

# SMAIL/ Gmail SMTP OAuth2 認証の利用・設定

2020/11/01 初版  
2021/06/22 第 2 版  
2021/09/10 第 3 版  
2022/03/27 第 4 版  
2022/04/05 第 5 版

## — 目次 —

1. 概要	
1) 準備	3
2) 設定	3
3) 動作確認	3
2. 詳細	
①Google API Console で OAuth 2.0 クライアント用の[クライアント ID]および [クライアントシークレット]の取得	4
1) 新しいプロジェクトの作成	4
2) OAuth 同意画面を[作成]	5
3) OAuth クライアント ID を作成	8
4) デスクトップ のクライアント ID の JSON ファイルのダウンロード	9
②Google OAuth API で[リフレッシュトークン]を取得	1 1
1) SmailRefreshToken.exe を起動させます	1 1
③SmailRefreshToken.exe で正しくリフレッシュトークンが取得できれば、 実行フォルダーに、smail_refresh token.ini が生成されます	1 3
④smail_refresh token.ini を smail.exe と同じフォルダーにコピーし、smail.exe を起動して、 smtp 認証に OAUTH-XOAUTH2 を選択すれば利用できます	1 4
3. 動作確認	1 5
4. OAUTH 関連エラー例	1 6

## 1. 概要

### 1) 準備

Google アカウントが必要です。(本参考例は Gmail 無料アカウントです。G Suite アカウントもほぼ同じです)

**※実際の設定方法は2. 詳細を参照して下さい。**

- ① Google API Console で OAuth 2.0 クライアント用の[クライアント ID]と[クライアントシークレット]の取得をブラウザアクセスで行います。
- ② Google OAuth API で[リフレッシュトークン]を取得  
付属の SmailRefreshToken.exe で行います。認証はデフォルトのブラウザが起動します。

ここまでで、SMAIL で OAuth2 認証を利用するための各種情報が取得できます。

この操作は一度のみです。ただし、Google アカウントのパスワード変更等を行った場合はリフレッシュトークンの再取得(SmailRefreshToken.exe)が必要です。

### 2) 設定

- ③ SmailRefreshToken.exe で正しくリフレッシュトークンが取得できれば、実行フォルダーに、smail\_refreshtoken.ini が生成されます。
- ④ smail\_refreshtoken.ini を smail.exe と同じフォルダーにコピーし、smail.exe を起動して、smtp 認証に OAUTH-XOAUTH2 を選択すれば利用できます。  
※SMTP ユーザ ID,SMTP パスワードを設定していても無視されます。(OAUTH-XOAUTH2 では必要が無いため)

### 3) 動作確認

以降、SMAIL は、リフレッシュトークンを利用してアクセストークンを自動で取得し、OAuth 2.0 認証でメールを送信できるようになります。

#### OAuth 2.0 認証の動作 OS:

OS	Windows Server 2008 sp2	Windows Server 2008 R2	Windows Server 2012	Windows Server 2016	Windows xp	Windows 7 pro	Windows 8.1	Windows 10 pro
OAuth 2.0 認証	△(*1)	○	○	○	×	○	○	○

(\*1)

KB4019276 パッチを利用し、TLS1.2 を利用可にすれば動作可能です。(Windows Server 2008 sp2 64BIT 版検証済み)

<https://support.microsoft.com/ja-jp/help/4074621/add-rds-support-for-tls-1-1-and-tls-1-2-in-windows-server-2008-sp2>

## 2. 詳細（実際の取得方法）

### ① Google API Console で OAuth 2.0 クライアント用の[クライアント ID]および[クライアントシークレット]の取得

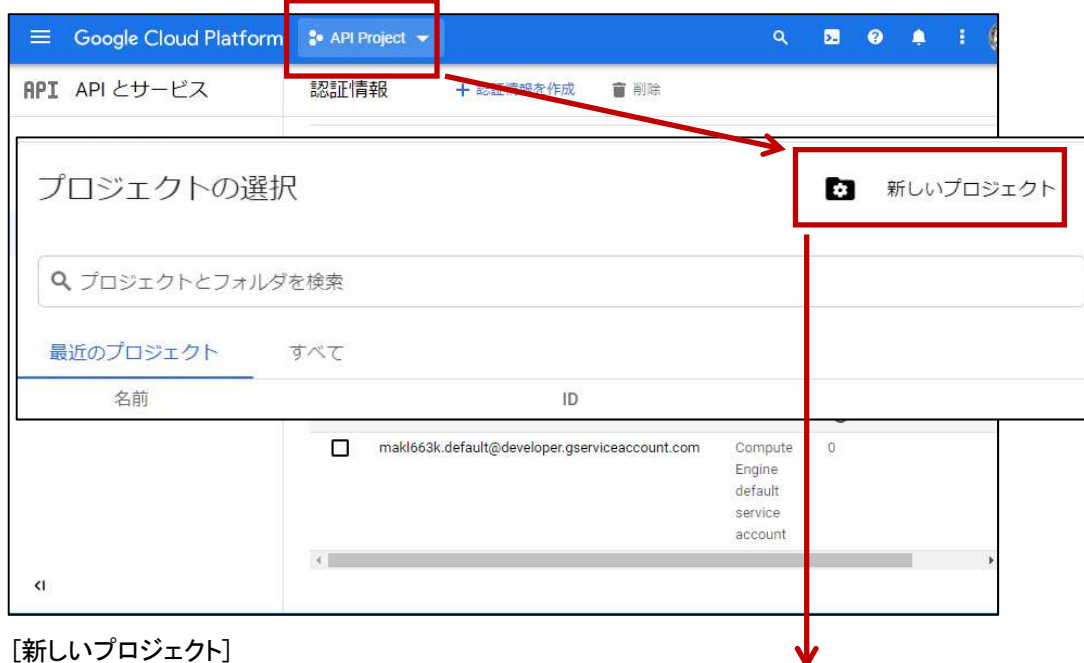
Google API Console (<https://console.cloud.google.com/apis/>)にアクセスする。

Google アカウントでログインします。

（本参考例は Gmail 無料アカウントです。G Suite アカウントもほぼ同じです。この処理は一度だけです。）

#### 1.) 新しいプロジェクトの作成

プロジェクト名は任意の名前です。



[新しいプロジェクト]



## 2) OAuth 同意画面を[作成]。

- ・外部を選択。G Suite であれば [User Type]、内部が選択できます。



## OAuth 同意画面

- ・アプリ名 (必須)

名前は任意の名前ですが短すぎたり一般名称を登録しようとすると、「アプリの保存中にエラーが発生しました」と保存時エラーが出た場合はアプリ名を変更して試してください。例では **smail20220327** としました。

- ・ユーザサポートメール(必須) 適宜設定してください。
- ・デベロッパーの連絡先情報(必須) 適宜設定してください。

Google Cloud Platform

search for gmail

検索 プロダクト、リ...

API API とサービス

有効な API とサービス

ライブラリ

認証情報

OAuth 同意画面

ドメインの確認

ページの使用に関する契約

アプリ登録の編集

1 OAuth 同意画面

2 スコープ

3 テストユーザー

4 概要

アプリ情報

この情報は同意画面に表示されるため、デベロッパーのユーザー情報とデベロッパーへの問い合わせ方法をエンドユーザーが把握できます。

アプリ名\*

small20220327

同意を求めるアプリの名前

ユーザー サポートメール\*

@gmail.com

ユーザーが同意に関して問い合わせるために使用

アプリのロゴ

参照

ユーザーがアプリを認識できるように、同意画面に 1 MB 以下の画像をアップロードします。使用できる画像形式は、JPG、PNG、BMP です。最適な結果を得るには、ロゴを 120 x 120 ピクセルの正方形にすることをおすすめします。

アプリのドメイン

デベロッパーとユーザーを保護するために、Google では、OAuth を使用するアプリのみに認可ドメインの使用を許可しています。同意画面では、次の情報がユーザーに表示されます。

アプリ登録の編集

アプリケーションのホームページ

ホームページへのリンクをユーザーに提供します

[アプリケーション プライバシー ポリシー] リンク

一般公開のプライバシー ポリシーへのリンクをユーザーに提供します

[アプリケーション利用規約] リンク

一般公開の利用規約へのリンクをユーザーに提供します

承認済みドメイン

同意画面または OAuth クライアントの構成でドメインが使用されている場合は、ここで事前登録する必要があります。アプリの検証が必要な場合は、[Google Search Console](#) にアクセスして、ドメインが承認済みであるかどうかを確認してください。承認済みドメインの上限の[詳細](#)をご覧ください。

+ ドメインの追加

デベロッパーの連絡先情報

メールアドレス\*

@gmail.com

これらのメールアドレスは、プロジェクトの変更について Google からお知らせするために使用します。

保存して次へ

キャンセル

「保存して次へ」

6

## OAuth 同意画面 - スコープ

Google Cloud Platform

search for gmail

検索 プロダクト、リ...

API API とサービス

有効な API とサービス

ライブラリ

認証情報

OAuth 同意画面

ドメインの確認

ページの使用に関する契約

アプリ登録の編集

OAuth 同意画面

2 スコープ

3 テストユーザー

4 概要

スコープとは、アプリのユーザーに許可を求める権限を表します。スコープを定めることで、プロジェクトからユーザーの Google アカウントにある特定の種類のプライベートなユーザーデータへのアクセスが可能になります。[詳細](#)

スコープを追加または削除

非機密のスコープ

API ↑	範囲	ユーザー向けの説明
表示する行がありません		

機密性の高いスコープ

機密性の高いスコープとは、プライベートユーザーデータへのアクセスをリクエストするスコープです。

API ↑	範囲	ユーザー向けの説明 ↑
表示する行がありません		

制限付きのスコープ

制限付きのスコープとは、機密性の高いユーザーデータへのアクセスをリクエストするスコープです。

API ↑	範囲	ユーザー向けの説明
表示する行がありません		

保存して次へ

キャンセル

何も変更せずそのまま「保存して次へ」

## OAuth 同意画面 - スcope - テストユーザ



[+ADD USERS]でテストユーザ(gmail アカウントのメールアドレス)を入力して保存します。

ここでは、この画面にログインしている google アカウントの gmail アドレスを登録します。

以後の **SmailRefreshToken.exe** で登録する gmail アカウントもこれと同じになります。

テストユーザを追加できたら「保存して次へ」

### 3) OAuth クライアント ID を作成

[認証情報] - [認証情報を作成] - [OAuth クライアント ID]をクリック。





[アプリケーションの種類]-[デスクトップアプリ]を選択します。

The screenshot shows the 'OAuth 2.0 Client ID' creation page in the Google Cloud Platform API Project console. The left sidebar contains navigation links: 'API とサービス', 'ダッシュボード', 'ライブラリ', '認証情報', 'OAuth 同意画面', 'ドメインの確認', and 'ページの使用に関する契約'. The main content area is titled 'OAuth クライアント ID の作成'. It includes a description of Client ID, a dropdown for 'アプリケーションの種類\*' (Application type) with options: 'ウェブアプリケーション', 'Android', 'Chrome アプリ', 'iOS', 'テレビと入力が限られたデバイス', 'デスクトップアプリ' (selected), and 'ユニバーサル Windows プラットフォーム (UWP)'. Below the dropdown is a section for '承認済みの JavaScript 生成元' (Authorized JavaScript origins) with a text input and a '+ URI を追加' button.

[名前]は、任意な名前を入力し、「作成」をクリックします。

The screenshot shows the 'OAuth 2.0 Client ID' creation page. The 'アプリケーションの種類\*' dropdown is set to 'デスクトップアプリ'. The '名前\*' (Name) field is filled with 'smail クライアント'. Below the name field is a note: 'OAuth 2.0 クライアントのみ使用され、エンドユーザーには表示されません' (Used only for OAuth 2.0 clients, not displayed to end users). The '作成' (Create) button is highlighted with a red box and an arrow pointing to a modal dialog. The modal dialog is titled 'OAuth クライアントを作成しました' (OAuth client created) and contains the following information: 'クライアント ID' (674173188309-2d9akg0ab143vraa66242hkjv3cgoggu.apps.googleusercontent.com), 'クライアント シークレット' (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), and a button 'JSON をダウンロード' (Download JSON). The modal also includes a note: 'OAuth アクセスは、OAuth 同意画面に表示されているテストユーザーに制限されます' (OAuth access is limited to test users shown on the OAuth consent screen).

JSON ファイルをダウンロードしてください。

後で SmailRefreshToken.exe で登録に使用

します。(JSON ファイルは認証情報画面で

いつでもダウンロード可能です)

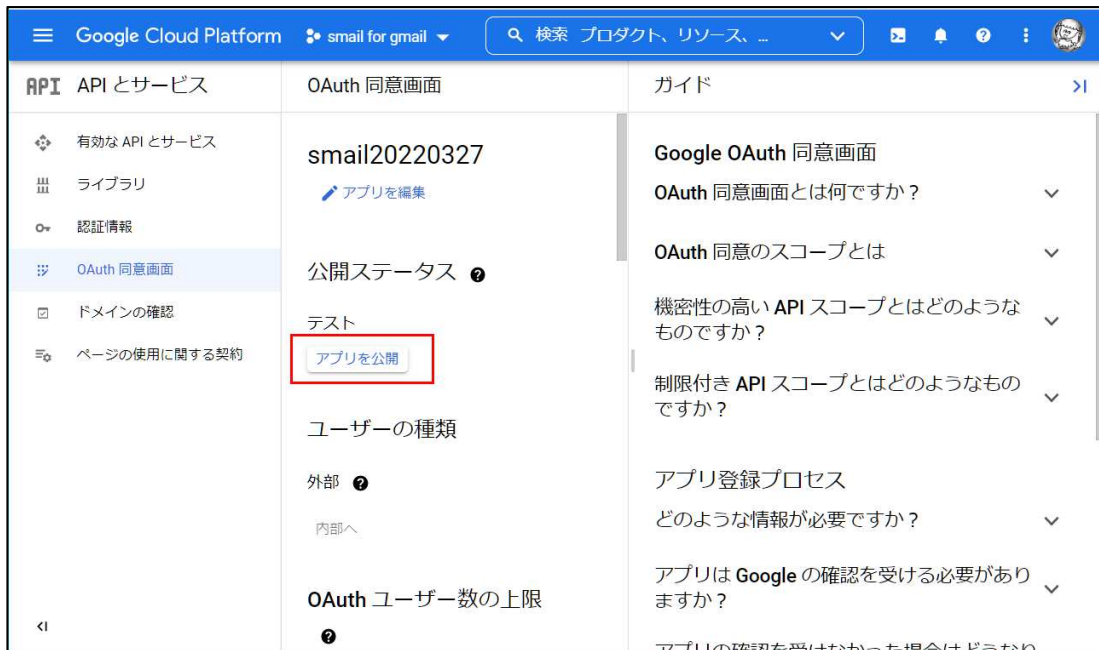
クライアント ID、クライアント シークレット

等の必要な情報が記載されています。

- ・ファイル名: **client\_secret\_xxxxxxxxxgoogleusercontent.com.json**

これで、Google API Console (<https://console.cloud.google.com/apis/>)で登録する作業は終了です。

Oauth 同意画面で公開ステータスを「テスト」から「本番環境」に変更します。

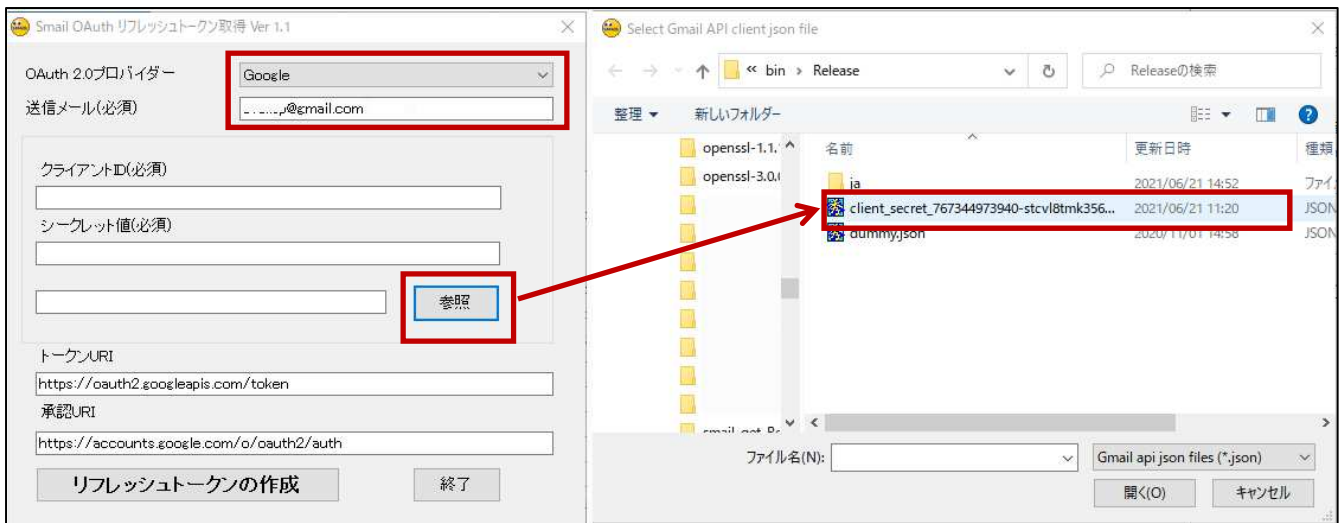


※公開ステータスが「テスト」のままの場合、リフレッシュトークンの有効期限が7日となり、  
期限切れの場合に SmailRefreshToken.exe でリフレッシュトークンの再取得が都度必要になります。  
かならず、テスト以外は「本番環境」に設定してください。

## ② Google OAuth API で[リフレッシュトークン]を取得

1) Smail 付属アプリの SmailRefreshToken.exe を起動させます。

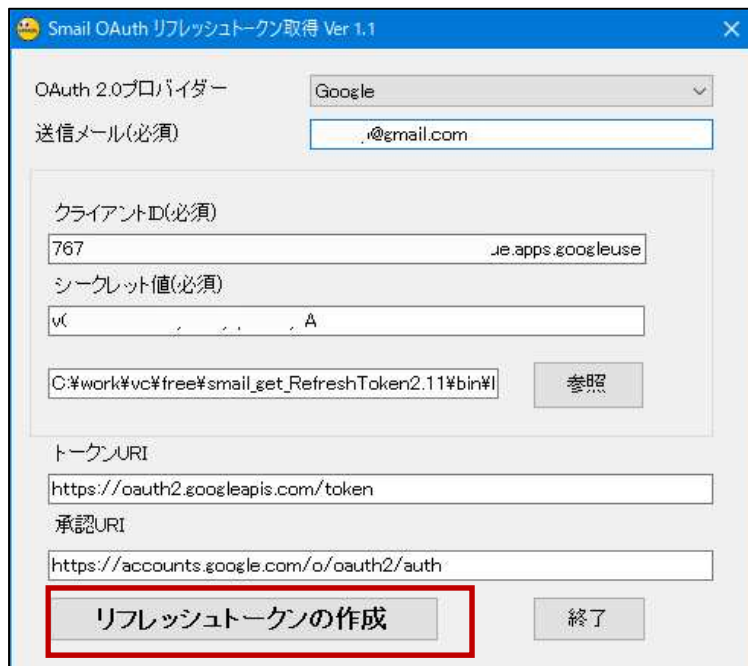
① で使用したメールアドレスと取得した設定用 J S O N ファイルを取り込みます。



正しく、J S O N ファイルが取り込めると、クライアント ID,シークレット値が表示されます。

トークン URI,承認 URI も自動的に更新されます。(通常は初期表示の内容と同じで、見た目は変わりません)

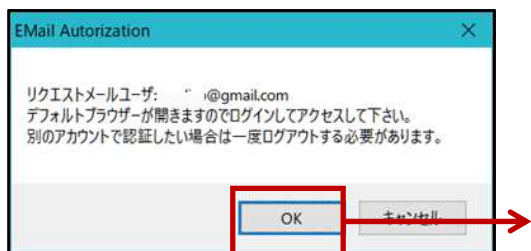
(※設定用 J S O N ファイルを使用せず、クライアント ID,シークレット値を直接入力しても問題ありません)



[リフレッシュトークンの作成] をクリックします。

(このボタンをクリックすることで、Google API サーバへ承認接続し、リフレッシュトークンを取得します。)

確認ダイアログが表示されますので、[OK] ボタンをクリックしてください。(リクエストのメールアドレスは正しいか、再確認してください)



[続行]をクリックしてください。

※承認エラーが出る場合は「OAuth 同意画面 - ストユーザ」で正しくユーザが登録されているか確認してください



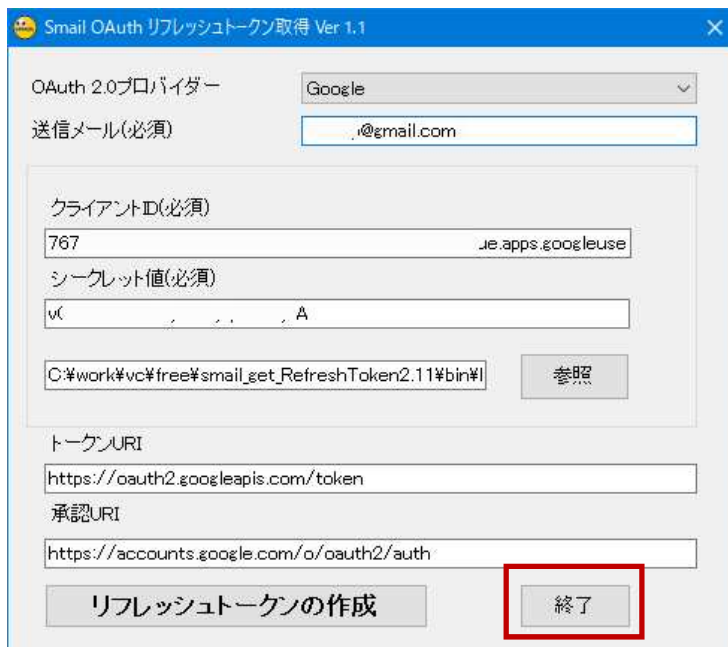
[続行]をクリックしてください。

③ SmailRefreshToken.exe で正しくリフレッシュトークンが取得できれば、実行フォルダーに、`smail_refreshtoken.ini` が生成されます。

- ・成功するとデフォルトのブラウザーに以下のように表示されます。



- ・ SmailRefreshToken.exe 以下のダイアログメッセージが表示されます。  
[OK]を押して[終了]させてください。



※リフレッシュトークンの再取得が必要な時はこの操作をします。通常は一度だけ取得すれば OK ですが、Google アカウントのパスワード変更の時は再取得が必要となります。



尚、ここまでの操作は SMAIL を動作させない PC 上で行っても問題ありません。

④smail\_refreshtoken.ini を smail.exe と同じフォルダーにコピーし、smail.exe を起動して、smtp 認証に OAUTH-XOAUTH2 を選択すれば利用できます。

- 1) 実行フォルダーにリフレッシュトークン等が記載された INI ファイル、**smail\_refresh token.ini** が生成されていますので、**smail.exe** または **smail.ini** のあるフォルダーにコピーして下さい。

以後、`smail_refreshtoken.ini` のファイルが SMAIL を実行する PC に必要です。

例) smail\_refreshtoken.ini

```
[OAUTH2]
UserID=?????????@gmail.com
ClientID=88888888888888888888-stcvxxxxxxxxxxxxxxxxxxxxapps.googleusercontent.com
ClientSecret=vXXXXXXXXXXXXXXXXXXXXXyq0VUOjLA
RefreshToken=1//0exxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
TokenURI=https://oauth2.googleapis.com/token
```

- 2) SMAIL.exe を起動し SMTP 認証 : OAUTH-XOATH2 を選択して保存して下さい。

**SMAIL初期設定 ver 4.26**

■SMTPサーバ

SMTPポート番号  ※SMTPサーバはコマンドラインから指定

SMTP認証 **AUTH-PLAIN**

SMTPユーザID

SMTPパスワード  ※保存後、自動的に暗号化されます

SMTP/SSL暗号化 ☒ 使用する ☐ STARTTLS使用する

ポート番号:  TLS1.3-1.0,SSLv2,v3自動

■POP3サーバ

POPサーバ  ポート番号:

POP before SMTP ☐ 使用する ☐ APOPを使う

POPユーザID

POPパスワード  ※保存後、自動的に暗号化されます

遅延待ち時間  msec タイムアウト時間  sec

POP3/SSL暗号化 ☒ 使用する ポート番号:

設定ファイル名:

保存先(E):

**保存(S)** 終了(X)

[保存]して終了して下さい。

refreshtoken.ini が見つからない場合は以下の警告が出ます。正しく INI ファイルを作製して下さい。



smail.ini に設定が保存されます。(既存の smail.ini に引き継がれます)

### 3. 動作確認

例) c>smail -hsmtp.gmail.com -s"TEST MAIL" -fxxxxxx@gmail.com dummy@example.jp -d -i

```
#----- SMAIL 4.26 Information By Eva -----
#設定ファイル      : [C:\work\vc\free\smail_v4.26_free_ssl_gui-vc2010\Release\smail.ini]
#POP before SMTP   : 使わない
#APOP              : 使わない
#SMTP 認証         : 使う AUTH XAUTH2 を利用
#設定ファイル (OAUTH 用) : [C:\work\vc\free\smail_v4.26_free_ssl_gui-vc2010\Release\smail_refreshtoken.ini]
#API アカウント    : [xxxxxxx@gmail.com]
#SMTP アカウント   :
#SMTPS (SMTP over SSL) : 使う [OpenSSL 1.1.1h]
#SSL 方式          : TLS1.3-1.0, SSLv2, v3 自動
#STARTTLS         : 使わない
#本文ファイル     : なし
#添付ファイル     : なし
#メールサーバ     : smtp.gmail.com
#メール送り元     : xxxxx@gmail.com
#メール送り先     : dummy@example.jp
#サブジェクト内容 : TEST MAIL
#キャラクターセット : ISO-2022-JP
#エンコーディング : 7bit
SSL connection using TLS_AES_256_GCM_SHA384 - TLSv1.3
X509 subject_name: /C=US/ST=California/L=Mountain View/O=Google LLC/CN=smtp.gmail.com
X509 issuer_name: /C=US/O=Google Trust Services/CN=GTS CA 101
Recv:[52 bytes] 220 smtp.gmail.com ESMTP 10sm8435678pjt.50 - gsmt
Send: EHLO xxxxxx
Recv:[220 bytes] 250-smtp.gmail.com at your service, [xx.xx.xx.xx]
250-SIZE 35882577
250-8BITMIME
250-AUTH LOGIN XOAUTH2 PLAIN-CLIENTTOKEN OAUTHBEARER XOAUTH
250-ENHANCEDSTATUSCODES
250-PIPELINING
250-CHUNKING
250 SMTPUTF8
[250 message ok!]
[Try get access token from https://oauth2.googleapis.com/token] ; 認証サーバにアクセストークン要求
[access_token=ya29.A0AfH6SMDzH_2_0...] ; アクセストークン取得
Send: AUTH XOAUTH2 ***** ; XOAUTH2 で接続要求
Recv:[20 bytes] 235 2.7.0 Accepted
Send: MAIL FROM:<XXXXXX@gmail.com>
Recv:[40 bytes] 250 2.1.0 OK 10sm8435678pjt.50 - gsmt
Send: RCPT TO:<xxxxx@xxxxxxxxxxx.jp>
Recv:[40 bytes] 250 2.1.5 OK 10sm8435678pjt.50 - gsmt
Send: DATA
Recv:[41 bytes] 354 Go ahead 10sm8435678pjt.50 - gsmt
Send: From: XXXXXX@gmail.com
To: dummy@example.jp
Subject: TEST MAIL
X-Mailer: SMAIL 4.26
Mime-Version: 1.0
Content-Type: text/plain; charset=ISO-2022-JP
Content-Transfer-Encoding: 7bit
Date: Sun, 01 Nov 2020 15:44:39 +0900
Send:
#//メールヘッダー送信完了 (1.609sec)....
Send:
.
Recv:[52 bytes] 250 2.0.0 OK 1604213082 10sm8435678pjt.50 - gsmt
Send: QUIT
Recv:[56 bytes] 221 2.0.0 closing connection 10sm8435678pjt.50 - gsmt

C:\work\vc\free\smail_v4.26_free_ssl_gui-vc2010\Release>
```

・SMAIL は SSL 通信に OPENSLL ライブラリーを使っていますが、OAUTH のアクセストークン取得部分のみ、

google への https プロトコルは、標準 Windows 関数を使用しております。従って、使用端末が tls1.2 以降をサポートしている必要があります。

#### OAuth 2.0 認証の動作 OS:

OS	Windows Server 2008 sp2	Windows Server 2008 R2	Windows Server 2012	Windows Server 2016	Windows xp	Windows 7 pro	Windows 8.1	Windows 10 pro
OAuth 2.0 認証	△(*1)	○	○	○	×	○	○	○

(\*1) KB4019276 パッチを利用し、TLS1.2 を利用可にすれば動作可能です。(Windows Server 2008 sp2 64BIT 版検証済み)

<https://support.microsoft.com/ja-jp/help/4074621/add-rds-support-for-tls-1-1-and-tls-1-2-in-windows-server-2008-sp2>

・プロキシ対応です。httpsアクセスに端末のプロキシ情報を利用しています。

#### 4. OAUTH 関連エラー例

エラー番号 28 が発生した時は、オプションに -d を付けて通信内容をコンソールに表示して詳細なエラーを取得して下さい。

##### 1. Error HttpSendRequest [12157]

```
[250 message ok!]  
[Try get access token from https://oauth2.googleapis.com/token]  
Error HttpSendRequest [12157]  
ERR-0028:OAUTH 認証サーバからアクセストークンが取得できませんでした。
```

12157 セキュリティ・チャネル・エラー

SSL ライブラリのロード中に、アプリケーションに内部エラーが発生しました

##### 2. Error HttpSendRequest [12045]

```
[250 message ok!]  
[Try get access token from https://oauth2.googleapis.com/token]  
Error HttpSendRequest [12045]  
ERR-0028:OAUTH 認証サーバからアクセストークンが取得できませんでした。
```

12045 無効な認証局です

サーバーSSL 証明書は、未知または無効な認証局から発行されたものです。

PC が TLS1.2 をサポートしていない場合のエラーです。

##### 3. Error HttpSendRequest [12029]

```
[250 message ok!]  
[Try get access token from https://oauth2.googleapis.com/token]  
Error HttpSendRequest [12029]  
ERR-0028:OAUTH 認証サーバからアクセストークンが取得できませんでした。
```

12029 サーバー接続の試みに失敗しました

プロキシ設定が間違っており接続できない場合もこのエラーが出ます。



#### 4. Error InternetCrackUrl [12006]

[250 message ok!]

[Try get access token from https://oauth2.googleapis.com/token]

Error InternetCrackUrl [12006]

ERR-0028:OAUTH 認証サーバからアクセストークンが取得できませんでした。

12006 不正な URL です

smail\_refreshtoken.ini 等が参照できず TokenURI が参照できない場合発生します。

もしくは smail\_refreshtoken.ini 中の TokenURI の URL の記述が間違っています。

#### 5. Error HTTP\_STATUS [400]

[250 message ok!]

[Try get access token from https://oauth2.googleapis.com/token]

Error HTTP\_STATUS [400]

ERR-0028:OAUTH 認証サーバからアクセストークンが取得できませんでした。

OAuth 同意画面でプロジェクトの公開ステータスがテストで7日間の有効期限切れでリフレッシュトークン切れです。公開ステータスを本番環境に設定して、smail\_refreshtoken.exe で再度リフレッシュトークンを作成してください。

またはリフレッシュトークンが6カ月間使用されなかった場合の期限切の場合もあります。  
(smail で gmail を使ってメール送信しない期間が6カ月間あった場合は再取得が必要です)