TUTORIAL · MARCH 24, 2018

Reading Twitter Feeds in a Delphi VCL app

By Dr. Holger Flick

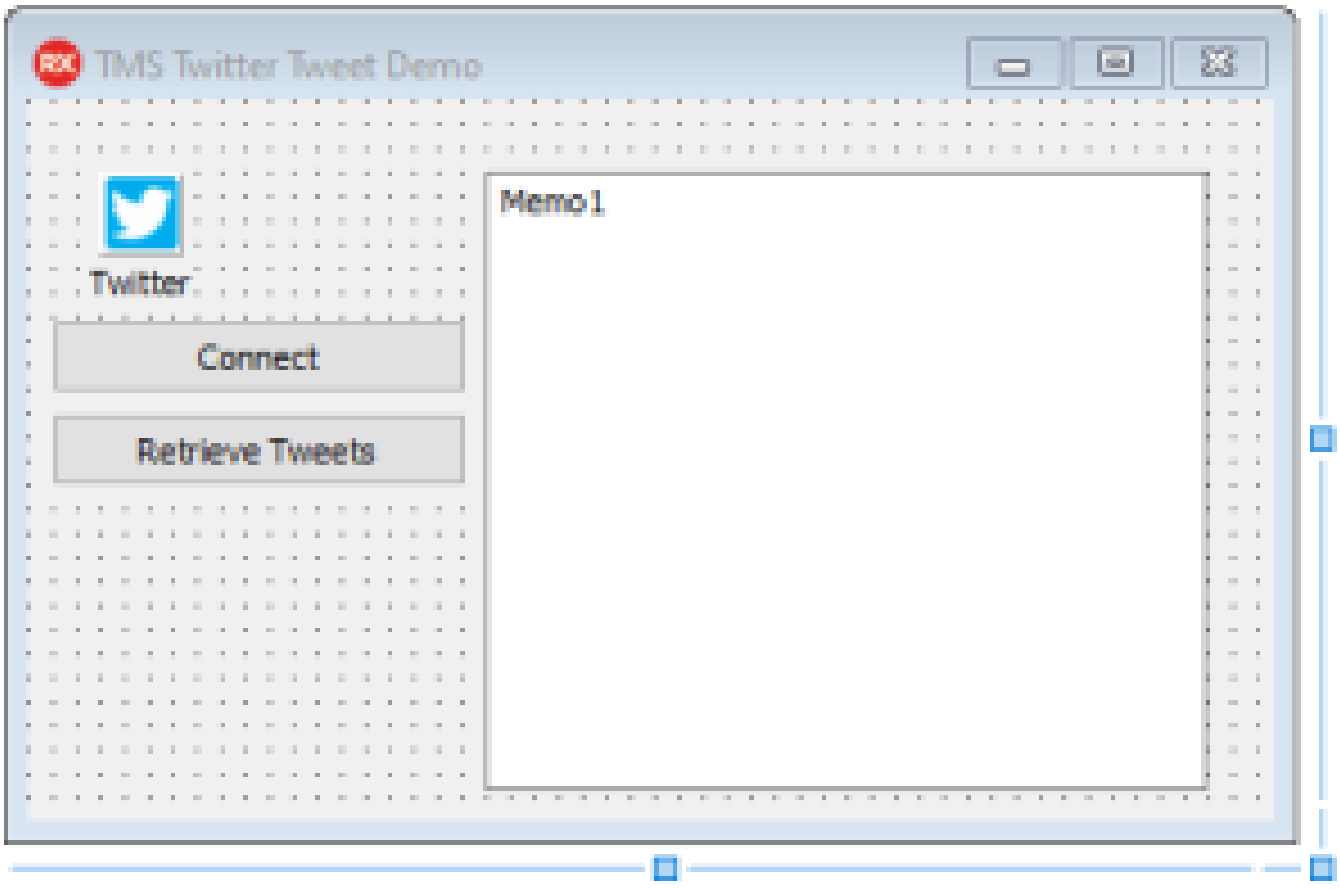
After a long time, a new 5-Minute-Snack for Delphi desktop application developers that still love the VCL as much as I do… (I can also tell you about one thing that I 100% dislike: the new WordPress editor aka Gutenberg. What a mess! I am a developer, so please let me type my post and be done with it. These new blocks are ridiculous.)

Actually, I have to confess that you will need the fabulous TMS VCL Cloud Pack to reproduce this snack. Even though Delphi is an amazing tool, it would take you longer than 5 minutes to write an interface to the Twitter API.

So, without further ado, let's look at the GUI we are going to use to access Twitter. We need two buttons:

- one to connect to Twitter and authenticate access to your account and
- another button that actually retrieves the tweets.

We are going to dump all the tweets into a `TMemo` control:



Screenshot

Finally, we need to drop the `TAdvTwitter` control from the TMS VCL Cloud Pack. This component will provide methods to read tweets from any Twitter account. However, you will also need a Twitter account yourself and you will have to create a developer account. The steps for this are described here: <https://www.tmssoftware.com/site/cloudkey.asp#twitter>

After being provided with a key and a secret, add them to the `App` property in the Object Inspector of the `TAdvTwitter` control, which I named Twitter. Another option is to set these properties in code if you want to retrieve the information from a database etc.

Let's look at the code to connect to Twitter and I'll elaborate a bit more on the tricky parts:

```
procedure TForm1.Button2Click(Sender: TObject);
begin
  Twitter.PersistTokens.Location := plIniFile;
  Twitter.PersistTokens.Key := '.\twitter.ini';
  Twitter.PersistTokens.Section := 'tokens';
  Twitter.LoadTokens;

  if not Twitter.TestTokens then
  begin
    Twitter.DoAuth;
  end;

  btnRetrieve.Enabled := Twitter.Connect;
end;
```

The Twitter API expects you to enter you username and password when you want to access it. This has to be done manually. Thus, when you connect for the first time, a browser window will open and you will be presented with the Twitter login dialog. In case you password is correct, you will be handed a token from the API that will allow you continuous access without entering the information over and over again. Thus, we need to tell our app where to store this info. This example sets up the component to save the tokens in an ini file.

The code basically loads the tokens from the file. If the file does not exist, the tokens are expired, or you simply use the app for the first time, the call to `Twitter.TestToken` will return false. On the other hand, it will return true of the tokens are valid. Furthermore, `Twitter.DoAuth` will only be called if necessary to bring up the browser window that allows you to enter username and password.

The final step is to call `Twitter.Connect`. It will return true when connected, false otherwise. We use this to enable or disable the retrieve button, which is disabled by default.

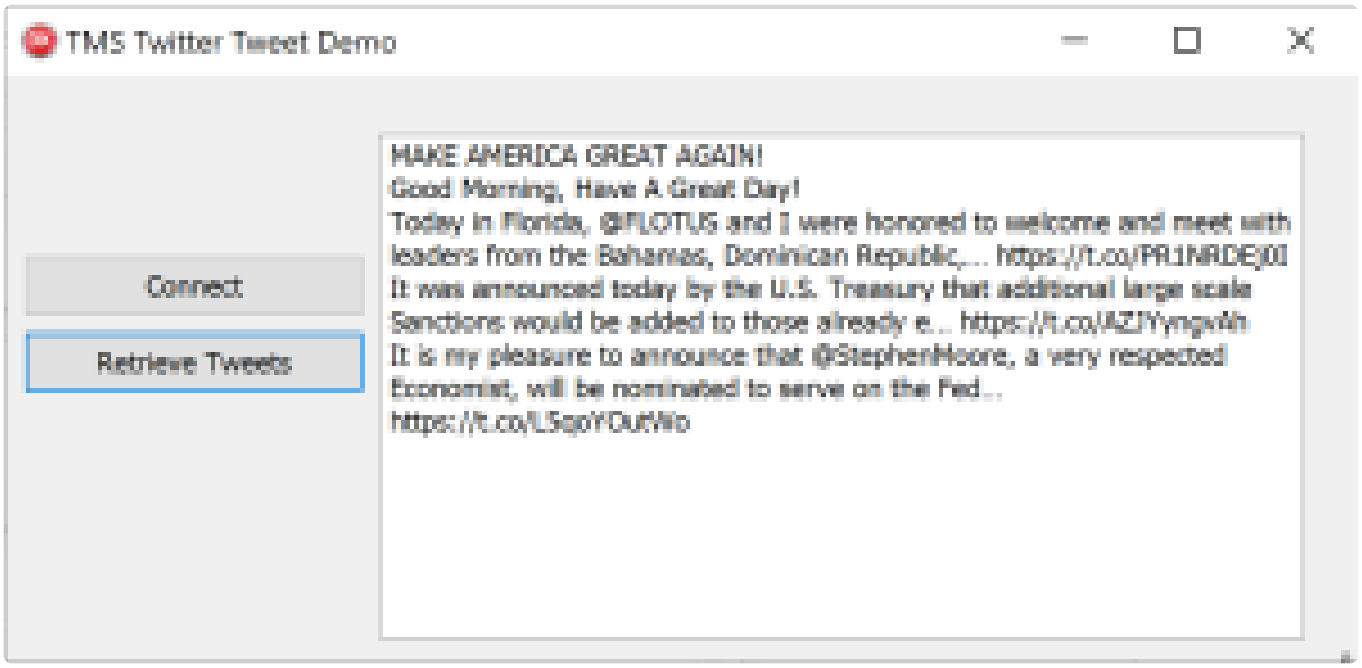
The code to retrieve the tweets could not be any simpler:

```
procedure TForm1.btnRetrieveClick(Sender: TObject);
begin
  Memo1.Clear;
  Twitter.Statuses.Clear;
  Twitter.GetStatuses(5, -1, -1, -1, 'realDonaldTrump' );
  begin
    for var i : Integer := 0 to Twitter.Statuses.Count -1 do
    begin
      var LStatus : TTWitterStatus;

      LStatus := Twitter.Statuses[i];
      Memo1.Lines.Add( LStatus.Text );
    end;
  end;
end;
```

We clear the memo and also clear any previously retrieved tweets. Tweets are called "status changes" in the TMS documentation and thus the property to access the tweets is a list of `TTwitterStatus`. In order to fill the list with data you invoke `Twitter.GetStatuses`. The method name is chosen a bit unlucky, as the plural of 'status' is also 'status', but code-completion will assist us there. You can specify how many tweets you want to retrieve, starting with the most-recent. Then you have different parameters to retrieve tweets by date and numerical user id. I decided to use the Twitter handle to specify the feed I want to retrieve. This could well be another TEdit control on the form to make it variable.

After retrieving all the tweets, we simply iterate the list and add the tweets to the memo.



Screenshot

Above you see the code output showing the last 5 tweets from the President of the United States on March 24th, 2018.

Try connecting to Twitter a second time after reopening the application. You will not need to provide your credentials again. This shows the token mechanism works properly as well.

INDEPENDENT · AD-FREE · READER-SUPPORTED

Free to read. Not free to make.

Every tutorial, article, and line of example code on Holger's Code is published without a paywall and without ads. This page is about what keeps it that way.

Professional work, published free

The content here is held to the same standard as professional client work: every tutorial is built as a real project, tested against current versions, documented step by step, and revisited when the frameworks move on. A single in-depth tutorial routinely takes days — and that's before the bills behind it: servers, storage, build infrastructure, licenses, and the hardware everything is verified on.

If an article or tutorial here has saved you an afternoon of debugging, taught you a technique, or helped you ship, it created real value for you. Supporting it isn't charity — it's paying for professional work you found useful, at whatever price you choose. And if you can't, keep reading anyway. That's what the free part means.

Ways to support

KO-FI

Buy me a coffee

A one-time tip — no account, no subscription, any amount.

BOOKS & COURSES

Buy the paid work

The paid tier of everything published free here — and you get something lasting in return.

SPREAD THE WORD

Share an article

Send a tutorial to a colleague, cite an article, subscribe to the RSS feed.

Nothing here goes behind a paywall, whether you chip in or not. But if something on this site earned its keep today, this is where to say so.

Support on Ko-fi —