



# Critico SOAP Web Services

Web Services API | Version 1.12

# User Guide

The information contained within this document is the property of PageOne Communications Ltd t/a Critico and may not be copied used or disclosed in whole or in part, except with the prior written permission of Critico.

## Table of Contents

<b>Critico SOAP Web Services .....</b>	<b>1</b>
<b>User Guide.....</b>	<b>1</b>
<b>Amendment History .....</b>	<b>3</b>
<b>1. Overview .....</b>	<b>4</b>
<b>2. Introduction.....</b>	<b>4</b>
<b>3. Getting Started .....</b>	<b>4</b>
3.1 Registering for the service.....	4
3.2 TLS/SSL Cryptographic Compliance .....	4
3.3 DNS Resilience .....	5
<b>4. Using the Oventus Web Service - API .....</b>	<b>6</b>
4.1 Oventus Soap Structure.....	6
<b>5. Oventus SOAP Functionality Overview .....</b>	<b>6</b>
<b>General/Synchronous SOAP Services .....</b>	<b>7</b>
5.1 Message Service.....	7
5.1.1 Authentication .....	8
5.1.2 Monitoring.....	9
5.1.3 Send.....	10
5.1.4 Retrieval.....	13
5.1.5 Account Management.....	15
5.2 Contact Service.....	16
<b>Asynchronous SOAP services .....</b>	<b>19</b>
6.1 2-Way Functionality (Push) .....	19
6.2 CallbackService (Push).....	20
<b>7. Advanced SOAP services .....</b>	<b>22</b>
7.1 Location Service .....	23
7.2 LocationCallbackService .....	24
7.3 Network Lookup Service .....	26
7.4 NetworkLookupCallbackService.....	28
<b>8. Response Codes .....</b>	<b>29</b>
8.1 Send Message Result Codes.....	29
8.2 Delivery Report Result Codes.....	30
8.3 Voice/TTS Message Status Codes .....	31
<b>9. Supported Source Addresses.....</b>	<b>31</b>
<b>10. Pager Message Types .....</b>	<b>32</b>
<b>11. Supported Character Sets.....</b>	<b>32</b>
<b>12. Help and Support .....</b>	<b>33</b>

## Amendment History

Date	Update
24/08/2010	Change of WSDL url. Addition of result codes for low credits.
10/01/2011	Update of Status codes
25/02/2011	Addition of TTS service
01/03/2011	Removal of Callback service parameters
12/05/2011	Pager Callback service new type field.
16/05/2011	Send message valid/accepted source address formats
20/06/2011	Pager callback request to have new statusID to statusText fields
12/07/2011	Spelling error correction of field "latitude" tag in pager callback request
06/07/2012	Updated IP addresses and SSL certificate information
17/09/2014	Added Responder App message types to callback
12/10/2016	Updated security certificate information
11/09/2020	Updated security certificate information
31/05/2023	Update Sender ID/Alphatag rules

## 1. Overview

'Oventus' is Critico's core multi-channel messaging gateway which supports its range of Connect, Paging, Responder, Flare and Connect messaging services.

Critico provides a SOAP web service interface to enable developers to integrate SMS functionality into their applications. All services are accessible using HTTP or HTTPS protocols

The SOAP API uses Web Services Description Language, which is an XML-based language that provides a model for describing Web services.

WSDL is often used in combination with SOAP and XML Schema to provide web services over the Internet. A client program connecting to a web service can read the WSDL to determine what functions are available on the server. Any special data types used are embedded in the WSDL file in the form of XML Schema. The client can then use SOAP to actually call one of the functions listed in the WSDL.

## 2. Introduction

This service, like most web services, requires that you have access to a programming language, such as Java, C#, PHP, or ASP, as well as basic knowledge of how to program in that language. You will not be able to access the service using only HTML.

This developer's guide assumes that you have a reasonable working knowledge of Microsoft Visual Studio and of .NET and HTTP development and are comfortable working in C# or VB.NET.

Critico already have customers who have implemented our Web Services using Java, C#, .NET, PHP, and VB.NET and sample code in these programming languages is available if required by visiting [www.ovens.com](http://www.ovens.com) or contacting our Customer Support Team (details at the end of the document).

## 3. Getting Started

### 3.1 Registering for the service

In order to use the Critico messaging gateway you need a Critico account and at least one registered connection between your application and our gateway. Please follow the steps below to get started:

1. Contact us with your request to obtain an account.
2. Critico will provide you with your user-id and password.
3. Program your application to connect to our web servers and call the required functions.
4. Receive and process the results of the functions back into your application.

### 3.2 TLS/SSL Cryptographic Compliance

It is important to note that once developed and implemented SOAP services maybe isolated from internet TLS/SSL environment updates that are issued from time to time. Failure to update SOAP environments may result in communication failures and/or compromise data security.

Critico use a GlobalSign Extended Validation Certificate to encrypt server communications. It is important that the SOAP environment Key-Stores are updated from time to time for GlobalSign nested path routing.

To ascertain and acquire the appropriate certificate chain proceed as follows;

- Open Internet Explorer
- Go to <https://m2m.omentus.com>.
- Click on the padlock next to the browser URL location bar. Click View Certificates, click Certification Path, this shows the nested path from Globalsign Root -> Globalsign Extended Validation CA – G3 -> www.omentus.com

All of these files can be saved in various formats from the browser for use in Key-Stores in SOAP development environments.

**TLS Cryptographic Certificates are only valid for a finite period. Globalsign renews its certificates are renewed every 12 months. Customers should use the procedure outlined above to ascertain periodicity, validity and expiration date and update their certificate as necessary.**

### 3.3 DNS Resilience

Critico use DNS resilience for soap/web based services using addresses within the following IP prefixes:

- 91.220.160.32/27
- 91.220.160.160/27

It is essential that all these ranges are included within the customer firewalls and proxy servers to ensure continued connectivity to Critico messaging platforms.

## 4. Using the Oventus Web Service - API

### 4.1 Oventus Soap Structure

Every web service request (excluding login) must contain an oventus header with a session-id, the session-id maintains authenticity with the client, and without a valid session-id all requests are rejected.

## 5. Oventus SOAP Functionality Overview

The Critico SOAP service currently provides functionality through the following services and associated methods:

Service	Description
---------	-------------

### General/Synchronous SOAP Services

MessageService	Methods to send SMS individually and in batches
	Methods to send SMS for delivering content via WAP individually and in batches
	Methods to retrieve delivery reports and inbound messages
	Methods to send text-to-speech messages
	Methods to authenticate, retrieve account and session information
ContactService	Methods to create and manage a list of contacts and contact groups within your Oventus account

### Asynchronous SOAP Services

RegisterCallbackService	Register your Push server to receive SMS delivery receipts and inbound messages from Critico
CallbackService	Specify whether you want to receive SMS delivery receipts, inbound SMS messages and 2way pager messages to your registered push server
LocationService	Register your Push server to receive location based service updates and trigger location based queries
LocationCallbackService	Receive location service authorisation commands and the location information for individual or multiple devices
NetworkLookupService	Registers your Push server to receive network updates for a GSM device and triggers a network lookup query
NetworkLookupCallbackService	Receive network lookup information for a device when it is available

## General/Synchronous SOAP Services

### 5.1 Message Service

Contained within the MessageService are methods to: authenticate to the Critico SOAP service, send messages, retrieve delivery receipts and account information and monitor the status of a session.

These operations are summarised in the following table:

Function	Method	Description
Authentication	login	Login to Critico SOAP services
	logout	Logs off user's SOAP session
Session Monitoring	ping	Checks if the session is valid
	isActive	Checks if the user has a valid session
Send	sendMessage	Sends a single message to up to 100 destinations
	sendMultimessages	Sends up to 20 messages and up to 100 destinations per message in one SOAP request
	pushWap	Sends a single WAP message
	pushMultiWap	Send multiple WAP messages
	sendTTSMMessage	Send text-to-speech message
Retrieval	getReceipt	Gets a single delivery receipt
	getReceipts	Gets up to 20 delivery receipts
	getInbound	Gets a single inbound message
	getInbounds	Gets up to 20 inbound messages
Account Management	getMsisdn	Gets a list of MSISDNs registered for the account
	getCredits	Gets remaining credits in the account(prepay)

All services are defined in:

<https://m2m.ventus.com/LiquidWS/MessageService?WSDL>

## 5.1.1 Authentication

Authentication provides two functions:

<b>login</b>	Authenticates a user onto the Oventus server.
<b>logOff</b>	Logoff allows users to securely log off the Oventus server.

**Login** and **logoff** allow users to securely log on and off the Oventus server.

### 5.1.1.1 login

Authenticates a user onto Oventus.

#### Request Parameters

Parameter	Description
user-id	Your unique username assigned by Critico
pwd	The password for your account
service-id	optional parameter that specifies the service.
version-id	optional parameter that specifies the version

#### Response Parameters

Parameter	Description
status	Status code representing the result of login
description	A textual description of the status

### 5.1.1.2 logoff

Ends the current session of the user.

#### Request Parameters

The **logoff** SOAP request body does not have any parameters.

#### Response Parameters

Parameter	Description
result	An integer indicating the logoff status



## 5.1.2 Monitoring

Session monitoring provides two functions:

<b>isActive</b>	Checks if the user has a valid session
<b>ping</b>	Checks if the session is valid

**IsActive** and **Ping** are monitoring methods that check if a user is active or whether the user's session is active or has been logged off.

### 5.1.2.1 *isActive*

**isActive** Checks if the user has a valid session.

#### Request Parameters

Parameter	Description
user-id	Your unique user username assigned by Critico

#### Response Parameters

Parameter	Description
isActiveResponse	Specifies if the user has a valid session
description	A textual description of the status

### 5.1.2.2 *ping*

**ping** request checks if the sessionID is still valid.

#### Request Parameters

The **Ping** request body does not contain any parameters.

#### Response Parameters

Parameter	Description
PingResponse	Specifies if session is valid.
Description	A readable description A textual description of the result.

### 5.1.3 Send

The **MessageService** provides four functions to send a message:

<b>sendMessage</b>	Sends a single message up to 100 destinations
<b>sendMultimessages</b>	Sends up to 20 messages and up to 100 destinations per message in one SOAP request
<b>pushWap</b>	Sends a single WAP message
<b>pushMultiWap</b>	Sends multiple WAP messages
<b>sendTTSMessage</b>	Sends a text-to-speech message

#### 5.1.3.1 SendMessage

**sendMessage** sends a single message to a list of destinations.

The request consists of a list of addresses, a message and optional parameters.

#### Request Parameters

Parameter	Description
sourceAddress	source msisdn number
destinationAddress	destination
Message	message text
ExpireAt	Specifies when the message should be expired if not delivered
flashsms	Specifies if the message is flash SMS
deliverAt	Specifies when the message should be delivered

#### Response Parameters

Parameter	Description
transactionID	Gives the ID assigned to the send message transaction
status	status code

#### 5.1.3.2 SendMultimessages

**sendMultimessages** sends up to 20 messages and up to 100 destinations per message in per SOAP request.

#### Request Parameters

The request contains one or more of the following groups.

Parameter	Description
FlashSMS	Specifies if the message is Flash SMS .
requestID	Helps the user match the multiple messages with the status values in the SOAP response.
sourceAddress	msisdn number of the account

destinationAddress	Destination
message	The message text
expireAt	When message is expired if not delivered to the destination
deliverAt	Specified when the message should be delivered

### Response Parameters

The response contains multiples of messagePacketResponse elements that each has following elements.

Parameter	Description
requestID	request ID specified in the SOAP request.
transactionID	Unique ID assigned to the message.
Status	status code for the operation
Description	A textual description of the status

### 5.1.3.3 pushWap

*pushWAP* sends a single WAP Push message.

### Request Parameters

Parameter	Description
sourceAddress	The address from which the message was sent.
destinationAddress	The address where the message is delivered
title	A brief caption/descriptor for the URL.
url	The URL of the WAP message.
expireAt	When the message should expire if not delivered to destination.

### Response Parameters

Parameter	Description
transactionID	Unique ID assigned to the Push WAP message
Status	The status code
Description	Textual description of the status

### **5.1.3.4 pushMultiWap**

**pushMultiWap** sends a batch of WAP Push messages.

#### **Request Parameters**

The request consists of a list of messagePacket that each has the following parameters.

Parameter	Description
requestID	Unique ID to match the message status in response.
sourceAddress	The address from which the message was sent. This is a string parameter.
destinationAddress	The address where message is delivered .
title	A brief caption/descriptor for the URL.
url	The URL of the WAP message.
expireAt	When the message should expire if not delivered to destination.

#### **Response Parameters**

The response is a list of messagePacketResponse elements that each has the following parameters.

Parameter	Description
requestID	request ID specified in the request.
transactionID	Unique ID assigned to the Push WAP request
Status	status code
Description	A textual description of the status

### **5.1.3.5 sendTTSMessage**

**sendTTSMessage** sends a text-to-speech message (a message sent as text, but where the recipient will be called and the message will be read out to the recipient).

#### **Request Parameters**

Parameter	Description
description	description of the message
sourceAddress	source msisdn number (not currently supported)
destinationAddress	destination
message	message text
deliverAt	Specified when the should be delivered (not currently supported)

#### **Response Parameters**

The response is a list of messagePacketResponse elements that each has the following parameters.

Parameter	Description
transactionID	Gives the ID assigned to the send message transaction
Status	status code

## 5.1.4 Retrieval

Retrieval provides four functions to get single or multiple message delivery receipts and inbound messages:

<b>getReceipt</b>	Retrieves the next available delivery receipt
<b>getReceipts</b>	Retrieves the next 'n' available delivery receipts
<b>getInbound</b>	Retrieves the next inbound message
<b>getInbounds</b>	Retrieves the next 'n' inbound messages

### 5.1.4.1 getReceipt

The **getReceipt** method will retrieve the next available delivery receipt available in the queue, which contains the receipts. (Only messages in the last 6 hours can be popped).

#### Request Parameters

Request body does not have any parameters.

#### Response Parameters

Parameter	Description
transactionID	transaction ID assigned to the original message sent
source	msisdn number
destination	destination address
sentAt	when the message was sent
updatedAt	when the message was received on the device
resultCode	An integer giving the status code

#### Response Parameters

Parameter	Description
transactionID	The ID assigned to the send message transaction
status	The status code

### **5.1.4.2 getReceipts**

The **getReceipts** method will retrieve up to the last 20 delivery receipts available in the queue, which contains the receipts. (Only messages in the last 6 hours can be popped).

#### **Request Parameters**

Request body does not have any parameters.

#### **Response Parameters**

The response is a list of “receipts” elements that each contains the following parameters.

Parameter	Description
transactionID	transaction ID assigned to the message
source	The msisdn from which the message was sent.
Destination	destination address
sentAt	The time that the message was sent.
updatedAt	The time that the message was received on the device.
resultCode	The status code

### **5.1.4.3 getInbound**

The **getInbound** method will retrieve the next available MO (inbound) message available in the queue that contains the inbound messages.

#### **Request Parameters**

The request body does not have any parameters.

#### **Response Parameters**

Parameter	Description
messageID	ID assigned to message.
source	Address from which the message originated.
destination	The address where the message was received.
timeReceived	The time the message was received
message	The message text

### **5.1.4.4 getInbounds**

The **getInbounds** method will retrieve up to the last 20 available MO (inbound) messages available in the queue that contains the inbound messages.

#### **Request Parameters**

The request body does not have any parameters.

#### **Response Parameters**

The response is a list of messages that each contains the following parameters.

Parameter	Description
messageID	The ID assigned to message.
Source	Address from which the inbound originated.
Destination	Address where the inbound was received.
timeReceived	The time the message was received
message	The message text

## 5.1.5 Account Management

This service provides two functions to return account information:

<b>getMsisdn</b>	Returns a list of msisdns registered to the account
<b>getCredits.</b>	Returns the number of message credits remaining (prepay accounts only)

### 5.1.5.1 getMsisdn

*getMsisdn* will return a list of MSISDNs belonging to the account.

#### Request Parameters

The request body does not have any parameters.

#### Response Parameters

Parameter	Description
MSISDN	A list of msisdn numbers of the account

### 5.1.5.2 getCredits

*getCredits* returns the number of remaining message credits from a prepay account.

#### Request Parameters

The request body does not have any parameters.

#### Response Parameters

Parameter	Description
creditsRemaining	the remaining credits in the account

## 5.2 Contact Service

The ContactService provides five functions to create and manage an address book and address book groups within an Oventus account:

Method	Description
listGroups	Lists the contact groups registered for the account
add	Adds a single contact to the specified group
remove	Removes a single contact from the specified group
list	Lists all contacts in the specified group
clear	Deletes all contacts in the specified group

All services are defined in:

<https://m2m.ovevus.com/LiquidWS/ContactServiceV2?WSDL>

### 5.2.1 listGroups

Allows users to return a list of contact groups registered for the account.

#### Request Parameters

The request body does not have any parameters

#### Response Parameters

Parameter	Description
contactGroupID	Unique ID of the group. This parameter can be a list.
alias	A string parameter that describes the group type

### 5.2.2 add

Adds a single contact entry to the specified contact group.

#### Request Parameters

Parameter	Description
contactGroupID	Unique ID of the Contactgroup
firstname	String
surname	String
address	One or more addresses of different types
Notes	String

#### Response Parameters

Parameter	Description
contactID	The ID assigned to the record which was added
value	The result of the addition operation



description	A textual description of the result
-------------	-------------------------------------

### **5.2.3 remove**

Removes a single contact from a specified contact group.

#### **Request Parameters**

Parameter	Description
contactID	ID of the record to be removed
contactGroupID	Unique groupID of the record

#### **Response Parameters**

Parameter	Description
Value	result of the remove operation.
Description	A textual description result

### **5.2.4 list**

Lists all contacts in the specified contact group.

#### **Request Parameters**

Parameter	Description
contactGroupID	The long ID of the group where the contacts should be listed from
ResultSetID	Optional ID to fetch the next set of contacts when the result list is too big for a single response.

#### **Response Parameters**

The response returns a list of contact elements that each contains the following parameters.

Parameter	Description
contactID	contactID for the record
firstname	Firstname
surname	Surname
Address and type	The address together with the type. This can be a list
Notes	notes

### **5.2.5 clear**

**Clear** deletes all contacts from the specified contact group.

#### **Request Parameters**

Parameter	Description
-----------	-------------

ContactGroupID	The unique group ID to be cleared
----------------	-----------------------------------

### Response Parameters

The response returns a list of contacts that have the following parameters each.

Parameter	Description
Description	The textual description of the clear operation
Value	status

# Asynchronous SOAP services

## 6.1 2-Way Functionality (Push)

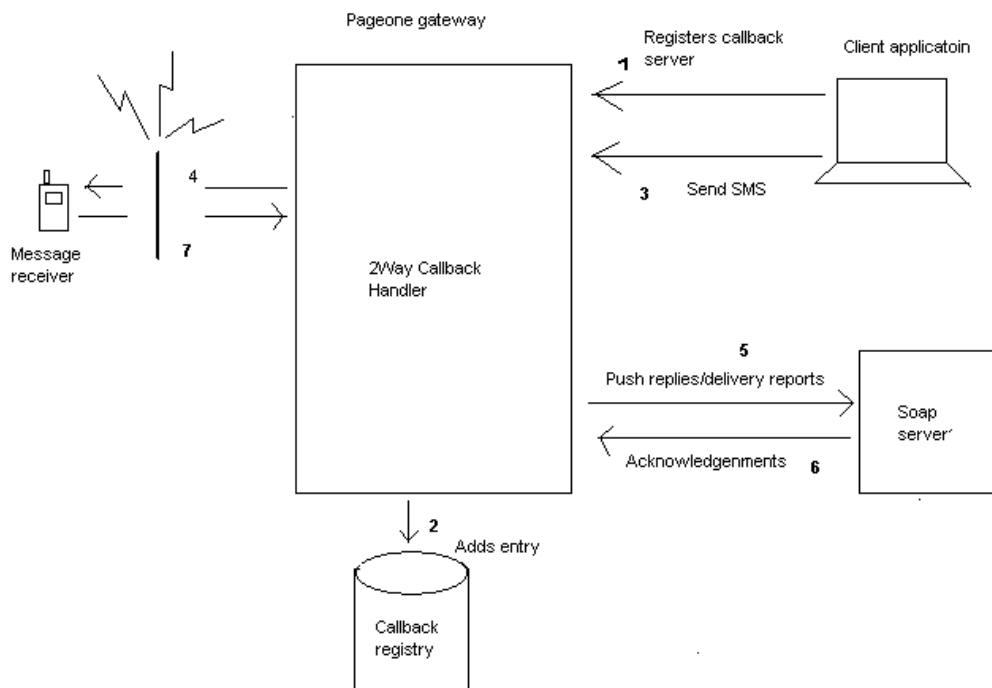
### 6.1.1 Implementing Push

Users can register with Oventus to enable Push.

When a user has registered, message delivery status and SMS replies will be automatically pushed from the Critico soap server to the client server.

Push is an example of a Business-to-Business type communication.

The following diagram illustrates how Push works.



The following is a typical Push scenario.

1. The client registers a host address with Critico server (Oventus) using port range 9110 – 9119. This is the address that the client expects Critico to send any notifications related to delivery status or message replies.
2. Oventus registers the host address in the Push registry
3. The client sends an SMS to a mobile phone (or other device) through Oventus.
4. Oventus sends the SMS to the mobile phone through the SMS receiver's network.
5. Oventus pushes the delivery status of the SMS sent back to the client. This is based on the delivery status sent by the SMS receiver's network.
6. The client-Soap server acknowledges the delivery status notification pushed from Critico. If the client does not acknowledge, Oventus will send the notification to the client again.

7. The mobile (or other destination device) sends a reply back to the client through his network.
8. Oventus pushes the reply notification to the client.
9. Client sends the acknowledgement back to Oventus.

### To use Push

The client is required to implement a server process capable of handling the requests from Oventus and sending back a response as specified in the contract and as defined in the following sections.

## 6.2 CallbackService (Push)

This describes the SOAP server that needs to be implemented on the client side in order to enable Push. Once this has been implemented, please contact Critico to register your client-side SOAP server with Critico. Until this is done, nothing will be pushed out to your server.

<https://m2m.ovevus.com/LiquidWS/CallbackServiceV3?WSDL>

Method	Description
onDeliveryReport	Receives a delivery report push from Critico
onMessageReceived	Receives inbound message pushed from Critico
onPagerMessageReceived	Receives pager message push from Critico

### 6.2.1 Delivery Report

*onDeliveryReport* - Called with the delivery details of a message.

#### Request Parameters

Parameter	Description
Source	The msisdn from which the message was sent
destination	The destination
ReceiptTime	The time the message was sent/updated
resultCode	The delivery status of the message
transactionID	Unique ID assigned to message

### 6.2.2 Inbound Messages

*onMessageReceived* - Called when a new message is received.

#### Request Parameters

Parameter	Description
source	The address where message originated from
destination	The address where the message was received

messageTime	The time the message was received
Text	The message contents
transactionID	Unique ID assigned to message

### **6.2.3 Inbound Responder 2-Way Messages**

***onPagerMessageReceived*** - method is invoked when a message is received from a responder 2-Way device.

#### **Request Parameters**

Parameter	Description	Optional?
battery	The battery level as a percentage	
delivered	1=message delivered, 0=message not delivered	
message_read	1=message read, 0=message not read	
position.latitude	The latitude of the device	
position.longitude	The longitude of the device	
position.accuracy	Circle of accuracy	
position.time	Time of the position	
request_text	The message in the original outbound request	yes
response_text	Message received from the device	
type	Message type (appendix section 10)	
destination.address	Adress/ID assigned to the pager	yes
destination.firstname	Associated first name	yes
destination.surname	Associated surname	yes
source.firstname	Associated first name	yes
source.surname	Associated surname	yes
source.address	Source device address	
time_received	The time the message was received	
transactionID	Unique transaction ID for the message	
statusText	Customer specific status mode of device e.g. 'available'	yes
statusID	An ID for status mode [0-9]	yes

## Responder App options by message type

Field Name	Acknowledgement	Delivery Receipt	Read Receipt	Lone Worker <ul style="list-style-type: none"> <li>● Started</li> <li>● Updated</li> <li>● Stopped</li> </ul> SOS
battery	✓	✓	✓	✓
delivered	✓	✓	✓	✓
message_read	✓	✓	✓	✓
position.latitude	Optional	Optional	✗	✗
position.longitude	Optional	Optional	✗	✗
position.accuracy	Optional	Optional	✗	✗
position.time	Optional	Optional	✗	✗
request_text	✓	✓	✓	✓
response_text	✓	✓	✓	✓
type	180	182	200	160/162/161/32
destination.address	✓	✓	✓	✓
source.address	✓	✓	✓	✓
time_received	✓	✓	✓	✓
transactionID	✓	✓	✓	✓

## 7. Advanced SOAP services

### 7.1 Location Service

Users will need to subscribe to the Location Service before use

The Location Service provides four functions to administer the client side server.

Method	Description
registerLocationListener	Registers a push server for location updates
unregisterLocationListener	Unregisters locationPush server
listLocationlisteners	Lists all push servers registered for location messages
locate	Triggers locating a device

The service is defined in:

<https://m2m.omentus.com/LiquidWS/LocationService?WSDL>

#### **7.1.1 Registerlocationlistener**

##### Request Parameters

Parameter	Description
url	Address of the client's push server

##### Response Parameters

Parameter	Description
status	Status code of the register operation
description	A textual description of the status.

#### **7.1.2 unregisterlocationlistener**

##### Request Parameters

Parameter	Description
Id	Unique ID assigned to the server

##### Response Parameters

Parameter	Description
status	Status code of the operation
description	A textual description

### **7.1.3 listLocationlisteners**

#### **Request Parameters**

The list request body does not have any parameters.

#### **Response Parameters**

Parameter	Description
url	The address of the Push server
id	Unique ID assigned to the server

### **7.1.4 locate**

#### **Request Parameters**

Parameter	Description
address	The address to be located

#### **Response Parameters**

Parameter	Description
status	Status of the operation
description	A textual description

## **7.2 LocationCallbackService**

The Location Push Service provides two functions

Method	Description
onAuthorised	Pushed to client server when status related to a client's authorisation for location service is received
onLocated	Pushed to client on receiving location information

The service is defined in

<https://m2m.oventus.com/LiquidWS/LocationCallbackService?WSDL>



### **7.2.1 onAuthorised**

#### **Request Parameters**

Parameter	Description
deviceSearched	The address that was searched
status	The status code
description	A textual description status code

### **7.2.2 onLocated**

#### **Request Parameters**

Parameter	Description
deviceSearched	The address that was searched
latitude	Latitude of the location
longitude	Longitude of the location
accuracy	The accuracy
city	City
region	Region
postcode	Postcode

## 7.3 Network Lookup Service

This command enables users to check our coverage of a network or number, without sending a message to that number.

The Network Lookup Service provides four functions to

Method	Description
registerNetworkLookupListener	Registers a Push server for network lookup updates received
unregisterNetworkLookupListener	Unregisters network Push server
listNetworkLookupListeners	Lists all Push servers registered for network lookup updates
locateNetwork	Triggers network lookup task

The service is defined in:

<https://m2m.oventus.com/LiquidWS/NetworkLookupService?WDSL>

### 7.3.1 RegisterNetworkLookupListener

#### Request Parameters

Parameter	Description
url	Address of the client's Push server

#### Response Parameters

Parameter	Description
Status	Status code of the register operation
description	A textual description of the status

### 7.3.2 unregisterNetworkLookupListener

#### Request Parameters

Parameter	Description
id	Unique ID assigned to the server

#### Response Parameters

Parameter	Description
status	Status code of the operation
description	A textual description

### **7.3.3 ListNetworkLookupListeners**

#### **Request Parameters**

The list request body does not have any parameters.

#### **Response Parameters**

Parameter	Description
url	The address of the Push server
id	Unique ID assigned to the server

### **7.3.4 locateNetwork**

#### **Request Parameters**

Parameter	Description
address	The address to be located

#### **Response Parameters**

Parameter	Description
status	Status of the operation
description	A textual description

## 7.4 NetworkLookupCallbackService

The Network Lookup Service provides one function

Method	Description
onNetworkFound	Pushed to client when network information is found

The service is defined in

<https://m2m.oventus.com/LiquidWS/NetworkLookupCallbackService?WSDL>

### 7.4.1 OnNetworkFound

#### Request Parameters

Parameter	Description
address	address (mobile number)
NetworkCode	NetworkCode
networkName	NetworkName
networkType	NetworkType
organisation	Organization
location	Location
countryCode	CountryCode
countryName	CountryName
status	status

## 8. Response Codes

### 8.1 Send Message Result Codes

The following list of error messages are generated by the Critico gateway during a validation phase before we accept the message.

Code	Description	Description
201	Accepted	Message request has been successfully received by Critico
400	No Access to this Service	You have not signed up to the service you are trying to use
402	Service currently unavailable	Service connection is currently unavailable
507	Insufficient Credits	Your account does not presently possess enough message credits to send your request
552	Inactive Subscriber	The device number is inactive
553	Max Limit Reached	The set amount of messages for this service has been reached (Generally applies to our trial customers)
554	Please Contact Service Provider	Operator network failure
555	System Error	Service connection is currently unavailable
556	Unknown	Final status is unknown
558	Failure to Replace	MT network unable to replace the SMS on the MT customer's handset

## 8.2 Delivery Report Result Codes

These are message statuses that are generated after the Critico gateway has accepted the message for delivery.

Code	Description	In Detail
200	Sent	Message has been delivered to handset
201	Accepted	Message has been received and is being sent to handset
202	Group Accepted	Group message has been received and sent to the group device number
203	Valid Login	Login Successful
210	IM Accepted	Message has been accepted as type IM
211	Email accepted	Message has been accepted as type Email
212	Pager accepted	Message has been accepted as type Pager
213	Location Accepted	Message has been accepted as type location
214	TTS Accepted	Message has been accepted as type TTS
400	No Access to this Service	You have not signed up to the service you are trying to use
401	Message Not Sent	Rejected by Remote Provider
402	Service currently unavailable	Service connection is currently unavailable
403	Could not send e-mail	Unable to convert message / device into e-mail
404	Unable to deliver	Unspecified protocol error on the MT handset
501	Bad Character	Content related error
502	Invalid Number	The device number you are sending to is incorrect
503	Unknown Called Address	The device number you are sending to does not exist
504	Unknown Pager	The device number you are sending to does not exist
505	Protocol Error	Permanent operator error
507	Insufficient Credits	Your account does not presently possess enough message credits to send your request
551	Failed Security	MT handset is listed as an Illegal device on the MSC.
552	Inactive Subscriber	The device number is inactive
553	Max Limit Reached	The set amount of messages for this service has been reached (Generally applies to our trial customers)
554	Please Contact Service Provider	Operator network failure
555	Systems Error	Service connection is currently unavailable
556	Unknown	Final status is unknown
557	Expired by Provider	Unable to deliver within the allocated time frame
558	Failed to replace	MT network unable to replace the SMS on the MT customer's handset
559	Insufficient Funds	Credit related - message has not been retried by operator

### 8.3 Voice/TTS Message Status Codes

These are message statuses that are generated after the Critico gateway has accepted Voice and Text-to-Speech messages for delivery.

Code	Description	In Detail
601	Answered or Connected	Answered, but recipient not accepted message (retry)
602	Accepted	Answered and recipient accepted message (retry)
603	Declined	Answered and recipient declined message (retry)
604	Rejected	Answered but recipient rejects message (no retry)
605	Reserved for future use.	
606	No Answer	No answer (retry)
607	Number Busy	Engaged (retry)
608	Network reject	Network Reject (retry)
609	Reserved for future use.	
610	Max Retries Reached	Max Retries Reached
620	Stopped - by system	Stopped by user
621	Stopped - Bad Number	Stopped by user
622	Could not send e-mail	Stopped by user
623	Stopped - No answer	Stopped by user
624	Stopped - Not accepted	Stopped by user
630	Confirmed (positive)	Recipient gives positive confirmation to message
631	Confirmed (negative)	Recipient gives negative confirmation message

## 9. Supported Source Addresses

When sending a message, the source address (alphanum) specified should conform to the following standard:

1. The maximum length is 11 characters
2. The supported characters are restricted to those in the following table below
3. Alphanums need to be pre-registered against your account or risk being blocked by certain mobile networks

Hex	Dec	Chr	ISO/IEC 10646-1:2000 Character Name
20	32		SPACE
26	38	&	AMPERSAND
2D	45	-	HYPHEN-MINUS

2E	46	.	FULL STOP
2F	47	/	SOLIDUS
30	48	0-9	ALL DIGITS FROM 0 -9
41-5A 61-7A	41-5A 61-7A	A-Z and a-z	LATIN CAPITAL LETTER A TO Z AND LATIN SMALL LETTER a-z
5F	95	_	UNDERSCORE

## 10. Pager Message Types

Service	Description
1	Auto acknowledgement
2-9	Fixed text response
17	Location request response
32	SOS
35	SOS test
38	SOS update (Later version 2.43 onwards)
48	Status switch
64	Remote tracking
100	Lone worker
110	Free text response

## 11. Supported Character Sets

The Critico SOAP service uses an encoding system that supports international language character sets. A message, which contains characters outside of the 7bit GSM default alphabet, will be processed as an international message using UCS2. This will reduce the maximum number of potential characters from 160 (if the message is entirely constructed from the 7 bit GSM alphabet) to 70 per SMS packet.

For example, an 84-character message, which contains a character outside of the 7bit GSM alphabet, will be sent as 2 separate SMS messages

**Note:** Characters that start with a decimal value of 27 are an extension of the default character set. As such, the inclusion of these characters will reduce the standard 160-character packet size.



## 12. Help and Support

For more information and support, please contact customer support:

*Tel:* 0333 200 5033

*email:* [customersupport@criticogroup.com](mailto:customersupport@criticogroup.com)

*Company website:* [criticogroup.com](http://criticogroup.com)

*Developer website:* [www.oventus.com](http://www.oventus.com)