Outbound Web Service
Version 3.9.4

Developer Documentation

The MTI Mark and Design are registered in U.S. Patent and Trademark Office.

Table of Contents

Overview

The outbound web service simplifies and streamlines the process of placing automated outbound calls through the Message Technologies, Inc (MTI) hosting environment. MTI clients can easily execute outbound calls by sending secure http requests to the web service. Each request that is sent to the web service will initiate one outbound call using one of MTI's hosting platforms. The web service will create a unique Session ID to identify each call initiation request. The web service can be invoked using an HTTP GET, HTTP POST, or a SOAP request. All of these access methods will yield the same result, but are included for ease of use and compatibility with multiple programming languages.

Standard Usage

The following web methods are available for use through vgOutbound:

- **GetCall** – requests the status details of an outbound call which is identified by the Session ID. Session ID is one of the values returned from the **MakeCall** method.

- **MakeCall** - initiates a request of the IVR platform to place an outbound call.

The recommended way to utilize the web service consists of a user invoking the **MakeCall** method with a defined PostbackURL. The defined PostbackURL will receive the call status information at the end of the call attempt. Based on the call status information returned by the IVR platform to the PostbackURL, the web service user (client) can then perform further processing.

Web Method Reference

GetCall Method

The **GetCall** web method allows the client to query the status of a previous call that was made using the **MakeCall** method of the web service. **GetCall** has three arguments:

Argument	Type	Description
Username	String	This is the username given to the client by MTI.
Password	String	This is the password given to the client by MTI.
SessionID	String	The unique web service SessionID that identifies the call of interest. This Session ID was returned to the client inside the response from the **MakeCall** web method. Use of this argument is necessary in order to get the call status information for a given call.

The **GetCall** method will return the call status results in the following XML format, where all xml tags between `<CallStatus>` and `</CallStatus>` represent properties of the call.

```
<?xml version="1.0" encoding="utf-8" ?>
<CallStatus xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns="http://ws.messagetech.com/vgOutbound">
  <SessionID>fc500852-0efc-4f4a-af7c-ff0c18d715d7</SessionID>
  <MakeCallRequestDateGMT>2008-07-17T15:01:01</MakeCallRequestDateGMT>
  <NumberCalled>5551234567</NumberCalled>
  <Duration>5</Duration>
  <Url>http://192.168.4.51/VXMLTest/TestOutbound.txt</Url>
  <UserReferenceID>987654</UserReferenceID>
  <Uui>123456</Uui>
  <CallRequestResult>
        fc500852-0efc-4f4a-af7c-ff0c18d715d7 Accepted 00: Call was accepted by the platform
  </CallRequestResult>
  <CallConnectionStatus>CONNECTED</CallConnectionStatus>
  <CallDropResult>APPL_END</CallDropResult>
  <MediaPlatform>vgslvs21.messagetech.com</MediaPlatform>
  <PostbackUrl>http://localhost/PostbackApplication/PostbackProcessor.aspx</PostbackUrl>
  <PostbackEncoding>URL</PostbackEncoding>
</CallStatus>
```

NOTE: The content of the XML tag `<PostbackURL>` will be empty if the PostbackURL value was not populated when the **MakeCall** method was invoked. If the

PostbackURL is defined but the PostbackEncoding type has not been set, the default encoding type will be URL Encoding (0).
.

The <CallConnectionStatus> tag will contain one of the following values:

1. ***** : Call did not connect to a person or answering machine
2. **CONNECTED**: Successfully connected.
3. **UNKNOWN_STATUS X**: X will be a status number.

The <CallDropResult> tag will contain one of the following values:

1. Possible values **before** a connection is made.
    a. **BUSY**: Line is Busy.
    b. **NO_ANSWER**: No answer
    c. **NO_RESOURCES**: no free channel or media resource. **
    d. **CALL_FAILED**: Call failed. **
    e. **URLTIMEOUT**: Fetch URL timeout. **
    f. **NO_LICENSE**: no licenses available. **
    g. **RESTRICTED_TELNO**: Restricted telephone number.
    h. **UNSUPPORTED_URL**: URL is unsupported.
    i. **INVALID_TELNO**: Invalid telephone number.
    j. **VXML_DECLINE:** Voicexml Interpreter declined the call. **
    k. **BADURI:** Invalid URI type. **
    l. **NOAUTH:** Network denied. **
    m. **GKREJECT:** H.323 Gatekepper returned ARJ. **
    n. **SHUTTINGDOWN:** Interpreter is shutting down. **
    o. **NETWORKTIMEOUT**: Network timeout. **
    p. **BADDEST**: Invalid number. **
    q. **NOCIRCUIT:** No Circuit. **
    r. **INEFFECTIVE_OTHER:** Ineffective other. **
    s. **SIT:** Destination number returned Special Information Tones.
    t. **UNKNOWN_REASON<sp><internal disconnect reason number>:** unknown reason. **
    NOTE: <SP> indicates a space in the format of the return value.
    u. **CARRIER_DECLINED:** Carrier did not accept call. **

    ** MTI leaves the action to be taken for all of the above return codes up to the clients' discretion. Return values denoted with a "**" *may* indicate an issue that would require action to be taken by MTI personnel; our initial suggestion is to retry the failed call. If the problem persists, please contact your MTI account manager for further assistance.

2. Possible values **after** a connection is made.
    a. **USER_END**: The called party hung up
    b. **APPL_END**: The application ended the call
    c. **TIMELIMIT_END**: Time limit of the call was reached

MakeCall Method

The **MakeCall** web method allows the client to initiate an outbound call in the MTI environment. Each successful call of this method will result in an outbound call being placed by an IVR platform. Once a platform accepts the call, **MakeCall** returns the details to the client. The return details include the Session ID that uniquely identifies this call. The Session ID is the necessary input for the use of the **GetCall** web method. The **MakeCall** web method returns information to the client in order to indicate if the IVR platform made the call.

> **RETURN FORMAT:** *<SessionID><sp><status><sp><code>:<sp><data>*
> **NOTE:** *<sp>* indicates a space in the format.
>
> Sample return value from the **MakeCall** web method:
> *4eed9c60-c703-4077-9792-2728def899dd Accepted 00: Call was accepted by the platform*
>
> Please see Appendix D which defines the return values for the *<status>, <code>, and <data> tags.*

All arguments of the **MakeCall** web method are required. Some of these arguments go through stringent validation. For example, the URL supplied in the URL argument will be requested by the web service in order to validate that the URL is valid before initiating the call request.

The table below defines the arguments for the **MakeCall** web method:

Argument	Type	Description
Username	String	This is the username given to the client by MTI.
Password	String	This is the password given to the client by MTI.
Environment	String	Indicates the type and version number of the IVR platform to send the outbound call through. Valid values are: STAGING_7.2 STAGING_7.2_INTERNATIONAL PRODUCTION_7.2 PRODUCTION_7.2_INTERNATIONAL **NOTE:** In order to use the STAGING_7.2_INTERNATIONAL environment, users must first contact your MTI account manager.

VxmlInterpreter	String	Indicates which VXML Interpreter should be used to run the VXML application specified in the URL argument. Valid values are: LEGACY
NumberToCall	String	This is the phone number for the Voice Genie platform to call (DNIS). This value can be supplied in the following formats: • "*<dnis>*" • "**+**<dnis>*" • "**sip:**<dnis>**@**<sip gateway>*" This parameter can consist of up to 1023 characters.
CallerID	String	This is the phone number that will show up on the caller ID (ANI). This number can not contain any alpha characters or spaces and must be less than 32 characters. **NOTE:** Users must follow all federal, state, local, and carrier laws, rules and regulations regarding the value used for the CallerID argument. **NOTE:** We suggest Toll Free Numbers (TFN) not be used for this field. Many carriers will reject calls with TFNs as the caller ID.
URL	String	The URL of the VXML application to run once the call is answered. **NOTE:** This URL will be requested by the web service during the argument validation process and again by the IVR platform when the call is connected. The HTTP User Agent will be "MTI vgOutbound Webservice".
CpaInfo	String	This setting enables Call Progress Analysis (CPA). Valid values are 'y' or 'n'. See Appendix H for additional details about CPA and how to use it.

		y - turns on call progress analysis n - turns off call progress analysis **NOTE:** Unless using the MtiToAnalyze setting 'a', CpaInfo should be set to 'n'.
MtiToAnalyze	String	Setting this property to 'a' will allow users to enable call progress analysis which is used to determine if the outbound call is answered by a person or voicemail system. See Appendix H for more details and for info on how to retrieve the call progress analysis results in your VXML application. **NOTE:** If MtiToAnalyze is set to "a", then CpaInfo must also be set to "y".
UUI	String	This is User-to-User Information. This value cannot exceed 64 characters, and can be an empty string.
UserReferenceID	String	This is an ID number that the client associates with the call. This value cannot contain Spaces and cannot exceed 254 characters, but can be an empty string.
PostbackURL	String	This is the URL for the web page that the Call Status information will be posted to. This argument must either be an empty string or be a valid URL. **NOTE:** This URL will be requested by the web service during the argument validation process and again when posting the call results to the URL. When the URL is requested by the, the HTTP User Agent will be "MTI vgOutbound Webservice"

PostbackEncoding	String	This argument indicates how to encode the Call Status information before posting it to the URL provided in the PostbackURL argument. Valid values are 0, 1,2 or 3: URL encoding = 0 Hex encoding = 1 Base64 encoding = 2 Text/XML (Plain XML) = 3
RnaTimeout	String	This argument allows users to control the amount of time in seconds that a call will ring before it is considered a no answer. The minimum value is 6, maximum value is 120, and the default is 60. *Customers should leave this parameter blank. Should you have a need to change this setting, please contact your MTI account manager.
CPASilenceDuration	String	The CPA Silence Duration is an *approximate* value for how many seconds the callee can wait silently before the web service returns 'silence' when MTIToAnalyze is being used. The minimum value that can be passed for the CPA Silence Duration parameter is 2 seconds, and the max value is 600 seconds If developers try using a value outside of this range, the following Call Reject message will be presented: Rejected 33: Invalid CPA Silence Duration
UseShortURL	String	Valid values: y n When this parameter is set to "y", the API creates a short URL that it is sent to the media platform (instead of the long URL from the MakeCall request). The short URL redirects to the application's long URL once the outbound call is answered. This setting should only be set to 'y' when the length of the application URL exceeds 450 characters. When this feature does not need to be used, set "UseShortURL" to '' or 'n'.

If the PostbackURL and PostbackEncoding arguments are being used, then once the call attempt is complete, the web service will post the call status information to the PostbackURL. The call status information posted by the web service to the PostbackURL is the exact same information that is returned when using the **GetCall** method. Please see the description of the **GetCall** method above for a detailed description of the information returned by the **GetCall** method. In order for the web service to post the call status results to the URL specified in the PostbackURL argument, the XML call status information will be encoded by the web service using URL encoding, Hex encoding, or Base64 encoding. As an alternative, users can set the PostbackEncoding to '3' to receive plain XML.

When the call status information is posted to the URL that was specified in the PostbackURL argument, the data will be stored in a form item named "CallStatus". The data may need to be decoded using the encoding type that was provided in the PostbackEnconding argument before it can be viewed as XML. It is the responsibility of the PostbackURL web page to decode and process the information.

NOTES: A Carriage-Return Line-Feed (CRLF) may be attached to the end of the encoded call status information when it is received by the PostbackURL web page. Depending on how this data will be processed, this CLRF may need to be removed by the PostbackURL web page.

Arguments of the MakeCall method should never contain spaces.

Appendix A: WSDL Location

You can find the WSDL file in the following location:

https://ws.messagetech.com/vgoutbound/vgoutbound3_9.asmx?WSDL

Appendix B: Session ID Format

The Session ID is a 36 character GUID.

Example Session ID:
4eed9c60-c703-4077-9792-2728def899dd

Appendix C: HTTP GET request format

The most common way to access the web service is by using an HTTP GET request. All argument values in the request should be URL encoded. See "Appendix D: URL Encoding" section near the end of this document for more information.

The **MakeCall** HTTP GET request must be structured in the following manner:

https://ws.messagetech.com/vgoutbound/vgoutbound3_9.asmx/MakeCall?Username =string&Password=string&Environment=string&VxmlInterpreter=string&NumberToC all=string&CallerID=string&Url=string&CpaInfo=string&MtiToAnalyze=string&Uui=str ing&UserReferenceID=string&PostbackURL=string&PostbackEncoding=string&RnaTi meout=string&CPASilenceDuration=string&UseShortURL=string

The **GetCall** HTTP GET request must be structured in the following manner"

https://ws.messagetech.com/vgoutbound/vgoutbound3_9.asmx/GetCall?Username=string&p assword=string&SessionID=string

For more information about how to make HTTP GET, HTTP POST, and SOAP requests, please go to https://ws.messagetech.com/vgoutbound/vgoutbound3_9.asmx

Appendix D: Return codes for MakeCall

RETURN FORMAT: *<SessionID><sp><status><sp><code>:<sp><data>*
NOTE: *<sp>* indicates a space in the format.

Status	Code	Error Data
Accepted	00	Call was accepted by the platform
Rejected	01	Please use a secure connection request to use this function. (HTTPS)
Rejected	02	Invalid Username and/or password.
Rejected	03	CpaInfo must be "y" or "n"
Rejected	04	Socket Not Found
Rejected	05	No Reference Number
Rejected	06	Too Many Calls in Progress
Rejected	07	Incorrect Telephone Format
Rejected	08	Incorrect Caller ID Format
Rejected	09	Incorrect URL Format
Rejected	10	Incorrect UserReferenceID Format
Rejected	11	Call Manager Not Available
Rejected	12	Could not connect to Host
Rejected	13	Could not resolve host name
Rejected	14	

		The following arguments are required: Username, Password, Environment, VxmlInterpreter, NumberToCall, CallerID, URL, CpaInfo, and MtiToAnalyze
Rejected	15	NumberToCall must be a 10 digit telephone number without any formatting characters or spaces.
Rejected	16	An unexpected response was received from the platform after sending analysis to remdial. Contact your MTI account manager.
Rejected	17	A valid URL must be provided
Rejected	18	Environment must be *<Lists of all the available environments>*
Rejected	19	A valid PostbackURL must be provided
Rejected	20	Postback Encoding Type must be 0(UrlEncoding), 1(HexEncoding), 2(Base64Encoding), 3(Text/XML)
Rejected	21	An international DNIS must start with a "+" and followed by a telephone number less than or equal to 15 digits without any formatting characters or spaces
Rejected	22	If MtiToAnalyze is set to "a" then CpaInfo must also be set to "y"
Rejected	23	Invalid VxmlInterpreter value. Valid values are "LEGACY" or "NGI"
Rejected	24	Invalid VxmlInterpreter value. NGI Interpreter only supported in 7.2+ enviornments.
Rejected	25	Invalid user permissions. Contact your MTI account manager.

Rejected	26	No SIP gateways available. Contact your MTI account manager.
Rejected	27	Your concurrent call limit has been reached. Please try again later.
Rejected	28	The systems concurrent call limit has been reached. Please try again later.
Rejected	29	The environments concurrent call limit has been reached. Please try again later.
Rejected	30	Invalid user permissions. RNA Timeout could not be changed. Contact your MTI Account Manager
Rejected	31	Invalid RNA Timeout. The RNA Timeout must be between 6 and 120
Rejected	32	Spaces are not allowed in arguments
Rejected	33	Invalid CPA Silence Duration
Rejected	34	Unprintable characters are not allowed in arguments
Rejected	35	UseShortURL must be "y" or "n"
Rejected	36	Number To Call specified is unreachable. Please try again later.
Error	96	Call Drop occurred at start of call
Error	97	No outbound servers available

Rejected	98	*<An unexpected response was received from the platform after sending the call through the socket. Contact MTI>*
Error	99	*<An unknown error occurred in the web service. Contact MTI>*

Appendix E: URL Encoding

A URL has reserved characters that can only be used in certain sections of the URL. The argument portion (also known as the query string portion) of a URL cannot contain the following reserved characters: **;**, /,**&**, **?, =**. To be able to use these characters in the argument portion of a URL, you must encode these characters (URL Encoding). Arguments such as the URL argument of the **MakeCall** method must be encoded so it can be transmitted and processed without error. URL encoding is simply replacing the character in the URL with a percent sign plus the hex value of the character that needs to be encoded. For example, the following URL has two arguments and needs to be encoded:

Original URL:
http://www.messagetech.com/vxmltest/start.vxml?arg1=value1&arg2=value2

Encoded URL:
http%3a%2f%2fwww.messagetech.com%2fvxmltest%2fstart.vxml%3farg1%3dvalue1%26arg2%3dvalue2

Helpful URL Encoding resources:
http://www.blooberry.com/indexdot/html/topics/urlencoding.htm
http://www.w3schools.com/tags/ref_urlencode.asp

Appendix F: Using a .NET proxy class to communicate with the web service

Add a web reference using the following URL:
https://ws.messagetech.com/vgoutbound/vgoutbound3_9.asmx

In the solution explorer, you will now see a section in your project called "Web References". In this section, you should see the "com.messagetech.ws" with a "vgOutbound.wsdl" file in the next level down.

Using the above web reference in .NET as proxy, an application/service can call this web service's web methods as it would normally do when calling one of its own methods or functions in the current client's programming language. For instance, in VB.NET, the function signatures that correspond to this web service's web methods would be as follows:

```
GetCall(ByVal Username As String, _
            ByVal password As String, _
            ByVal SessionID As String) As CallStatus


MakeCall(ByVal Username As String, _
            ByVal Password As String, _
            ByVal Environment As String, _
            ByVal VxmlInterpreter As String, _
            ByVal NumberToCall As String, _
            ByVal CallerID As String, _
            ByVal Url As String, _
            ByVal CpaInfo As String, _
            ByVal MtiToAnalyze As String, _
            ByVal Uui As String, _
            ByVal UserReferenceID As String, _
            ByVal PostbackURL As String, _
            ByVal PostbackEncoding As String
            ByVal RnaTimeout As String
            ByVal CPASilenceDuration
            ByVal UseShortURL) As String
```

After the web reference is added, clients can write code to use the web service. The syntax for writing this code in VB.NET is show below:

```
Dim WS As New com.messagetech.ws.vgOutbound
Dim strReturn As String

'requesting the call to be placed
strReturn = WS.MakeCall(<username>, _
                    <password>, _
                    <Environment>, _
                    <VxmlInterpreter>, _
                    <NumberToCall>, _
                    <CallerID>, _
                    <Customer VXML Application URL>, _
                    <CpaInfo>, _
                    <MtiToAnalyze>, _
                    <Uui>, _
```

```
<UserReferenceID>, _
<PostbackURL>, _
<PostbackEncoding>,_
<RnaTimeout>,_
<CPASilenceDuration>
<UseShortURL>)
```

Appendix G: Example of a NULL GetCall Response

The XML example below illustrates the format of the response that the client will receive when an invalid Session ID is sent in the **GetCall** method via an HTTP GET or HTTP POST. If the **GetCall** method is invoked via a SOAP request with an invalid Session ID, the **GetCall** method will return a NULL value.

```
<?xml version="1.0" encoding="utf-8" ?>
<CallStatus xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xsi:nil="true"
xmlns="http://ws.messagetech.com/vgOutbound">
</CallStatus>
```

Appendix H: Using Call Progress Analysis (CPA) and Getting the CPA Result in a VXML Application

For outbound call attempts that connect, the web service includes the capability to determine whether a person or answering machine answered the call. This functionality is referred to as Call Progress Analysis (CPA). To use the CPA functionality, the *CpaInfo* argument needs to be set to 'y' and the *MtiToAnalyze* argument needs to be set to 'a'.

Here are details for how CPA behaves when MtiToAnalyze is set to 'a':

The first page of the VXML application (as defined in the URL argument of the MakeCall web method) will have the following three querystring arguments appended at the end:

- MTI_CPA_Result – The final result of the call progress analysis. The value will be one of the following. Note that the CPA functionality no longer checks for fax machines, and that *sit* is also no longer included in the set of possible MTI_CPA_Result values.
    - human
    - answeringmachine
    - silence (returned if the called party remains silent)

- MTI_Salutation_Length – The salutation duration. The unit of measure used is milliseconds.

- interpreter – the VXML interpreter used for the outbound call:
    - legacy
    - ngi

For example, if the URL in the MakeCall method is set to http://exampleurl.com/example.vxml, your VXML application can expect to see the MTI_CPA_Result, MTI_Salutation_Length, and interpreter arguments passed in the querystring:

http://exampleurl.com/example.vxml?interpreter=legacy&
MTI_CPA_Result=human&MTI_Salutation_Length=3000

Based on the query string information, your VXML application can capture the interpreter, MTI_CPA_Result, and MTI_Salutation_Length argument values, and execute accordingly.

When MtiToAnalyze is set to 'a', the CPA functionality will wait until the callee has completed her salutation before returning the MTI_CPA_Result to the IVR application. The callee's salutation could be as brief as "Hi" or "Hello, this is Jane" or as long as a typical Voice Mail salutation. The CPA is designed to return quickly once it has determined that the callee is a person *and* that the callee has finished speaking the initial salutation. If the CPA has determined that the callee is an answering machine, its behavior is then tailored to listen and wait until it's time to leave the voicemail message before returning the MTI_CPA_Result to the IVR

application. If the MTI_CPA_RESULT=answeringmachine, it is safe to assume that the machine has completed speaking and if desired, the user application can begin leaving a message immediately.

When MtiToAnalyze is set to 'a', if the party answering the phone remains silent the application will appear to pause. Analysis cannot begin until a person or Voice Mail system begins speaking. For example, if MtiToAnalyze is set to 'a' and a client places an outbound call but does not speak when he answers the phone, the callee will hear a couple seconds of silence. If the callee doesn't speak during the couple of seconds of silence, CPA will return an MtiToAnalyze value of "silence". This silence duration can be lengthened beyond 3 seconds if need-be using the MakeCall web method argument *CpaSilenceDuration*.

Appendix I: Call throttling in the vgOutbound Environment

In order to avoid potentially overwhelming any shared resources on the MTI platform, vgOutbound uses a call throttling mechanism in order to limit outbound traffic at the client, environment, and vgOutbound levels. The first step after selecting MTI as your outbound hosting provider is to determine how much outbound capacity will be required to support the application(s). MTI works closely with our clients to understand projected call volumes (by hour, day, week), average call duration, and peak concurrent call capacity. After determining the requirements, MTI will set the maximum number of concurrent calls that a client will be allowed to place. Please note that if your deployment requires a higher concurrent call rate or your needs have changed; please contact your MTI account manager for further assistance.

vgOutbound uses three different maximum concurrent call limits for throttling calls:

1) The **Outbound** maximum concurrent call limit - total number of outbound calls that can be in process at the same time across all outbound clients and environments.

2) The **environment** maximum concurrent call limit - total number of outbound calls that can be in progress at the same time for a specific environment. Please see page six (6) for a list of valid environments.

3) The **client** maximum concurrent call limit - total number of outbound calls that can be in progress at the same time for a specific client

Appendix J: Viewing call logs in the vgOutbound Environment

The MTIWeblogs portal allows clients to view raw logs from both the staging and production environments. In addition to viewing individual logs online, clients can bulk export logs to .csv format. If whole call recording is enabled within the application, whole call recordings are also made available on a per call basis. Bulk export of call recordings is not enabled at this time. Before enabling whole call recording, please speak to your MTI account manager.

The metrics logs for outbound calls are available at:
http://logs.messagetech.com

Log-In Credentials:
Please log into the portal using the username and password that was provided by your MTI account manager. If you do not know this information, please contact your MTI account manager. Note that the username and password for the log portal are the same as the credentials used to access the outbound web service.

Notes:

- When searching the staging environment, only calls from the last 24 hours are shown

- When searching the production environment, data is stored for a maximum of 96 hours. Due to the volume of calls, production logs will have a one-hour time delay

- The DNIS Search type is not valid for outbound calls