

Web Service (SOAP) Documentation

1 Preface

The information contained in this document explains the current version of the AddressDoctor Web Services (SOAP) as of the date of publication (14. July 2008). Because we must respond to changing market conditions, the information contained in this document should not be interpreted to be a commitment on the part of AddressDoctor, and AddressDoctor cannot guarantee the accuracy of any information presented after the date of publication.

This document is for informational purposes only. AddressDoctor GmbH does not make warranties, express or implied, in this document.

© 2000-2008 AddressDoctor® GmbH. All rights reserved.

2 Summary

AddressDoctor is the leading Global Address Verification System (AVS) on the Internet. First introduced in 1999 the system has pioneered the Web Service market for address verification. This document describes the most recent SOAP based version of the Web Service.

As an address verification system, AddressDoctor offers the full spectrum of processing steps required to improve the quality of address data. With a single Web Service call an address can be transliterated, parsed, corrected, formatted and enriched. Combining this with the ability to process more than 240 countries through just one API leads to a powerful component for address quality.



Process an address with just one Web Service call

While using the AddressDoctor Enterprise software components in a local install is possible, the most significant advantages of using the Web Services are the short implementation time and the fact that no software or data has to be installed and maintained. This lowers the deployment cost and as a result the total cost of ownership significantly.

At the same time the postal reference data is always current as AddressDoctor updates its server systems monthly. The AddressDoctor Web Services are hosted in Germany and redundantly connected to one of the leading internet backbones. This minimizes latency times regardless of where the actual user or application resides. The AddressDoctor Web Services are available 24/7 and we guarantee a high availability of the system. It has surpassed 99.9% in the last years. Service level agreements guaranteeing 99.5% or 99.7% uptime are available.

Utilizing the AddressDoctor Web Services is an alternative to using one of our sophisticated software products. For both we offer sample code and macros that ease the integration into existing software systems. For very high address volumes please take a look at our AddressDoctor Enterprise software (<http://www.AddressDoctor.com/en/products/software/default.asp>) or inquire about the local hosting of the AddressDoctor Web Service at your facility.

The AddressDoctor Web Services (SOAP) discussed in this paper allow physical (postal) addresses to be transmitted electronically for verification using a SOAP message. Your computer system transmits information to be verified via an HTTP/HTTPS POST operation thus traversing most corporate firewalls. The data is processed **synchronously** on one of the AddressDoctor validation servers. A few milliseconds later either corrected/improved data is sent back or a list of alternatives/suggestions is returned to the computer system where the communication originated.

3 Business

AddressDoctor offers a variety of products for address cleansing. The AddressDoctor Web Services are a very useful component that builds on the AddressDoctor software libraries and offers the most important features of these powerful software components.



3.1 Target Groups

The AddressDoctor Web Services can be used by anyone who wants to enhance and maintain the quality of an existing or planned address database. The services are used by online merchants who want to assist their customers when entering their address data. It is also used to prevent fraud by screening customer data in the background. Users intending to process data in batch mode will find a SOAP message suitable for their task of performing synchronous online batch verification. When used in call center environments the FastCompletion features can be used to speed up data entry.

Incorrect or incomplete postal addresses lead to returned/undeliverable merchandise and cause significant expense for restocking/reshipping returned items. Furthermore, the quality of a customer database suffers for subsequent activities such as mailings or performing effective customer relationship management.

Studies show that customers are annoyed when they receive incorrectly addressed parcels or mail. Thus using a validation tool minimizes this problem.

The use of the AddressDoctor Web Services significantly lowers back office expenses by completely eliminating the need for any further data quality improvement. The easy integration keeps setup cost down and the low per address verification fees minimize the commercial risk involved in any software purchase.

3.2 Business Model

AddressDoctor Web Services are a set of Web Services designed to perform several postal address verification and enhancement tasks. Using the Simple Object Access Protocol (SOAP) and other message formats, data is transmitted to the AddressDoctor servers and returned synchronously.

All required postal and other reference data is hosted by AddressDoctor or its suppliers and is queried in real-time. AddressDoctor maintains its servers and ensures high availability and performance. The use of the system is charged on a transactional basis with prices dependent on countries accessed and services requested. Two standard price plans are available.

- Pay as you go
- Prepaid

For the pay as you go pricing model a monthly fee is charged in addition to a per address validation charge. As an alternative a transaction package can be purchased in advance. Please contact an AddressDoctor sales representative (sales@AddressDoctor.com) to learn more about the price plan options available in your region.

3.3 Distribution

The Web Services can be accessed from virtually any computer system connected to the Internet. As standard SOAP messages are used most existing SOAP toolkits are suitable for connecting to the Web Services. Also a number of software applications directly integrate the AddressDoctor Web Service, thus enhancing data quality within these systems.

To access the AddressDoctor Web Service a user account needs to be created. Please go to <http://www.AddressDoctor.com> to access or create an account.

3.4 Future Developments

The services discussed in this paper are the second major revision of the very successful AddressDoctor Web Service. The services are based on the AddressDoctor Enterprise engine version 4.1 and have support for UTF-8 characters in languages such as Greek, Chinese and Japanese.

The service offering has been extended and will continue to be extended to include features such as movers check (NCOA), mail suppression lists, credit scores and others. As access to most of these personalized services is mandated by national privacy laws, it is required that some legal documents are signed. Please contact an AddressDoctor sales representative more information.

AddressDoctor is an innovation leader in global addressing technology. Please look for future enhancements and innovations.

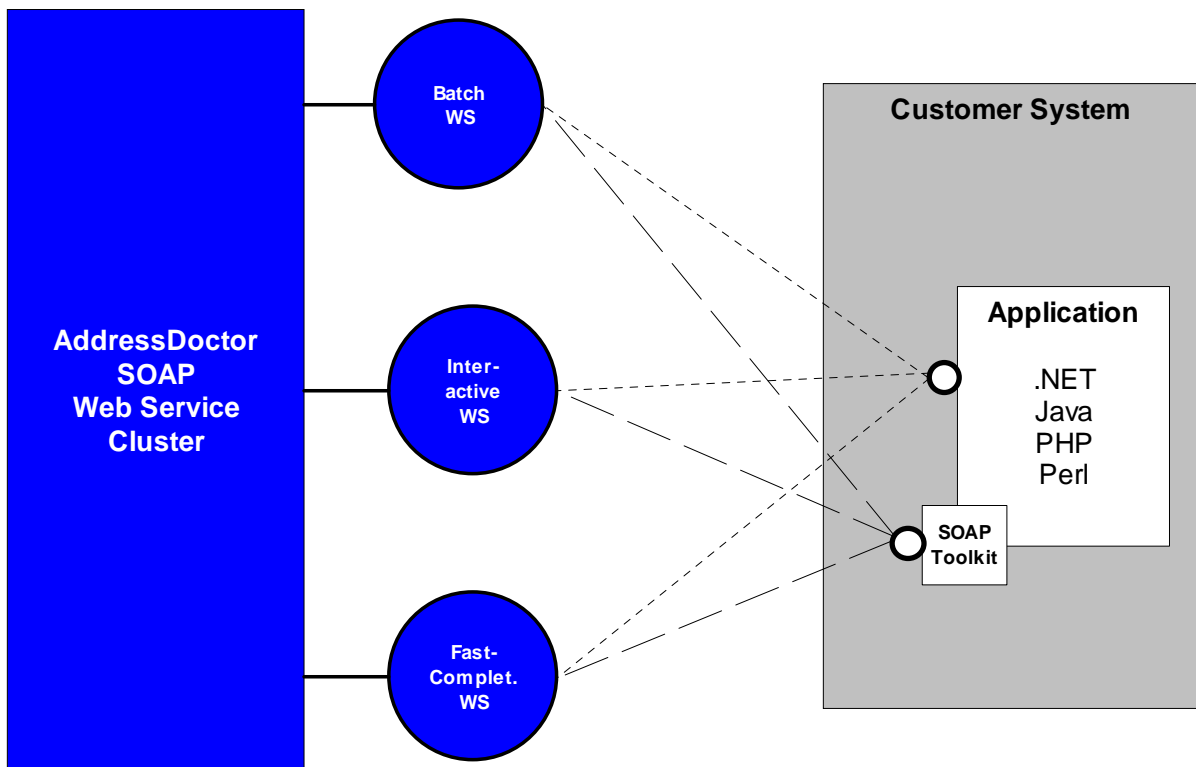
4 Technology

The AddressDoctor Web Service follows the widely accepted SOAP standard as published by the World Wide Web Consortium (www.W3.org). It can be accessed using a variety of SOAP toolkits or by creating and analyzing the XML content directly.

4.1 How it works

The AddressDoctor Web Services (SOAP) are an interface to the AddressDoctor validation technology made available on the Internet. By accessing the AddressDoctor Service cluster it is ensured that the most up to date postal reference data and software is always used for address verification.

Before accessing the Web Services make sure you have created a user account. User accounts are available for trial and development purposes without charge. The following illustration gives a general overview how the AddressDoctor services are typically accessed. An application on the Customer system either directly creates a SOAP request or uses one of the widely available SOAP toolkits to create a SOAP request. This request is transmitted via the internet (SSL is available) to the AddressDoctor Web Service Cluster. AddressDoctor will then process the address(es) and return a proper SOAP response that is then used by the requesting application. All function calls are **synchronous**, that is they immediately return a response.



Basic illustration of a customer application accessing the AddressDoctor Web Services

4.2 User Authentication

AddressDoctor users are identified by a user name and password embedded in the XML data stream. As the HTTP protocol transmits the data unencrypted this data could be intercepted by a malicious attacker. For this reason we recommend using SSL encryption for your requests.

4.3 Data Transmission and Security

All data is transmitted via the HTTP or HTTPS protocol on their default ports (80/443). This is done via a regular HTTP POST operation where the SOAP request is part of the transmitted data. All services are hosted on fault tolerant, redundant servers with the Domain Name System (DNS) name validator2.AddressDoctor.com. In rare occasions we have seen DNS problems on the Internet where resolving the names of the AddressDoctor host was interrupted. For mission critical operations we therefore recommend that you use the IP Address of the service instead of the DNS name. Please notify an AddressDoctor support representative (support@AddressDoctor.com) if you intend to access the

AddressDoctor service via the IP address. This will ensure that you are notified should there be any changes of the IP address of the server systems.

Using an HTTPS request may significantly delay the response because of the encryption overhead. The SSL encryption is using 128 bit.

4.4 Performance

All requests are processed in a synchronous mode. The AddressDoctor Web Service system runs on a Windows 2003 cluster. It uses load balancing for maximum performance and fault tolerance. A typical request will return in less than 100ms. Batch requests containing multiple addresses may take longer.

Tests with the AddressDoctor for Excel Add-In have shown that the Batch Web Service is capable of processing anywhere between 65000 and 250000 addresses per hour. The actual performance will depend on your Internet connection speed.

The AddressDoctor Web Services are hosted in Germany. The time to process a request consists of the actual processing time plus any Internet transmission times. We have observed the following Internet transmission times (packet round trip).

Germany:	50 ms
United States:	200 ms
United Arab Emirates:	200 ms
India:	240 ms
Argentina:	350 ms
Brazil:	320 ms
Malaysia	390 ms
Taiwan:	500 ms

In addition to the basic processing time of approx. 100ms you can, therefore, expect processing from Brazil to take about 420ms.

4.5 Update procedure

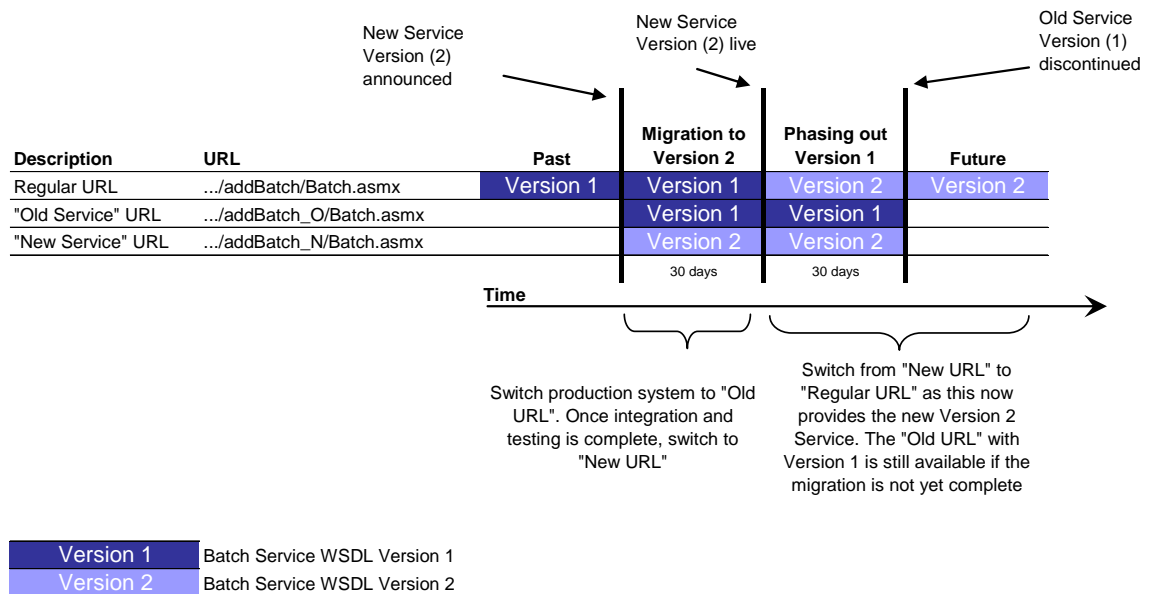
Even though it is our aim to provide a stable API, it sometimes becomes necessary to make improvements to the product offering. In a few cases this leads to a change in the WSDL. Whenever a change is required to the WSDL the new version will be published for testing and evaluation purposes approx. 4-6 weeks before its official release. An underscore character followed by the letter N will be inserted into the URL of the affected service. A new version of the batch service would carry the following name for instance:

http://validator2.AddressDoctor.com/addBatch_N/Batch.asmx

At the same time the already existing Web Service will continue to be available under its existing URL. In addition the existing service will also be accessible through an URL that has been changed to contain the letter O preceded by an underscore such as

http://validator2.AddressDoctor.com/addBatch_O/Batch.asmx

At the end of the migration period, the existing Web Service will be replaced by the new service. The old service will only be available through the service with the _O in its name. The following graphic illustrates this procedure.



4.6 Selecting the appropriate Web Service

AddressDoctor offers three Web Services for three distinct purposes. All services use similar request and response structures that vary only where necessary. The Web Services are

- AddressDoctor Batch Web Service
- AddressDoctor Interactive Web Service
- AddressDoctor FastCompletion Web Service

The service descriptions can be found at

<http://validator2.AddressDoctor.com/addBatch/Batch.asmx>
<http://validator2.AddressDoctor.com/addInteractive/Interactive.asmx>
<http://validator2.AddressDoctor.com/addFastCompletion/FastCompletion.asmx>

The Batch Web Service allows for the validation of up to ten (10) addresses in one single request. Each address will either be flagged as correct, it will be corrected or receive a status flag indicating its deliverability rating. There are no suggestions/alternatives returned for batch transactions.

The Interactive Web Service is similar to the AddressDoctor XML Connector that was offered previously. It accepts one address for correction and will return suggestions/alternatives for ambiguous records. Each suggestion is structured as such that a user that is presented with this suggestion can "pick with one click". Depending on the outcome of the validation operation, an address might be submitted again for validation if the first or preceding validation attempt did not lead to a final status (V or C). Addresses that can be submitted again for processing contain a job token for reprocessing the record at no additional charge. Up to three attempts per address are possible.

The FastCompletion Web Service was designed for call center environments, where truncated data is entered to save data entry time. The AddressDoctor service will automatically complete the data entered or generate meaningful suggestions that a user can pick from. In some countries such as the UK the postal code is often sufficient for generating a full address.

The following table compares the features of the three Web Services

	Batch	Interactive	FastCompletion
Addresses per request	Up to 10	1	1
Maximum Request size	30 KB (50 KB)	3 KB (5 KB)	3 KB (5 KB)
Suggestions returned	No	Yes (up to 20)	Yes (up to 20)
Submit truncated input data	No (only missing elements allowed)	No (only missing elements allowed)	Yes
Parse input data	Yes	Yes	No
Submit unstructured data	Yes	Yes	No
Data enrichment	Yes	Yes	No
More than 240 countries	Yes	Yes	Yes

4.7 Example of a typical session

The following example explains a typical session between your host and the AddressDoctor Interactive Web Service. Usually a SOAP toolkit is used to generate the XML request shown below. You may, however, generate the request yourself.

If you do construct the request yourself, be advised, that XML namespaces are case sensitive (i.e. "http://validator2.addressdoctor.com/addinteractive/interactive" is not correct and will result in the rejection of your request. Also, you must make sure that you send the following HTTP header, so that the server will recognize the request as a SOAP request (example to access the Interactive Web Service):

```
SOAPAction: http://validator2.AddressDoctor.com/addInteractive/Interactive/Validate
```

Also make sure that you set the default XML namespace in the <Validate> tag as shown below. If you bind that namespace to a special name, you must provide that name in every following tag. For example, if you specify the tag as follows:

```
<ad:Validate xmlns:ad="http://validator2.AddressDoctor.com/addInteractive/Interactive">
```

then the following **will not work**:

```
<addInteractiveRequest>
<Authentication>
```

You must then specify all following tags as:

```
<ad:addInteractiveRequest>
<ad:Authentication>
```

Here is a sample request

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" >
  <soap:Body>
    <Validate xmlns="http://validator2.AddressDoctor.com/addInteractive/Interactive">
      <addInteractiveRequest>
        <Authentication>
          <CustomerID>12345</CustomerID>
          <DepartmentID>0</DepartmentID>
          <Password>myPassword</Password>
        </Authentication>
        <CampaignID>myCampaignID</CampaignID>
        <JobToken></JobToken>
      </addInteractiveRequest>
    </Validate>
  </soap:Body>
</soap:Envelope>
```

```

<Parameters>
  <CountryOfOrigin>COO_FIN</CountryOfOrigin>
  <StreetWithHNo>true</StreetWithHNo>
  <CountryType>NAME_EN</CountryType>
  <LineSeparator>LST_LF</LineSeparator>
  <PreferredLanguage>PFL_NO_CHANGE</PreferredLanguage>
  <Capitalization>NO_CHANGE</Capitalization>
  <FormattedAddressWithOrganization>>false</FormattedAddressWithOrganization>
  <ParsedInput>NEVER</ParsedInput>
  <RemoveDiacritics>>false</RemoveDiacritics>
</Parameters>
<Address>
  <RecordID>998887</RecordID>
  <Organization>AddressDoctor GmbH</Organization>
  <Department>Sales</Department>
  <Contact>Steffen Preuschoff</Contact>
  <Building></Building>
  <Street>Röntgenstr. 9</Street>
  <HouseNumber></HouseNumber>
  <POBox></POBox>
  <Locality>Maxdorf</Locality>
  <PostalCode>67133</PostalCode>
  <Province></Province>
  <Country>Deutschland</Country>
  <Residue></Residue>
  <CountrySpecificLocalityLine></CountrySpecificLocalityLine>
  <DeliveryAddressLines></DeliveryAddressLines>
  <FormattedAddress></FormattedAddress>
</Address>
</addInteractiveRequest>
</Validate>
</soap:Body>
</soap:Envelope>

```

The request above will return the following result

```

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" >
  <soap:Body>
    <ValidateResponse xmlns="http://validator2.AddressDoctor.com/addInteractive/Interactive">
      <ValidateResult>
        <ErrorCode>0</ErrorCode>
        <ErrorMessage></ErrorMessage>
        <CampaignID>myCampaignID</CampaignID>
        <ValidationStatus>V</ValidationStatus>
        <ResultCount>1</ResultCount>
        <Results>
          <Result>
            <Address>
              <RecordID>998887</RecordID>
              <Organization>AddressDoctor GmbH</Organization>
              <Department>Sales</Department>
              <Contact>Steffen Preuschoff</Contact>
              <Building />
              <Street>Röntgenstr. 9</Street>
              <HouseNumber>9</HouseNumber>
              <POBox />
              <Locality>Maxdorf</Locality>
              <PostalCode>67133</PostalCode>
              <Province>Rheinland-Pfalz</Province>
              <Country>Germany</Country>
              <Residue />
              <CountrySpecificLocalityLine>67133 Maxdorf</CountrySpecificLocalityLine>
              <DeliveryAddressLines>Röntgenstr. 9</DeliveryAddressLines>
              <FormattedAddress>AddressDoctor GmbH
Sales
Steffen Preuschoff
Röntgenstr. 9
67133 Maxdorf

```

```
Germany</FormattedAddress>
  </Address>
  <ResultPercentage>81</ResultPercentage>
  <ElementMatchStatus>03334001</ElementMatchStatus>
  <ElementResultStatus>33336002</ElementResultStatus>
</Result>
</Results>
</ValidateResult>
</ValidateResponse>
</soap:Body>
</soap:Envelope>
```

Please see below for a detailed explanation of the elements used in the SOAP request. The HTTP connection will be closed after each request has been processed.

5 Batch Web Service – Reference

The AddressDoctor Batch Web Service was designed for processing larger quantities of addresses without any user interaction. The AddressDoctor system will attempt to confirm or correct the submitted data. If the input data is ambiguous or insufficient the service will perform a deliverability assessment. Please see section 8 for an explanation of the deliverability assessment and status codes.

The WSDL of the Batch Web Service can be found at

<http://validator2.addressdoctor.com/addBatch/Batch.asmx?WSDL>

5.1 Structure of a Batch request

Each request to the AddressDoctor Batch Web Service may contain up to 10 addresses transmitted at the same time. Bundling multiple addresses in one request increases throughput and eliminates latency times. Please note that only one Batch request (connection) can be performed for each user account at the same time. The AddressDoctor servers will enforce this and potentially refuse additional connections. Refused connections will receive the HTTP status 403 (forbidden).

A batch request looks as follows (simplified).

```
<addBatchRequest>
  <Authentication>
</Authentication>
  <CampaignID>string</CampaignID>
  <OtherServices>
</OtherServices>
  <Parameters>
</Parameters>
  <AddressCount>long</AddressCount>
  <Addresses>
    <Address>
</Address>
    <Address>
</Address>
    ...
  </Addresses>
</addBatchRequest>
```

The <addBatchRequest></addBatchRequest> tags encapsulate tags for authentication, job control and the addresses. Please note that the <Addresses></Addresses> tag can contain up to 10 <Address></Address> tags each with one postal address. It is possible, however, to send any number of addresses between 1 and 10.

The maximum size of an <addBatchRequest> is limited to

Version 2.2.X 30 KB

Version 2.3.X 50 KB

5.2 Authentication

In order to use the AddressDoctor Web Service a Customer ID and password is required. For billing purposes a customer may “create” several department records that can have different passwords. Each department is assigned a number by the AddressDoctor system. The AddressDoctor account management system will show transactions for each department. You must use 0 as a default department if no departments have been created.

```
<Authentication>
  <CustomerID>string</CustomerID>
  <DepartmentID>long</DepartmentID>
  <Password>string</Password>
</Authentication>
```

Please refer to the activation eMail you have received after signing up for an AddressDoctor account to locate your Customer ID. A Customer ID may also be provided by an AddressDoctor representative. Please note that the Customer ID may be referred to as Account ID during the registration process.

5.3 Miscellaneous tags

The request may/must contain some additional tags for billing purposes and job control

```
<CampaignID>string</CampaignID>
<AddressCount>long</AddressCount>
```

The tag <CampaignID> may be passed to the AddressDoctor system for billing purposes. By utilizing the account management features it is possible to see how many transactions were performed for a specific campaign/job. The tag is limited to 20 characters. The campaign ID will apply to all addresses contained in a request.

The <AddressCount> tag is required and has to match the number of occurrences of the <Address> tags. Values between 1 and 10 are valid.

5.4 OtherServices

Other services provide additional information about addresses. This information is not directly related to the validation of the address itself. Currently two extra services are available in the Batch and Interactive Web Service: National Change of Address (NCOA) for Germany and Geo Coding. Extra services can be requested through the OtherServices tag. It is optional to request other services and requires the account to be set up accordingly.

The OtherServices tag may include any number of Service tags, which consist of the two tags "type" and "payload". These both contain string values. The type must exactly match one of the supported services. The payload will usually just be the String "1" for the request, though this depends on the service being requested. Other service types may be available. Please contact an AddressDoctor sales representative for more information.

The order in which other services are requested is irrelevant. All requested services will be processed in a specific, predefined order. For example, if requested, Change of Address is requested first and all other services will be processed for the new address, if applicable.

Please note that several restrictions apply, which will be detailed in each service's chapter. One restriction that is common to all extra services is that the Web Service will only perform the extra service for addresses which have been validated (V) or corrected (C). In order to supply the additional information the system relies on correct addresses.

```

<OtherServices>
  <Service>
    <type>NCOA_DEU</type>
    <payload>1</payload>
  </Service>
  <Service>
    <type>GeoCoords</type>
    <payload>1</payload>
  </Service>
</OtherServices>

```

5.4.1 National Change of Address for Germany (NCOA_DEU)

The National Change of Address service will provide the new address of the contact if the person has moved.

Typically there is one provider of this information per country and you must sign a contract with each provider before you can access NCOA. An AddressDoctor sales representative will assist you in obtaining the proper contracts. Additionally, NCOA is not available for every country. Contact the AddressDoctor sales team for more information on the available countries and to obtain the contracts necessary to access a country's NCOA service.

When you request NCOA for a country for which you are not cleared contractually, the Web Service will tell you so with one of the tags in the result, and will not request an NCOA check for that address. This way you can send mixed countries in your request and the Web Service will only request the NCOA for the countries where you are allowed to request it.

Please note that only address marked as correct or corrected (ValidationStatus "V" or "C") will be sent to the NCOA provider.

5.4.2 GeoCoords

The geo coding service will provide geographical coordinates in the WGS84 system for the address. In addition to the regular address validation transactions you will need to purchase geo coding transactions if you wish to use this service. Not all countries are supported yet, but AddressDoctor will add more countries as they become available. Contact the AddressDoctor sales team for more information about available countries and to purchase geo coding transactions.

Please note that only addresses marked as correct or corrected (ValidationStatus "V" or "C") and which are street addresses (i.e. no PO boxes) will be processed by the geo coding service. In addition the Batch Web Service will process P records (P3, P2, P1) as well. Countries where no reference data is available for correction will, therefore, never be processed for geo coding.

5.5 Parameters

The <Parameters> tag controls the input and output of some of the values of the <Address> tag. As each batch request can contain up to ten <Address> tags parameters will apply to **all** addresses contained in a request.

```

<Parameters>
  <CountryOfOrigin></CountryOfOrigin>
  <StreetWithHNo></StreetWithHNo>
  <CountryType></CountryType>
  <LineSeparator></LineSeparator>
  <PreferredLanguage></PreferredLanguage>
  <Capitalization></Capitalization>
  <FormattedAddressWithOrganization></FormattedAddressWithOrganization>
  <ParsedInput></ParsedInput>
  <RemoveDiacritics></RemoveDiacritics>
  <FieldDimensions>
    <Organization>

```

```

    <MaxLines></MaxLines>
    <MaxLineLength></MaxLineLength>
</Organization>
<Contact>
    <MaxLines></MaxLines>
    <MaxLineLength></MaxLineLength>
</Contact>
<Building>
    <MaxLines></MaxLines>
    <MaxLineLength></MaxLineLength>
</Building>
<Street>
    <MaxLines></MaxLines>
    <MaxLineLength></MaxLineLength>
</Street>
<POBox>
    <MaxLineLength></MaxLineLength>
</POBox>
<Locality>
    <MaxLines></MaxLines>
    <MaxLineLength></MaxLineLength>
</Locality>
<PostalCode>
    <MaxLineLength></MaxLineLength>
</PostalCode>
<Province>
    <MaxLines></MaxLines>
    <MaxLineLength></MaxLineLength>
</Province>
<Country>
    <MaxLineLength></MaxLineLength>
</Country>
<CountrySpecificLocalityLine>
    <MaxLineLength></MaxLineLength>
</CountrySpecificLocalityLine>
<DeliveryAddressLines>
    <MaxLines></MaxLines>
    <MaxLineLength></MaxLineLength>
</DeliveryAddressLines>
<FormattedAddress>
    <MaxLines></MaxLines>
    <MaxLineLength></MaxLineLength>
</FormattedAddress>
</FieldDimensions>
</Parameters>

```

5.5.1 CountryOfOrigin

The <CountryOfOrigin> tag controls the **output** of the <FormattedAddress>. Whenever mail or other items are sent cross border it is necessary to place the country of destination in the address. As an example a mail piece or shipment going from Germany to the USA should be addresses as follows

ABC Company, Inc.
 1234 Any Street
 ANYTOWN AT 12345
 USA

The same shipment can be addressed as

ABC Company, Inc.
 1234 Any Street
 ANYTOWN AT 12345

when mailed within the United States. It is recommended that you set the CountryOfOrigin to the country from which the shipment will originate. Allowed values are

Parameter value	Explanation
COO_ALWAYS_USE_DESTINATION_COUNTRY (default)	The country of destination will always be used in the FormattedAddress
COO_NEVER_USE_DESTINATION_COUNTRY	The country of destination will never be used in the FormattedAddress
COO_ABW	Country of origin is Aruba
...	
COO_XXX	Country of origin is XXX where XXX is equal to the ISO3 country code.

5.5.2 StreetWithHNo

This parameter can be set to true or false. Please note that these parameters are case sensitive.

Parameter value	Explanation
True	Output the Street field including a House or Building number. For example Hauptstr. 9 Or 128 Main Street
false (default)	Output the Street field without a house or building number. Only the name is returned as part of the Street field. For example Hauptstr. Or Main Street Please note that there is a separate field that returns the house or building number regardless of the value of this parameter

5.5.3 CountryType

The CountryType parameter controls the way the country name is **returned** in the FormattedAddress and the Country property. Allowed values are

Parameter value	Explanation
ISO_2	Return the country as the ISO2 country code
ISO_3	Return the country as the ISO3 country code
ISO_NUMBER	Return the country as the numeric ISO code
ABBREVIATION	Return the country abbreviation as often used in car license plates
NAME_CN	Return the name in Chinese
NAME_DA	Return the name in Danish

Parameter value	Explanation
NAME_DE	Return the name in German
NAME_EN	Return the name in English
NAME_ES	Return the name in Spanish
NAME_FI	Return the name in Finnish
NAME_FR	Return the name in French
NAME_GR	Return the name in Greek
NAME_HU	Return the name in Hungarian
NAME_IT	Return the name in Italian
NAME_JP	Return the name in Japanese
NAME_KR	Return the name in Korean
NAME_NL	Return the name in Dutch
NAME_PL	Return the name in Polish
NAME_PT	Return the name in Portuguese
NAME_RU	Return the name in Russian
NAME_SA	Return the name in Arabic
NAME_SE	Return the name in Swedish

5.5.4 LineSeparator

When passing data to the FormattedAddress property and when retrieving it the line separation character can be set. There are also other properties such as DeliveryAddressLines that can contain multiple lines. The line separator is always used in these cases.

Parameter value	Explanation
LST_LF	Use a linefeed as the line separation character. Also use this value when you would prefer a carriage return linefeed. The carriage return linefeed is not supported for technical reasons.
LST_SEMICOLON	Use a semicolon as the line separation character
LST_COMMA	Use a comma as the line separation character
LST_TAB	Use a TAB character as the line separation character
LST_NO_SEPARATOR	Use no special line separation character. Fields will only be separated by a space.

5.5.5 PreferredLanguage

The PreferredLanguage parameter controls the overall language of an address. In general it is possible to return an address in Latin script (romanized) or in its original format.

Parameter value	Explanation
PFL_DATABASE	This will output the address according to the data in the AddressDoctor reference database.
PFL_LATIN_SCRIPT (default)	Can be used to request a Latin script output. Submitting an address in Greek will return the same address romanized. Does not work for Addresses in Status Nx
PFL_PREFERRED_BY_POSTAL_ADMIN	Will return the Address in the format preferred by the postal administration. This is especially helpful in countries with more than one language. In Switzerland for instance, the name Genf and Genève are both valid. AddressDoctor will return the spelling preferred by the postal administration
PFL_PRIMARY	Uses the primary language for countries with more than one alphabet. For Japan it will return the addresses in Kanji.
PFL_SECONDARY	Uses the secondary language for countries with more than one alphabet. For Japan it will return the addresses in Kana.
PFL_LANG_EN	Will always return the English spelling of a place name (if available) as well as of the province name (if available and applicable). As an example it will return MUNICH instead of München as well as BAVARIA instead of Bayern for the province. Please note: All output will be in upper case unless mixed casing is explicitly requested.

5.5.6 Capitalization

The Capitalization parameter controls the casing used for an address.

Parameter value	Explanation
NO_CHANGE (default)	Outputs all postal data as stored in the reference database and does not change the casing of fields that do not originate from the reference database.
UPPER_CASE	Outputs all data in upper case.
LOWER_CASE	Outputs all data in lower case.

Parameter value	Explanation
MIXED_CASE	Outputs all data in mixed case. Please note that not all data providers make their data available in mixed case. In these situations, AddressDoctor uses its own casing algorithm that will attempt to proper case all address elements. If, however a data provider does not supply extended characters such as accents in France, the mixed case output will not contain these accents as well.

5.5.7 FormattedAddressWithOrganization

Whenever data is passed to the AddressDoctor Web Service it is parsed using the advanced parsing features. Data whether it is fielded correctly, incorrectly or even if it does not have any structure will be separated into individual fields. When formatted address lines are output they will contain the Organization and Contact elements by default. This tag allows to suppress this information in the <FormattedAddress>.

Parameter value	Explanation
true (default)	Use company and contact elements in the <FormattedAddress> tag
false	Suppress the output of company and contact elements.

5.5.8 ParsedInput

In most cases the AddressDoctor Web Service can provide an address correction or even confirm the input data. In some cases this is not possible and no output is generated. In these situations it is possible to retrieve the parsed input data.

Parameter value	Explanation
NEVER (default)	Will never return parsed data. If there are no corrections the <Address> tag will be empty. Please note that for addresses with status P3 an address may or may not be returned.
ONLY_FOR_P	The parsed data is only returned when no suggestions or corrections can be generated. You can distinguish a parsing output from a suggestion by checking the values of the <ElementResultStatus> which will be all zeroes (0) if a parsing result is returned.
ONLY_PARSED	Will not validate the addresses, but only parse the elements.

5.5.9 RemoveDiacritics

This parameter allows eliminating diacritical characters such as ä, é or ß and replacing them with their proper equivalent. Only works on Latin alphabets.

Parameter value	Explanation
True	Eliminate diacritics

Parameter value	Explanation
false (default)	Leave diacritics unchanged.

5.5.10 FieldDimensions

Typically the AddressDoctor Web Service will return data as it is stored in its reference databases or as parsed from the input fields. Data fields that have been generated as the result of a parsing operation will have the length of the parsed input. In some cases it might be necessary to limit the length of certain fields. The <FieldDimensions> tag allows controlling the maximum number of lines that are returned for a certain element as well as the length of each line.

If no value is supplied, each field will contain the number of lines necessary with a potentially unlimited length per line. Setting MaxLineLength to zero or any other value smaller than the minimum length allowed will be treated as unlimited length as well. The maximum value allowed is 10 for MaxLines and 99 for MaxLineLength. Supplying a greater value will have the same effect as setting the values to the maximum allowed. Please see the list below for minimum values for MaxLineLength per field:

Organization:	20
Contact:	20
Building:	20
Street:	20
POBox:	20
Locality:	15
PostalCode:	5
Province:	2
Country:	50
CountrySpecificLocalityLine:	20
DeliveryAddressLine:	20
FormattedAddress:	20

Please note that depending on the data and the country selected return values might still exceed the selected minimum value. This happens if there is no useful way to abbreviate the values further. An example would be to abbreviate the following postal code from United Kingdom "AB123AD" to 5 characters. The return value will still be contains 7 digits: "AB123AD"

5.6 Address data

The <Address></Address> tags comprise the description of one postal address to be verified. All elements are strings. Some of them may be omitted. The presence of some elements overrides the presence of others.

```
<Address>
  <RecordID>string</RecordID>
  <Organization>string</Organization>
  <Department>string</Department>
  <Contact>string</Contact>
  <Building>string</Building>
  <Street>string</Street>
  <HouseNumber>string</HouseNumber>
  <POBox>string</POBox>
  <Locality>string</Locality>
  <PostalCode>string</PostalCode>
  <Province>string</Province>
  <Country>string</Country>
  <Residue>string</Residue>
  <CountrySpecificLocalityLine>string</CountrySpecificLocalityLine>
  <DeliveryAddressLines>string</DeliveryAddressLines>
  <FormattedAddress>string</FormattedAddress>
</Address>
```

5.6.1 RecordID

The RecordID is a string than can be passed in to identify the record. The same string will be returned in the output.

5.6.2 Organization

Organization is the name of a company or organization. It is often not passed in when using individual fields but is returned when passing in a FormattedAddress

5.6.3 Department

Name or description of the department.

5.6.4 Contact

Contact information or name of the person. Often not used when passing in data but when retrieving data elements that have been passed in as part of a <FormattedAddress> tag.

5.6.5 Building

This element contains the building and / or subbuilding name.

5.6.6 Street

This element contains the street name. If the parameter StreetWithHNo is used the street field is also expected to contain the house or building number.

5.6.7 HouseNumber

Contains the house or building number.

5.6.8 POBox

Contains the post office box.

5.6.9 Locality

Contains the locality or city name.

5.6.10 PostalCode

Contains the postal code. When an address is returned the postal code will be properly formatted for this country.

5.6.11 Province

This field contains the province name. Please note that a province name is returned whenever AddressDoctor has data available. This does not indicate that the province is required as part of an address.

5.6.12 Country

This field contains the name of the country

5.6.13 Residue

This field should not be used when inputting data to the system. However, when a parsing operation has identified that an address contains superfluous elements, they will be returned in the Residue field.

5.6.14 CountrySpecificLocalityLine

Many address databases contain several data elements in one field. Examples would be

12345 Berlin

Or

MOUNT VERNON WA 98273

AddressDoctor considers these fields to be "country specific locality lines". Using this field they can be set or retrieved

5.6.15 DeliveryAddressLines

The DeliveryAddressLines are similar to the CountrySpecificLocality line, but they do not contain locality or postal code elements. An Example of DeliveryAddressLines would be

```
172 MAIN ST  
8TH FLOOR  
SUITE 15
```

5.6.16 FormattedAddress

The FormattedAddress element is very useful both for passing data to AddressDoctor as well as when retrieving results. When data is sent to AddressDoctor the system expects each line to be separated with a line separation character. Processing of totally unstructured data is not possible. Please note, however, that either a FormattedAddress can be passed to an AddressDoctor Web Service or structured address elements.

5.7 Structure of a batch <ValidateResponse>

After a batch request has been processed, the same number of addresses will be returned as part of the validate result that have been submitted. The elements will appear in the same order as they were submitted. Also the <RecordID> of each address can be used to identify the addresses

```
<ValidateResult>  
  <ErrorCode>long</ErrorCode>  
  <ErrorMessage>string</ErrorMessage>  
  <CampaignID>string</CampaignID>  
  <ResultCount>string</ResultCount>  
  <Results>  
    <Result>  
    </Result>  
    <Result>  
    </Result>  
    ...  
  </Results>  
</ValidateResult>
```

The Validate Result may contain an ErrorCode and an ErrorMessage. Please see section 9 for an explanation of the error messages.

The CampaignID is returned unchanged.

5.7.1 Description of a Result

A response to a batch request will contain the same number of addresses that were passed to the system. For each result (address) status values indicate the result of the validation process.

```
<Result>  
  <Address></Address>  
  <ValidationStatus>long</ValidationStatus>  
  <ResultPercentage>long</ResultPercentage>  
  <ElementMatchStatus>string</ElementMatchStatus>  
  <ElementResultStatus>string</ElementResultStatus>  
  <OtherServices></OtherServices>  
</Result>
```

Please see section 8 for an explanation of the different status codes.

5.7.2 OtherServices Results

If the request contained an <OtherServices> tag, the response will include one for every <Result>, which will in turn include one <Service> tag for each one present in the input.

```
<OtherServices>  
  <Service>  
    <type>string</type>  
    <payload>string</payload>
```

```
</Service>  
...  
</OtherServices>
```

The type will be exactly the same as the ones in the request. This will also include all unknown or invalid types the request may contain.

Please see section 8.4 for an explanation of the payloads for the various services.

6 Interactive Web Service – Reference

The AddressDoctor Interactive Web Service was designed for processing one address at a time in an interactive fashion. This service is often used when validating data on the Internet. An address is submitted and either confirmed to be correct or deemed incorrect. If AddressDoctor is able to generate suggestions, they will be returned as part of the response of the system.

The Interactive Web Service will process **only one** address in each request. It is, however, possible, to submit an unlimited number of requests in parallel for each user account. Unlike for the Batch Web Service the system will not block multiple requests. Interactive requests have a higher system priority and will return faster than batch requests.

The WSDL of the Interactive Web Service can be found at

<http://validator2.addressdoctor.com/addinteractive/interactive.asmx?WSDL>

6.1 Structure of an Interactive request

Each request to the AddressDoctor Interactive Web Service can contain one address. This address will either be corrected or suggestions/alternatives will be returned to the requesting system. Up to twenty (20) suggestions might be returned for each ambiguous address. Along with the suggestions the AddressDoctor system will return a JobToken that allows resubmitting an address without incurring an extra charge.

An interactive request looks as follows (simplified).

```
<addInteractiveRequest>
  <Authentication>
</Authentication>
  <CampaignID>string</CampaignID>
  <JobToken>string</JobToken>
  <OtherServices>
</OtherServices>
  <Parameters>
</Parameters>
  <Address>
</Address>
</addInteractiveRequest>
```

The <addInteractiveRequest></addInteractiveRequest> tags encapsulate tags for authentication, job control and the actual address.

The maximum size of an <addInteractiveRequest> is limited to

Version 2.2.X 3 KB

Version 2.3.X 5 KB

6.2 Authentication

Please see section 5.2 for an explanation of the authentication elements.

6.3 Miscellaneous Tags

The request may/must contain additional tags for billing purposes and job control

```
<CampaignID>string</CampaignID>
<JobToken>string</JobToken>
```

The tag <CampaignID> may be passed to the AddressDoctor system for billing purposes. By utilizing the account management features it is possible to see how many transactions were performed for a specific campaign/job. The tag is limited to 20 characters.

The <JobToken> tag can be used to revalidate an address if it has already been passed to the system and lead to suggestions. By using the same JobToken the address can be processed again without charge. An example of this might be an address that has been passed to AddressDoctor without a building number. The Web Service will return up to 20 suggestions for this address. However, the valid building number range might not be part of the suggestions. For this reason it would be helpful to revalidate the address once the missing building number has been entered by the user.

6.4 OtherServices

Please refer to section 5.4 for an explanation of the available other services.

6.5 Parameters

Please see section 5.5 for an explanation of the possible parameter settings. The parameter <ParsedInput> is not supported by the Interactive Web Service.

6.6 Address data

Please see section 5.6 for an explanation of the <Address> element structure.

6.7 Structure of an Interactive ValidateResponse

After an Interactive Web Service request has been processed, the address is returned along with some status flags and potentially a number of suggestions if the address could not be confirmed. Up to 20 <Result> elements can be contained in the <Results> tag.

```
<ValidateResult>
  <ErrorCode>long</ErrorCode>
  <ErrorMessage>string</ErrorMessage>
  <CampaignID>string</CampaignID>
  <JobToken>string</JobToken>
  <ValidationStatus>string</ValidationStatus>
  <OtherServices></OtherServices>
  <ResultCount>long</ResultCount>
  <Results>
    <Result>
    </Result>
    ...
    <Result>
    </Result>
  </Results>
</ValidateResult>
```

The number of <Result> elements can be found in the <ResultCount> tag.

The <OtherServices> tag is only present if there was one in the request and it will contain one <Service> tag for each one in the request.

6.7.1 Description of a Result

A response to an interactive request may contain zero (0) to twenty (20) results (suggestions). If the address was correct or could be corrected automatically, only one result will be returned. If, however, the data was ambiguous, the system will generate suggestions. Each suggestion will be contained in one <Result> element. Each suggestion has its own ResultPercentage and several status flags.

```
<Result>
  <Address></Address>
  <ValidationStatus>long</ValidationStatus>
  <ResultPercentage>long</ResultPercentage>
```

```
<ElementMatchStatus>string</ElementMatchStatus>
<ElementResultStatus>string</ElementResultStatus>
</Result>
```

Please see section 8 for an explanation of the different status codes.

6.7.2 OtherServices results

If the request contained the <OtherServices> tag, the response will include one as part of the <ValidateResult> tag. This will contain one <Service> tag for each one present in the request.

```
<OtherServices>
  <Service>
    <type>string</type>
    <payload>string</payload>
  </Service>
  ...
</OtherServices>
```

The type will be exactly the same as the ones in the request. This will also include all unknown or invalid types the request may contain.

Please see section 8.4 for an explanation of the status codes.

7 FastCompletion Web Service – Reference

The AddressDoctor FastCompletion Web Service was designed to assist users in Call Center environments in the data entry process. By entering truncated data elements, a user can save significant key strokes when entering address data. AddressDoctor will either automatically complete the address data provided or generate a list of up to 20 suggestions.

The WSDL of the FastCompletion Web Service can be found at

<http://validator2.addressdoctor.com/addFastCompletion/FastCompletion.asmx?WSDL>

7.1 Structure of a FastCompletion request

A FastCompletion request is similar to the requests used in the other AddressDoctor Web Services. As the data entered by a user in a call center is typically limited, the <Address> tag elements are different.

```
<addFastCompletionRequest>
  <Authentication>
</Authentication>
  <CampaignID>string</CampaignID>
  <JobToken>string</JobToken>
  <Parameters>
</Parameters>
  <Address>
</Address>
</addFastCompletionRequest>
```

The maximum size of an <addFastCompletionRequest> is limited to

Version 2.2.X 3 KB

Version 2.3.X 5 KB

7.2 Authentication

Please see section 5.2 for an explanation of the authentication elements. They are identical to the other Web Services

7.3 Miscellaneous Tags

The request may/must contain additional tags for billing purposes and job control

```
<CampaignID>string</CampaignID>
<JobToken>string</JobToken>
```

The tag <CampaignID> may be passed to the AddressDoctor system for billing purposes. By utilizing the account management features it is possible to see how many transactions were performed for a specific campaign/job. The tag is limited to 20 characters.

The <JobToken> tag can be used to revalidate an address if it has already been passed to the system and lead to suggestions or was incomplete. By using the same JobToken the address can be processed again without charge. The Web Service will return up to 20 suggestions for this address.

7.4 OtherServices

Currently no other services are available in the FastCompletion Web Service.

7.5 Parameters

Please see section 5.5 for an explanation of the possible parameter settings. The parameter <ParsedInput> is not supported by the FastCompletion Web Service

7.6 Address data

The address data submitted for a FastCompletion Web Service request is different from the address data submitted to the other two Web Services. This is mainly due to the fact that several data elements would not be used in the typical Call Center environment. Also it is not possible to submit unstructured data to the FastCompletion Web Service.

```
<Address>
  <RecordID>string</RecordID>
  <Street>string</Street>
  <HouseNumber>string</HouseNumber>
  <POBox>string</POBox>
  <Locality>string</Locality>
  <PostalCode>string</PostalCode>
  <Province>string</Province>
  <Country>string</Country>
</Address>
```

7.7 Structure of a FastCompletion ValidateResponse

After a FastCompletion Web Service request has been processed, the address is returned along with some status flags and potentially a number of suggestions if the address could not be completed automatically. Up to 20 <Result> elements can be contained in the <Results> tag.

```
<ValidateResult>
  <ErrorCode>long</ErrorCode>
  <ErrorMessage>string</ErrorMessage>
  <CampaignID>string</CampaignID>
  <JobToken>string</JobToken>
  <ValidationStatus>string</ValidationStatus>
  <ExtraServices></ExtraServices>
  <ResultCount>long</ResultCount>
  <Results>
    <Result>
    </Result>
    ...
    <Result>
    </Result>
  </Results>
</ValidateResult>
```

The number of <Result> elements can be found in the <ResultCount> tag.

7.7.1 Description of a Result

A response to a FastCompletion request may contain zero (0) to twenty (20) results (suggestions). Each suggestion will be contained in one <Result> element. Each suggestion has its own ResultPercentage and several status flags.

```
<Result>
  <Address></Address>
  <ValidationStatus>long</ValidationStatus>
  <ResultPercentage>long</ResultPercentage>
  <ElementMatchStatus>string</ElementMatchStatus>
  <ElementResultStatus>string</ElementResultStatus>
</Result>
```

Please see section 8 for an explanation of the different status codes. It is important to note that the structure of an <Address> in the Result is different from the structure of an <Address> in a request. This is true only for the FastCompletion Web Service.

8 Status codes

An address verification operation invoked through the Web Service can have many different outcomes. The input address may have been correct or the AddressDoctor Web Service can have corrected it. It is also possible that no correction can be done because of insufficient information on input or in the reference data. If the information provided is not sufficient to make a correction, AddressDoctor will make an attempt to guess the chances of deliverability of this address. The software uses unique algorithms to make this assessment.

The following status codes are available in a ValidationResponse to interpret the outcome of a call to the AddressDoctor Web Service.

- ValidationStatus
- ElementMatchStatus
- ElementResultStatus
- ResultCount

The ValidationStatus property contains an overall status value of the validation operation. The ValidationStatus describes whether the input address was correct, could be corrected or how likely it is to be deliverable if the correction could not be performed. The ElementMatchStatus and ElementResultStatus explain the matching and correction of each address element.

8.1 ValidationStatus codes

- V: Data correct on input (validated)
- C: Data corrected by AddressDoctor
- P3: Data cannot be corrected, but is very likely to be deliverable
- P2: Data cannot be corrected, but fair chance that the address is deliverable
- P1: Data cannot be corrected and unlikely to be deliverable
- N6: No validation performed because the transactions for this country are exhausted
- N4: Only parsing was requested and performed for this address
- N3: No validation performed because country is not available for this user account
- N2: No validation performed because reference data for this country is not available
- N1: No validation performed because country not recognized

Additional status values are used when working with the FastCompletion Web Service. These are

- N5: Insufficient information to generate suggestions
- Q3: Suggestions are available
- Q2: Suggested address is not complete (enter more information)
- Q1: No suggestions are available

Here is a more detailed explanation of the ValidationStatus codes

- V:** The data that was submitted to the Web Service was correct on input. The address returned in the ValidationResponse is more or less identical to the input data. However some changes are possible. Among these changes are simple standardizations and synonym replacements.
- C:** The data that was submitted to the Web Service has been corrected. The address was either incomplete or contained spelling mistakes. The address returned by the Web Service can now be used instead of the original input data.
- P3:** The data submitted to the Web Service may or may not be correct. However AddressDoctor could not come to an automatic decision because of ambiguities or incomplete data on input or in the reference file. Because of this a deliverability assessment has been made. If you use the **original input data** it is

very likely that the address can be delivered to. The Batch Web Service will return the parsed input data as the result. The Interactive Web Service may return suggestions that are possible corrections. However, these suggestions should be selected manually.

P2: The data submitted to the Web Service may or may not be correct. However AddressDoctor could not come to an automatic decision because of ambiguities or incomplete data on input or in the reference file. Because of this a deliverability assessment has been made. If you use the **original input data** it is fairly likely that the address can be delivered to. The Batch Web Service will return the parsed input data as the result. The Interactive Web Service may return suggestions that are possible corrections. However, these suggestions should be selected manually.

P1: The data submitted to the Web Service may or may not be correct. However AddressDoctor could not come to an automatic decision because of ambiguities or incomplete data on input or in the reference file. Because of this a deliverability assessment has been made. If you use the **original input data** it is not very likely that the address can be delivered to. The Batch Web Service will return the parsed input data as the result. The Interactive Web Service may return suggestions that are possible corrections. However, these suggestions should be selected manually.

8.2 ElementMatchStatus codes

All results share an ElementMatchStatus that describes how the address elements matched to the postal reference data.

Each address element has its own match status. The fact that individual components such as postal code and the locality could be matched to the reference data does not indicate that both elements are correct in combination.

The ElementMatchStatus is a string with eight characters. An example would be 42424000.

The positions within the string represent the following address components

- 1 = Postal code
- 2 = Locality
- 3 = Province
- 4 = Street
- 5 = Building number
- 6 = PO Box
- 7 = Building
- 8 = Organization

The values for each position can be

- 0: empty
- 1: not found
- 2: not checked (no reference data or no chance of success)
- 3: matched with errors
- 4: matched without errors

If no data is available for a specific country and only a parsing operation has taken place the following values are possible

- 0: empty
- 1: element present

8.3 ElementResultStatus codes

The ElementResultStatus indicates for each address component if and how it has been modified during the validation process. Each result or suggestion has its own value.

The ElementResultStatus is a string of length eight. An example looks as follows 53066000.

The positions within the string are the same as for the ElementMatchStatus. The values, however, are different.

- 0: empty
- 1: not checked
- 2: not checked but standardized
- 3: checked and corrected (changed or inserted)
- 4: validated, but changed (synonyms, old names)
- 5: validated, but standardized
- 6: validated and unchanged
- 7: no value given because of multiple matches

8.4 OtherServices payloads

Each <Service> tag in <OtherServices> will provide a string in its <payload> tag that contains the complete status and result information. This will be an XML fragment which is encoded for transport. If you use a SOAP Toolkit or an XML Reader, they will automatically decode the payload. Otherwise you must do this yourself. If your programming environment provides a HttpDecode function, this will work. Generally speaking, all characters that have a special meaning in XML will be encoded to their entity representations.

&	becomes	&
<	becomes	<
>	becomes	>

After the payload has been decoded, it will contain an XML fragment containing at least a <Status> tag and a <Message> tag. This <Status> tag will indicate the status of the operation while the <Message> tag may include a human readable message in English. It may, however be empty.

If the <Status> indicates success – the exact value for success depends on the service type – there may be other, service-dependent tags that provide the result values.

8.4.1 General status codes

The following <Status> values are possible for each type of other service. Each of these indicates that no results are available.

INVALID_SERVICE_TYPE	The <type> tag does not match any valid service type. Note that service types are case sensitive.
SERVICE_DOWN	The external service was temporarily not available. No results are available, but the request may yield a result if tried again later.
NOT_AVAILABLE	The service is not available for this country.
NOT_ALLOWED	You do not have permission to request this type of service. Usually this is the case when some special contract needs to be signed before access to the service is allowed. Contact an AddressDoctor sales representative.
NO_MORE_TRANSACTIONS	You have no more transactions of this service's type available.
NOT_REQUESTED	The address does not meet the requirements to be forwarded to the external service provider. Usually only correct or corrected addresses are forwarded to external providers. For some services types PO Box addresses are also not forwarded (e.g. for GeoCoding). The <Message> tag will provide a reason.

NO_MATCH The address was forwarded to the external service provider, but the provider could not match the address.

8.4.2 NCOA_DEU status codes

NO_MATCH No information was available for the address. This usually means that the person has not moved.
MOVED The person has moved and the new address has been provided. No other result tags are available.

8.4.3 GeoCoords status codes

NO_MATCH The AddressDoctor Web Service has requested geo coding, but the service could not provide coordinates for this address. The <Message> tag may contain a reason for the failure to provide coordinates.
GEO_COORDS_PROVIDED Coordinates have been provided for the address. These are available in the <Latitude> and <Longitude> tags. Please refer to the <Accuracy> tag for information on how accurate the coordinates are.

The <Accuracy> tag will contain string values that describe the accuracy of the provided coordinates. The following values may be provided:

ADD The coordinates are accurate to the address.
STR The coordinates are interpolated to the middle of the street.
ZIP The coordinates are interpolated to the middle of the postal code area
ZIPC The coordinates are interpolated to the middle of the postal code and locality area.
ZIPCD The coordinates are interpolated to the middle of the postal code, locality and district area.

9 Sample addresses for testing

The following addresses are provided for you to test your implementation of the Web Service. For each address the status code values are provided and explained. If not otherwise mentioned the addresses have been submitted to the InteractiveWeb Service (see chapter 6) as unstructured address data as part of the <FormattedAddress> tag with the line separator set to <LineSeparator>LST_SEMICOLON</LineSeparator>.

9.1 Addresses with StatusCode V

Addresses for which the processing results in a StatusCode of V were correct on input. Depending on other parameters some minor standardizations may take place.

9.1.1 Correct address

The following input address is entirely correct, the postal code is properly spaced and the address also is in the proper capitalization for a Swedish address. Because of this, no standardization will have to take place.

```
<FormattedAddress> VASAGATAN 22; 111 20 STOCKHOLM;SVERIGE</FormattedAddress>
```

ElementMatchStatus will be 44044000

With the PreferredLanguage set to Latin_Script

(<PreferredLanguage>PFL_LATIN_SCRIPT</PreferredLanguage>) the result will be:

```
<Street>VASAGATAN</Street>  
<HouseNumber>22</HouseNumber>  
<POBox />  
<Locality>STOCKHOLM</Locality>  
<PostalCode>111 20</PostalCode>  
<Province>STOCKHOLMS LÄN</Province>
```

```
<Country>Sweden</Country>
```

The ElementResultStatus is 66366000

9.1.2 Address with synonym replaced

The following input address is written with the English synonym for München (Munich). Because this is a correct name for the city the overall status value will be V.

```
<FormattedAddress>Theatinerstr. 12;80333 Munich;Germany</FormattedAddress>
```

ElementMatchStatus will be 44044000

With the PreferredLanguage set to English (<PreferredLanguage>PFL_LANG_EN</PreferredLanguage>) the result will be.

```
<Street>THEATINERSTR.</Street>
<HouseNumber>12</HouseNumber>
<POBox />
<Locality>MUNICH</Locality>
<PostalCode>80333</PostalCode>
<Province>BAVARIA</Province>
<Country>Germany</Country>
```

The ElementResultStatus is 64356000

With the PreferredLanguage set to Latin_Script (<PreferredLanguage>PFL_LATIN_SCRIPT</PreferredLanguage>) the result will be:

```
<Street>Theatinerstr.</Street>
<HouseNumber>12</HouseNumber>
<POBox />
<Locality>München</Locality>
<PostalCode>80333</PostalCode>
<Province>Bayern</Province>
<Country>Germany</Country>
```

The ElementResultStatus is 64366000

9.2 Addresses with StatusCode C

Addresses that AddressDoctor can automatically correct will result in a StatusCode of C. This indicates that either some address components were missing or incorrect. The returned address can be used instead of the original input address.

9.2.1 Address with missing postal code

The following input address is basically correct, but it is missing the postal code. The AddressDoctor Web Service will automatically append the correct postal code and return a status value of C for the address.

```
<FormattedAddress> 2827 yonge street; toronto on; Canada</FormattedAddress>
```

ElementMatchStatus will be 04444000

With the PreferredLanguage set to Latin_Script (<PreferredLanguage>PFL_LATIN_SCRIPT</PreferredLanguage>) the result will be:

```

<Street>YONGE ST</Street>
<HouseNumber>2827</HouseNumber>
<POBox />
<Locality>TORONTO</Locality>
<PostalCode>M4N 2J4</PostalCode>
<Province>ON</Province>
<Country>Canada</Country>

```

The ElementResultStatus is 35556000

9.2.2 Address with misspellings in street and city name

The following input address is basically correct, but has misspellings in the street and city name. The AddressDoctor engine will automatically correct these misspellings and return a value of C for the address.

```

<FormattedAddress>100 GOULD ST;Neu York NY 10038;Estados Unidos</FormattedAddress>

```

ElementMatchStatus will be 43434000

With the PreferredLanguage set to Latin_Script

(`<PreferredLanguage>PFL_LATIN_SCRIPT</PreferredLanguage>`) the result will be:

```

<Street>GOLD ST</Street>
<HouseNumber>100</HouseNumber>
<POBox />
<Locality>NEW YORK</Locality>
<PostalCode>10038</PostalCode>
<Province>NY</Province>
<Country>UNITED STATES</Country>

```

The ElementResultStatus is 63636000

10 Error codes and messages

The AddressDoctor Web Services may return the following Error codes in a ValidationResponse.

Error number	Error Text	Explanation
0		No Error
101	Authentication failed	The user credential passed in the <code><Authentication></code> tag are invalid
102	Insufficient transactions	All previously purchased transactions are used up. Please purchase additional transactions. This applies to Batch, FastCompletion and Interactive transactions when used through the corresponding service. Insufficient Transactions for OtherServices are returned as part of their status code outlined in section 8.4.1
103	Transactions expired	This error code is not yet available. It is going to be returned when the usage time of your transactions has expired. Currently the error code 101 will be returned in this case.

Error number	Error Text	Explanation
200	<add#Request> tag must be supplied	# may be Batch/Interactive/FastCompletion. This tag is required for a validation.
201	<Parameters> tag must be supplied	
202	<AddressCount> must be greater than zero	Used in Batch only. Please submit at least one address
203	<AddressCount> must be less than 11	Only 10 or less addresses can be submitted in one batch request.
204	<Addresses> tag must be supplied	Used in Batch only
205	<AddressCount> does not match number of supplied addresses	Used in Batch only
206	<Address> tag must be supplied	Not used in Batch
299	Request size exceeds maximum allowance of #KB	See section 4.6 for an explanation of the maximum request size
300	Operation timed out	Please contact AddressDoctor for assistance
399	Server Error	Please contact AddressDoctor for assistance

11 Sample code

Sample code is available for download on the AddressDoctor web site at http://www.addressdoctor.com/en/support/web_doku.asp

12 Version History

This chapter details changes from one interface version to the next.

12.1 Changes 2.2.X to 2.3.X

- Removed the <ExtraServices> tag and its child tags from the requests and responses.
- Added the <OtherServices> tag instead of the <ExtraServices> tag.
This change will make future introductions of other services possible without changing the WSDL again. Instead of an enumeration, which explicitly lists all available services, the <OtherServices> tag can hold any number of <Service> tags, which will define the type of service requested with a generic String.
- The Web Services may from now on return Error Number 103 if the user's transactions have expired. Implementers should expect this status code and react accordingly.
- The Web Services will now return a Validation Status value of N3, if the user's transactions packages do not include this country. In this case the user's input will be returned unchanged. The Batch Web Service will process a mixed request, where some addresses are from allowed countries and others are not. Allowed addresses will be validated, others will return the input and N3.
- The Web Services will now return a Validation Status value of N6, if the user's transactions for the country are exhausted. The behavior will be identical to the N3 case.
- Added the <Reserved> tag to the Request and Response.
Integrators should ignore this tag and not provide it in their requests. It is intended for AddressDoctor's internal use only.

13 Company Information

AddressDoctor is the leading product for international online verification of postal addresses. It is developed and maintained by AddressDoctor GmbH, a German company.

Further information is available online at <http://www.AddressDoctor.com>

AddressDoctor® and AddressDoctor products are registered trademarks of AddressDoctor GmbH or its suppliers. All other names are trademarks or registered trademarks of the respective companies.

EN046 – 140708 – 2008/736