2024

API Version 1.1

PacificEast
8625 SW Cascade Avenue, Suite 250
Beaverton, OR 97008

Table of Contents

PacificEast Development

Overview

PacificEast web services provide access to the best data in the industry. With access to multiple vendors and data sources, PacificEast strives to provide the most relevant and up-to-date responses to your queries. PacificEast is committed to providing the best solution to your data needs by combining our sophisticated algorithms with a wide array of data sources.

Most PacificEast web services are accessible using either the Simple Object Access Protocol (SOAP) version 1.1 or Representational State Transfer (REST).

This document describes accessing these services using both SOAP and REST. Further information on SOAP can be found on the World Wide Web Consortium's web site at www.w3.org. More information on REST can be found in various places including Wikipedia at http://en.wikipedia.org/wiki/Representational_state_transfer.

Architecture

PacificEast web services are hosted in several data centers throughout North America and are configured to provide rapid failover in the event of issues. PacificEast uses a combination of load balancing, failover and external monitoring to maintain high service uptimes.

Security

PacificEast web services use Transport Layer Security (TLS) for all communication between client applications and the PacificEast servers. Each request to an PacificEast web service must include the account key that was assigned by PacificEast.

Production vs. Development

All development work must be done on PacificEast's client development servers. Once development has been completed, clients may request that their account on the production servers be enabled by contacting their account manager or by emailing devsupport@pacificeast.com.

Access to the development servers is granted solely for the purpose of testing code integration with the PacificEast web services. The development servers are not to be used for stress or load testing. Queries to the development servers should not exceed 100 queries per day or 1,000 queries total. Queries exceeding these totals may be billable. If additional queries are needed, please contact your account manager or email devsupport@pacificeast.com. Excessive use of the development servers will result in access being revoked. Access to the development servers will be disabled after the production account has been enabled.

The development servers have access to the same data sources as the production servers. However, the production servers are quicker and have redundancy and load-balancing features implemented. While we do our best to ensure that the development servers are available at all times, there may be times when a web service on a development server is unavailable because of maintenance. These periods will usually

be very brief and we will do our best to perform maintenance outside of regular business hours. However, please note that there will not be maintenance notifications issued for our development servers. If you have a question about the status of a particular service on a development server, please email devsupport@pacificeast.com.

Production servers should be available at all times. Using multiple data centers and load-balancing, we strive to provide uninterrupted service to our clients. However, there may be times when maintenance, either our own or at our downstream providers, will affect our services. Whenever possible, we will - provide maintenance notifications at least a week before the scheduled maintenance.

Support

During development, support for PacificEast web services can be obtained by emailing our development support group at devsupport@pacificeast.com. Please be sure to include details about the nature of the problem and your contact information.

Support for production systems can be obtained by emailing devsupport@pacificeast.comFor emergency support, please call our 1-800 number.

Errors

Errors in SOAP-based queries will be indicated using SOAP faults. See the SOAP Fault specification at http://www.w3.org/TR/2000/NOTE-SOAP-20000508/#_Toc478383507 for more information. Information about the error is returned in the fault as an IdiciaError object (see *IDICIAError* below).

Account Configuration

By default, your Forward Phone Append account is configured to return only a single result that best satisfies the input criteria. If you would like your account to be configured to return multiple results (up to 30) please contact your account manager or email devsupport@pacificeast.com.

Query Tuning Recommendations

Queries against API services may not operate exactly like the same query against a batch service. Real-time/API services that query consumer information are typically designed to provide back more specific answers with less “fuzziness” than a batch query. Fortunately, you aren’t charged for consumer information queries that don’t find a result so making a few attempts, each time slightly altering the query, can be a good strategy to improve your odds of finding the correct result.

When you implement searches against this API, we recommend that you develop a search strategy that allows you to execute additional queries if your first query doesn’t produce a result. Specifically, we suggest you start with a very focused query using as much information as you know. Then, if no match is found, start reducing the amount of information you pass in the query arguments.

As an example, if you execute a search for a person using their full name and address but don’t find a match, try first removing their first name from the query. Our systems try to allow for nicknames but the individual’s phone number may be associated to another person in the household, especially for landlines. Removing the first name will broaden the search to any phone associated to that surname at that address.

Original Query: John Smith, 123 Main Street, Apt 1A, Springfield, KY, 25124

Household Version: Smith, 123 Main Street, Apt 1A, Springfield, KY, 25124

Making the above change to the query (using the Household Version if the Original Query fails) may provide a record for someone in the household. Of course, this is all dependent on your particular use case and whether or not a "household" record is acceptable.

Another option for tuning a query is to remove all or part of the address. A phone number may not be found because it is associated to a prior or newer address. If you have already verified the address you are searching with is the current address, try removing parts of the address and re-executing the query. We suggest you modify the address portion of a query in the following order to allow you to get as tight of a match as possible:

1. Leave the primary address intact but remove any secondary addresses like unit numbers, suite numbers or apartment numbers. (It's possible the unit number provided isn't correct but a record may be found with the correct unit number).
2. If executing option #1 doesn't produce a match, remove the city/state but leave the postal code.
3. If option #2 doesn't produce a match, put the city/state back in, but remove the postal code.
4. Finally, if your use case allows for matches at different addresses, remove the entire address and execute the search.

Note that if your name is fairly common, as in our example, you may find a result but it may not be the person for whom you are searching. Searching for an uncommon name in a city may help you find the correct record but searching for a very common name, especially in a large city or town, is more likely to find a record that may not be the correct person.

Using our above example, here are the four versions of queries you might try:

1. Smith, 123 Main, Springfield, KY, 25124
2. Smith, 123 Main, 25124
3. Smith, 123 Main, Springfield, KY
4. Smith, Springfield, KY, 25124

Altering the query arguments and re-executing the query in the above order should provide you with the best matches first. However, there is, of course, always a chance that the record just doesn't exist at all. Query latency adding up with all these searches can be a concern. However, our systems generally return a not-found condition very quickly. The majority of time spent for processing your query is typically used in formatting and filtering the answers from searches that produce many results (especially when your account is configured to return multiple results) so executing additional queries that don't produce results should not normally add significantly to your aggregate transaction processing time.

We realize that different customers may have different business purposes and may not be able to accept different kinds of query results so some of these examples may not be appropriate for your particular use. We offer them only to provide ideas of how you might try modifying queries if you aren't getting the

results you expect. If you are having challenges finding the records you are searching for we're happy to assist with advice and suggestions about query execution plans that may improve your search success within the limits of your particular use case. If you need advice please reach out to your account manager to schedule a consultation with our data analysts or developers. Its free and available to any customer of an PacificEast API service.

Authentication

Service authentication is performed using a custom header and the account key assigned by PacificEast. Each request must include a header with the name *X-PacificEast-Acct* and the account key as the value.

For example, if your account key is "abc123" then the header included with each request should look like:

X-PacificEast-Acct : abc123

Requests not containing the *X-PacificEast-Acct* header will be rejected with a 403 (Unauthorized) response.

Transaction Requests

The service Web Service Description Language (WSDL) is available from our web site. Please contact your account manager or email devsupport@pacificeast.com.

GetContacts

The GetContacts transaction allows for contact information to be retrieved based on input name and address information.

Table 1 – GetContacts Parameters

Property	Description	Type	Required
queryParameters	A QueryParameters object (see below).	QueryParameters	Y
appendParameters	An AppendParameters object (see below)	AppendParameters	Y

The QueryParameters object encapsulates the query parameters that are common to all PacificEast web service requests.

Table 2 – QueryParameters

Parameter	Description	Type	Required
consentCode	The consent code indicating what consent, if any, was given for retrieving the requested data. Consent code values are mutually agreed upon by PacificEast and its customers. Resellers should pass a consentCode in two parts. Part 1 is their contractually established consent code. After a delimiting colon (":") reseller should add a unique indicator of their own choosing that indicates which of their customers had initiated the transaction. A typical end user consent code might look like "ABC123". A typical consent code from a reseller might look like "ABC123:0129".	Text	N
jobCode	An identifier (twenty characters maximum) that can be included for reporting purposes. Queries with the same job code will be grouped together on the invoice.	Text	N
purpose	Indicates what the retrieved data will be used for. Valid values are: **LE** law enforcement **FR** commercial fraud and risk prevention **AD** administrative (including demo and testing) **CC** contractual consent (including HIPAA uses)	Text	N
referenceID	An identifier that can be used by the client application to uniquely identify the transaction. This text will be returned in the response.	Text	N

Table 3 –AppendParameters

Property	Description	Type	Required
queryType	Specifies the type of query to be performed: 0 Standard 1 Prioritize landline 2 Prioritize wireless	Text	Y
firstName	First name of the contact.	Text	N
lastName	Last name of the contact.	Text	N
businessName	Business name of the contact.	Text	N
address	Street address of the contact.	Text	N
city	City of the contact.	Text	N
state	Two character state or province abbreviation.	Text	N
postalCode	The five digit ZIP code for US contacts or the six character postal code (do not include spaces) for Canadian contacts.	Text	N
dob	Reserved for future use.	Text	N
ssn	Reserved for future use.	Text	N

Append Parameter Combinations

The following table lists the valid combinations of append parameters. Queries submitted using a combination of parameters other than those listed below will result in *Invalid query* (error code 2) error.

Table 4 - Parameter Combinations

First Name	Last name	Business name	Address	City	State	Postal
o	X		X	X	X	
o	X		X			X
o	X		X	X	X	X
o	X		X		X	
X	X					X
		X	X	X	X	
		X	X			X
		X	X	X	X	X
		X	X		X	
		X				X
		X			X	
			X	X	X	X
			X	X	X	
			X			X
			X		X	

X - designates a parameter for which a value is supplied.

o – parameter value is optional.

Transaction Responses

IDICIAError

The IDICIAError object is returned when a transaction fails and contains information on why the transaction failed.

Table 5 - IDICIAError

Field	Description	DataType
status	Indicates if the query was successfully executed or not. The status should have a value of -1 if an IDICIAError object is received	Integer

errorInfo	An ErrorInfo object (see below)	ErrorInfo

Table 6 - ErrorInfo

Field	Description	DataType
code	The error code describing the type of error that occurred. See Appendix A - Error Codes for a list of possible error codes.	Integer
description	A text description of the error.	Text
internalCode	For PacificEast internal use. You may be asked to provide this value when working with PacificEast support to resolve an issue.	Text

Response

A successful transaction is returned in a Response object.

Table 7 - Response

Field	Description	DataType
status	Indicates if the query was successfully executed or not. -1 An error occurred executing the query. 0 Query executed successfully.	Integer
referenceID	The referenceID that was included in the input.	Text
lookupResult	The result of the contact search. Possible values are: -1 – An error occurred executing the search. 0 – No contacts were found for the input query information. 1 – One or more contacts were found.	Integer
contactsFound	The number of contacts found.	Integer
contacts	An array of ContactInfo objects (see below).	ContactInfo[]
matchInfo	An array of MatchInfo objects (see below)	MatchInfo[]

Table 8 - ContactInfo

Field	Description	DataType
phoneInformation	A PhoneInfo object (see below).	PhoneInfo
nameInformation	A NameInfo object (see below).	NameInfo
addressInformation	An AddressInfo object (see below).	AddressInfo

The ContactInfo object contains the detailed information about all contacts found that match the input criteria. The contact information is divided into three separate objects the list the phone, name and address information.

Table 9 - PhoneInfo

Field	Description	DataType
phoneNumber	The ten digit phone number of the contact.	Text
carrier	Not available. Will always be blank.	Text
contactType	The record type of the contact data. Possible values include: • R – residential • B – business • BR – business/residential If the record type is unknown the field will be left empty.	Text
startDate	The earliest date known for the contact if available. The data will be in YYYYMMDD format.	Text
transactionDate	The last date a change was made to the contact if available. The data will be in YYYYMMDD format.	Text
source	The source of the contact data. Possible values include: • Billing • Bureau • DA • Delisted • NonDA • Hosted If the source is unknown the field will be left empty.	Text
phoneServiceType	Not available. Will always have a value of “Unknown or multiples”.	Text

Table 10 - NameInfo

Field	Description	DataType
firstName	The first name of the contact.	Text
lastName	The last name of the contact.	Text
businessName	The business name of the contact.	Text

Table 11 - AddressInfo

Field	Description	DataType
address	The street address of the contact.	Text
unit	The unit number or secondary address.	Text
city	The city of the contact.	Text
state	The two character state or province abbreviation.	Text

postalCode	The five digit ZIP (US) or six character postal code (Canada).	Text
country	The country of the contact.	Text
latitude	Latitude with a maximum of six decimal accuracy.	Text
longitude	Longitude with a maximum of six decimal accuracy.	Text

MatchInfo

The MatchInfo object contains the following fields:

firstName
lastName
businessName
overallName
streetAddress
secondaryAddress
overallAddress
city
state
postal
location

The *overallName*, *overallAddress* and *location* values are composite scores that can be used to simplify evaluation of the matches. The *overallName* score is based on the individual scores for the first, last and business names. The *overallAddress* score is based on the street and secondary address scores. The *location* score is based on the city, state and postal scores.

Each field will have one of the following values:

Table 12 - Match Scores

Value	Description
-1	Not Compared – the field was not compared because either the corresponding input parameter was empty or no value was found to compare the parameter to.
0	None – there was no match between the input and the found value.
2	Low – there was a low degree of similarity between the input and the found value.
8	High – there was a high degree of similarity between the input and the found value.
10	Exact- the input and the found value were an exact match.

A Note About Data Fields

The Forward Append Phone service uses many different data sources to obtain the best results possible. Not all of the sources provide all of the fields in the PhoneInfo, NameInfo and AddressInfo objects listed above. The service will return as much data as possible but some of the fields listed above may be blank.

It should also be noted that the order that the fields appear may not remain consistent. Parsing of the PhoneInfo, NameInfo and AddressInfo objects should not rely on the order of the fields as listed above. Also, additional fields may be added in future versions.

Samples

The following samples show various transactions using the service.

Note: the data in the responses is fictional and is just a representation of the data that would be returned by the service.

SOAP samples: for the sake of brevity and readability, some of the SOAP envelope and namespace information has been removed. Only the relevant XML has been left.

REST samples: only the last portion of the resource and the parameters are shown. The JSON response has been formatted for easier readability.

GetContacts Samples

Sample #1 – Unauthorized Access

This request shows the IdiciaError response that is returned when an error occurs. In this case, the error is caused by an invalid account key.

REST Request

```
…/Append.svc/0?referenceID=Test%20query&lastName=Smith&address=123%20Main%20St&city=Seattle&state=WA
```

REST Response

```
{
  "status": -1,
  "errorInfo":   {
    "code": 1,
    "description": "Unauthorized access.",
    "internalCode": 1235
  }
}
```

SOAP Request

```
<AppendQuery>
        <queryParameters>
                <referenceID>Test Query</referenceID>
        </queryParameters>
        <appendParameters>
                <queryType>0</queryType>
                <lastName>Smith</lastName>
                <address>123 Main St</address>
                <city>Seattle</city>
                <state>WA</state >
        </appendParameters>
</AppendQuery>
```

SOAP Response

```
<faultcode>a:Forbidden</faultcode>
<faultstring>Forbidden</faultstring>
<detail>
        <IdiciaError>
                <status>-1</status>
                <errorInfo>
                        <code>1</code>
                        <description>Unauthorized access.</description>
                        <internalCode>1235</internalCode>
                </errorInfo>
```

```
        </IdiciaError>
    </detail>
```

Sample #2 – Simple Request

This sample shows a basic request for the last name ‘Smith’ with a street address of ‘123 Main St’ in Seattle, WA using query type 0 (standard).

REST Request

```
…/Append.svc/0?referenceID=Simple%20Request&lastName=Smith&address=123%20Main%20St&city=Seattle&state=WA
```

REST Response


```
{
    "contacts": [
        {
            "nameInformation": {
                "businessName": "",
                "firstName": "John",
                "lastName": "Smith"
            },
            "addressInformation": {
                "address": "123 Main St",
                "city": "Seattle",
                "country": "US",
                "latitude": "",
                "longitude": "",
                "postalCode": "981051234",
                "state": "WA",
                "unit": "Apt 2A"
            },
            "phoneInformation": {
                "carrier": null,
                "contactType": "R",
                "phoneNumber": "2062344321",
                "phoneServiceType": "Unknown or multiples",
                "source": "DA",
                "startDate": "20190901",
                "transactionDate": "20190901"
            }
        }
    ],
    "contactsFound": 1,
    "lookupResult": 1,
    "matchInfo": [
        {
            "businessName": -1,
            "city": 10,
            "firstName": -1,
            "lastName": 10,
            "location": 10,
            "overallAddress": 10,
            "overallName": 8,
            "postal": -1,
            "secondaryAddress": -1,
            "state": 10,
            "streetAddress": 10
        }
    ],
    "referenceID": "Simple Request",
    "status": 0
}}
```


SOAP Request

```
<GetContactsQuery>
```

```
            <queryParameters>
                <referenceID>Simple Request<referenceID>
            <queryParameters>
            <appendParameters>
                <queryType>0</queryType>
                <lastName>Smith</lastName>
                <address>123 Main St</address>
                <city>Seattle</city>
                <state>WA</state >
            </appendParameters>
        <GetContactsQuery>
```

SOAP Response

```
<Response>
  <contacts>
    <ContactInfo>
      <nameInformation>
        <businessName/>
        <firstName>John</firstName>
        <lastName>Smith</lastName>
      </nameInformation>
      <addressInformation>
        <address>123 Main St</address>
        <city>Seattle</city>
        <country>US</country>
        <latitude/>
        <longitude/>
        <postalCode>981051234</postalCode>
        <state>WA</state>
        <unit>Apt 2A</unit>
      </addressInformation>
      <phoneInformation>
        <carrier i:nil="true"/>
        <contactType>R</contactType>
        <phoneNumber>2062344321</phoneNumber>
        <phoneServiceType>Unknown or multiples</phoneServiceType>
        <source>DA</source>
        <startDate>20190901</startDate>
        <transactionDate>20190901</transactionDate>
      </phoneInformation>
    </ContactInfo>
  </contacts>
  <contactsFound>1</contactsFound>
  <lookupResult>1</lookupResult>
  <matchInfo>
    <MatchInfo>
      <businessName>-1</businessName>
      <city>10</city>
      <firstName>-1</firstName>
      <lastName>10</lastName>
      <location>10</location>
      <overallAddress>10</overallAddress>
      <overallName>8</overallName>
      <postal>-1</postal>
      <secondaryAddress>-1</secondaryAddress>
      <state>10</state>
      <streetAddress>10</streetAddress>
    </MatchInfo>
  </matchInfo>
  <referenceID>Simple Request</referenceID>
  <status>0</status>
</Response>
```

Appendix A - Error Codes

The following are possible error codes that may be returned in the ErrorCode field of the ErrorInfo object.

Table 13 - Error Codes

Error Code	Description
0	Unknown error.
1	Unauthorized access.
2	Invalid query. The *description* field may provide more information.
3	Query execution error. Retrying the query may provide a successful result.
4	Unauthorized data source. A data source to which the account does not have access has been requested.
5	Unauthorized permissible purpose. The use of the requested permissible purpose has not been granted to the account.

Appendix B – Document History

API Version	Doc. Version	Date	Description
1.0	1.0.0	2020/10/9	Initial release
1.0	1.0.1	2020/10/22	Minor text edits
1.0	1.0.2	2021/09/08	Branding Update
1.1	1.1.0	2024/11/13	Authentication information update
1.1	1.1.1	2024/12/11	Authentication information update