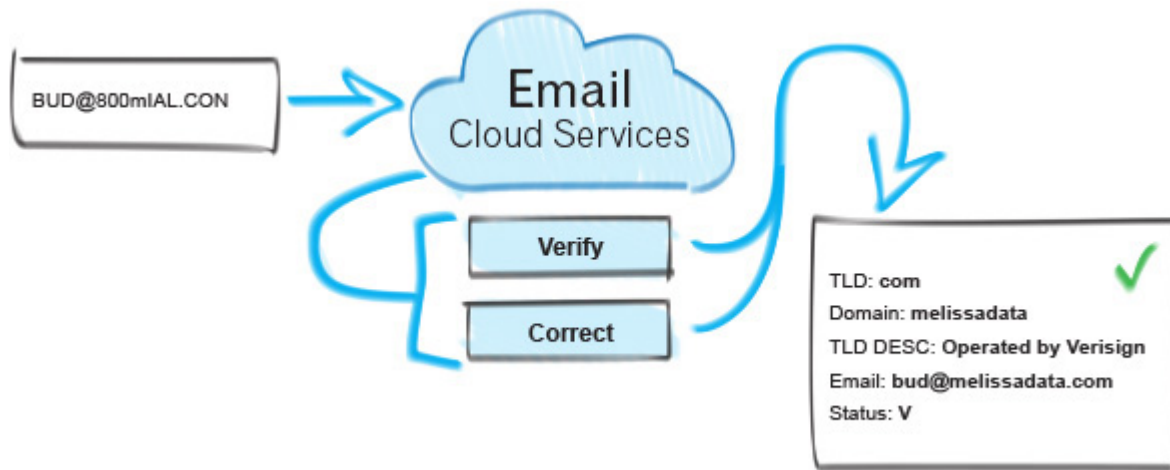


## Global Email Cloud Service: Programmer's Quick Start

### Overview

The Global Email Cloud Service features real-time Email Mailbox Verification that captures up to 95% of bad emails, helps improve campaign deliverability, increases conversions, and prevents bounce-backs and black listing.

Global Email validates your email address information and will allow your Web sites and custom applications to update email addresses in your database files while verifying and correcting misspelled domain names.



### You can use Global Email to:

- Verify and correct syntax errors; checks for illegal characters and extra "@" characters.
- Verify and correct top-level domain names.
- Check for and correct misspelled domain names.
- Check for and correct domains that have changed.
- Standardize casing in email addresses.
- Perform Real Time Mailbox Validation.

### Global Email Real Time Mailbox Validation has the ability to detect:

- The email account (the inbox) does not exist at the given domain.
- The email address format is incorrect.
- If mailbox exists on mail server.
- Invalid, bad email.
- MX Server Error.
- The email address is associated with a domain that doesn't exist.
- Unknown or catch all domains.
- Disposable/Spamtrap domains; Email address is disposable and the Inbox will self-destruct.
- Role-based emails.
- Valid email address.

The Global Email service provides two options for email validation: **PREMIUM** and **EXPRESS** at two different pricing and subscription models.

The **PREMIUM** option always does the most comprehensive type of real-time checking by using domain-specific logic as well as SMTP commands and other proprietary mechanisms to validate email boxes. Use the **PREMIUM** option when you absolutely require the highest level of Mailbox verification. Bear in mind it can sometimes take up to 12 seconds to verify an email box. Applications that are time sensitive in nature should be programmed to call the **EXPRESS** mode instead.

The **EXPRESS** option of Global Email will use cached mailbox validations, so any submitted emails will be checked against a database of known bad and known good emails. Any emails that are sent into the **EXPRESS** will also be cached so they can be looked up during a 90-day cycle. Use this option for increased cost savings or where speed and performance are of upmost importance.

---

## FIELDS INPUT AND OUTPUT FROM THE SERVICE

INPUTs	Description
Transmission Reference	A unique string value identifying the request
Customer ID	License String from Melissa Data
Options (Express, Premium)	Control Options for the Web Service
Record ID	Unique ID if processing multiple records
Email	Email Address to validate

OUTPUTs	Description
Transmission Reference	A unique string value identifying the request
Transmission Results	Results of the transmission (success/fail)
Record ID	Unique ID if processing multiple records
Results	Results (status of the validation)
Email Address	Email Address (full address)
Mailbox Name	Mailbox Name (before the @ sign)
Domain Name	Domain Name (yahoo)
Top Level Domain	Top Level Domain (com)
Top Level Domain Name	Top Level Domain Name (commercial)

### License String

You should have been provided an encrypted and unique license string or Customer ID from Melissa Data. This is necessary for including with each request to the Global Email Cloud Service. This value should be put into the CustomerID element in each Web service request.

If you do not have a license string, please contact your Melissa Data sales representative at 1-800-MELISSA (1-800-635-4772).

## Sample REST Requests

1. <https://globalemail.melissadata.net/v3/WEB/GlobalEmail/doGlobalEmail?id=12345678&email=surfbud@gmail.com&opt=VerifyMailBox:Express&format=json>
  2. <https://globalemail.melissadata.net/v3/WEB/GlobalEmail/doGlobalEmail?id=12345678&email=surfbud@gmail.com&opt=VerifyMailBox:Premium&format=xml>
- 

## Sample JSON Response

```
{"Version":"3.0.0.21","TransmissionReference":"","TransmissionResults":  
", "TotalRecords": "1", "Records": [{"RecordID": "1", "Results": "ES01,ES22", "EmailAddress": "surfbud@gmail.com", "MailboxName": "surfbud", "DomainName": "gmail", "TopLevelDomain": "com", "TopLevelDomainName": "commercial"}]}
```

---

## Sample XML Response

```
<Response xmlns="urn:mdGlobalEmail" xmlns:i="http://www.w3.org/2001/XMLSchema-instance">  
<Version>3.0.0.21</Version>  
<TransmissionReference/>  
<TransmissionResults></TransmissionResults>  
<TotalRecords>1</TotalRecords>  
<Records>  
  <ResponseRecord>  
    <RecordID>1</RecordID>  
    <Results>ES01,ES22</Results>  
    <EmailAddress>surfbud@gmail.net</EmailAddress>  
    <MailboxName>surfbud</MailboxName>  
    <DomainName>gmail</DomainName>  
    <TopLevelDomain>com</TopLevelDomain>  
    <TopLevelDomainName>commercial</TopLevelDomainName>  
  </ResponseRecord>  
</Records>  
</Response>
```

---

## Single vs. Batch

Melissa Data cloud services are capable of both single record real-time processing and batch processing. The difference is simply in the number of records sent in each request. Melissa Data cloud services take an array of records. This array can contain a single record or 100 records. For a real-time process like a Web form entry or a call center application, send in a request with one record. For a batch processing scenario like a database, send requests of up to 100 records until all the records are processed. Note: Make sure each record in the request has a unique Record ID.

## Sample Batch XML Response

```
<?xml version="1.0" ?>
<Request xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <CustomerID xmlns="urn:mdGlobalEmail">xxxxxxx</CustomerID>
  <Options xmlns="urn:mdGlobalEmail">VERIFYMAILBOX:EXPRESS</Options>
<Records xmlns="urn:mdGlobalEmail">
<RequestRecord>
  <Email>support@melissadata.com</Email>
  <RecordID>1</RecordID>
</RequestRecord>
<RequestRecord>
  <Email>support2@melissadata.com</Email>
  <RecordID>2</RecordID>
</RequestRecord>
.....
<RequestRecord>
  <Email>support100@melissadata.com</Email>
  <RecordID>100</RecordID>
</RequestRecord>
</Records>
<TransmissionReference xmlns="urn:mdGlobalEmail" />
</Request>
```

---

## Global Email Cloud Service URLs [Global Email Cloud Service Endpoint URLs](#)

### Choosing a Web Service Protocol

The Melissa Data Global Email Cloud Service supports REST, JSON, XML, and SOAP. For the undecided, here are some Pros and Cons of one Web Service protocol over the other.

#### **REST**

**Pros:** REST is lightweight and relies upon HTTP to do its work. If you don't need a strict API definition, this is the way to go. REST is also format-agnostic so you can use XML or JSON as responses.

**Cons:** REST can only be used for sending of single records and doesn't support strict contracts or more involved security. The Response is an XML or JSON document.

#### **XML**

**Pros:** XML allows record set structures of more than one record at a time and has very good support with most languages and browsers. Supports namespaces.

**Cons:** Developers need to use tools to serialize/de-serialize the XML structure.

#### **JSON**

**Pros:** JSON relies on simple object serialization based on JavaScript's object initialization. It is very simple to use with JavaScript and easily parsed and understood by developers.

**Cons:** No support for formal definitions. No namespace support. Not much support in Web Service clients with some platforms.

#### **SOAP**

**Pros:** SOAP (using a WSDL) is a heavy-weight XML standard that is centered around document passing. The advantage with this is that your requests and responses can be very well structured.

**Cons:** SOAP documents are very verbose and hard to consume without a SOAP toolkit and generally carry more overhead.

## Basic Order of Operations (Pseudo Code)

1. Choose SOAP, XML, or the REST service.
  2. Create an instance of the request object.
  3. Populate the request element CustomerID with your Product License.
  4. Set the options for the Web service.
  5. Add input email addresses to the <Records> array with anywhere from 1 to 100 <RequestRecord> items. (SOAP, XML)
  6. Call the method and pass in the request to the service using the SOAP endpoint for SOAP request and the WEB endpoint for XML or JSON requests.
  7. Examine and parse the response from the reply object back from the service.
  8. Interpret the results.
- 

## Interpreting Results

Melissa Data's Global Email Cloud Service uses Results Codes to determine if an email is good or bad. The Melissa Data Cloud Services use the following Results conventions:

1. CLOUD SERVICE ERRORS: SExx
2. CLOUD TRANSMISSION ERRORS: GExx
3. EMAIL ADDRESS STATUS CODES: ESxx
4. EMAIL ADDRESS ERROR CODES: EExx

For Example: An ES01 Results Code means that the email address is good and an ES02 is an invalid email.

The EE01-EE04 Results Codes will explain the ES02 (Invalid) Results. Please check the documentation for any additional information on Results.

## Results Codes

The service returns a series of results codes to tell you of the status of the Email and any changes or errors found during the verification process.

For a full list of the Results Codes returned by the Global Email Cloud Service, see [Global Email Result Codes](#).

## Sample Code

Fully working examples are available on the wiki pages:  
[Click here to go to the Global Email Cloud Service Wiki Page](#)

## Wiki Page

A product support Wiki is available for your convenience. In the Wiki, you will find documentation about the service in more detail.  
[Click here to go to the Global Email Cloud Service Wiki Page](#).

## Misc. Considerations

### Firewall

If you are behind a firewall, you may need to allow specific IP addresses access in order to communicate with the service. For a full list of IP Addresses, see [IP Address Information](#).

### Appendix A: Email Validation Logic flowchart

