# Outbound Contact Deployment Guide

## Pre-Dial Validation

12/18/2025

# Pre-Dial Validation

## Contents

This topic provides an overview of the pre-dial validation feature that is introduced in Outbound Contact 8.0.001. It describes how to provision and implement pre-dial validation of the dialing records over secure and non-secure HTTP connections.

# Pre-Dial Validation Over Non-Secure Connection

## Preface

Pre-dial validation is an optional first step in the processing of the record. With pre-dial validation in place, OCS connects to the specific Web or Application Server using the HTTP or HTTPS and delivers a specifically formed HTTP `POST` request for each record before dialing this record. The body of the HTTP `POST` request contains properties of the individual record, just as an outbound call contains record-specific attached data. Based on the information received in HTTP `POST` request, Web or Application Server makes a decision whether or not the record should be dialed, and replies to OCS with either an HTTP `200 OK` or a `409 Conflict` response. As a next step in processing, OCS dials the records that were validated successfully (`200 OK` response) and attempts to apply treatments (without dialing) to those records that were validated negatively (`409 Conflict` response). Only those records that have successfully passed the validation are dialed by OCS.

## Provisioning

Pre-dial validation is controlled by a number of OCS options, which can be set at various configuration levels. Unless the options are set correctly, pre-dial validation will not take place. By default OCS behaves consistently with the previous release where pre-dial validation is not possible.

Configure the pre-dial validation feature by using the following new options:

- `pre-dial-validation`
- `validation-uri`
- `validation-timeout-call-result`
- `http-response-timeout`
- `http-connection-pool-size`

## Recommendations for Configuration

The pool size and HTTP request handling timeout (`http-connection-pool-size` and `http-response-timeout` options, respectively) must be set in accordance with the anticipated load on Web or Application Server. OCS will reuse open connections (although request pipelining is not supported by OCS) and will never exceed the connection pool size. After a connection has been opened, OCS will not attempt to close it and will reuse it as long as it's available (no request which has not yet been responded to is currently submitted for this connection) and valid (Web or Application Server has not closed this connection). In case an HTTP request needs to be sent out and no available (spare) connection is found in the pool and no new connection can be opened, OCS will queue the request internally until either the spare connection appears or until the handling timeout for the given request expires. Whenever the request is queued internally, log message `93100` is produced (see also Specific Log Messages for log messages description). When the request completes due to the timeout, log message 93202 is generated (see also Specific Log Messages for log messages description). The presence of 93100 and/or 93202 messages indicates that the setting is too low for the timeout, a slow responsiveness on Web or Application Server side, and/or insufficient connection pool size.

OCS creates a separate connection pool for each host:port pair it needs to maintain a connection to. This means that HTTP and HTTPS types of the connections will have separate connection pools, even if the host name is the same for both (this is due to different port numbers, 80 and 443 [defaults]). Short and fully qualified domain names for the host are also qualified as separate hosts by OCS; for example host1and host1.subdomain1.domain1.com are treated as different hosts by OCS and will therefore be assigned separate connection pools.

## Specific Log Messages

OCS logs the following standard messages when processing connections to Web or Application Server:

| | |
|---|---|
| 2102 | (data send error) |
| 4500 | (connecting) |
| 4501 | (server contacted) |
| 4502 | (cannot connect) |
| 4503 | (connected) |
| 4504 | (connection lost) |
| 4541 | (message received) |

In addition to the standard messages, OCS also logs the result of the pre-dial validation for each dial attempt (all pre-dial validation results messages are of Trace Level):

| | |
|---|---|
| 93200 | Pre-dial check completed with positive result for phone <phone number> |
| 93201 | Pre-dial check completed with negative result for phone <phone number> |
| 93202 | Pre-dial check aborted (timeout elapsed) for phone <phone number> |

Whenever the connection pool limit is reached the following Standard Level message is logged:

| | |
|---|---|
| 93100 | Maximum connections limit <number of connection> reached for server <Chost:port> |

## Pre-Dial Validation Protocol Description

This section provides a description of the pre-dial validation HTTP requests and responses.

### Requests for Pre-Dial Validation

OCS delivers POST request to Web or Application Server that contains BODY in the application/json format (JavaScript Object Notation). This BODY holds all key-value pairs of the record subject to validation in the same fashion that the outbound call produced by OCS contains key-value pairs in its attached data. Similarly to the outbound call, OCS packs in the JSON `BODY` some mandatory key-value pairs (those whose keys are prefixed with `GSW_`) and all user-defined fields configured for delivery using the `send_attribute` field-level option.

### Example: Validation Request with BODY

The following example is a request with the `BODY` (mandatory fields begin with `GSW_` and user-defined fields begin with `USR_`):

```
POST /validation/validate.php HTTP/1.1
Accept: */*
User-Agent: OCS/8.0.001
Host: host1.domain1:80
ETag: 390
Content-type: application/json
Content-length:720

        {
                "GSW_TZ_OFFSET":0,
                "GSW_PHONE":"01282663420",
                "GSW_CALLING_LIST":"PFR_CL_01",
                "GSW_CAMPAIGN_NAME":"Campaign One",
                ...
                "GSW_RECORD_HANDLE":390,
                ...
                "GSW_CALL_ATTEMPT_GUID":"00S0VQKQK0DHT2O518838SDAES000098",
                "USR_FIELD1": "John Doe",
                "USR_FIELD2": "501-12-4312",
                ...
                "USR_FIELDT": "2010-05-06 13:25:10.003",
                "USR_FIELDN": 1970
        }
```

Notice that the `ETag` header of the request is always present and holds the value of record handle of the record being populated.

### Processing the Pre-Dial Validation Request

Web or Application Server needs to make a decision about whether or not the record that has been delivered to it in the `POST` request by OCS is allowed to be dialed. This decision is made based on the individual record properties passed in the JSON `BODY` of the `POST` request, as required by the business logic. Web or Application Server should reply with `200 OK` for a positive validation result and with `409 Conflict` for a negative validation result.

Both positive and negative validation results may contain a `BODY` that must also be in the application/json format. This `BODY` can contain mandatory and user-defined fields that are to be updated. For example, a negative validation response might include a call result value that will be assigned to record or a positive validation response might contain the timestamp of the validation attempt.

### Positive Response to the Validation Request

### Example: Positive Validation Response

The following example is a positive validation response. Notice that the `ETag` header must be present in the response and must contain the value of the record handle of the record being validated.

```
HTTP/1.1 200 OK
Date: Fri, 30 Apr 2010 19:40:58 GMT
Server: Apache/2.2.14 (Win32) mod_ssl/2.2.14 OpenSSL/0.9.8k PHP/5.2.8
X-Powered-By: PHP/5.2.8
ETag: 390
Content-Length: 112
Content-Type: application/json
        {
                "USR_FIELDT": "2010-05-06 15:00:00.000"
        }
```

As a result of this response received by OCS, the record with a record handle `390` will be dialed. The user-defined field of the record with the `send_attribute` option set to `USR_FIELDT` will be updated in the Calling List table with the new string value `"2010-05-06 15:00:00.000"`. Notice that the `BODY` part of the positive response is optional and should be provided only if some fields of the record require an update.

## Negative Response to the Validation Request

### Example: Negative Validation Response

The following is an example of a negative validation response. Notice that the `ETag` header must be present in the response and must contain the value of the record handle of the record being validated.

```
HTTP/1.1 409 Conflict
Date: Fri, 30 Apr 2010 19:40:58 GMT
Server: Apache/2.2.14 (Win32) mod_ssl/2.2.14 OpenSSL/0.9.8k PHP/5.2.8
X-Powered-By: PHP/5.2.8
ETag: 390
Content-Length: 62
Content-Type: application/json
        {
                "GSW_CALL_RESULT":53,
                "USR_FIELDT": "2010-05-06 15:00:00.000"
        }
```

As a result of this response received by OCS, the record with a record handle `390` will be marked with call result `53` (Wrong Number) and will not be dialed; the user-defined field of the record with the `send_attribute` option set to `USR_FIELDT` will be updated in the Calling List table with the new string value `"2010-05-06 15:00:00.000"`. After assigning call result `53` to the record, OCS attempts to apply the treatment to specified call result (if such treatment is configured). Notice, that the `BODY` part of the response is also optional. If no call result is explicitly specified, call result `52` (Cancel Record) is applied.

## Timeout While Processing the Validation Request

It is possible that the Web or Application Server will not be able to handle a validation request during the time period that is specified by using the `http-response-timeout` option. In this situation, the validation request will be aborted, and OCS will treat the timeout situation as a negative validation outcome (no dialing will take place). OCS will apply the call result that is specified in the `validation-timeout-call-result` option (default, 3 [General Error]) and will attempt to apply the treatment to that call result.

# Pre-Dial Validation Over Secure Connection

This section describes pre-dial validation over a secure connection, including information about provisioning and secure connection-specific log messages.

## Preface

For pre-dial validation, OCS supports communication over HTTPS, or strictly speaking, HTTP over Transport Layer Security (TLS) connection, using a Genesys TLS implementation.

For a detailed description of a Genesys TLS implementation, see the *Genesys Security Deployment Guide*, Part 3, "Server Integrity - Transport Layer Security".

For information about the operating systems that are supported by a Genesys TLS implementation, see the *Genesys Security Deployment Guide*, in the "Environment Prerequisites" section.

For the installation procedure of Genesys Security Pack on UNIX, see the *Genesys Security Deployment Guide*, "Security Pack Installation". On Windows platforms, support for SSL/TLS is integrated into the operating system.

## Provisioning

To force OCS to open a secure connection to the validation service, the protocol (scheme) part of the URI that is specified in the `validation-uri` option must have a value of "`https://`". In accordance with HTTPS definitions, OCS will connect to port `443` (instead of `80`) if a port number is not explicitly specified in the URI.

In addition to that, the user must create and configure a `Host` configuration object, with the same `Name` property as the host name which is specified in the `validation-uri` option.

For a description of the required configuration steps for a `Host` configuration object, see the *Genesys Security Deployment Guide*, "Assigning a Certificate to a Host" section. Please pay attention to the fact that configuration is different for OCS deployments on UNIX and Windows OS. Also keep in mind that the referenced document describes only the most typical configuration with both a self-signed root certificate from the Certification Authority (CA) and a CA-signed client certificate. Certificate and private key files for a UNIX deployment have to be Privacy Enhanced Mail (PEM)-encoded. The `Common Name` property of the client certificate should match the corresponding host name in the Genesys configuration.

## Secure Connection-Specific Log Messages

When a secure connection to the Web or Application Server is established, OCS prints a specific trace-level message into the log output. As shown in the following example, this message contains the properties of used certificates:

```
8103 Secure connection is established. type 'client', info '1600-135.225.58.18:443',
issuer 'C=US, S=California, L=Daly City, O=Genesys, OU=Outbound, CN=Outbound
Certificate Authority.
```

If a secure connection cannot be established, the following standard-level messages are logged:

| | |
|---|---|
| 8100 | Certificate is expired |
| 8101 | Certificate is not valid |
| 8102 | Secure connection error |