

# Adding Google Contact Lookup to FreePBX CIDSuperfecta

## Create and configure a new Google Project

Create a new google project if you don't already have one

- Goto <https://console.cloud.google.com/projectcreate>

Enter a Project Name:

Example: FreePBX Google Contacts Lookup

## Enable Google People API

- Goto <https://console.cloud.google.com/apis/library/people.googleapis.com>
- and click [ENABLE]

## Configure OAuth Consent Screen (Before adding any credentials you will need to configure Consent screen)

- Goto <https://console.cloud.google.com/apis/credentials/consent/edit>
- Enter an App name. Example: FreePBX Google Contacts Lookup
- Enter your support email address
- Enter your domain under Authorized Domains: <<myfreepbxfqdn>>
- Enter your email address under Developer Contact Information
- Click [SAVE AND CONTINUE]
- Add a scope
  - Click "ADD OR REMOVE SCOPES"
  - Under "Manually add scopes" add: <https://www.googleapis.com/auth/contacts.readonly> and click "ADD TO TABLE"
  - Click [UPDATE]
- Add Test Users. While publishing status is set to "Testing", only test users are able to access the app. Allowed user cap prior to app verification is 100, and is counted over the entire lifetime of the app
  - Click on + ADD USERS

- Click [SAVE AND CONTINUE]
- Alternatively to adding a test user you can **Publish** your project to **Production**. A project in Production does not need to have an explicit list of test users. However, the cap of 100 email addresses is still enforced unless the project is verified.
  - Goto <https://console.cloud.google.com/apis/credentials/consent>
  - Click on [PUBLISH APP]
  - The status will change to “In Production” and “NEEDS VERIFICATION” as Verification Status.
  - If you are planning to have more than 100 unique email addresses registered to this feature then you will need to verify the project. Otherwise just leave it unverified.

## Add Credentials

- The new integration supports both Web Application Credentials as well as Desktop Application credentials. The Web Application credentials are the new preferred way

### Add Web Application Credentials

- Click on Credentials or goto <https://console.cloud.google.com/apis/credentials>
- Click on [+ CREATE CREDENTIALS] and Select OAuth client ID, or goto <https://console.cloud.google.com/apis/credentials/oauthclient>
- Select **Web application** as the Application type.
- Enter a name for your client or leave it as the default
- Under Authorized redirect URIs add the following two entries
  - <https://127.0.0.1> (This is the local loopback. This is needed only if you use the Web Application credentials without the “-UseWebApp” option in the Google ID field when configuring Google Contacts as a source under FreePBX CIDSperfecta. See below for more details)
  - <https://<<myfreepbxfqdn>>/admin/config.php?display=superfecta&action=oauth2callback>
    - Please replace <<myfreepbxfqdn>> with your real FreePBX FQDN. IT CANNOT BE an IP Address
    - **oauth2callback.php** is a new php helper script that will make it easier to read and copy the new Google Code that is required to complete the setup on FreePBX
- Click [SAVE]
- Please make a note of the new **Client ID** and **Client Secret** associated with this Web Application credential. You could also download a json file with that information. Please note that the client secret will only be displayed once. You will have to reset it if you didn't make a note of it.


## Add Desktop Application Credentials

- Click on Credentials
- Click on [+ CREATE CREDENTIALS] and Select OAuth client ID, or goto <https://console.cloud.google.com/apis/credentials/oauthclient>
- Select **Desktop app** as the Application type.
- Enter a name for your client or leave it as the default
- Click [CREATE]
- Please make a note of the new Client ID and Client Secret associated with this Desktop Application credential. You could also download a json file with that information. The only redirect URI that works with Desktop Application Credentials is **<https://127.0.0.1>**

## Install new .php files

- Replace the following three files in the directory **`/var/www/html/admin/modules/superfecta/includes/oauth-google`**
  - **Google\_Service\_ReadContacts.php**
  - **GoogleAuthManager.php**
  - **Oauth2callback.php**
- **`/var/www/html/admin/modules/superfecta/sources`**
  - **source-GoogleContacts.module**
- **`/var/www/html/admin/modules/superfecta`**
  - **pages.superfecta.php**
- Copies of the files are attached to <https://issues.freepbx.org/browse/FREEPBX-23790>

## Configure your FreePBX SuperFecta profile with the new Google Contact Lookup

- Navigate to your FreePBX CID Superfecta screen  
**`https://<<myfreepbxfqdn>>/admin/config.php?display=superfecta`**
- Add a new Scheme and give it a name and optionally specify your DID rules and CID rules and all other fields.
- Select the new Caller ID Superfecta Scheme
- Enable GoogleContacts as a data source.
- Configure Google Contact lookup integration
  - Click on the  Wrench icon next to Google Contacts to configure the new source
  - Please note the new fields that were added to this interface
  - **Google ID** - No Change
  - **Google Code** - field should be left “XXX” first time around

- **Access Token** should be left blank
- **Redirect URI** - This is a new field. If you are using Google OAuth2.0
  - Desktop Application Credentials - set to <https://127.0.0.1>
  - Web Application Credentials - Set to either
    - <https://<yourservername>/admin/config.php?display=superfe cta&action=oauth2callback>, or
    - <https://127.0.0.1>
    - The preferred method is <https://<yourservername>/admin/config.php?display=superfe cta&action=oauth2callback>.
    - If you use a URI that includes the localhost address (127.0.0.1) you might first need to revoke the existing credentials on <https://myaccount.google.com/permissions?continue=https%3A%2F%2Fmyaccount.google.com%2Fsecurity> before granting access again.
- **Client ID** - the one you just created
- **Client Secret** - the one you just created
- Click **Save**
- Click on **Debug/Test Run Scheme**
- Click on **Run This Scheme**
- You should see a section like that

Google Code did not result in authentication.

## To set or reset authorisation

Click the link in the next line, then copy the code google give into the second box on the settings area for Google Contacts. Then click submit in the settings, and then use another test number to check all is working.

To start that process, **click here**

- Click on the link and complete the authorization.
- At this point if everything was setup correctly you should see a Sign with Google screen
- 

Sign in with Google

Choose an account

to continue to <YOUR FREEPBX SYSTEM FQDN>

- Select the account from the list or add a new one

- If you were using a web application credentials AND you properly configured your Google Project and updated your FreePBX system with the new .php files you should see a screen like below

1. Copy the code: **4/0XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX**

2. Go back to configure Google in FreePBX

2a. Paste this code into the **Google Code** box replacing the **XXX**

2b. Hit **Save**

2c. Now run debug again and it should work

3. You can close this window once done

- If you don't see a page like that it means your setup is not complete. However, you might still be able to get it to work depending on the issue.
- Look at the url that you are trying to display. If it looks like any of the formats below you will be able to complete the setup.

`https://<<myfreepbxfqdn>>/admin/config.php?display=superfecta&action=oauth2callback?code=XXXXXXXXXXXXXXXXXXXXXXXXXXXX&scope=https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fcontacts.readonly`

Or

`https://127.0.0.1/?code=XXXXXXXXXXXXXXXXXXXXXXXXXXXX&scope=https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fcontacts.readonly`

- Copy the code portion with the URL
- Go back to configure Google in FreePBX - Paste this code into the **Google Code** box replacing the **XXX**
- - Hit [Save]
- - Now run debug again and it should work