WhatsApp for uniGUI/VCL
via
TWhatsAppBridges Component Package fo Delphi

Introduction

These examples/components/projects allows you to create a WhatsAppWeb session and control the sending/receiving of messages over the WhatsApp network using a phone authenticated via QrCode in the same session, just like in a conventional browser. The CEF4Delphi and TInject components used here are "Open Source" projects and have the originals published on GitHub (links at the end of this doc).

My three-component package called WhatsAppBridges makes the connection avoiding Middle wares, APIs and other tricks to make this connection between the parts (CEF4Delphi + TInject + your app).

Connecting WhatsAPP with uniGUI/VCL

Several modifications made by me allow to publish the QrCode in a uniGUI project and in this way start a WhatsApp session in a browser and remotely control a the same session via a *StandAlone* uniGUI server. This allows you to create applications as such automated attendants, chat support, operation bots, chat-bots, sales, invoice, ERP, CRM, “notifiers” by WhatApp Messages etc. The primary objective is to enable this to happen transparently and for integration with Delphi applications.

For VCL, the TInject example itself already allowed this without major difficulties, although some adjustments were needed to “bridge” its functions to 3rd party applications like your projects. Hence the incompatibilities of the TInject component using here and the original that you can find on GitHub.

The main problem is that you have to build your app with Tinject + CEF4Delphi inserted in your application and such approach generates a high-level risk of maintained and dependency. There were many occasions when WhatsApp developer team made changes to que JS code and rendered inoperative for many days until the WA backstage community made some “workaround”.

Scenario/Description

Picture this: you can connect several phones with Whatsapp session and control from a single point via your browser interface. This way you can optimize procedures, services, sales, order execution, execution of scripts created by you, that is, the possibilities are many, as only the WhatsAPPWeb interface does allow this level of control.

However, the goal here IS NOT TO REPLICATE WHATSAPPWEB AND ALL ITS FUNCTIONS, but to create a form of integration to serve as a gateway for any Delphi VCL, uniGUI, Intraweb program and its current, legacy systems or even expand them.

So don't come up with absurd questions like "Does it send my audios?", "Does it send GIFs?", "Sends stickers images?", "Sends videos or make audio and video calls?". It will do if you add to your project using the code here provided. But, "*why the heck*" you would like to make another WhatsApp to use WhatsApp?

This is not the purpose of these projects as you can do the same with the WhatsAppWeb session opened on another computer. Now you can use the same WhatsAPP in four simultaneous Web sessions.

You can open as many as four different numbers in ONE WhatsAppWeb session as you like, so this kind of question does not make sense. Learn to use WhatsAppWeb then chose the features you might want to add to your app and use TWhatsAppbridges package to connect the apps. Is a 4 to 1 and a 1 to 4 solution !

Moreover, yes, you can send files like PDF, MP3, MP4 (Video), gifs etc. See DEMO WhatsAppTandem VCL that is the main provider here of all functions and features.

How it Works

I made (or better, "adapted") three components to facilitate the integration of WhatsAPPWeb (via CEF4Delphi - Chromium) with your Delphi application but can be used to bridge to any application. This package is called **WhatsAPPBridges**: One in TANDEM format and two components to provide SOCKETs connection (Client and Server), not explored in this release.

>**TANDEM (TWATandemBridge)**: On the same computer, you will run your program using the TWATandemBridge component that connects with the APP VCL that makes the WhatsAPPWeb host service. This way, you can easily integrate your projects with the features of using messaging, sending to groups and as you would use in the WhatsApp application without worrying about redoing resources, reconfigures and using paid APIs. Therefore, they work together on the SAME COMPUTER.
NOTE: Both programs must be on the same computer with WINDOWS OS. No linux here !

>**SOCKET (TWANetworkBridge:Client/Server)**: Use the Client / Server format. It may be configured for use over a local network or over the internet. The HOST and CLIENT application can be on separate computers, but with access via TCP/IP using direct Socket and HTTP connection.

WHY USE CEF4Delphi

This component allows your Delphi application to integrate the CHROMIUM engine (the same as Chrome) into an APP VCL. In this way, literally insert JavaScript commands to manipulate or create a session like in WhatsAppWeb.

WHY USE TInject

This component allows the publication to Pascal of properties and events as well as "injecting" into CEF4Delphi, the JavaScript code to do the operations that are made available in a WhatsAppWeb session, hence its name. It is he who "bridges" Delphi/Pascal and JavaScript to run a browser session inside itself. It is only tailored to work with WhatsApp JS Code provided in many resources at GItHub.

The Demo ***WhatsAppTandem*** shows how to send and receive a message via *StandAlone* uniGUI Server and be easily expanded using the properties and features shown in the DEMO VCL (WhatsAppTandem.exe) in the CEF folder.
In short, is CEF4DELPHI (provides the browser) that accepts the WhatsApp WEB JavaScript code via TInject in one VCL Application. At this same application, it connects to yours by TWhatsAppBridges or to any other application making a "bridge" between all of them.

You always need the 3 components package and, at least, two apps!

Caveats and Warnings

1-Use only the material indicated and published here in this package. I do not advise using the original **TInject** from this point forward due to several changes I have made to the TInject component, the VCL Demo and the JavaScript Code. If you want to use the original component, use the DEMO that comes with it and you will not have uniGUI integration available in it, but you can put my **TWATandembridge** component and do the same as the example published here show.

2-Visit the TInjects group on Telegram, buy courses or material from it to learn about its features. I do not provide support to TInject and it is a free project. If you want more support, subscribe to the paid format of this component. Moreover, I am well aware this is poorly documented and maintained material.

3-This is not a project for *newcomers* to Delphi, uniGUI or WhatsAPP use. If you never have used these tools, I advise you to get informed first, "play" with the TInject distribution, CEF4Delphi from GitHub, and then venture into this package.

4- I do not have the patience for those who do not have the patience to read the manual or test the demos of the material mentioned above. Please avoid problems and only search this material if you really have mastered the tools and topics mentioned.

5- FMX and Linux are not supported by TWhatsAppBridges Package.

How to install for further modifications/adapt. your projects

--->> Based on the \WhatsApp\ Folder

1- Install CEF4Delphi

In this folder in WhatsApp\CEF_files install the package \WhatsApp\CEF_files\packages for Delphi: CEF4Delphi.dpk

Be sure to place the ..\Sources folder in the path of this package.

Compile and install in your IDE

2- Install TInject

In this WhatsApp\TWhatsApp folder, open the WhatsApp\TWhatsApp\Package folder for Delphi: TWhatsAppDelphi.dpk

Be sure to place the ..\Sources folder in the path of this package.

Compile and install in your IDE

3- Install TWhatsAppBridges

In this WhatsApp\TWhatsApp folder, open the \WhatsApp\TWhatsAppBridge\Package folder for Delphi: TWhatsAppBridges.dpk

Be sure to place the ..\Sources folder in the path of this package.

Compile and install in your IDE

4- Open and Compile the WhatsAppTandem Project in the main folder of this package.

Make sure the \WhatsApp\CEF_files\Source, WhatsApp\TWhatsApp\Source and \WhatsApp\TWhatsAppBridge\Package folders are

in the SEARCH properties of the project.

This project uses the three component packages.

Your executable must ALWAYS be placed in the folder

5- Open and Compile the uniGUIStarter Project

Make sure the \WhatsApp\TWhatsAppBridge\Package folder is in the project's SEARCH properties.

This uniGUI project only uses the WhatsAppBridges package

6- Open the WhatsAppDemos.groupproj Project group and compile uniGUIStarter.

WhatsAppTandem.exe is already in the \WhatsApp folder

\CEF_files\CEF and will run there.

StandAlone uniGUI uniguistarter.exe will be compiled into this folder too. Always PLACE in CEF folder.

7-Run and press the START WHATSAPP WEB button

8-Read the QrCode and authenticate the WhatsAppWeb session. WhatsAppTandem and uniGUIStarter will be open

9-To synchronize, select the last CONFIG TAB of WhatsAppTandem and click on the WA TEST TANDEM button.

10-A confirmation message will appear in both programs with your WA ID number confirming the connection between the two Software.

NOTE: WhatsAppTandem must always be open for uniGUIStarter to work sending and receiving messages. When moving whole project to another computer, delete ConfTInject.ini from CEF folder before run WhatsAppTandem.

You can test the main application without needing to recompile. Just run the **WhatsAppTandem.exe** application in the CEF folder and make sure to install the uniGUI RunTime. This uniGUI RunTime is X.X.X.1545 version. Avoid doing this in the same computer with a previous or newer uniGUI version due to reinstallation of uniGUI RunTime. Another option is just recompile uniGUIStarter.exe with your uniGUI version BEFORE test or any usage. Just open Project Demo Group and recompile and run.

This uniGUIStarter.exe is only to demonstrate de feasibility of this project and to show you how to send and receive messages between then two app. A more complete demo (WhatsAPP uniGUI professional) is in development stage as describe below.

CEF4Delphi

It uses many stuff and folder structure's dependents. You might configure to your need but get this rule: PUT YOUR APP INSIDE CEF FOLDER!

You just have to do that. There are many Demos for Cef4Delphi projects. You do not need to mess with it for now. Do not worry.

Also, be aware that this is the only solution to use WhatsApp network with your projects without paying for 3rd party API. It is not a fetish to make you dependent on a lot of stuff but the only way for now.

TInject - Features

Modifications to the Original TInject

a- Publishing and integration of the QrCode image for reading in StandAlone uniGUI in the TEMP folder of the uniGUI server and saving there for future use. This allows you to remotely run this session, start or end.

b- Several small modifications to the component and a new organization of units of the TInject component to facilitate its use and installation.

c- Added the property to load only Broadcast groups that have a different behavior from the sending groups.

d- Removal and addition of Country Code and Area Code properties to facilitate their international use, which are the correct terms. This makes this component here incompatible with the distribution version of TInject. See ROADMAP 2021.

e- Removed of a lot of not working properties, procedures, revamped function and recode.

TInject main Features available

You might add more of these features with TWhatsAppTandem easily to your app as is done with the sending and message receiving. See the list below:

Login/Logout

-- Sending

Send text message with buttons
Send text messages to numbers outside the phonebook
Send contacts
Send MP3
Send MP4
Send IMG
Send RAR
Send Link with Preview
Send location
List contacts
List Chats
Battery status
Simulate typing
Receiving new messages notification

-- Others

WA Number validation (if permitted by contact)
Connection check
Download files
Download profile Picture
Country and Code Area code normalize

-- Groups

Send messages to groups
Send messages to Broadcast groups
Create group
Leave group
Add participant to group
Remove group member
Promote participant group administrator
Demote participant group administrator
List all groups
List group participants
Load Broadcast Groups only
Get group invitation link
Join group via invitation link

Note: You won't need to implement all of this in your projects. But it may be done with TWhatsAppBridges.

ROADMAP 2021 / Modification and future implementations

(Edited on 10/10/2021)

a- I intend to create a new component that is simpler and more effective to use that allows better integration with **uniGUI**, **Intraweb**, **Delphi VCL**. Therefore, there will be removal of the TInject component and creation of a new one in near future. However, **WhatsAppBridges components** will continue to be compatible with previous DEMOS, VCL, UNIGUI and INTRAWEB versions.

b- Greater integration with uniGUI for immediate use in your projects (**TWANetworkBridge** and maybe **Horse**) via CEF4Delphi DLL.

c- Several improvements to properties and the way the JS code is used and updated.

d- I don't intend to use any Middleware like RestDataWare as I consider it completely unnecessary for integration with any uniGUI or VCL project format. Some solutions have emerged using middleware's that is an initiative for REST solutions. However, with uniGUI, it is unnecessary as demonstrated here. Just use **TWATandemBridge** and see how simple it is to integrate anything Delphi with WhatsAPP.

e- Publication of Sample for **TWANetworkBridge** that allows the use on separate computers but connected by Internet or local network.

f- **uniGUI WhatsApp Professional:** Support for various WA and basic management of contacts, groups, sending messages and examples of advanced functions. This example will be published until 11/01/2021 !
Some features:

- WA dynamic manager. You may connect many (as 10 or more) WA ID
- and manager with shortcuts,
- auto-messages,
- scripts
- auto-response messages
- Send files (PDF, MP3, MP4, images etc)
- Scheduled messages

g- **uniGUI WhatsAPP Center:** Mega uniGUI application with support for audio, video, audio chat, video chat, custom scripts, Static PIX, Dynamic PIX, automatic response management and TWANetworkbridge component format support (Several numbers in a chat control center Whatsapp with all VCL features). All contact and group management will be done as well as importing and focus on mass messaging.

Some features:

- All professional features +
- Audio recording and sending
- Video recording and sending
- Direct Audio chat support - WebRTC
- Direct Video chat support - WebRTC
- PIX (For Brazilians users) Estático e Dinâmico
- QrCode manager (generates and decodes QrCodes received).
- Links manager.
- Predictive typing messages

- Macro substitution: Ex. Today is %date%   -> Today is 10/10/2021 and a lot of macro predefined and you can define your own.
- Message templates manager.
- Script manager.
- Voice recognition to send messages (Chrome).
- Voice Speech Syntheses.
- WA Session control per WA ID and script.

Important: uniGUI ! Until 12/01/2021 this application will be ready. This app will be based in the same components here published but largely improved with all my work with uniGUI. It will be sold separately and you might rebate all value you paid. Price not defined yet.

Price – uniGUI/WhatsApp + TWhatsappBridges and Examples

USD: $ 150,00.
Included:
>> One year UPDATE
>> ROADMAP 2021 until **item G.**
>> You have support via WhatsApp/Telegram in published projects examples here only and implementation in uniGUI. It is not support service to TInject or CEF4Delphi.
Buy now and you guarantee all items in Roadmap.

After 16/10/2021
Price USD: $200,00
>>> All items ROADMAP 2021 except item G uniGUI WA Center. After 16/10/2021, no more **item G** for free. You still gets uniGUI WA Professional (item F).

URLS related to this Project

All components here provided are free of charge or "open source", except TWhatsAppBridges.

TInject : https://github.com/mikelustosa/Projeto-TInject

TInject 2: https://github.com/mikelustosa/TInject-corporate

CEF4Delphi :https://github.com/salvadordf/CEF4Delphi

JavaScript Sources: https://github.com/pedroslopez/whatsapp-web.js

JavaScript Sources: https://github.com/open-wa/wa-automate-nodejs/releases/tag/4.22.1

uniGUi BR: https://www.unigui.**com**.br

Telegram: https://t.me/uniguiBrazil

WhatsApp/Telegram: +5511986207676