



# DOTS Email Validation

Developer's Guide

Version 1.0.0  
March 16, 2006  
Glenn Wilson

# Table of Contents

Introduction	3
Web Service Structure	3
Operation Definitions	4
ValidateEmail	4
ValidateEmailNoSMTP	6
ValidateEmailFullSMTP	7
ValidateEmailFastSMTP	9
Error Codes	10
Integration	11
Conclusion	13

## Introduction

DOTS Email Validation (EV) is a publicly available XML web service that provides validity and metadata information about an email address. The service provides common data elements such as syntax validity along with more refined data such as SMTP failures and vulgarity flags.

EV can provide instant data verification to websites or enhancement to contact lists. However, the output from EV must be considered carefully before the validity or invalidity of an email is decided.

## Web Service Structure

Web services are methods that integrate with other applications via the web, and encapsulate tricky business logic. Web services are too large of a topic to cover in this document, but ServiceObjects has developed its web services to be as easy to integrate and as accessible as possible.

EV is a public XML web service that supports SOAP, POST and GET operations. Note that SOAP is done via POST, only with special XML markup in the post-body.

The host path, or physical location of the web service is here:

<http://ws.serviceobjects.com/ev/EmailValidate.asmx>

The location of the WSDL, or Web Service Definition Language document, is here:

<http://ws.serviceobjects.com/ev/EmailValidate.asmx?WSDL>

(This is also accessible via the “Service Definition” link.)

This XML is the definition of the web service, meaning its inputs, outputs, operations, and the like. Most likely, you will have another tool read this WSDL and make the operations available to you in your application. Whenever your utilities or IDE asks for a WSDL path to EV, you can provide this one.

Every web service has *operations* that it offers to subscribers – methods that do different work and return different output. Examining the link above, you will notice several of these operations available, which are described in detail later on.

**ValidateEmail** – provides several flags on the validity of an email address, such as the syntax validity, DNS validity and SMTP validity.

**ValidateEmailNoSMTP** – like ValidateEmail, only the SMTP check is not performed

**ValidateEmailFastSMTP** – like ValidateEmail, only the outputs have been standardized to a universal “true”/“false” and the SMTP check has been streamlined and cached.

**ValidateEmailFullSMTP** – like ValidateEmailFastSMTP, except the SMTP check does the full mailbox query.

Each of these operations will be described in detail later in this document.

## Operation Definitions

This document defines the input, output and behavior of the web service operations in EV. Each operation has its own unique behavior and output, although some of the operations are very similar.

One crucial element of email validation in general is the behavior of email servers, known as SMTP servers. Because of the growth of spam and email-mining tools, SMTP servers often respond to requests for information in defensive ways. SMTP servers will respond very slowly to information requests, provide unhelpful data, or sometimes no data at all. This means that the SMTP validation done in EV can be inconsistent, or even time out. You must take into account that SMTP servers are telling EV this misinformation, or not responding in a timely manner, and it is outside the service’s control. Any SMTP check reporting a “timeout” doesn’t necessarily mean the email is invalid, it means only that the SMTP server was protecting itself.

### ValidateEmail

This is the basic operation for validating an email. It runs through several checks to determine if an email is valid. If one step fails, subsequent checks cannot be done because they have been logically eliminated; if the syntax on an email fails, neither the DNS nor the SMTP check is done.

#### Step 1: Syntax Check

The email is run through formatting strings to verify whether it has a valid structure, such as an “@” symbol, a domain, and no odd characters that aren’t allowed in email addresses.

#### Step 2: DNS Check

The email's domain is verified to make sure that it exists. Once the domain is found, the domain's DNS records are scanned for the location of the SMTP server for that domain.

### Step 3: SMTP Check

Using the data gathered from Step 2, EV contacts the SMTP server to verify that it really is there and is accepting connections. EV will then ask the SMTP server to start a new email to the target recipient, but then stop and quit before the email is sent. No email is sent to the email being verified, ever.

After these three steps are complete, EV searches other data lists for more information about the target email. For instance, EV has lists of known free email domains, such as gmail.com, yahoo.com and hotmail.com.

### ValidateEmail Inputs

Name	Type	Description
Email	String	The email address to validate. Validation will not be performed on empty strings or null values.
LicenseKey	String	Your license key to use the service. Sign up for a free trial key at <a href="http://www.serviceobjects.com">www.serviceobjects.com</a> .

### ValidateEmail Outputs

Name	Type	Values	Description
SyntaxCheck	String	Passed Failed	The status of the Syntax Check.
DNSCheck	String	Passed Failed	The status of the DNS Check.
SMTPCheck	String	Passed Failed	The status of the SMTP Check.
AgeIndex	String	Always "0"	Deprecated, no longer used.
ValidityIndex	String	Always "0"	Deprecated, no longer used.
IsFree	String	True False Unknown	Whether the domain of the email is a public-register domain, where users can sign up for email accounts for free.
IsEstablished	String	"0" or "1"	Whether the email is well known in large bulk-marketing lists.
Exception	String	Varies	If an error occurred when

			processing the email, the error's description will be displayed here.
Error – Desc	String	Varies	If there was an internal web service error, the description will be displayed here.
Error – Number	String	"1", "2", "4"	See "Error Codes" below.
Error -- Location	String	Always null	Deprecated, no longer used.

## ValidateEmailNoSMTP

This operation is almost exactly like ValidateEmail, but no SMTP Check is performed. The inputs are exactly the same, and the outputs are identical save for the SMTPCheck value. This method was created for a fast alternative to ValidateEmail that wouldn't time out – for instances where real time processing was crucial. A better option for this is to use the newer ValidateEmailFastSMTP method, described below.

### ValidateEmailNoSMTP Inputs

Name	Type	Description
Email	String	The email address to validate. Validation will not be performed on empty strings or null values.
LicenseKey	String	Your license key to use the service. Sign up for a free trial key at <a href="http://www.serviceobjects.com">www.serviceobjects.com</a> .

### ValidateEmailNoSMTP Outputs

Name	Type	Values	Description
SyntaxCheck	String	Passed Failed	The status of the Syntax Check.
DNSCheck	String	Passed Failed	The status of the DNS Check.
AgeIndex	String	Always "0"	Deprecated, no longer used.
ValidityIndex	String	Always "0"	Deprecated, no longer used.
IsFree	String	True False Unknown	Whether the domain of the email is a public-register domain, where users can sign up for email accounts for free.
IsEstablished	String	"0" or "1"	Whether the email is well known in large bulk-marketing lists.

Exception	String	Varies	If an error occurred when processing the email, the error's description will be displayed here.
Error – Desc	String	Varies	If there was an internal web service error, the description will be displayed here.
Error – Number	String	"1", "2", "4"	See "Error Codes" below.
Error -- Location	String	Always null	Deprecated, no longer used.

## ValidateEmailFullSMTP

In an effort to normalize output parameters and improve the SMTP check, ServiceObjects built new functions to give the same great validation data, but with internal logic that would save EV from doing unnecessary, lengthy algorithms for data it already had access to.

ValidateEmailFullSMTP was born from this effort. The method it uses to check syntax and DNS is the same as ValidateEmail, but improves upon the earlier methods by using cached data instead of making labor-intensive connections. For example, once ValidateEmailFullSMTP knows that "a.com" does not exist, it won't need to make a connection out to look for "a.com" ever again.

However, this method does not cut the SMTP process off. It will check whether the given SMTP server accepts any email address, and won't check for mailboxes on that SMTP server if that server will accept anything.

All of the outputs have also been normalized to "true"/"false". This will help processing on your side, as there is no need to look up what the outputs may be.

## ValidateEmailFullSMTP Inputs

Name	Type	Description
Email	String	The email address to validate. Validation will not be performed on empty strings or null values.
LicenseKey	String	Your license key to use the service. Sign up for a free trial key at <a href="http://www.serviceobjects.com">www.serviceobjects.com</a> .

## ValidateEmailFullSMTP Outputs

Name	Type	Values	Description
SyntaxCheck	String	True False	The status of the Syntax Check.
DNSCheck	String	True False	The status of the DNS Check.
SMTPCheck	String	True False	The status of the SMTP Check.
IsVulgar	String	True False	Whether the email address contained obvious vulgar words.
IsBogus	String	True False	Whether the email was an obviously bogus email – such as <a href="mailto:a@a.com">a@a.com</a> .
IsFree	String	True False Unknown	Whether the domain of the email is a public-register domain, where users can sign up for email accounts for free.
IsEstablished	String	True False	Whether the email is well known in large bulk-marketing lists.
IsAlias	String	True False	Whether the email matches alias rules for different sites. For example, any email with a “+” in it at gmail.com is an alias.
IsCatchAllDomain	String	True False	Whether the domain for the email accepts any email address as valid.
PossibleFirstName	String	Varies	A common first name that was found in the email address. It is possible this is the user’s first name, but not guaranteed.
PossibleLastName	String	Varies	A common last name that was found in the email address. It is possible this is the user’s last name, but not guaranteed.
Exception	String	Varies	If an error occurred when processing the email, the error’s description will be displayed here.
Error – Desc	String	Varies	If there was an internal web service error, the description will be displayed here.
Error – Number	String	“1”, “2”, “4”	See “Error Codes” below.
Error -- Location	String	Always null	Deprecated, no longer used.

## ValidateEmailFastSMTP

This method is almost exactly like ValidateEmailFullSMTP, but will use heuristics to make the SMTP resolution much faster.

For example, FullSMTP will do full mailbox resolution, which slows the operation down considerably in some circumstances. With FastSMTP, this check is skipped in the light that the SMTP server will probably be a catch-all server and give no useful information in the mailbox resolution step anyway.

There are no other differences in the service, except that the output for the service contains “FastSMTP” instead of “SMTPCheck”.

### ValidateEmailFastSMTP Inputs

Name	Type	Description
Email	String	The email address to validate. Validation will not be performed on empty strings or null values.
LicenseKey	String	Your license key to use the service. Sign up for a free trial key at <a href="http://www.serviceobjects.com">www.serviceobjects.com</a> .

### ValidateEmailFastSMTP Outputs

Name	Type	Values	Description
SyntaxCheck	String	True False	The status of the Syntax Check.
DNSCheck	String	True False	The status of the DNS Check.
FastSMTP	String	True False	The status of the Fast SMTP Check (SMTP Check sans mailbox resolution.)
IsVulgar	String	True False	Whether the email address contained obvious vulgar words.
IsBogus	String	True False	Whether the email was an obviously bogus email – such as <a href="mailto:a@a.com">a@a.com</a> .
IsFree	String	True False Unknown	Whether the domain of the email is a public-register domain, where users can sign up for email accounts for free.

IsEstablished	String	True False	Whether the email is well known in large bulk-marketing lists.
IsAlias	String	True False	Whether the email matches alias rules for different sites. For example, any email with a "+" in it at gmail.com is an alias.
IsCatchAllDomain	String	True False	Whether the domain for the email accepts any email address as valid.
PossibleFirstName	String	Varies	A common first name that was found in the email address. It is possible this is the user's first name, but not guaranteed.
PossibleLastName	String	Varies	A common last name that was found in the email address. It is possible this is the user's last name, but not guaranteed.
Exception	String	Varies	If an error occurred when processing the email, the error's description will be displayed here.
Error – Desc	String	Varies	If there was an internal web service error, the description will be displayed here.
Error – Number	String	"1", "2", "4"	See "Error Codes" below.
Error -- Location	String	Always null	Deprecated, no longer used.

## Error Codes

Error codes in EV are the same for all operations. They are as follows:

Error Code 1 – "Input cannot be less than zero length"

This error means the web service did not get any input. The connection to the service was made, and data was transferred, but no parameters were passed that the service could understand.

This error often happens when input is passed to the service with namespaces that the service does not understand. Applying a namespace to any of the parameters (Email or LicenseKey, in this service) will cause this error. Additionally, requests made in the "rpc/encoded" format will cause this error. The only namespace that should appear in any element is the "<http://www.serviceobjects.com>" namespace on the root ValidateEmail element as so:

```
<ValidateEmail xmlns="http://www.serviceobjects.com/">
```

**Note, however, that the namespace is not applied to the ValidateEmail element, it is only present.**

Error Code 2 – Various descriptions

This error code appears when various internal errors occur, but are of the expected nature. Service timeouts, connection errors, and the like will occur with an error code 2.

Error Code 4 – Various descriptions

An error code 4 is a fatal error and it means something has seriously gone wrong. You will never see an error code 4 in a live production environment.

## Integration

Integrating EV into your application should be easy and straightforward. If you are using a common platform, ServiceObjects may already have sample code built that you can use:

[http://www.serviceobjects.com/support/dots\\_example\\_code.asp](http://www.serviceobjects.com/support/dots_example_code.asp)

However, if you are using a common platform that does not already have sample code, you can ask ServiceObjects to build you an example. Email [support@serviceobjects.com](mailto:support@serviceobjects.com) for more details.

### Which Operation Should You Use?

Picking which operation you want to use should be decided carefully. Depending on your environment and needs, you will need to use different operations for their corresponding strengths:

ValidateEmailFullSMTP – stronger for slower, offline systems or batches, and care about metadata that could be important for judging borderline emails' validity

ValidateEmailFastSMTP – stronger for real-time or online systems that cannot handle long delays, but also need metadata regarding an email's validity.

ValidateEmail – stronger for slower, offline systems or batches that need definitive validation, but not all metadata about an email address.

ValidateEmailNoSMTP – stronger for real-time, fast systems that cannot handle long delays but don't need all metadata about an email address.

The best suggestion is to try out each of the operations to find the data you need. If you need lots of metadata, one of the Full/FastSMTP operations would be best. If you need the operation to be fast and have minimal timeouts, FastSMTP or NoSMTP would be best.

### **The Sample Code is Giving Strange Errors or is Crashing!**

Most likely, the sample code cannot connect to ServiceObjects. Many environments will not allow you to connect out on port 80, or will clip out XML data from these requests/responses.

The easiest way to check for this is to open a browser on the machine running the sample code. In your browser, navigate to:  
<http://trial.serviceobjects.com/ev/EmailValidate.aspx>

Then try to run one of the operations with your trial key. If you get a browser error, or get no data back, then the sample code isn't able to connect, either. Contact your systems administrator to resolve why you are not able to connect to ServiceObjects.

### **Email Validation Says An Email is Bad When I Know It's Good!**

More than likely, the SMTP check timed out or came from a questionable SMTP server. **This does not mean the email was bad.** It means EV could not get the right information from the email's SMTP server. Please understand SMTPCheck's output with the consideration that timeouts happen, and they're outside the control of EV.

### **Email Validation Says An Email is Good When I Know It's Bad!**

More than likely, the SMTP server will say any email given to it is valid, primarily to deter spammers from running scripts to gather lists of email addresses. You can glean this from the FullSMTP/FastSMTP operations' IsCatchAllDomain parameter, which shows if the SMTP server will accept any email you provide.

Please review the other output fields from the web service to make the determination if the email is truly invalid – IsVulgar, IsBogus, IsEstablished, etc. provide great insight into whether the email is probably invalid.

### **I'm Not a Programmer. How Do I Use DOTS EmailValidation?**

ServiceObjects runs batches for you! A free batch trial is available at <http://www.serviceobjects.com/batch/signup1.asp>.

## Conclusion

ServiceObjects is proud to offer you a free trial of DOTS Email Validation.

Sign up today for a free trial at:

[http://www.serviceobjects.com/products/dots\\_email.asp](http://www.serviceobjects.com/products/dots_email.asp)

Other technical questions or concerns can be directed to [support@serviceobjects.com](mailto:support@serviceobjects.com).

If you are interested in purchasing DOTS Email Validation, please contact [sales@serviceobjects.com](mailto:sales@serviceobjects.com).

We welcome your feedback! Please do not hesitate to let us know what you think of our web services, documentation, or customer support.

[www.serviceobjects.com](http://www.serviceobjects.com)