

# WhatsApp Cloud API

---

WhatsApp Cloud API client — Meta-hosted WhatsApp Business messaging, templates and media for Delphi.

## Overview

---

WhatsApp Cloud API client — Meta-hosted WhatsApp Business messaging, templates and media for Delphi. The component is part of the sgcWebSockets library.

## At a glance

---

### COMPONENT CLASS

`TsgcWhatsApp_Client`

### STANDARDS / SPEC

WhatsApp Cloud API — Meta for Developers

### TRANSPORTS

TCP, TLS

### PLATFORMS

Windows, macOS, Linux, iOS, Android

### FRAMEWORKS

VCL, FireMonkey, Lazarus / FPC

### EDITION

Standard / Professional / Enterprise

## Features

---

- Native Delphi implementation with full ANSI/Unicode support.

# Technical specification

---

Standards & specs	<a href="#">WhatsApp Cloud API — Meta for Developers</a>
Component class	<code>TsgcWhatsApp_Client</code> (unit <code>sgcWhatsApp_Client</code> )
Frameworks	VCL, FireMonkey, Lazarus / FPC
Platforms	Windows, macOS, Linux, iOS, Android

---

## Main properties

The principal published / public properties used to configure and drive the component. Consult the online help for the full list.

<code>ServerOptions</code>	Published or public property used to configure or query the component.
<code>TLSoptions</code>	Published or public property used to configure or query the component.
<code>OnBeforeSubscribe</code>	Published or public property used to configure or query the component.
<code>LogOptions</code>	Published or public property used to configure or query the component.
<code>WhatsAppOptions</code>	Published or public property used to configure or query the component.
<code>OnBeforeSendMessage</code>	Published or public property used to configure or query the component.
<code>OnRawMessage</code>	Published or public property used to configure or query the component.
<code>OnMessageReceived</code>	Published or public property used to configure or query the component.
<code>OnMessageSent</code>	Published or public property used to configure or query the component.
<code>Log</code>	Published or public property used to configure or query the component.

---

## Main methods

The principal public methods exposed by the component.

<code>SendMessageText()</code>	Public function exposed by the component.
<code>SendMessageInteractiveList()</code>	Public function exposed by the component.

---

---

<code>SendMessageTemplate()</code>	Public function exposed by the component.
<code>StartServer()</code>	Public procedure exposed by the component.
<code>StopServer()</code>	Public procedure exposed by the component.
<code>SendTest()</code>	Public function exposed by the component.
<code>SendMessageImage()</code>	Public function exposed by the component.
<code>SendMessageDocument()</code>	Public function exposed by the component.
<code>SendMessageAudio()</code>	Public function exposed by the component.
<code>SendMessageVideo()</code>	Public function exposed by the component.

---

## Quick Start

---

Drop the component on a form, configure the properties below and activate it. The snippet that follows shows the typical **Get Started** configuration sourced from the online help.

**About this scenario.** To send and receive a first message using a test number, complete the following steps:

### Delphi (VCL / FireMonkey)

```
oClient := TsgcWhatsapp_Client.Create(nil);
oClient.WhatsappOptions.PhoneNumberId := '107809351952205';
oClient.WhatsappOptions.Token := 'EAA040pgZAs98BAGj3nCFGr...ZB2t8mmLB2LRXJkte2Y5PMNh2';
oClient.SendTest('34605889421');
```

### C++ Builder

```
TsgcWhatsapp_Client oClient = new TsgcWhatsapp_Client();
oClient->WhatsappOptions->PhoneNumberId = "107809351952205";
oClient->WhatsappOptions->Token = "EAA040pgZAs98BAGj3nCFGr...ZB2t8mmLB2LRXJkte2Y5PMNh2";
oClient->SendTest("34605889421");
```

### .NET (C#)

```
TsgcWhatsapp_Client oClient = new TsgcWhatsapp_Client();
oClient.WhatsappOptions.PhoneNumberId = "107809351952205";
oClient.WhatsappOptions.Token = "EAA040pgZAs98BAGj3nCFGr...ZB2t8mmLB2LRXJkte2Y5PMNh2";
oClient.SendTest("34605889421");
```

## Common scenarios

---

The following scenarios are lifted verbatim from the online help. Each shows the configuration and method calls needed to drive the component through a specific real-world flow.

### 1 · Received Messages

Every time a new message is received the event `OnMessageReceived` is called, where you can access to the content of the `Message` and mark the message as read.

Delphi (VCL / FireMonkey)

```
procedure OnWhatsAppMessageReceived(Sender: TObject; const aMessage: TsgcWhatsApp_Receive_Messag
var
  vText: string;
  vTo: string;
begin
  if aMessage.Contacts.Count > 0 then
  begin
    vTo := aMessage.Contacts.Contact[0].WaID;
    if aMessage.Messages.Count > 0 then
    begin
      if aMessage.Messages._Message[0]._Type = wapmrtText then
      begin
        vText := 'ECHO ==> ' + aMessage.Messages._Message[0].Text.Body;
        WhatsApp.SendMessageText(vTo, vText);
        aMarkAsRead := True;
      end;
    end;
  end;
end;
```

C++ Builder

```

void OnWhatsAppMessageReceived(TObject *Sender, const TsgcWhatsApp_Receive_Message *aMessage, re
{
    if (aMessage->Contacts->Count > 0)
    {
        string vTo = aMessage->Contacts->Contact[0]->WaID;
        if (aMessage->Messages->Count > 0)
        {
            if (aMessage->Messages->_Message[0]->_Type = wapmrtText)
            {
                vText = "ECHO ==> " + aMessage->Messages->_Message[0]->Text->Body;
                WhatsApp->SendMessageText(vTo, vText);
                aMarkAsRead = true;
            }
        }
    }
}

```

.NET (C#)

```

void OnWhatsAppMessageReceived(TsgcWhatsApp_Client Sender, TsgcWhatsApp_Receive_Message Message,
{
    DoLog("Message Received: [" + Message.From + "]" + Message.Text);
    MarkAsRead = true;
}

```

## 2 · Create Endpoint

Before you can start receiving notifications you will need to create an endpoint on your server to receive notifications.

Delphi (VCL / FireMonkey)

```

oClient := TsgcWhatsapp_Client.Create(nil);
oClient.ServerOptions.WebhookOptions.Endpoint := '/webhook';
oClient.ServerOptions.WebhookOptions.Token := 'MySecretToken';
oClient.StartServer();

```

C++ Builder

```

TsgcWhatsapp_Client oClient = new TsgcWhatsapp_Client();
oClient->ServerOptions->WebhookOptions->PhoneNumberId = "/webhook";
oClient->ServerOptions->WebhookOptions->Token = "MySecretToken";
oClient->StartServer();

```

.NET (C#)

```
TsgcWhatsapp_Client oClient = new TsgcWhatsapp_Client();
oClient.ServerOptions.WebhookOptions.PhoneNumberId = "/webhook";
oClient.ServerOptions.WebhookOptions.Token = "MySecretToken";
oClient.StartServer();
```

### 3 · Image Messages

Call the method `SendMessageImage` and pass the following parameters:

Delphi (VCL / FireMonkey)

```
oClient := TsgcWhatsApp_Client.Create(nil);
oClient.WhatsappOptions.PhoneNumberId := '107809351952205';
oClient.WhatsappOptions.Token := 'EAA040pgZAs98BAGj3nCFGr...ZB2t8mmLB2LRXJKte2Y5PMNh2';
oClient.SendFileImage('34605889421', 'c:\images\image.png', 'image/png');
```

C++ Builder

```
TsgcWhatsapp_Client oClient = new TsgcWhatsapp_Client();
oClient->WhatsappOptions->PhoneNumberId = "107809351952205";
oClient->WhatsappOptions->Token = "EAA040pgZAs98BAGj3nCFGr...ZB2t8mmLB2LRXJKte2Y5PMNh2";
oClient->SendFileImage("34605889421", "c:\images\image.png", "image/png");
```

.NET (C#)

```
TsgcWhatsapp_Client oClient = new TsgcWhatsapp_Client();
oClient.WhatsappOptions.PhoneNumberId = "107809351952205";
oClient.WhatsappOptions.Token = "EAA040pgZAs98BAGj3nCFGr...ZB2t8mmLB2LRXJKte2Y5PMNh2";
oClient.SendFileImage("34605889421", "c:\images\image.png", "image/png");
```

### 4 · Interactive List

Interactive messages give your users a simpler way to find and select what they want from your business on WhatsApp. During testing, chatbots using interactive messaging features achieved significantly higher response rates and conversions compared to those that are text-based.

Delphi (VCL / FireMonkey)

```
oClient := TsgcWhatsApp_Client.Create(nil);
oClient.WhatsappOptions.PhoneNumberId := '107809351952205';
oClient.WhatsappOptions.Token := 'EAA040pgZAs98BAGj3nCFGr...ZB2t8mmLB2LRXJkte2Y5PMNh2';
oClient.SendMessageInteractiveList('34605889421', 'What Would you like to do today?', 'To begin,
```

C++ Builder

```
TsgcWhatsApp_Client oClient = new TsgcWhatsApp_Client();
oClient->WhatsappOptions->PhoneNumberId = "107809351952205";
oClient->WhatsappOptions->Token = "EAA040pgZAs98BAGj3nCFGr...ZB2t8mmLB2LRXJkte2Y5PMNh2";
oClient->SendMessageInteractiveList("</code><code class=
```

.NET (C#)

```
TsgcWhatsApp_Client oClient = new TsgcWhatsApp_Client();
oClient.WhatsappOptions.PhoneNumberId = "107809351952205";
oClient.WhatsappOptions.Token = "EAA040pgZAs98BAGj3nCFGr...ZB2t8mmLB2LRXJkte2Y5PMNh2";
oClient.SendMessageInteractiveList("</code><code class=
```

## 5 · Template Message Parameters

Templates can include parameters, see below an example of default template with parameters

Delphi (VCL / FireMonkey)

```

procedure SendSamplePurchaseFeedbackTemplate(const aName: string);
var
  oTemplate: TsgcWhatsApp_Send_Message_Template;
  oComponent: TsgcWhatsApp_Send_Message_Template_Component;
  oParameter: TsgcWhatsApp_Send_Message_Template_Parameter;
begin
  oTemplate := TsgcWhatsApp_Send_Message_Template.Create;
  Try
    oTemplate.Language.Code := 'en_US';
    oTemplate.TemplateName := 'sample_purchase_feedback';
    // ... header
    oComponent := TsgcWhatsApp_Send_Message_Template_Component.Create;
    oComponent._Type := wapctHeader;
    oTemplate.Components.Add(oComponent);
    oParameter := TsgcWhatsApp_Send_Message_Template_Parameter.Create;
    oParameter.Image.Link := 'https://www.esegece.com/images/esegece.png';
    oParameter._Type := wapptImage;
    oComponent.Parameters.Add(oParameter);
    // ... body
    oComponent := TsgcWhatsApp_Send_Message_Template_Component.Create;
    oComponent._Type := wapctBody;
    oTemplate.Components.Add(oComponent);
    oParameter := TsgcWhatsApp_Send_Message_Template_Parameter.Create;
    oParameter.Text := aName;
    oParameter._Type := wapptText;
    oComponent.Parameters.Add(oParameter);
    whatsapp.SendMessageTemplate('107809351952205', oTemplate);
  Finally
    oTemplate.Free;
  End;
end;

```

C++ Builder

```

void SendSamplePurchaseFeedbackTemplate(string aName)
{
    TsgcWhatsApp_Send_Message_Template *oTemplate = new TsgcWhatsApp_Send_Message_Template();
    Try
        oTemplate→Language→Code = "en_US";
        oTemplate→TemplateName = "sample_purchase_feedback";
        // ... header
        TsgcWhatsApp_Send_Message_Template_Component *oComponent = new TsgcWhatsApp_Send_Message_Tem
        oComponent→_Type = wapctHeader;
        oTemplate→Components→Add(oComponent);
        TsgcWhatsApp_Send_Message_Template_Parameter *oParameter = new TsgcWhatsApp_Send_Message_Tem
        oParameter→Image→Link = "https://www.esegece.com/images/esegece.png";
        oParameter→_Type = wapptImage;
        oComponent→Parameters→Add(oParameter);
        // ... body
        TsgcWhatsApp_Send_Message_Template_Component *oComponent = new TsgcWhatsApp_Send_Message_Tem
        oComponent→_Type = wapctBody;
        oTemplate→Components→Add(oComponent);
        TsgcWhatsApp_Send_Message_Template_Parameter *oParameter = new TsgcWhatsApp_Send_Message_Tem
        oParameter→Text = aName;
        oParameter→_Type = wapptText;
        oComponent→Parameters→Add(oParameter);
        whatsapp→SendMessageTemplate("107809351952205", oTemplate);
    __Finally
        oTemplate→Free();
    End;
}

```

```

void SendSamplePurchaseFeedbackTemplate(string aName)
{
    TsgcWhatsApp_Send_Message_Template oTemplate = new TsgcWhatsApp_Send_Message_Template();
    oTemplate.Language.Code = "en_US";
    oTemplate.TemplateName = "sample_purchase_feedback";
    // ... header
    TsgcWhatsApp_Send_Message_Template_Component oComponent = new TsgcWhatsApp_Send_Message_Template_Component();
    oComponent._Type = wapctHeader;
    oTemplate.Components.Add(oComponent);
    TsgcWhatsApp_Send_Message_Template_Parameter oParameter = new TsgcWhatsApp_Send_Message_Template_Parameter();
    oParameter.Image.Link = "https://www.esegece.com/images/esegece.png";
    oParameter._Type = wapptImage;
    oComponent.Parameters.Add(oParameter);
    // ... body
    TsgcWhatsApp_Send_Message_Template_Component oComponent2 = new TsgcWhatsApp_Send_Message_Template_Component();
    oComponent2._Type = wapctBody;
    oTemplate.Components.Add(oComponent2);
    TsgcWhatsApp_Send_Message_Template_Parameter oParameter2 = new TsgcWhatsApp_Send_Message_Template_Parameter();
    oParameter2.Text = aName;
    oParameter2._Type = wapptText;
    oComponent2.Parameters.Add(oParameter2);
    whatsapp.SendMessageTemplate("107809351952205", oTemplate);
}

```

## 6 · Text Messages

Call the method `SendMessageText` and pass the following parameters:

Delphi (VCL / FireMonkey)

```

oClient := TsgcWhatsApp_Client.Create(nil);
oClient.WhatsappOptions.PhoneNumberId := '107809351952205';
oClient.WhatsappOptions.Token := 'EAA040pgZAs98BAGj3nCFGr...ZB2t8mmLB2LRXJkte2Y5PMNh2';
oClient.SendMessageText('34605889421', 'Hello from sgcWebSockets!!!');

```

C++ Builder

```

TsgcWhatsapp_Client oClient = new TsgcWhatsapp_Client();
oClient->WhatsappOptions->PhoneNumberId = "107809351952205";
oClient->WhatsappOptions->Token = "EAA040pgZAs98BAGj3nCFGr...ZB2t8mmLB2LRXJkte2Y5PMNh2";
oClient->SendMessageText("34605889421", "Hello from sgcWebSockets!!!");

```

.NET (C#)

```
TsgcWhatsapp_Client oClient = new TsgcWhatsapp_Client();  
oClient.WhatsappOptions.PhoneNumberId = "107809351952205";  
oClient.WhatsappOptions.Token = "EAA040pgZAs98BAGj3nCFGr...ZB2t8mmLB2LRXJkte2Y5PMNh2";  
oClient.SendMessageText("34605889421", "Hello from sgcWebSockets!!!");
```

## Sources used to build this document

---

Every external claim links back to a primary source. The online-help references decode the canonical deep-link the company maintains for this component.

Primary standard / spec — WhatsApp Cloud API — Meta for Developers [developers.facebook.com/docs/whatsapp/cloud-api](https://developers.facebook.com/docs/whatsapp/cloud-api)

---

Delphi demo project (in the sgcWebSockets package) [Demos\50.Other\05.WhatsApp](#)

---

Component page [www.egegece.com/products/websockets/apis/whatsapp/](http://www.egegece.com/products/websockets/apis/whatsapp/)

---

Product page [www.egegece.com/products/websockets/](http://www.egegece.com/products/websockets/)

**Document scope.** This document covers the publicly-documented surface of the WhatsApp Cloud API component shipped with sgcWebSockets. For full property, method and event reference consult the online help linked above.