LINX

Getting started with Xero and Linx

linx.software

@LinxCode

Contents:

This guide provides step-by-step instruction to set up Linx and integrate it with the Xero API.

Integrating Xero and Linx will create a powerful platform for custom development, providing you with the right tool to create systems and processes that work in a seamless and harmonious way to enhance the user experience.

The plugin handles all the intricate details involved in working with the Xero API including feature-rich data integration between Xero and virtually any other database or application.

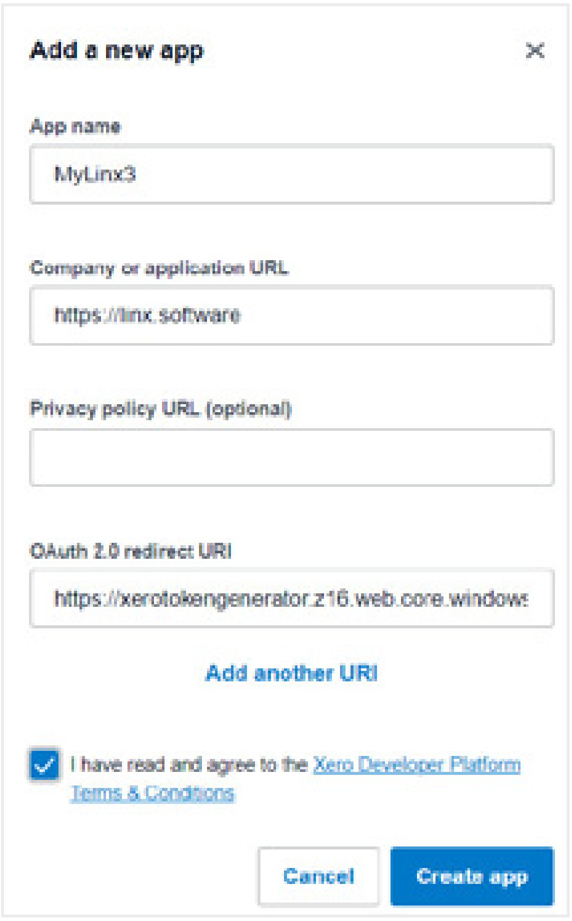
Disclaimer

These details are accurate at the time of writing this Help file. However, from time to time Xero may make further changes that are unknown to Linx.

Part 1: Creating a Xero App

- If you don't have an active Xero account, go to xero.com to register. Login to the Xero Developer portal: https://api.xero.com
- Go to the *My Apps* section, click *New App*.
- Provide the following details:
  - App name
  - Company or application URL
  - Privacy policy URL (optional)
  - OAuth 2.0 redirect URI
    - Use: https://xerotokengenerator.z16.web.core.windows.net/auth This redirect URI will enable you to use our Xero Token Generator to create a Refresh Token (which is required for connecting Linx to your Xero app – covered in Part 2, below).
- Click *Create app*.
- Once complete, your generated *Client Id* will be displayed.
- Click on *Generate a secret*.
- Both the *Client Id* and *Client Secret* values will be used later when you create your Linx solution (see Part 2, below). You can copy these values now, or return to your Xero app at any time to get these values.

Note:
Xero apps created after December 2019 use *OAuth2* authentication, as per the steps outlined on the left.


Add a new app

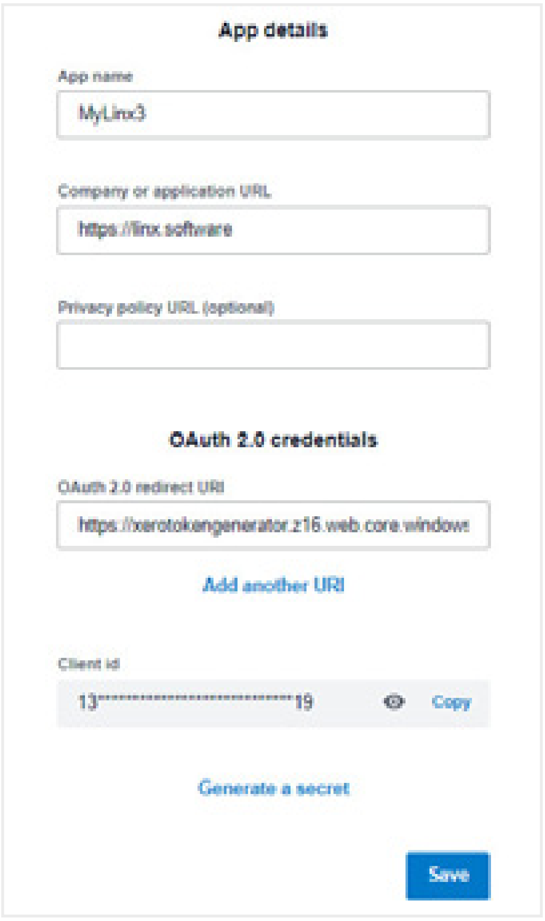
App name: MyLinx3

Company or application URL: https://linx.software

Privacy policy URL (optional):

OAuth 2.0 redirect URI: https://xerotokengenerator.z16.web.core.windows

Add another URI
I have read and agree to the Xero Developer Platform Terms & Conditions
Cancel
Create app


App details

App name: MyLinx3

Company or application URL: https://linx.software

Privacy policy URL (optional):

OAuth 2.0 credentials

OAuth 2.0 redirect URI: https://xerotokengenerator.z16.web.core.windows

Client id: 13**************************19

Part 2: Example Linx Process

To illustrate the power of Linx, we've created an example solution for Acme Co. This process will be automated and deployed to a server to run unattended.

In this example, we will:

- Integrate Linx with Xero.
- Pull and write the name and account number for Xero Customers to a csv file.
- Attach the text file to an email.
- Send the email to a specified address, at a specified time, on weekdays only.
- Deploy the solution to the Linx Server to automate this process.

Before you start...
This example requires that you have Contacts that are Customers in your instance of Xero. If not, you may create a demo company in Xero and add the appropriate contacts to complete the exercise.

2.1: Setting Up Linx

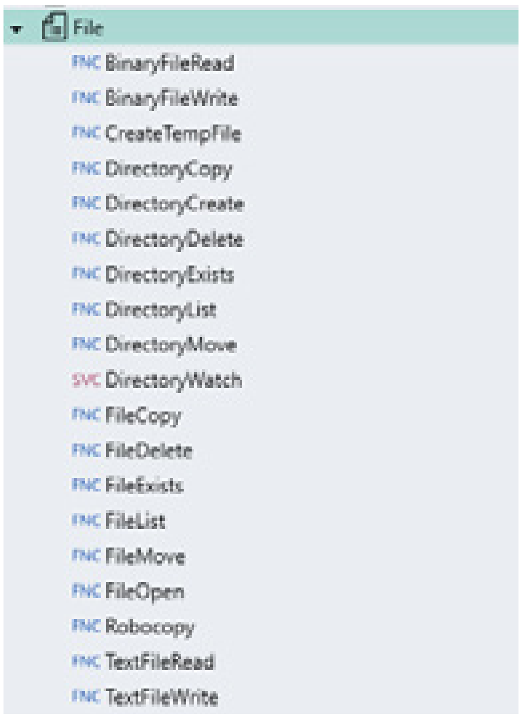
- Open the Linx Designer
- From the start screen select *New Solution* or alternatively start from the Ribbon bar by selecting *File –> New –> Solution*.
- Rename your new solution “XeroExample” by clicking on the field in the Properties panel (bottom left).
- Rename the project to “XeroCustomerDetailsToTextFile”.
- Rename the process (currently called “Process”) to “GetXeroCustomerContacts”.
- Open *Settings* in the ribbon bar and add the output file location and name so that you can refer to it later and not have to specify the physical path for each function where it will be used:
  - Name: FilePath
  - Type: String
  - Value: C:\Temp\CustomerExtract.csv
  - The values are automatically saved when you close the tab.

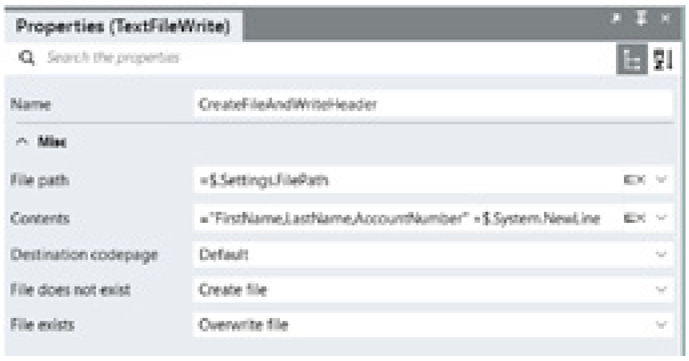
2.2: Build Your Process

Using the TextFileWrite function

- Click the *Add Plugins* button, then click the File plugin’s *Add* button.
- From the Plugin List on the right-hand panel, expand the File Plugin and select *TextFileWrite*.
- Drag this function onto the middle canvas — your design area.
- Update the properties of the Function in the Properties panel (bottom left):
  - Name: CreateFileAndWriteHeader
  - * - File path: =$.Settings.FilePath
  - Contents: =”First Name, Last Name, Account Number” + $.System.NewLine
  - Destination codepage: Default
  - File does not exist: Create file
  - File exists: Overwrite file

* As we previously set this earlier, you can either add this text directly into the field or use the built-in dropdown menu to select this value.




Properties (TextFileWrite)

Search the properties

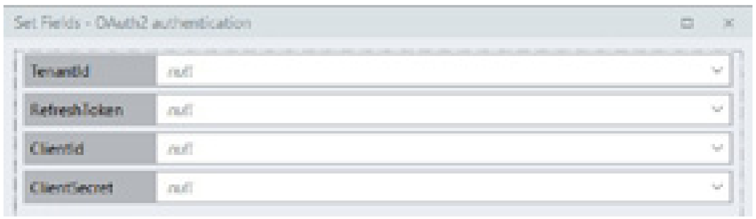
Name	CreateFileAndWriteHeader
^ Misc	
File path	=$.Settings.FilePath
Contents	="FirstName,LastName,AccountNumber" +$.System.NewLine
Destination codepage	Default
File does not exist	Create file
File exists	Overwrite file

2.3: Connect To Xero and Fetch Data

- From the Plugin list in the right-hand panel, expand the Xero Plugin to view a list of sub-sections. Expand the *Contacts* section.
- Select the *GetContacts* function (in the *Contacts* section) and drag it to the middle canvas below *CreateFileAndWriteHeader*.
- Update the properties of the function in the Properties panel (bottom left)
- The values you enter depend on your Xero app's authentication type:
  - for apps created after December 2019, *OAuth2* is used for authentication (see Part 1, above)
  - for apps created before December 2019, *OAuth1* was used for authentication

Connecting to Xero apps using OAuth2

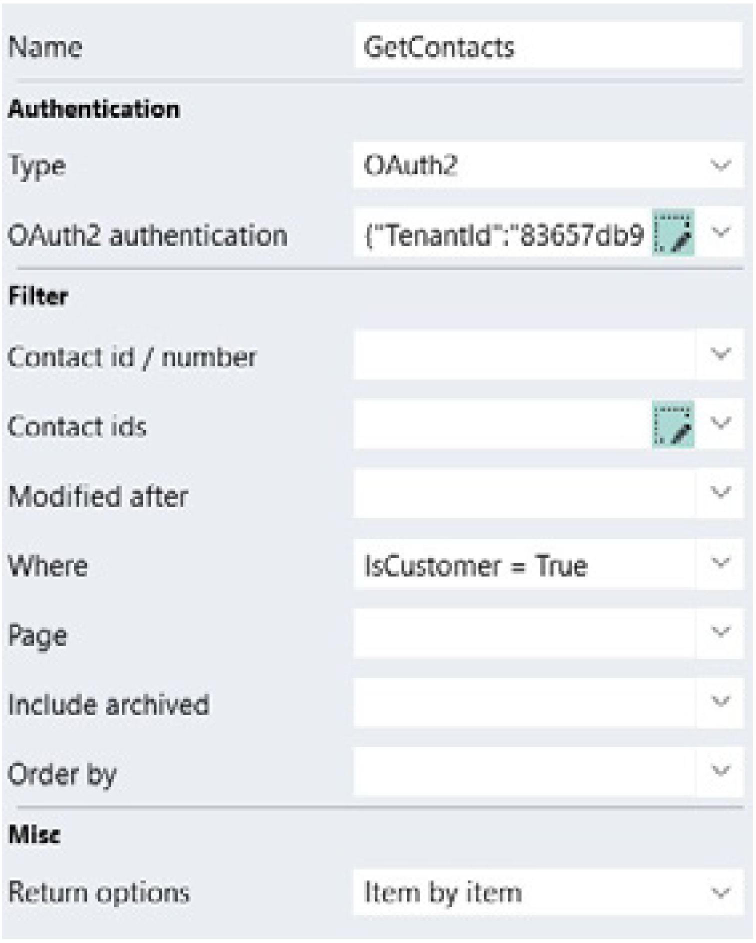
- Name: GetContacts
- Type: Select 'OAuth2'
- OAuth2 authentication: Click the icon:



Set Fields - OAuth2 authentication

TenantId	null
RefreshToken	null
ClientId	null
ClientSecret	null

- *ClientId* and *ClientSecret*:
  These values were generated when you created your Xero app – see Part 1, above.
- *RefreshToken* and *TenantId*:
  To generate these values, go to the Xero Token Generator and follow the steps:
  1. Select the scopes your Xero App requires.
  2. Provide the *Client Id* of your Xero App and connect to Xero. Xero will ask you to login and approve the connection and scopes.
  3. Provide the *Client Secret* of your Xero App and get the access tokens from Xero.
  4. The tenants connected to your Xero App are requested and displayed. Copy the relevant *TenantId* value.
- Where: IsCustomer = True
- Return options: Item by item



Name	GetContacts
Authentication	
Type	OAuth2
OAuth2 authentication	{"TenantId":"83657db9
Filter	
Contact id / number	
Contact ids	
Modified after	
Where	IsCustomer = True
Page	
Include archived	
Order by	
Misc	
Return options	Item by item

Connecting to Xero apps using OAuth1

- Name: GetContacts
- Type: Select ‘OAuth1’
- Authentication: Click the icon:
  - *CertificatePath*: Select the location where you created your SSL certificate. This should be a pfx file (e.g. C:\OpenSSL-Win64\bin\public_privatekey.pfx).
  - *CertificatePassword*: The password you specified for your SSL certificate.
  - *ConsumerKey* and *ConsumerSecret*: These values are for the Xero application that you added on the Xero Developer Portal.
- Where: IsCustomer = True
- Return options: Item by item

Using the TextFileWrite function to write records

- From the Plugin list on the right-hand panel, expand the File Plugin and select the *TextFileWrite* Function.
- Drag the function onto the middle canvas and place it under the *Loop* Function

About Process Flow: Loops and Branches

When adding Functions that have their own branch/loop, Linx indents such Functions on their own below your main process as a placeholder. This is because these Functions create their own scope and results. A user can then select another function to work with the data returned from the Loop/Branch.

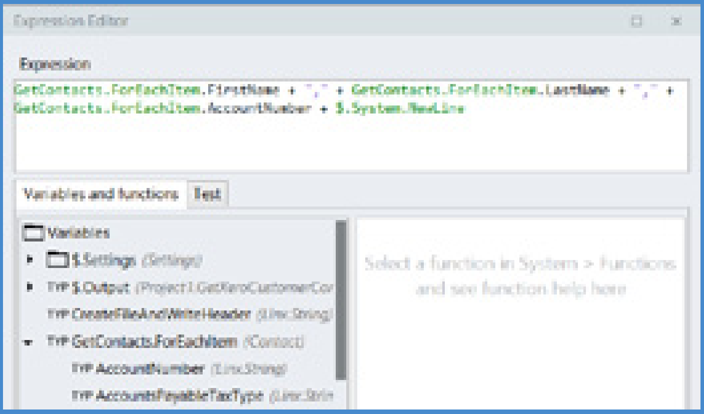
Example functions for this include: *ifElse*, doWhile, FileRead, TryCatch** and *ExecuteSQL*.
The *Return Option* property of components allow you to decide whether you want to receive a complete list as the return to use later (List of Items), or immediately loop through these items (Item by Item).

* These functions create a branch as opposed to a loop, yet have the same layout and effect as the other functions in this group.

- Update the properties of the Function in the Properties panel (bottom left)
  - Name: CreateFileAndWriteBody
  - File path: =$.Settings.FilePath
  - Contents: =GetContacts.ForEachItem.FirstName + “,” + GetContacts.ForEachItem.LastName + “,” + GetContacts.ForEachItem.AccountNumber + $.System.NewLine

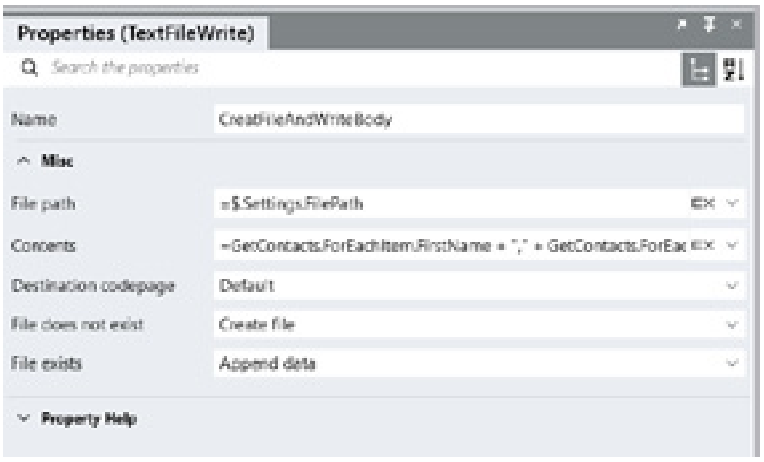
Note:
You can either add this text directly into this field or use the built-in Expression Editor to create the same. To launch the Expression Editor, simply click on the “EX” on the right-hand side of the entry field.

Additional reading: Using the **Expression Editor**


Expression Editor
Expression
Variables and functions
Test
Variables
$.Settings (Settings)
TYP $.Output (Project1.GetXeroCustomerCo
TYP CreateFileAndWriteHeader (Linx.String)
TYP GetContacts.ForEachItem (Contact)
TYP AccountNumber (Linx.String)
TYP AccountsPayableTaxType (Linx.Strin
Select a function in System > Functions and see function help here

```
GetContacts.ForEachItem.FirstName + "," + GetContacts.ForEachItem.LastName + "," +
GetContacts.ForEachItem.AccountNumber + $.System.NewLine
```

- Destination codepage: Default
- File does not exist: Create file
- File exists: Append data

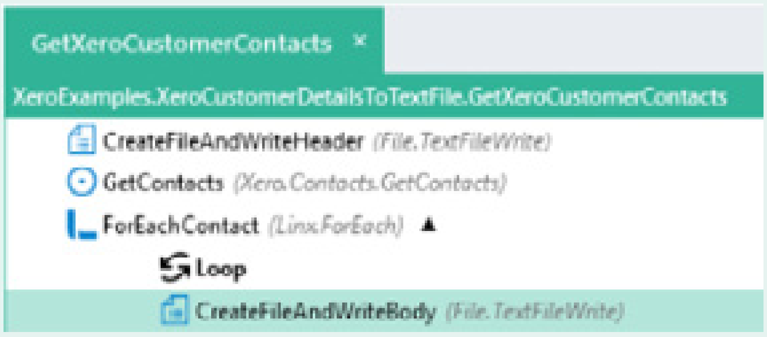


Name	CreatFileAndWriteBody
File path	=$.Settings.FilePath
Contents	=GetContacts.ForEachItem.FirstName + "," + GetContacts.ForEac
Destination codepage	Default
File does not exist	Create file
File exists	Append data

Process complete

Your completed process should look like this:

- In the ribbon bar click *Save* and save your solution to your local machine.


GetXeroCustomerContacts ×
XeroExamples.XeroCustomerDetailsToTextFile.GetXeroCustomerContacts
CreateFileAndWriteHeader (File.TextFileWrite)
GetContacts (Xero.Contacts.GetContacts)
ForEachContact (Linx.ForEach)
Loop
CreateFileAndWriteBody (File.TextFileWrite)

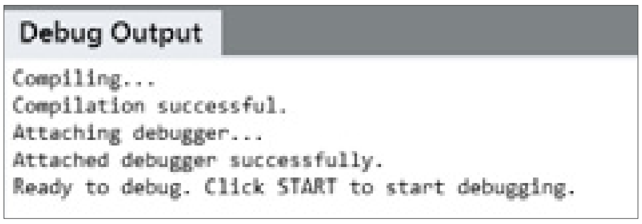
Debug for errors

The debug tool allows you to run a process while debugging. While debugging, you can see the values used in the execution of the process.

Using the Debug Tool:
When the debug tool is active, the application menu is replaced by the debug menu. The properties panel now becomes inactive as you cannot change the properties while debugging. Three other interface areas then also change to facilitate debugging. **Further reading.**

To allow you to step through a process to debug and test results, you first need to add breakpoints to your process. Click on the control you want to add your first breakpoint to (i.e. *CreateFileAndWriteHeader*), right click and select *Add breakpoint* or simply press F9. Now, when you debug, the process will stop execution on this control and wait for your command before continuing.

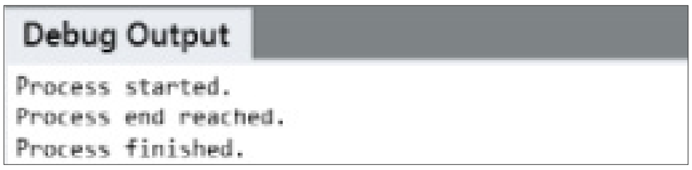
- Click *Debug* in the ribbon bar.
  Debug and allow Linx to compile and prepare your process to run.



```
Compiling...
Compilation successful.
Attaching debugger...
Attached debugger successfully.
Ready to debug. Click START to start debugging.
```

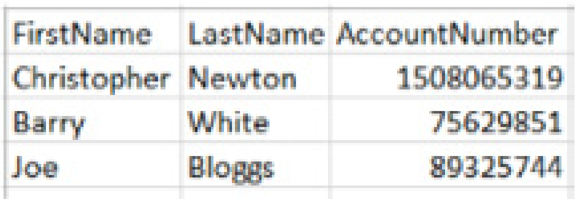
The process results

- After successful compilation click *Start*. The process should stop at the breakpoint you've created, and you can click *Resume* to step over to the rest of the process. A "Process finished" result signifies that process has run successfully.
- Click *Close* to exit the Debugger



```
Process started.
Process end reached.
Process finished.
```

- You can now check the contents of your file by navigating to the directory of the file path previously stipulated in your set up: **C:\Temp\CustomerExtract.csv**



FirstName	LastName	AccountNumber
Christopher	Newton	1508065319
Barry	White	75629851
Joe	Bloggs	89325744

2.4: Automating Your Process Using Services

The main advantage of Linx is the automation of your solutions (that you have built in the Designer). These solutions are deployed to the Server and run as a service, executing automated tasks that take place in the background.

The Linx Server is the key to this automation. The Server hosts and manages your Linx Solutions and web services. Once Solutions are added to the Server, you will run the process defined in your Solution – triggered by a one-time event or have unlimited repetitions based on the conditions defined in your solution.

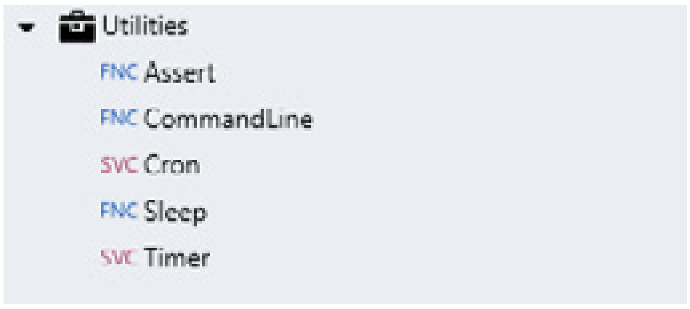
Setting up a Timer Service to send your results via email

A Timer Service allows the execution of a process at a specified time or interval. The user can specify a list of days on which the timer should run.

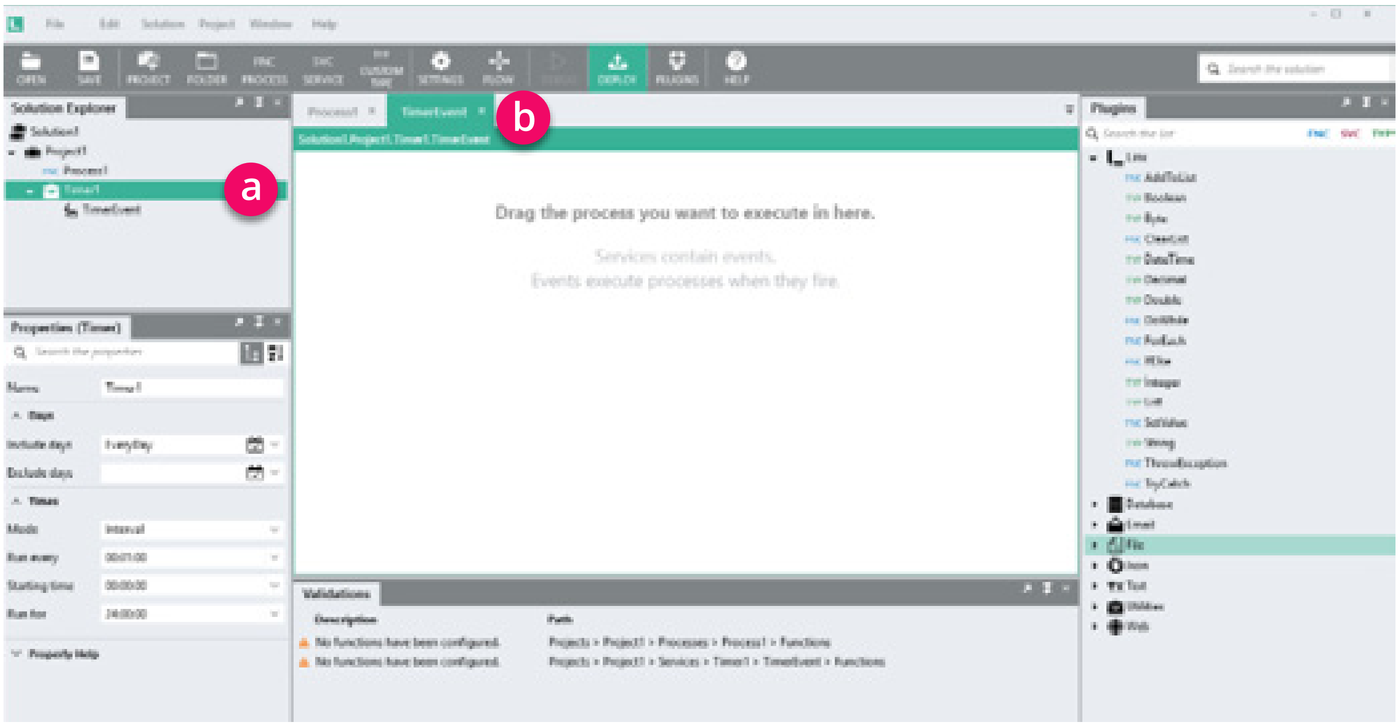
Services are the backbone of Linx and can be triggered on pre-defined conditions or system events and states. Find more information on Services and how to use them here.

Selecting a Timer service

- From the Plugin list on the right-hand panel, expand the Utilities Plugin and select the Timer service.
  Alternatively, select the Timer Service from the Service button in the ribbon bar.



- Once selected, Linx will
  a Create and place the new service in the Solution Explorer panel, and
  b Open a new tab in the middle canvas called "TimerEvent"



Solution1.Project1.Timer1.TimerEvent

Drag the process you want to execute in here.

Services contain events.
Events execute processes when they fire.

Properties (Timer)

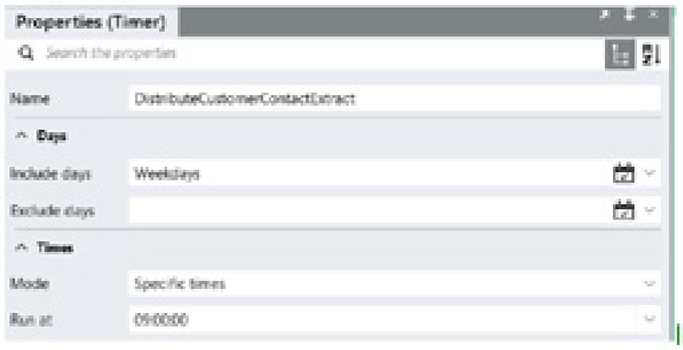
Name	Timer1
Days	
Include days	Everyday
Exclude days	
Times	
Mode	Interval
Run every	00:01:00
Starting time	00:00:00
Run for	24:00:00

Validations

Description	Path
No functions have been configured.	Projects > Project1 > Processes > Process1 > Functions
No functions have been configured.	Projects > Project1 > Services > Timer1 > TimerEvent > Functions

Setting the trigger

- Select *Timer1* in the Solutions Explorer panel and update the properties of the Timer:
  - Name: DistributeCustomerContactExtract
  - Include days: Weekdays
  - Mode: Specific Times
  - Run at: 09:00:00

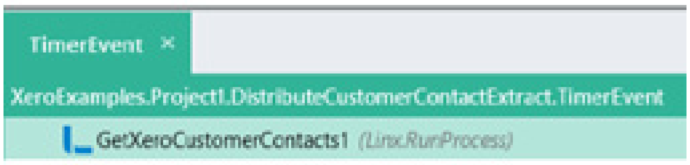


Properties (Timer)

Name	DistributeCustomerContactExtract
Days	
Include days	Weekdays
Exclude days	
Times	
Mode	Specific times
Run at	09:00:00

Attaching the process to be triggered

In this step, we drag the *GetXeroCustomerContacts* function from the Solution Panel on to the *TimerEvent* canvas. This defines the process we want to be triggered by the Timer.



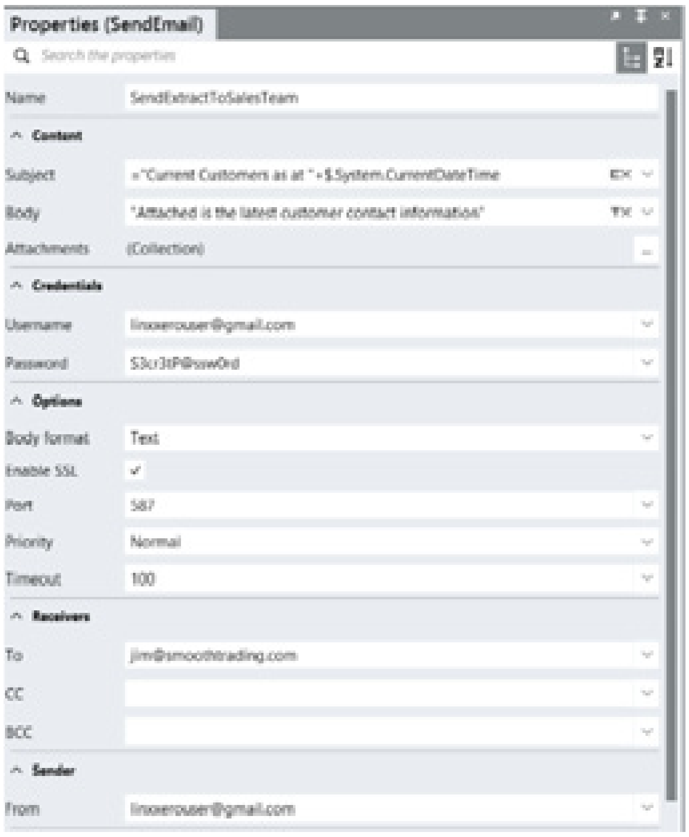
TimerEvent ×

XeroExamples.Project1.DistributeCustomerContactExtract.TimerEvent

GetXeroCustomerContacts1 *(Linx.RunProcess)*

To add a SendEmail timer event

- From the Plugin list on the right-hand panel, expand the *Email* Plugin and select the *SendEmail* Function.
- Drag the function onto the middle canvas and place on the *TimerEvent* canvas, under the *GetXeroCustomerContacts1* process.
- Update the properties of the *SendEmail* Function in the Properties panel (bottom left):
  - Name: SendExtractToSalesTeam
  - Subject: Any description
    Eg: Current Customer as at 20 October 2017 or =”Current Customers as at “+$.System.CurrentDateTime
  - Body: “Attached is the latest customer contact information”
  - Credentials: Username and password for the sending address of the SMTP server.
  - Body Format: Text
  - Attachments: Select Settings -> FilePath from the drop-down list
  - Sender/From: Sender’s email address
  - Receiver: Desired recipient email address
  - SMTP server: smtp.yourdomain.com

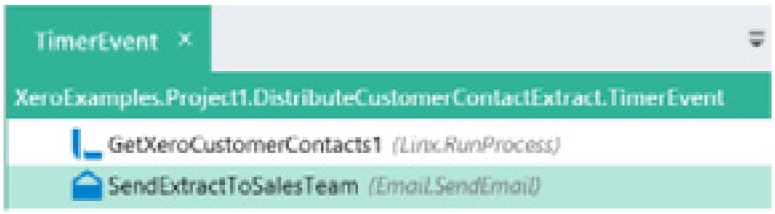


Properties (SendEmail)

Name	SendExtractToSalesTeam
Content	
Subject	="Current Customers as at "+$.System.CurrentDateTime
Body	"Attached is the latest customer contact information"
Attachments	(Collection)
Credentials	
Username	linxxerouser@gmail.com
Password	S3cr3tP@ssw0rd
Options	
Body format	Text
Enable SSL	✓
Port	587
Priority	Normal
Timeout	100
Receivers	
To	jim@smoothtrading.com
CC	
BCC	
Sender	
From	linxxerouser@gmail.com

Using email with 2FA:
For email providers using two-factor authentication (such as Gmail), you will need to complete additional steps to bypass the additional security settings.
Further reading: **Gmail and Linx**

- Your TimerEvent should now look like this:



TimerEvent
XeroExamples.Project1.DistributeCustomerContactExtract.TimerEvent
GetXeroCustomerContacts1 *(Linx.RunProcess)*
SendExtractToSalesTeam *(Email.SendEmail)*

- In the ribbon bar click *Save* and save your solution.
- Rerun the debug tool for any errors.

Summary

Every weekday morning at 9am your Linx solution automatically connects to your Xero app, retrieves a list of your Xero contacts and writes it to a CSV file. The file is then automatically emailed to a specific person.

This is made possible by creating a process that uses functions from relevant plugins, like the File, Xero and Email plugins. The process is automated by adding a Timer service to your solution and then hosting your solution on Linx Server.

LINX

For further information

https://linx.software

support@linx.software