

Paper 304-2013

Running SAS® programs using Skype

Romain Miralles, Genomic Health

ABSTRACT

Skype is a well-known method used to talk to friends, family members and co-workers. It is one of the best applications available to make voice calls over the Internet. In this paper we will present a new, innovative way to use SAS with Skype.

Here, we have developed a solution that allows users to run SAS remotely through Skype. After installing the DLL from the API on the application website, programmers can create scripts to control Skype. By sending a specific message to a pre-defined user, programmers can execute SAS on demand.

This paper will explain how to use Skype to run SAS programs. It will describe the installation and configuration of the DLL, provide the VB script needed to communicate with Skype, and illustrate a real-case scenario in which this technique is used.

INTRODUCTION

The Skype solution described in this paper was developed to resolve an issue that came up while implementing a new SAS application for a clinical trial. The application had to connect to an Oracle database to download the data in SAS format. However, this database could not be accessed directly from the server hosting the application; only computers situated in a specific location could connect to it. Therefore, we have developed a solution using SAS programs, VB scripts, and Skype to communicate between the two locations, and automatically run SAS programs.

SKYPE AND SKYPE API

Skype is a software that allows users to communicate with peers by voice, video and instant messaging over the internet. It has become a very popular tool, with more than 600 million registered users. Skype is also a very interesting application for developers. Indeed, the company behind this software released the Skype Application Programming Interface (API) for both experienced and novice programmers. The public API enables third party applications to communicate with Skype.

This API is a text based request/response interface contained inside the Skype client program. External programs that want to use the API need to be programmed in C++, which is not the easiest language to learn. Fortunately, for programmers, free products like SKYPE4COM allow developers to write programs in other programming languages such as Visual Basic, PHP, Java or Perl. SKYPE4COM is a wrapper that represents the Skype API as objects with properties, commands, events and notifications. Using SKYPE4COM, it becomes possible for SAS programmers to write scripts to communicate with Skype.

In this paper, we used Visual Basic, a powerful and relatively easy language to learn, but we could have developed the same thing in Java, PHP or any other languages compatible with the API.

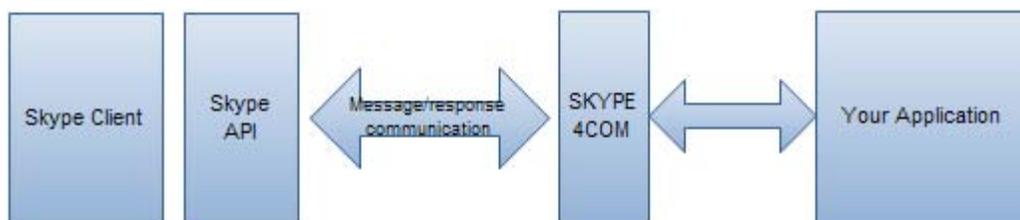


Figure 1. SKYPE4COM is an interface which represents the Skype API as objects.

INSTALLING AND CONFIGURING SKYPE TO RUN SAS PROGRAMS

Skype can be downloaded from the official Skype website and installed easily on any computer. After downloading and installing the application, a new account needs to be created.

The SKYPE4COM wrapper comes together with normal Windows Skype and gets installed in C:\Program Files\Common Files\Skype. Depending on your operating system and Skype version, you may need to undertake some additional steps in order to be able to use this shared DLL. It is not possible to describe all of the possible scenarios in this paper but Skype has a dynamic developer community where it is easy to get help.

RUNNING SAS PROGRAMS USING SKYPE: A SIMPLE EXAMPLE

In this section, we will use a simple example to illustrate the use of Skype to run SAS programs. The next section will describe a more complex real case scenario.

The idea is to run SAS programs on demand using Skype as an interface. The first step is to install Skype and SKYPE4COM on the machine where the SAS programs are located. We then need to add a VB script that will create the link between the Skype application and the programs. As explained previously, we chose Visual Basic as our programming language to interact with the public API, and wrote a script to listen and interpret incoming text messages on Skype. The script runs permanently in the background and triggers the SAS program when specific messages are received from a predetermined user.

DIAGRAM

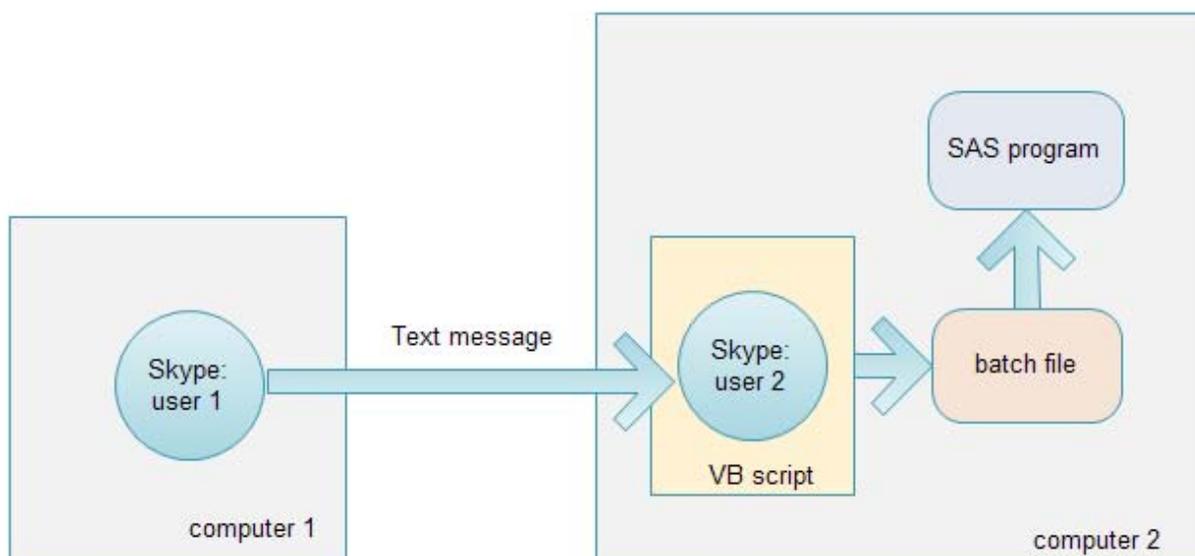


Figure 2. Simple example

On computer 2, we installed Skype, SKYPE4COM, a VB script (described below), and started the Skype session for the user as “user 2”. The SAS programs and the batch files are located on the server in a specific folder.

The VB script allows Skype to listen and interpret incoming messages from user “user 1” (we specified the username in the VB script).

On computer 1, we just installed Skype and logged in with the user “user 1”.

When “user 1” sends a specific text message to “user 2”, the VB script installed on computer 2 automatically interprets the message, and calls the corresponding batch file.

VB SCRIPT

The VB script is the key part of the system and SAS programmers should find it relatively easy to understand. Its role is to monitor all incoming text messages and trigger the corresponding SAS programs. The code below needs to be modified to work in your environment (program names and path) and then saved as a text file with the extension ‘.vbs’. It is activated by double-clicking on it.

```

DIM oSkype
CONST my_friend = "user1"
CONST cmsReceived = 2

'initialize_Skype
SET oSkype = WScript.CreateObject("Skype4COM.Skype","oSkype_") 'Create Skype4COM
object

IF NOT oSkype.Client.IsRunning 'Start Skype Client
    THEN oSkype.Client.Start
END IF

oSkype.Attach 'Connect to the Skype API

DO WHILE TRUE ' endless loop
    WScript.Sleep(60000)
LOOP

PUBLIC SUB oSkype_MessageStatus(BYVAL pMessage , BYVAL Status )
    DIM st' As String
    DIM stMess' As String
    DIM oFso 'As fileSystemObject
    DIM shell
    DIM fso
    DIM ObjStream
    CONST TristateFalse = 0
    CONST file_location = "C:\SASfiles\log\" 'location to save log files
    CONST file_name = "listen.log" 'name of the log file

    IF pMessage.FromHandle = my_friend AND Status = cmsReceived THEN 'If message
    is received from my_friend
    SELECT CASE pMessage.Body 'Get the message

CASE "Run program 1" 'If text message is equal to 'Run program 1' then it will
run the code below
    SET shell=createobject("wscript.shell")
    shell.run "c:\SASfiles\runpg1.bat" 'Run batch file
    SET shell=NOTHING
    SET Fso = CreateObject("Scripting.FileSystemObject")
    SET objStream = FSO.CreateTextFile(file_location & file_name, TRUE,
    TristateFalse) 'Write to the log file
    WITH ObjStream
        .WriteLine "Program 1 ran successfully."
        .Close
    END WITH

CASE "Run program 2" '2nd case
    SET shell=createobject("wscript.shell")
    shell.run "c:\SASfiles\runpg2.bat" 'Run batch file
    SET shell=NOTHING
    SET Fso = CreateObject("Scripting.FileSystemObject")
    SET objStream = FSO.CreateTextFile(file_location & file_name, TRUE,
    TristateFalse)
    WITH ObjStream
        .WriteLine "Program 2 ran successfully."
        .Close
    END WITH

CASE "Run program 3"
    SET shell=createobject("wscript.shell")

```

```

        shell.run "c:\SASfiles\runpg3.bat" 'Run batch file
SET shell=NOTHING
SET Fso = CreateObject("Scripting.FileSystemObject")
SET objStream = FSO.CreateTextFile(file_location & file_name, TRUE,
TristateFalse)
WITH ObjStream
    .WriteLine "Program 3 ran successfully."
    .Close
END WITH
END SELECT
END IF
END SUB

Function to initialize Skype
SUB init_Skype()
SET oSkype = WScript.CreateObject("Skype4COM.Skype","oSkype_")
IF NOT oSkype.Client.IsRunning THEN
    oSkype.Client.Start
END IF
WHILE NOT oSkype.Client.IsRunning
    WScript.Sleep(30000)
WEND
oSkype.Attach
END SUB

```

There are a few important things to understand about this code:

- 1: We specified the Skype user who will have the right to run the SAS programs. Any messages received from any other users will not trigger the programs.
- 2: The SKYPE4COM wrapper is used to create an object.
- 3: We defined a loop at the beginning of the program. Skype will run an infinite loop to ensure that the connection remains active, as we want to be able to run the SAS programs at any time.
- 4: We specified the text message that will trigger the batch files. It can be either a word or a sentence.
- 5: We can define multiple Skype messages in the script. Each of them will trigger a different batch file (and consequently a different SAS program).
- 6: The script can write in a log file if necessary

USING SKYPE TO RUN SAS PROGRAMS: REAL CASE SCENARIO

We were working on a new web-based application to manage queries for a clinical trial. The application was hosted internally on a dedicated server and ran SAS programs to generate a list of queries using the clinical database as an input. Unfortunately, this clinical database was hosted by a vendor at a different location and, to make the situation even more difficult, it was not possible to connect directly to the vendor's server to download the data. Indeed, only a computer with an IP address from a specific location could connect to the vendor database. This situation led us to ask how to automatically download the data and run the SAS programs for the web application. We wanted something fully automated that did not require any user intervention.

Different solutions were considered, including the use of Microsoft Outlook or Skype.

DIAGRAM

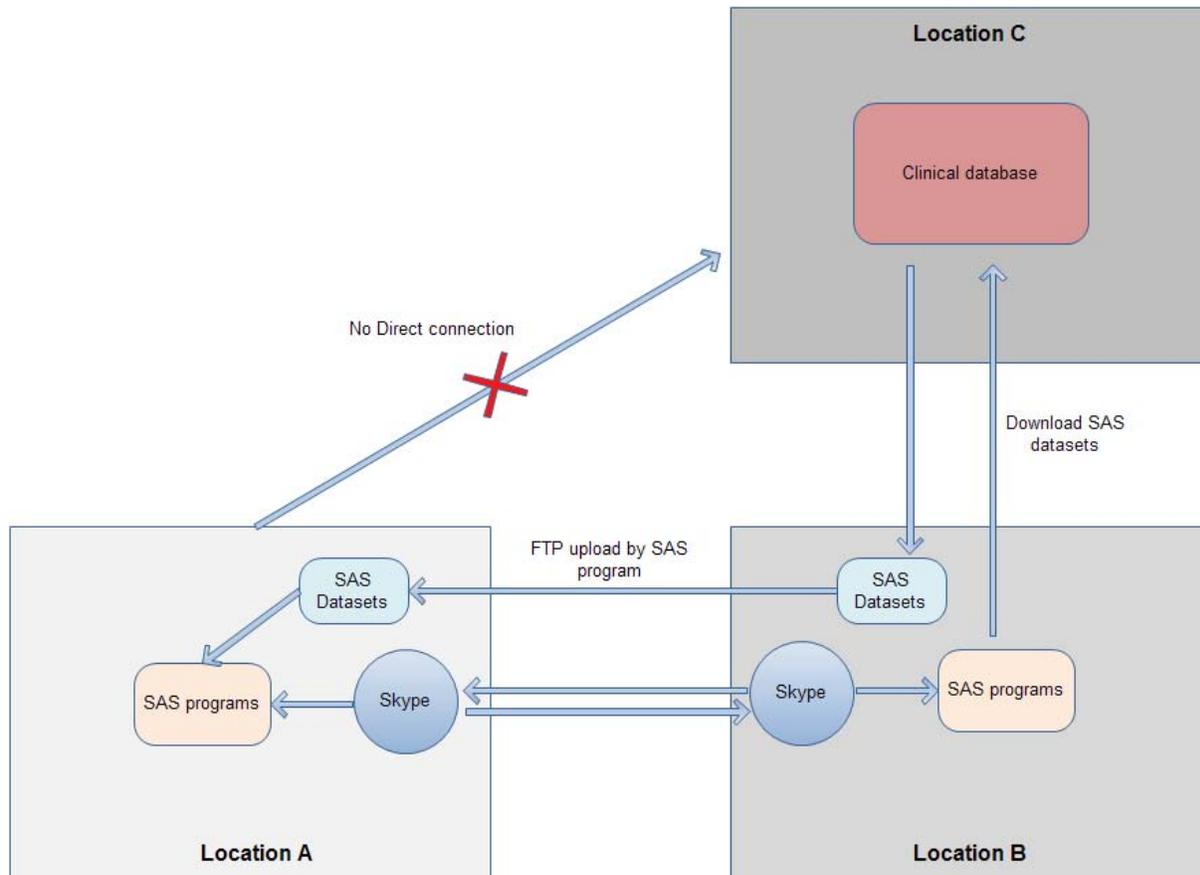


Figure 3. Real case scenario

The system is completely automatic and runs every day without any manual effort. It is started by a scheduled task in Windows and involves multiple steps, with programs and data located at three different sites (locations A, B and C in the diagram).

The clinical database for the study is located at site C, where only computers with an IP address from site B can access it. However, the SAS-based web application is located at site A and consequently has no direct access to the database.

Skype is running on both computers at sites A and B. A Skype message is sent from site A to site B, where the message is then interpreted by the VB script and triggers a SAS program. The program connects to the Oracle database at site C, and downloads the data in SAS format. After downloading the tables, another SAS program uploads the files on FTP and, at the same time, a Skype message is sent from site B to site A. Next, a VB script on the computer at site A reads the Skype message and executes the corresponding SAS program that will download the datasets from the FTP server.

For this project we used the VB script described in the previous section with the only difference being that the script was installed on both computers for bidirectional communication. Another VB script was also used to automatically send a Skype text message from user 1 to user 2 and user 2 to user 1. This script is triggered by a scheduled task and sends a text message to the Skype user defined in the code.

VB script to send a Skype message:

```
' // Create a Skype4COM object:
Set oSkype = WScript.CreateObject("Skype4COM.Skype")
```

```

DIM fso
DIM ObjStream
CONST TristateFalse = 0
CONST file_location = "C:\program\log\" 'location for the log file
CONST file_name = "script.log" 'name of the log file
SET Fso = CreateObject("Scripting.FileSystemObject")
SET objStream = FSO.CreateTextFile(file_location & file_name, TRUE, TristateFalse)

' // Create a UserCollection object and add the multichat users to this collection:
SET oMembers = WScript.CreateObject("Skype4COM.UserCollection")
oMembers.Add(oSkype.User("user1"))

' // check if user is online
lblOnlineStatus = oSkype.User("skpe_user1").OnlineStatus

' // Create a multichat:
SET oChat = oSkype.CreateChatMultiple(oMembers)

' // Open a chat window:
oChat.OpenWindow()

' // Send a chat message:
IF lblOnlineStatus=1 THEN
    SET oMsg = oChat.SendMessage("Run program 1")
    WITH ObjStream
        .WriteLine ""
        .Close
    END WITH
END IF
IF lblOnlineStatus=0 THEN
    SET oMsg = oChat.SendMessage ("User offline")
    WITH ObjStream
        .WriteLine "ERROR: User is offline."
        .Close
    END WITH
END IF

```

CONCLUSION

We knew how to use Skype to talk to our family and friends, and we now know how Skype can run SAS programs. It might seem unusual to use this software with SAS, but it was actually a very helpful solution to complete the project described previously.

It was also an interesting challenge and a great opportunity to learn new skills and techniques. Can this solution be used for other projects? Certainly, as it works perfectly well in this context and the VB script code is available in this paper. However, it might not always be possible to use it at work as some companies block software such as Skype. In any case, it is still a fun technique to try out and play with.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Name: Romain Miralles
 Enterprise: Genomic Health
 Address: Penobscot Drive
 City, State ZIP: Redwood City, CA 94063
 E-mail: mrom34@gmail.com

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are trademarks of their respective companies.