



Teams Wizard v0.5

by LyncWizard.com



Configuration Note: Contact Lookup External App

Date	Version	Remarks
2023-03-26	1.0	Initial Version

Configure External App for Contact Lookup

Overview

Contact Lookup External App allows you to attach Teams Wizard to almost all data source for Reverse Phone Number Lookup, like:

- RESTful
- SOAP
- LDAP
- SQL
- ...

If a contact lookup rule towards the External App is matching, Teams Wizard starts configured External App with 2 arguments:

- PhoneNumber
- File path to store search result as JSON

After External App is executed or configured timeout expired, Teams Wizard reads content of dynamic file path. If this file contains a valid JSON record, result will be presented.

External App can be any executable App, which does not need a user interaction to run, like:

- PowerShell Script
- Command Line
- EXE
- ...

To develop such an External App some developer knowledge is required. A sample PowerShell script, which executes a RESTful request is delivered with Teams Wizard.

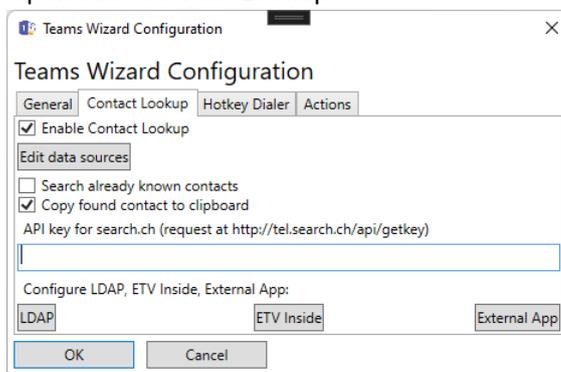
With this concept, you are no longer dependent on what methods Teams Wizard offers for Contact Lookup to attach your own data source.

External App is executed in context of Windows signed-in user.

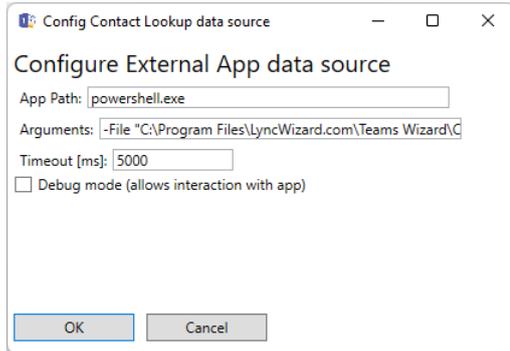
Configure External App

To configure Contact Lookup External App, open Teams Wizard Configuration with a right click to Systray Icon and choosing "Configuration".

1. Open Tab "Contact Lookup"



2. Click to “External App” button. A new window opens:



3. Configure following parameters:

a. App Path

- i. Full path to executable app.
- ii. Example: `powershell.exe`
- iii. For PowerShell scripts, add `"powershell.exe"`
- iv. For Command Line (Batch Files), enter `"cmd.exe"`.
- v. For VB- or Windows Scripts, use `CScript.exe`
- vi. If you have an own EXE file, enter full path like `"C:\Program Files\CustomTool\Lookup.exe"`

b. Arguments:

- i. Add all required parameters to run script or executable. There are 2 strings which will be replaced by specific values when app is started by Teams Wizard:
 1. `<PhoneNumber>`: This string will be replaced by phone number. Format matches to regex result configured in “Replaces” column of Contact Lookup search rules.
 2. `<OutputFile>`: This string is replaced by a dynamic prepared temp file, to which the external app must write result as JSON record.
- ii. Example: `-File "C:\ProgramData\LyncWizard.com\Teams Wizard>ContactLookupExternalApp\Get-ContactForTeamsWizard_Sample.ps1" -PhoneNumber "<PhoneNumber>" -OutputFile "<OutputFile>"`
- iii. In case of PowerShell script, please make sure, PowerShell Execution Policy allows running script (reference: <https://learn.microsoft.com/en-us/powershell/module/microsoft.powershell.security/set-executionpolicy>)

c. Timeout in milliseconds:

- i. How long Teams Wizard should wait for external app to finish. In case it is not finished, Teams Wizard kills process of external app and continues as it would have normal finished (see exceptions if Debug mode is enabled).

d. Debug mode:

- i. For production, this check box should not be enabled. This makes sure, External App is started in background without user interaction.
- ii. Enabling Debug mode may help to develop, test and debug an External App.
- iii. If enabled:
 1. External App Window will be opened visible and user interaction is possible.
 2. Process will not be killed after timeout expires, but Teams Wizard continues anyway.

Sample

A sample PowerShell script is installed to "C:\ProgramData\LyncWizard.com\Teams Wizard\ContactLookupExternalApp\Get-ContactForTeamsWizard_Sample.ps1".

```
# Sample Script to query contacts from a Web Service for Teams Wizard Contact Lookup
#
# Please make sure, PowerShell Execution Policy allows execution of this Script:
# Set-ExecutionPolicy -ExecutionPolicy Unrestricted
#
# After script execution is finished, Teams Wizard reads content of $OutputFile and deletes this file.
# OutputFile must contain JSON. Following sample shows which values are used:
# Valid Address Types: 0 = Unknown, 1 = Person, 2 = Organization
# {
#   "Type": 2,
#   "Organisation": "LyncWizard.com",
#   "Title": null,
#   "Firstname": "Stephan",
#   "Name": "Kaufmann",
#   "AddressLine1": "Application Boulevard 123",
#   "ZIP": "LW95123",
#   "City": "Wizard Town",
#   "Country": "Switzerland"
#   "TelephoneNumber": "+41991234567",
#   "MobileNumber": "+41997654321",
#   "DataSource": "External App: Static Entry",
# }
#
# If Web Service needs Authentication or Authorization, add corresponding Headers or Credentials:
# - Basic Authentication by UserID and Password:
# $UserID = "myUsername"
# $Password = "my!Password1"
# $Credentials = [System.Management.Automation.PSCredential]::new($UserID,(ConvertTo-SecureString $Password
-AsPlainText -Force))
# $Contact = Invoke-RestMethod -Credential $Credentials -Method Get -Headers $Headers -Uri
"$($BaseURL)?PhoneNumber=%2B41991234567"
#
# - Authorization with a Bearer Token:
# $Headers = @{'Authorization'=("Bearer {0}" -f $Token)}
#
# 18.03.2023 StK v1.00 Initial Code

# Params
param (
    [Parameter(Mandatory=$true)][string]$PhoneNumber,
    [Parameter(Mandatory=$true)][string]$OutputFile
)

# Script
$headers = @{}
$BaseURL = "https://www.lyncwizard.com/lyncwizard/SearchContact.json"
$PhoneNumberEncoded = [System.Web.HttpUtility]::UrlEncode($PhoneNumber)
$Contact = Invoke-RestMethod -Method Get -Headers $headers -Uri
"$($BaseURL)?PhoneNumber=$($PhoneNumberEncoded)"

$Contact | ConvertTo-JSON -Depth 100 | Out-File -FilePath $OutputFile -Encoding oem
```

Default configuration for External App points to this script. This is only an example, do not use it for production.

This sample executes an unauthenticated RESTful GET request. It also includes some hints how to add authentication and authorization or changing GET to POST. With such a script, it is easy to realize a RESTful API request towards any internet based services with OAUTH2 or any other authorization methods as well as any local LDAP or SQL requests.

Search Result

Search result must be stored in file defined in parameter <OutputFile>. Content must be a JSON record with following definition:

```
{
  "Type": "int",
  "Organisation": "string",
  "Title": "string",
  "Firstname": "string",
  "Name": "string",
  "AddressLine1": "string",
  "AddressLine2": "string",
```

```

"ZIP": "string",
"City": "string",
"Country": "string"
"TelephoneNumber": "string",
"MobileNumber": "string",
"CompleteAddress": "string",
"DataSource": "string",
}

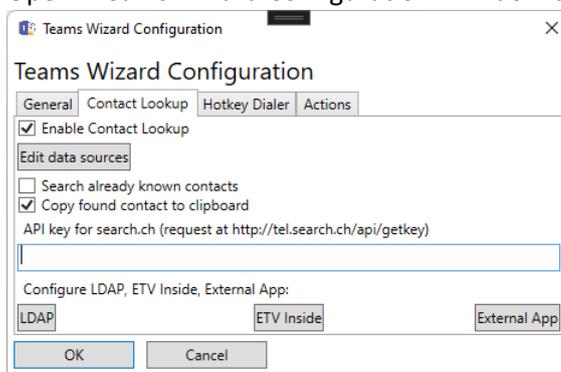
```

Key	Type	Value
Type	Integer	Type of address: 0 = undefined 1 = Person Organization
Organization	String	Name of organization, null or empty if not used/known
Title	String	Title of person, null or empty if not used/known
Firstname	String	Given name/first name of person, null if not used/known
Name	String	Surname of person, null or empty if not used/known.
AddressLine1	String	Postal address line 1, null or empty if not used/known
AddressLine2	String	Postal address line 2, null or empty if not used/known
ZIP	String	Postal code, null or empty if not used/known
City	String	City, null or empty if not used/known
Country	String	Country, null if not used/known
TelephoneNumber	String	Found phone number, null or empty if not used/known
MobileNumber	String	Found mobile number, null or empty if not used/known
CompleteAddress	String	Multi Line string containing full address block, if contact information are not available with specific fields. Null or empty if not used. Do not mix CompleteAddress with other specific fields except Type and DataSource
DataSource	String	Reference to used data base to fulfill copyright/term of use conditions. Some public reachable contact data bases require to present source of displayed result. Null or empty if not required.

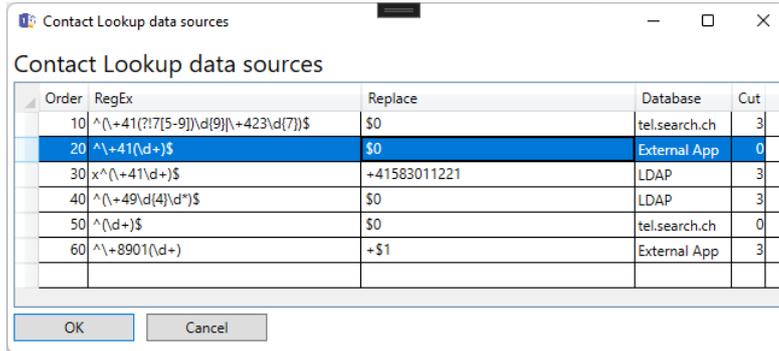
Configure Contact Lookup Rules

If External App is configured correctly, you can add this external app as data source in contact lookup.

1. Open "Teams Wizard Configuration" window tab "Contact Lookup":



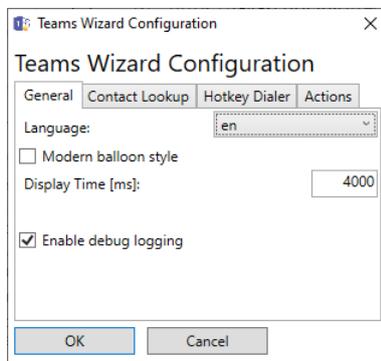
2. Click to “Edit Data sources”. A new window opens:



3. Define a rule with data base “External App”
 - a. Order: Value defines order of execution (from lowest to highest). First match stops execution.
 - b. RegEx: `^\(+41\d+$` = matches to all Swiss phone numbers (leading +41 followed by one or more digits). If you enter an invalid RegEx, you get a red marking.
 - c. Replace: `$0` = Full phone number, `$1`=first group (parenthesis), `$2`=second group, etc.
 - d. Data base: “External App”
 - e. Cut: In case no results can be found, how many times search can be repeated with removing a digit at the end of the phone number. For External App, 0 is preferred value, because this behavior can be developed more specific inside your External App.

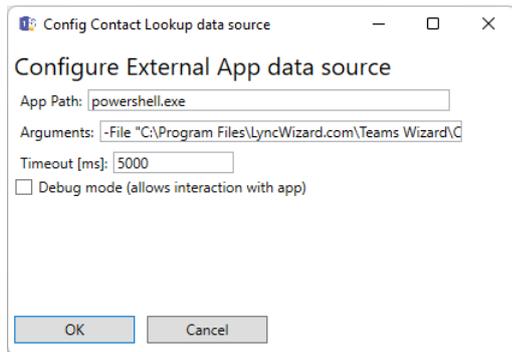
Testing and Debugging

To check overall function, activate debug logging on the “General” tab, and restart Teams Wizard to activate debug logging:

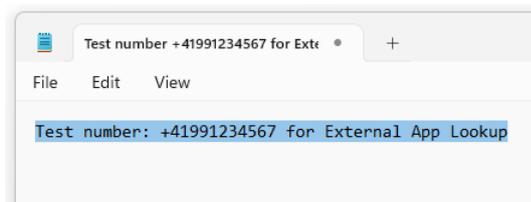


Teams Wizard creates in your temp Directory “C:\Users\\AppData\Local\Temp” a file “TeamsWizard.log” with a lot of useful information for analyzing. You can open it with any text editors like Notepad.

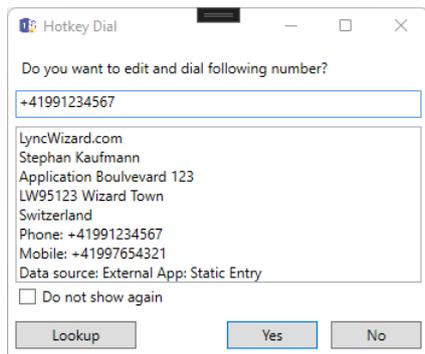
Enable Debugging mode on External App Configuration, to see window and content of External App. Don’t to forget to disable “Debug Mode” after testing.



For testing your External App, you can use Hotkey feature of Teams Wizard instead of making incoming calls. Just write phone number to a text file.



Highlight this number and press hotkey (default [F8]). Hotkey window opens. Click to "Lookup". Now contact lookup rules will be executed similar to an incoming call. Result will be presented in hotkey window result area instead of notification window.



In %Temp%\TeamsWizard.log you find some more details to verify and validate execution:

```

11302 3/26/2023 12:59:43 PM Durchgang :3 Anzahl:0
11303 3/26/2023 12:59:57 PM ContactLookupExternalApp: CmdLine: powershell.exe, Arguments: -File "C:\ProgramData\LyncWi
11304 3/26/2023 1:00:00 PM ContactLookupExternalApp: Search result: {
11305     "City": "Wizard Town",
11306     "Type": 2,
11307     "DataSource": "External App: Static Entry",
11308     "Firstname": "Stephan",
11309     "AddressLine1": "Application Boulevard 123",
11310     "ZIP": "LW95123",
11311     "Organisation": "LyncWizard.com",
11312     "Name": "Kaufmann",
11313     "TelephoneNumber": "+41991234567",
11314     "MobileNumber": "+41997654321",
11315     "Title": null,
11316     "Country": "Switzerland"
11317 }

```