Remote Repository Manager1.5 使い方

by nao jake.burst@gmail.com 2024-5
改訂　2026-10

0. Remote Repository Manager で何ができるか？各種のレポジトリ管理サービス（Github,Gitlab,Bitbucket に加えて Gitea）に対してそれぞれの API を利用して、レポジトリの一覧表示（List)、削除（Delete)、作成（Create)、さらにはサービス間での移行（Migrate）を統合的に行うことができます。統合開発環境である RAD Studio や Visual Studio の外部ツールとしても便利に使用できます。

1. 準備

最初にレポジトリ管理サービスの登録を行います。このためにはプログラムと同一階層に置いた、”noenc-urls-pats.txt”という名前のテキストファイルに、下記の項目を記述します。5 行で 1 サーバーのエントリーですが、以下に gitlab.com の例を示します。同梱した” sample-urls-pats.txt”を参考に作成し、”noenc-urls-pats.txt”という名前で RemoteRepositoryManager.exe と同じ階層においてください。

```
4
gitlab.com
https://gitlab.com/api/v4
Doctornao
glpat-SUxSE_q8-hDeJTu4RkgVWWM6MQpvOjE0Yg8.01.1703hbui2
```

最初がサーバー番号（ここでは 4 ですが任意です)、サービス名称（gitlab.com)、サーバーの API への base url（https://gitlab.com/api/v4)、id(Doctornao などあなたの id)、Personal Access Token(glpat- oJsXxYyZabcfLpq)です。
ここでサービス名称（名前）は管理サービスの種類を判別するために、github,gitlab,bitbucket,gitea などのキーワードを入れてください。キーワード以外は任意の文字列が使用できます。例えば、gitlab @ lan とか、gitlab.com 等と表記することで Lan 内部の gitlab 自前サーバーと本家の gitlab.com を区別できます。baseurl の具体例は、

Github　　https://api.github.com

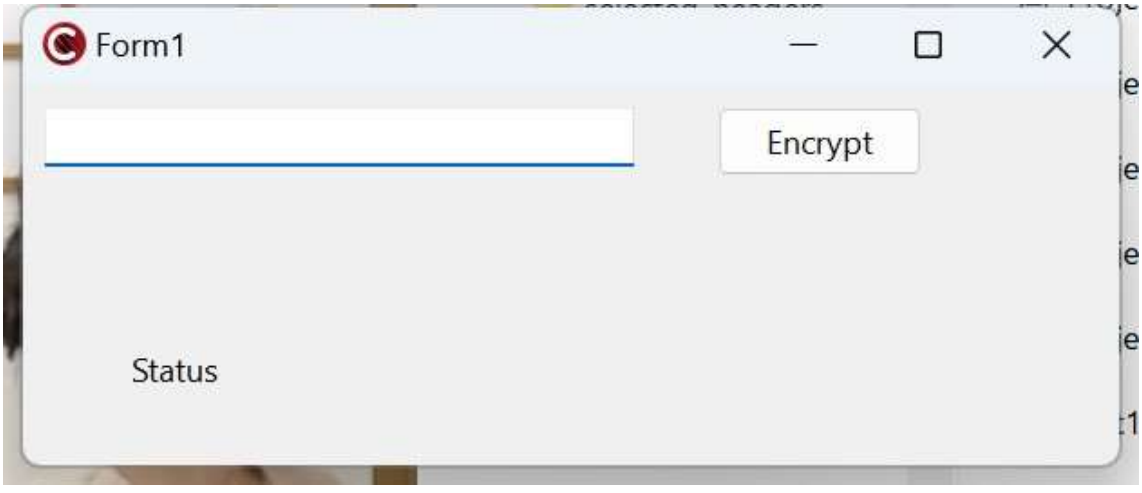
Gitlab　　https://gitlab.com/api/v4
Bitbucket　　https://api.bitbucket.org/2.0/repositories
Gitea　　https://gitea.com/api/v1

です。Gitea は公式サイトの例を示します。自前で運営しているサーバの IP アドレスやポートを適切に当てはめてください。LAN 内の IP でも可です。

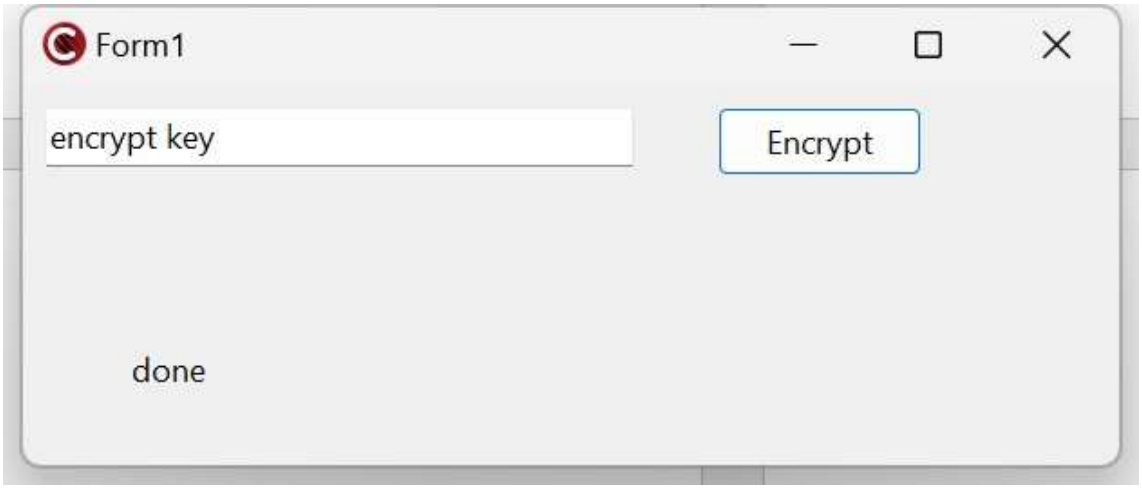
id or group は
ユーザ名や、サービスにログインする時の id です。作業によっては必要となります。

pat は Personal access Token で、それぞれのサービスにログインした後、api にアクセスする権限を持った pat を生成し、それを copy & paste してください。Github を筆頭に最近の管理サービスでは、この pat にスコープ(通用範囲)を持たせており、レポジトリの削除もできるようにするためには、レポジトリの書き込み権限も与えないといけません。 コメント行は行頭に'#'を置いてください。行全体を読み飛ばします。

複数のサーバーをお使いの場合は、複数のエントリーを書き込み。平文のままテスト的に使って、list,delete,create 等ができるかを確認してくだい。複数のユーザで使う場合に、サーバーや id,pat を自分だけが見ることができるようにするためには、この"noenc-url-pats.txt"を暗号化し、"url-pats.txt"を作成ためのアプリケーション simple-enctypt.exe を使います。これと同一の階層に、上記の"noenc-url-pats.txt"を置いて、プログラムを起動すると、



となりますので、エディットボックスに暗号化のキー(空白を含めることは可能)を入れて、Encrypt ボタンを押してください。例えば、



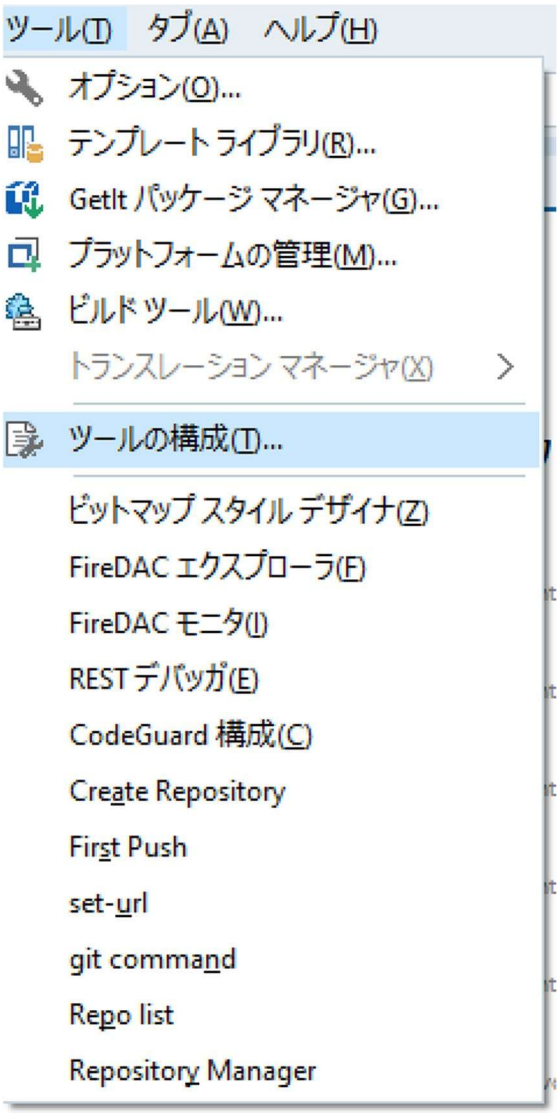
すると下部に図のように”done”との Status が出て、この時点でエッセンシャルな情報のみを暗号化したサーバーリストがアプリケーションのディレクトリに”urls-pats.txt” の名前で保存されます。同名のファイルがあっても強制的に上書きされます。このファイルをサーバーリストとして Remote Repository Manager で使用します。暗号化の文字列は忘れないようにしてください。長くても良いのですが、忘れ難いものを使用しましょう。すぐ後で使います。暗号化した”urls-pats.txt”を使う場合は、元の”noenc-urls-pats.txt”を”RemoteRepositryManager.exe”とは別の階層に置いてください。両方が存在する場合は、”noenc-urls-pats.txt”を優先的に使用します。

2. ツールとしての設定ないし登録

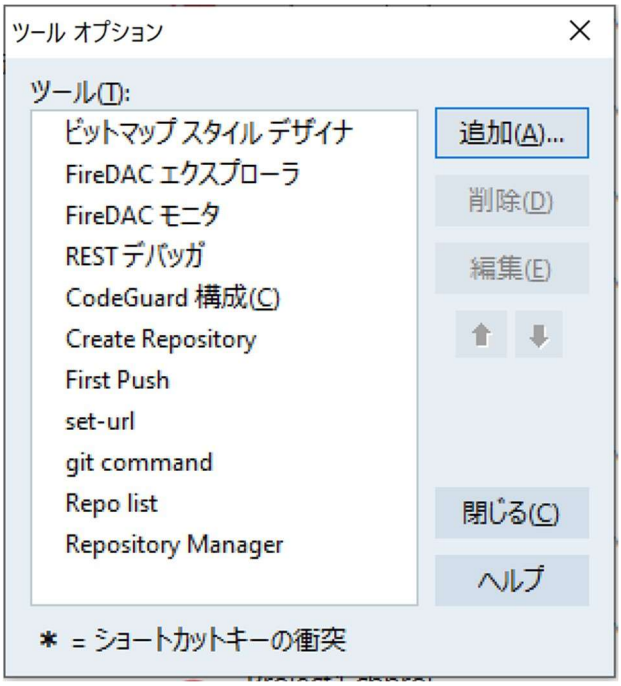
RAD Studio と Visual Studio でツールとして登録できるのは、主役の Remote Repository Manager に加えて、同梱している FirstPush.exe,GitStatus.exe,GitCommand.exe です。Remote Repository Manager を例に挙げると、

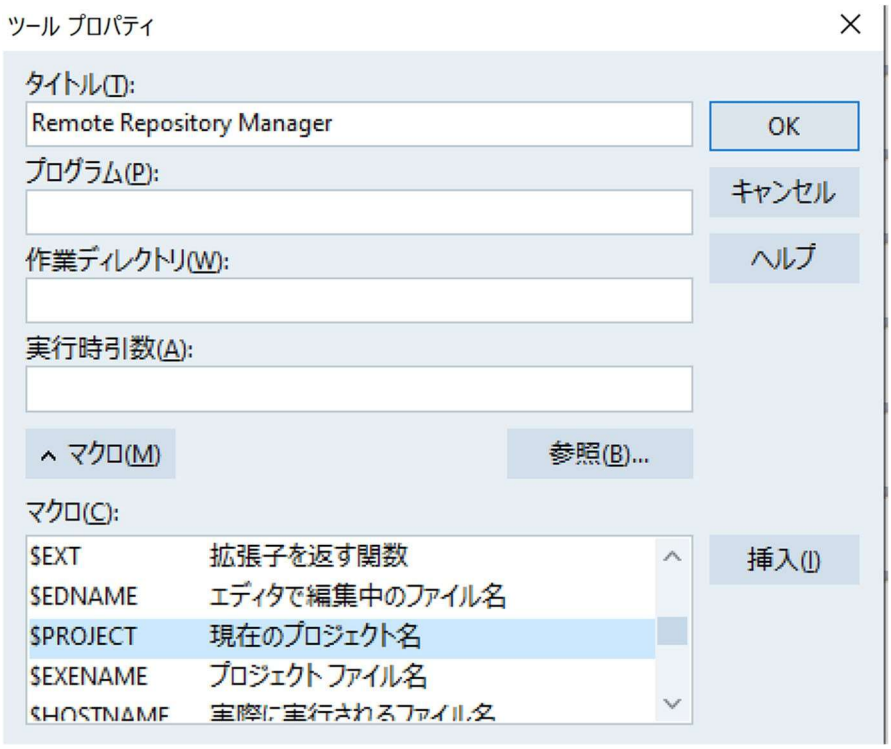
・RAD Studio の場合

IDE のトップメニューから、下図のようにツール->ツールの構成を選びます。



次に出てくるツールオプションで追加をクリックして、

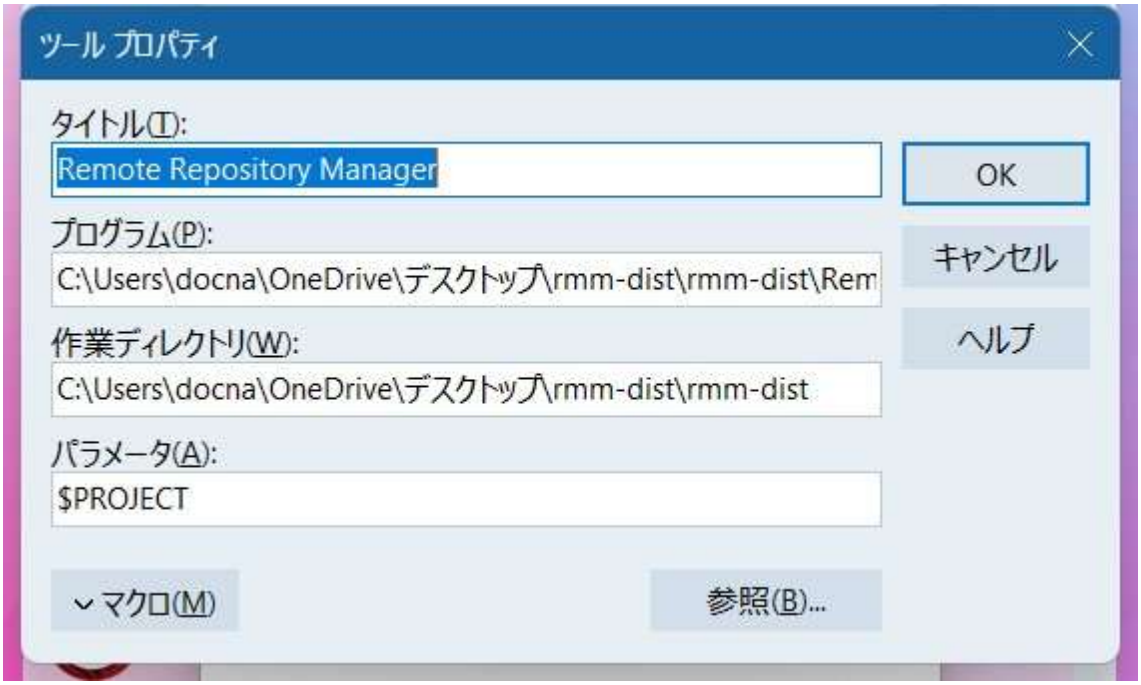




上図のようにタイトルを適宜設定し、マクロをクリックして出てくるメニューから図のように$PROJECT 現在のプロジェクト名を選択し、挿入をクリックしてください。さらに

参照をクリックして Remote Repository Manager.exe を選択してください。で、OK です。

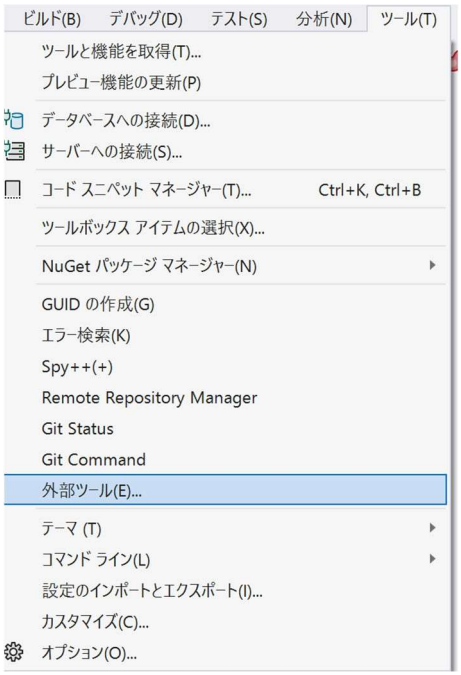
最終的には、



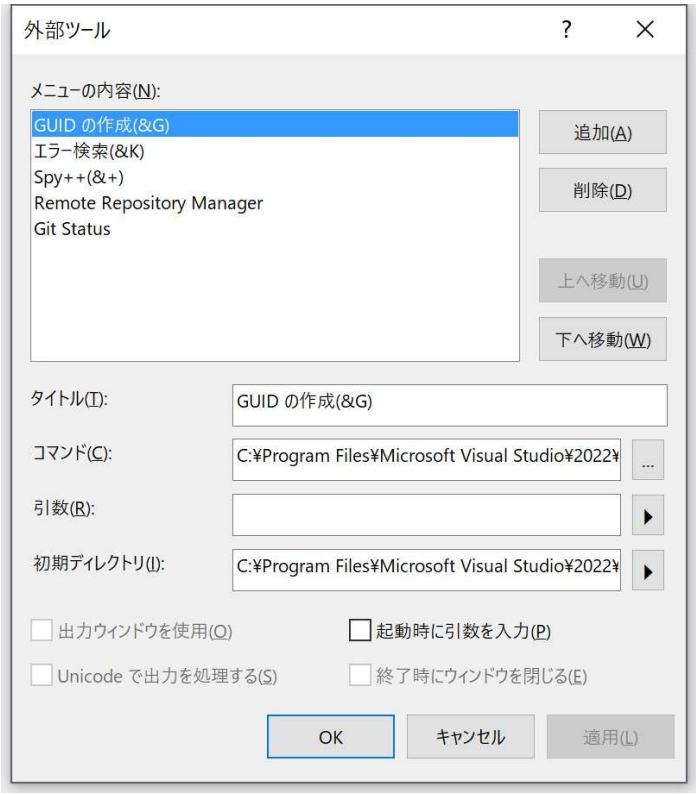
等のようになっているはずです。これで IDE からツールー＞Remote Repository Manager 経由で、いつでもリモートレポジトリの作成等が可能です。

・Visual Studio の場合

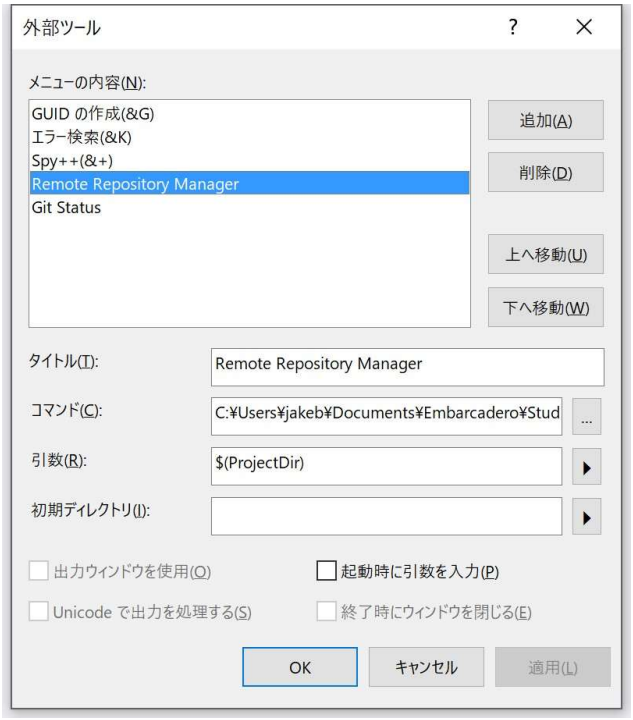
IDE のトップメニューから下図のようにツールー＞外部ツールを選び、



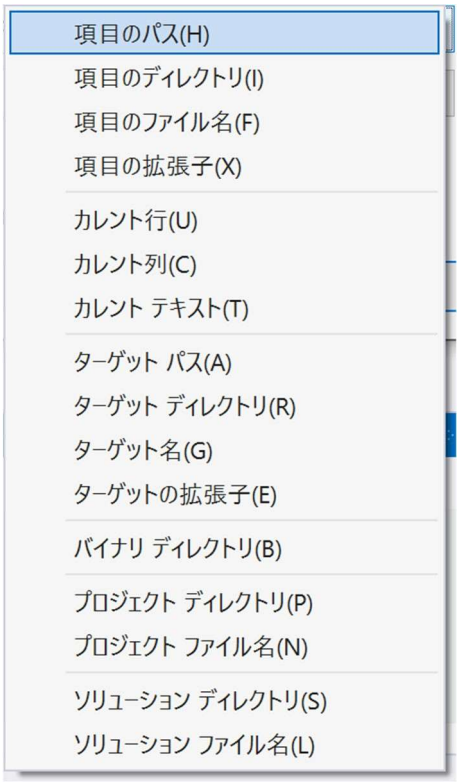
下図で”追加“をクリック。



下図のように入力します。



引数は、右向き矢印をクリックして出てくる下図で、“プロジェクト ディレクト”を選びます。

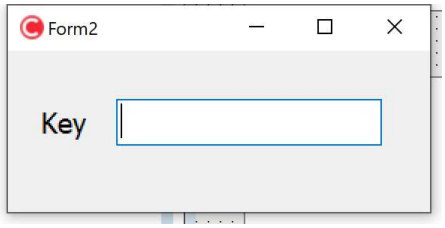


コマンドとして RemoteRepositoryManager.exe を選択して、“適応”－＞“OK”をクリックです。

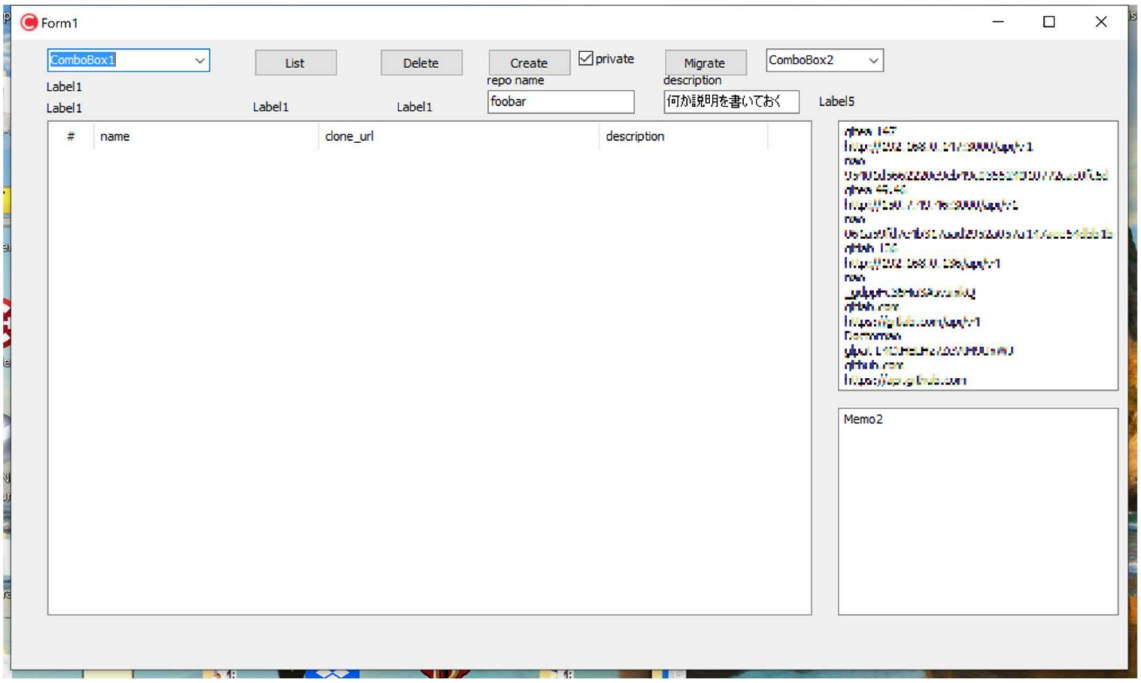
この作業を　FirstPush.exe,GitCommand.exe について繰り返してください。使う予定がなければ組み込む必要はありません。特に Visual Studio の場合は、FirstPush.exe は不要です。これら三つの補助プログラムでも実行時引数や引数は RemoteRepositoryManager.exe の場合と同じように設定してください。

3. いよいよ本題 Remote Repository Manager の使い方

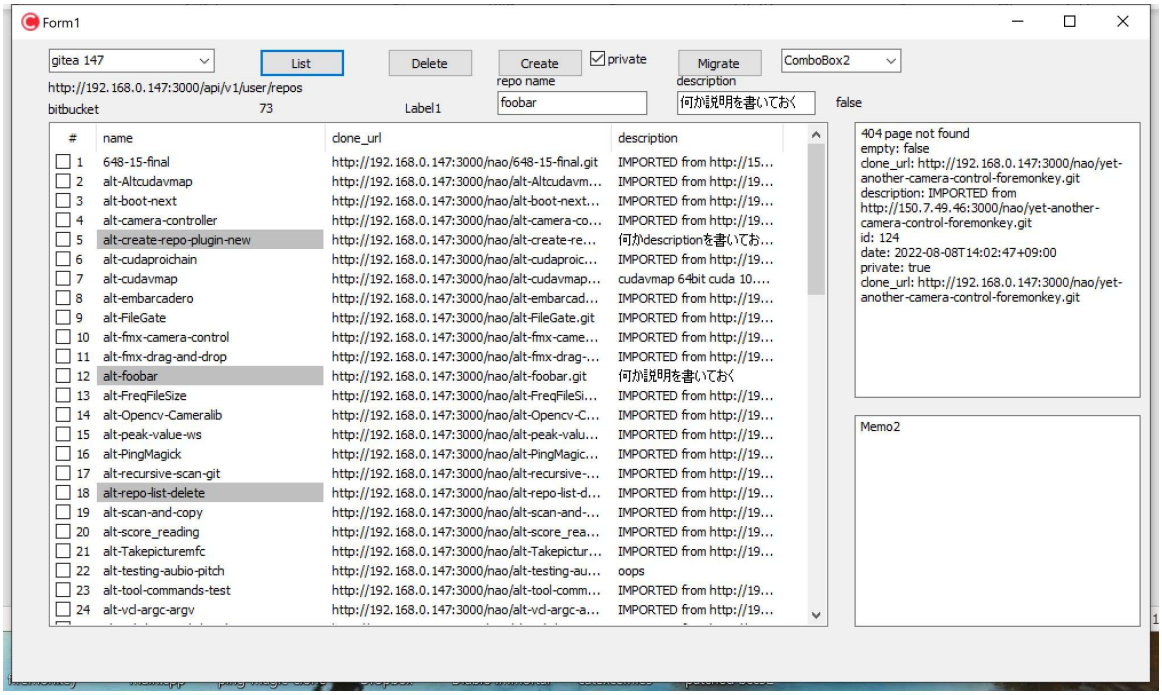
起動すると、



なるダイアログが出ますので、1 で”Encrypt”前に入れたキーの文字列を入れてください。”noenc-urls-pats.txt”を使う場合は、復号する必要がないために、このダイアログは出ません。



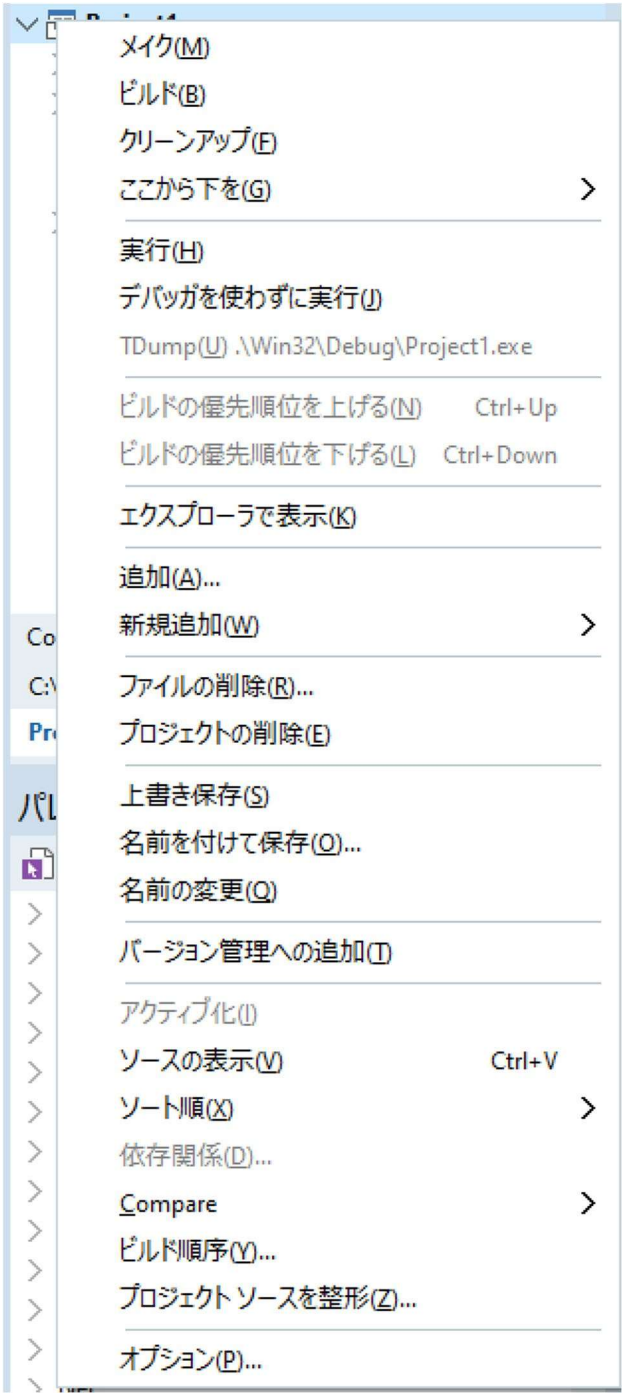
右上に復号化されたサーバーリストが表示されているはずです。上図ではモザイクで不可視にしてあります。
ここで左上のリスト（コンボボックス）からサーバを選んで、レポジトリの list,delete,create,migrate ができます。delete や migrate は選択したレポジトリに対して作用します。複数選択可能です。delete は安全のために ctrl-alt-click しないと動きません。Ctrl キーと Alt キーの両方を押しながら”Delete”ボタンをクリックです。
List の例を示します。

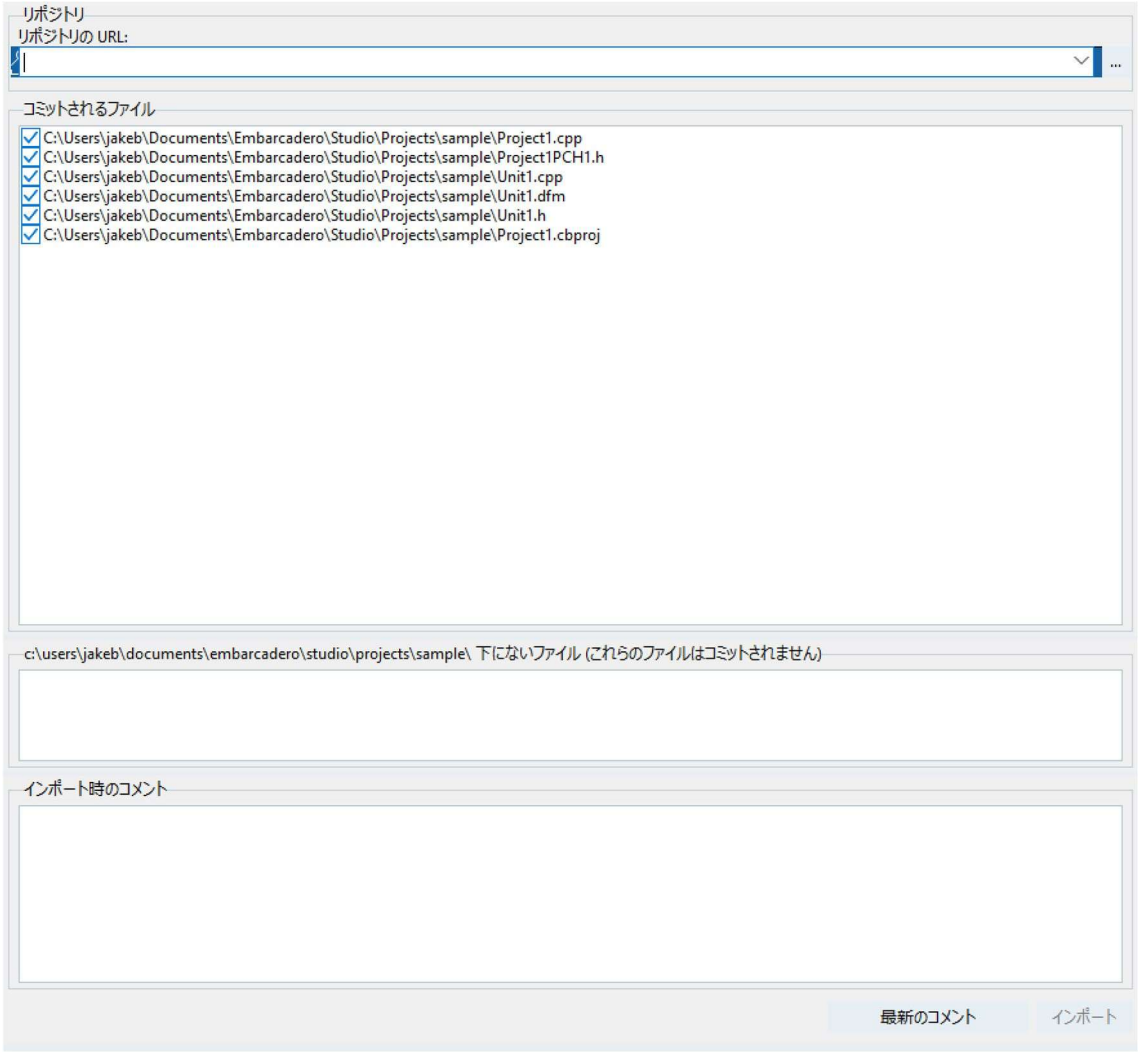


name が灰色なのは empty なレポジトリです。Delete などの場合に参考にしてください。
Create ボタンではレポジトリの作成が可能です。左上で選択したサーバ上に、name,descriptoin を指定したレポジトリが新たに作成可能です。プライベートかパブリックかの選択もチェックボタンから可能です。デフォルトはプライベートレポジトリです。

RAD Studio ないし Visual Studio からツールとして起動した場合は、現在開いているプロジェクトのディレクトリ名がレポジトリ名の初期値としてセットされるので、そのまま Create ボタンをクリックできます。説明(description)は適宜設定してください。Create に成功すると、clone_url がクリップボードに copy されますので、例えば RAD Studio の場合ならば、プロジェクトを右クリックし、出てくる下図の“バージョ

ン管理への追加（T）”を選び、次に出てくるダイアログの最上部の“レポジトリの url”にフォーカスを当てて、ペースト（ctrl-v）すればそのまま設定可能です。Visual Studio の場合にも同様に使えます。



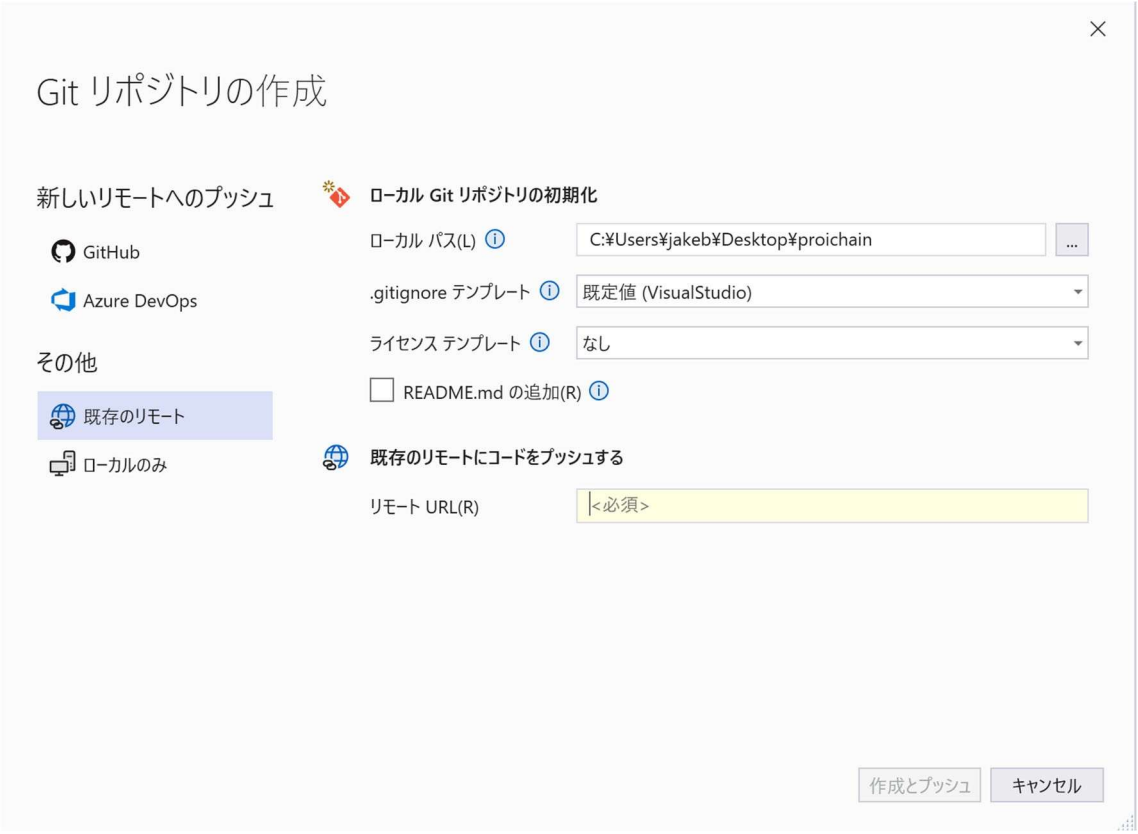


インポート後、最初の一回だけ同梱の First Push での操作が必要です。これは RAD Studio の IDE の git push が、”git push -u origin master(main)”をしてくれないためです。

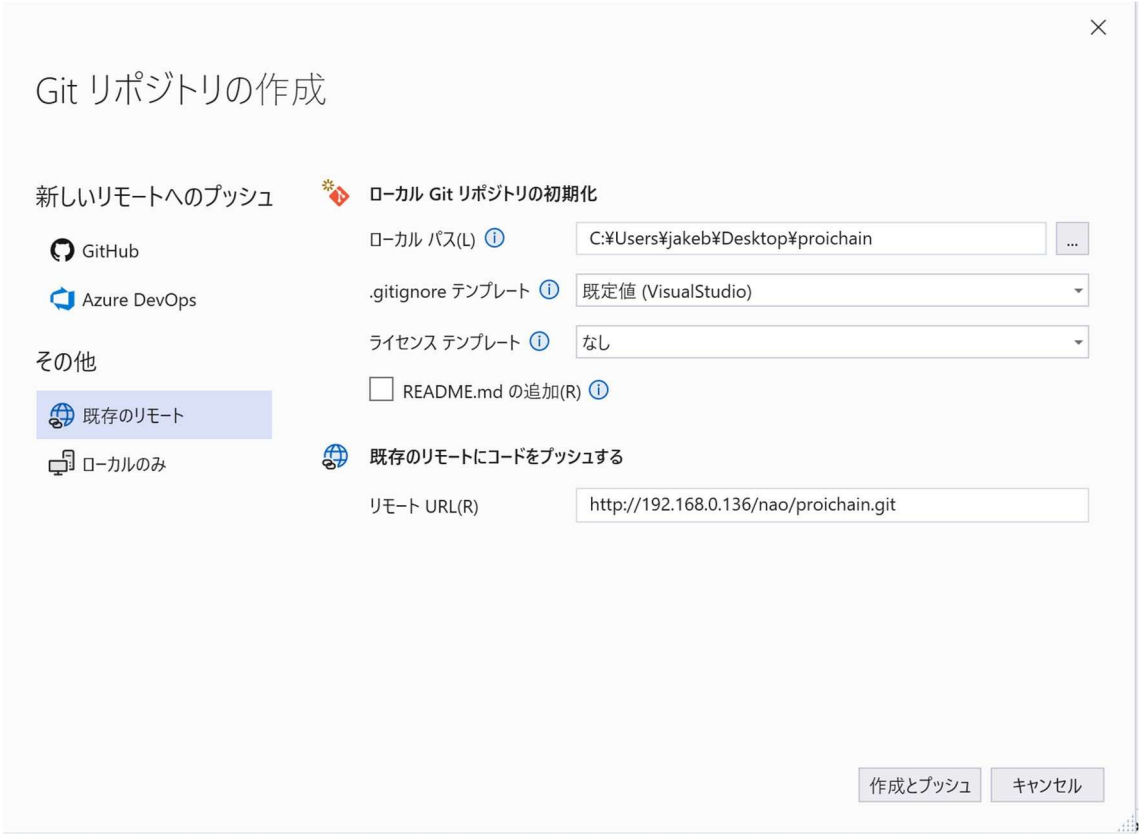
Visual Studio の場合は、下図のようにツールー＞Remote Repository Manager と選び、



適切な name と description でレポジトリを作成した後で、Git->Git レポジトリの作成から、既存のリモートを選択した下図の状態から



リモート URL をクリックして ctrl-v すれば良いです。

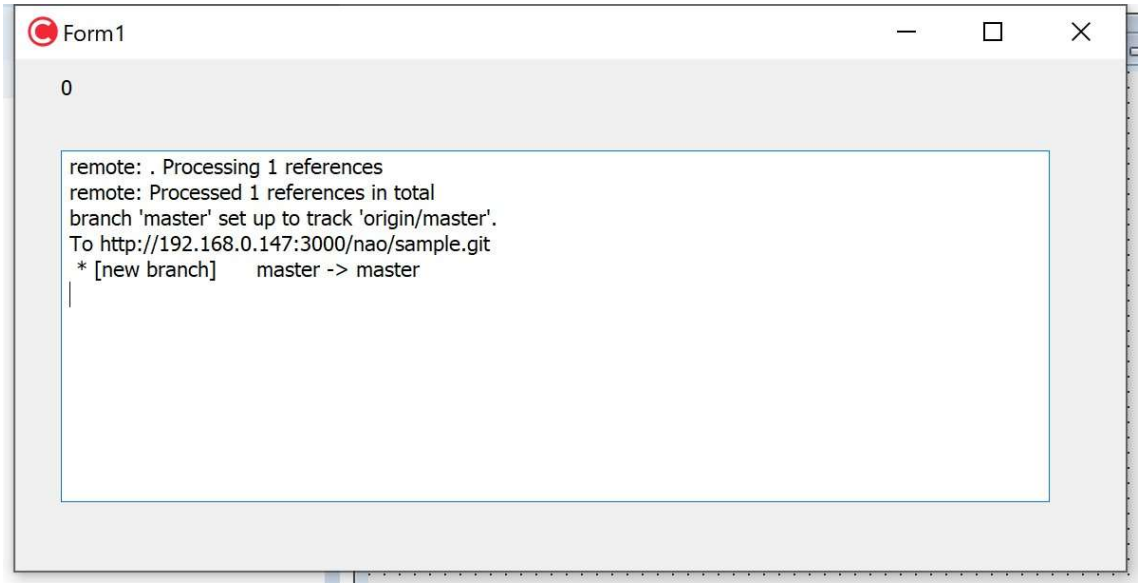


で、”作成と push”ボタンをクリックしてください。Visual Studio の場合は、First Push は不要です。

Migrate ボタンを使うと、任意のサーバのレポジトリを Gitea ないし Github のサーバへ複製できます。それぞれのサーバーにログインした後に、ui からも import は可能ですが操作方法が統一されていないので、こちらのやり方の方が便利かと思います。同じサーバー間での Migrate はできません。また LAN 上のサーバから WAN 上のサーバへ Migrate は普通はできません。WAN から LAN へは可能です。ルーターの DMZ とかポートフォワードの設定を適切に行えば、LAN から WAN への Migrate もできるはずです。Migrate は複製先のサーバーをボタンのすぐ右のリストから選択します。

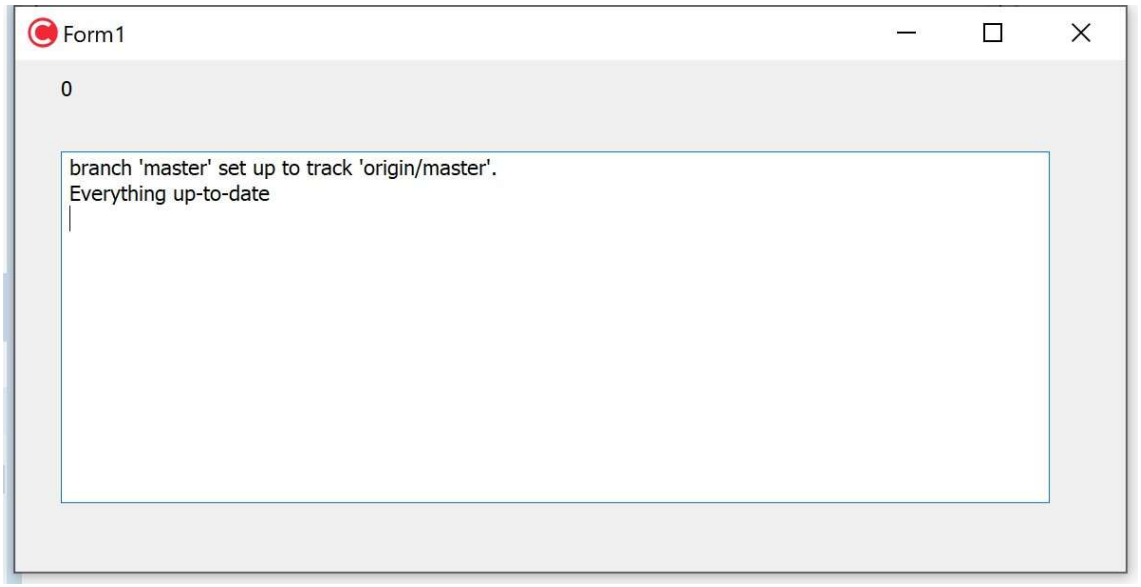
4. 補助ツールの使い方

FirstPush.exe
起動時の引数が正しくセットされていれば、そのディレクトリ内部から、
“git push -u origin master(or main 自動判別)”を実行して、その結果を表示します。
典型的な実行例は、



等となるはずです。

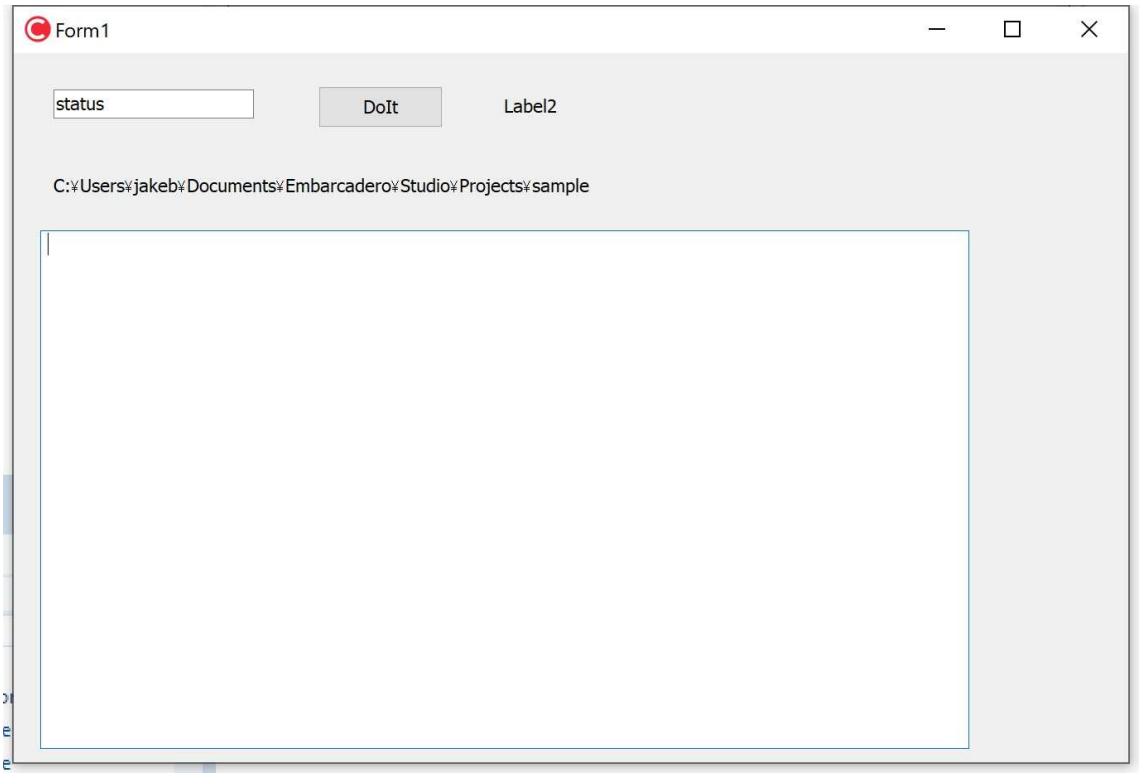
再度起動すると。



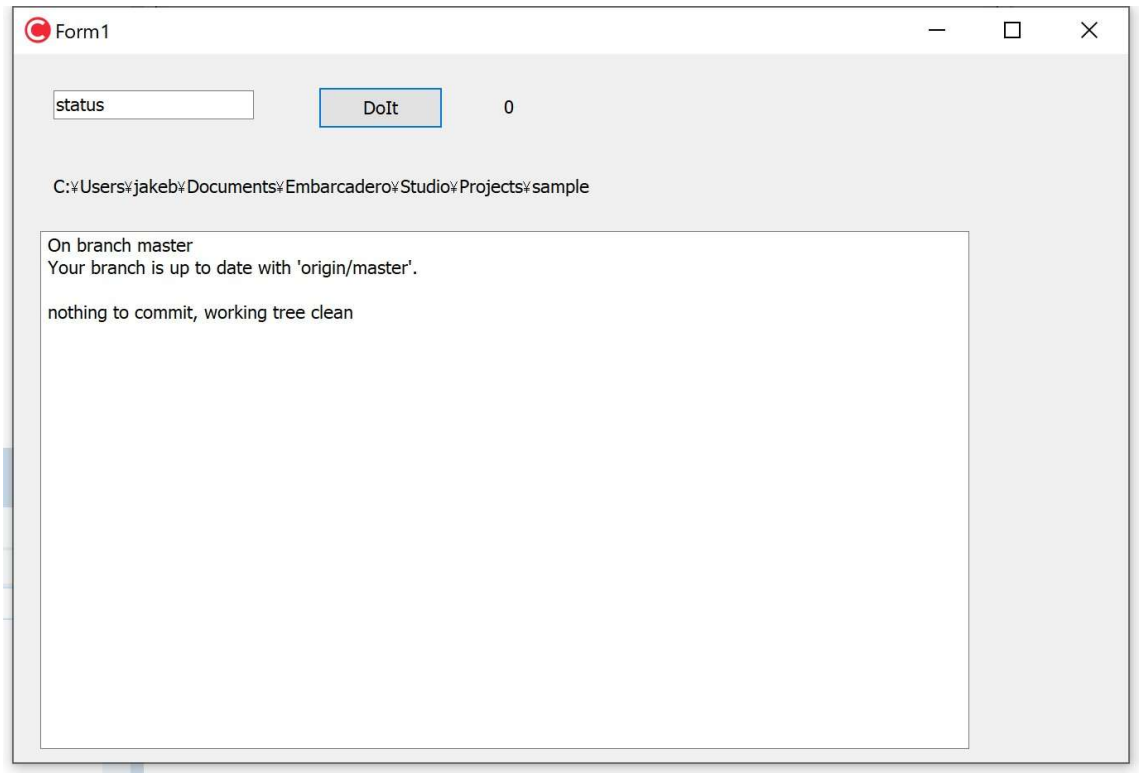
となります。本来は、ide からの git command で動作するはずですが、どうも”git push origin master”をやってるようで永久に、サイクルスタートしないので、このツールを同梱しました。

GitCommand.exe

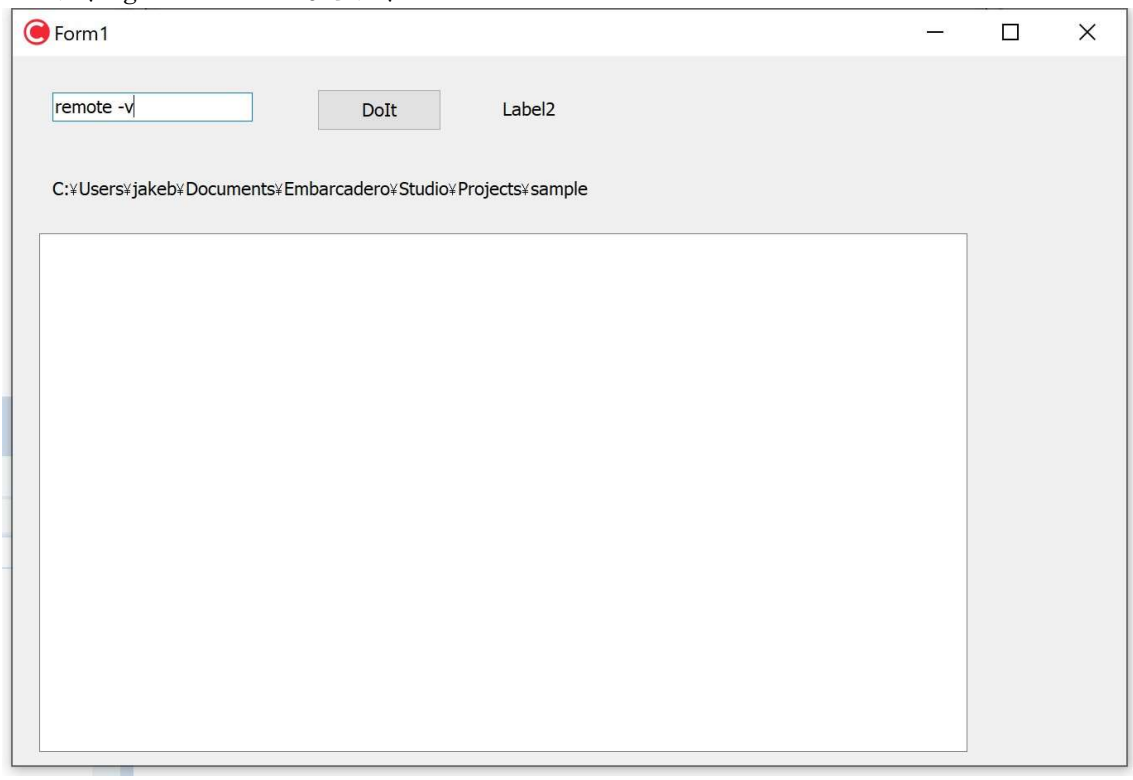
起動時の引数が正しくセットされていれば、そのディレクトリ内部から、指定した Git command を実行して、その結果を表示します。典型的な実行例は、



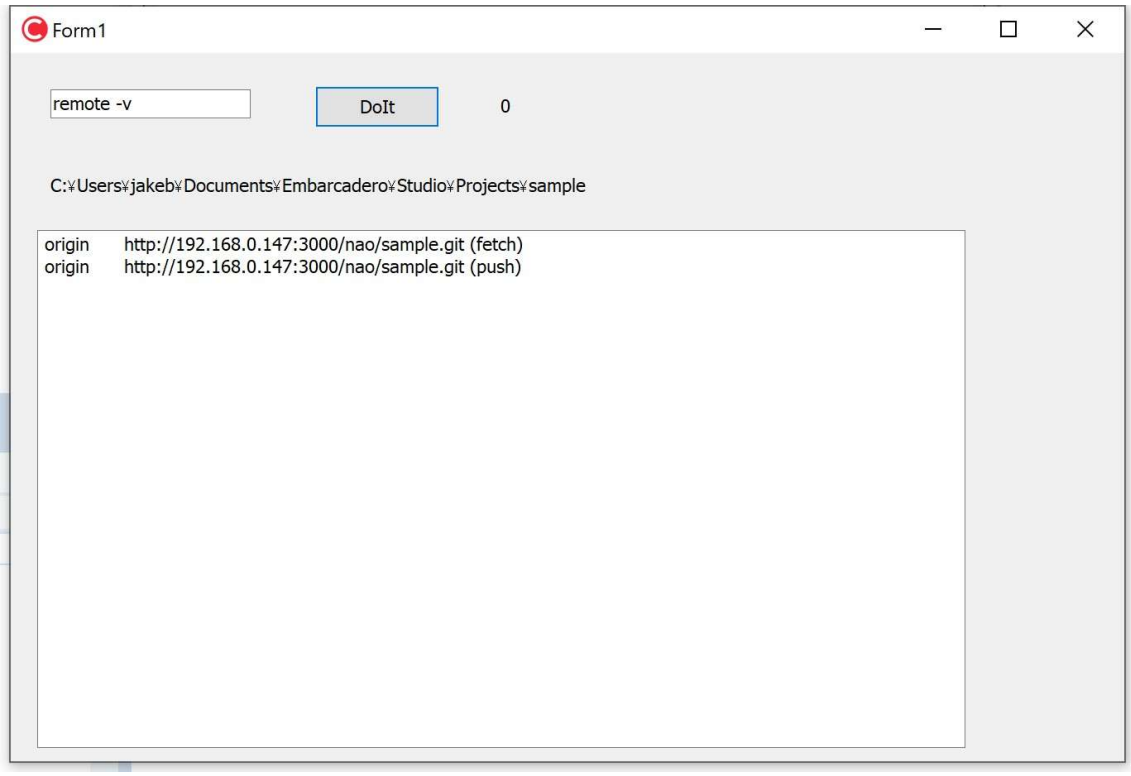
から”DoIt”クリックで、



となります。Editbox 内部を編集して”status”以外の git command も実行できます。例えば、”git remote -v”ならば、



から”DoIt”クリックで、



等と結果が表示されます。

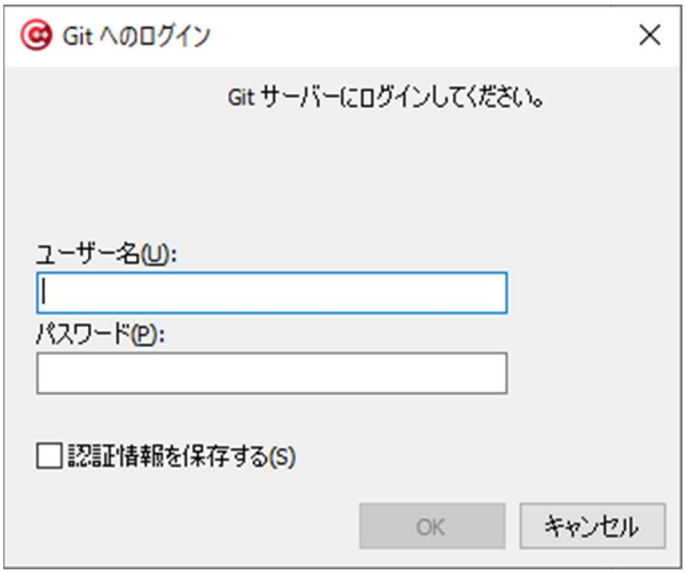
補足ないし注意

git push の際に、時として credential dialog が出てきて、id と password の入力を促されることがあります。下図参照。



この場合は、サイトにログインないしサインインするためのユーザ名とパスワードでは通らないことがあります。その場合は、id と pat を入れてみてください。

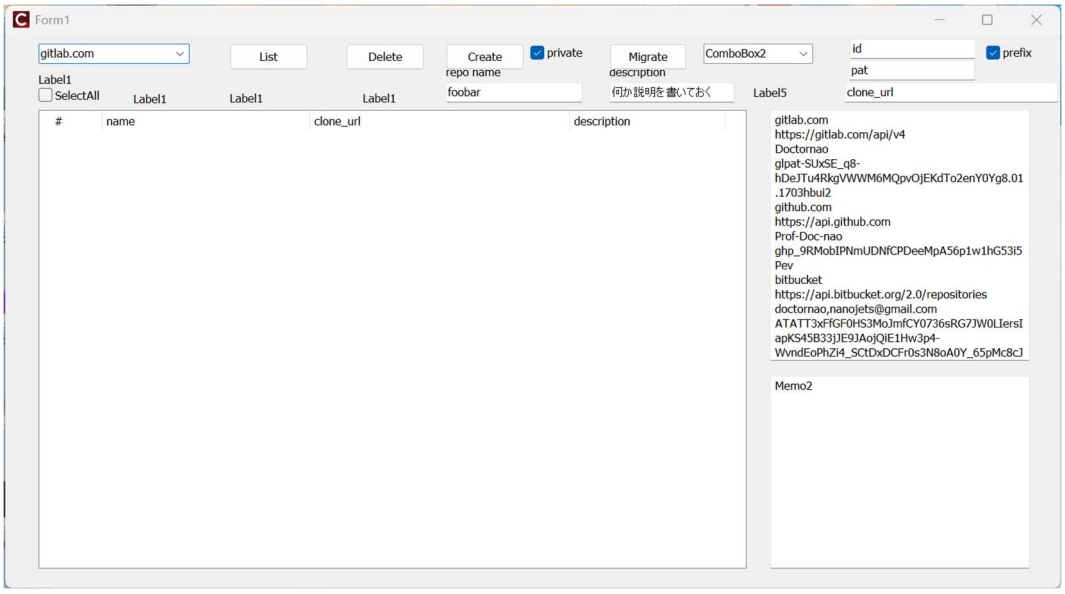
RAD Studio の IDE からのダイアログ



の場合は、同様に id と pat を入れて、認証情報を保存するにチェックを入れてから OK を押してください。

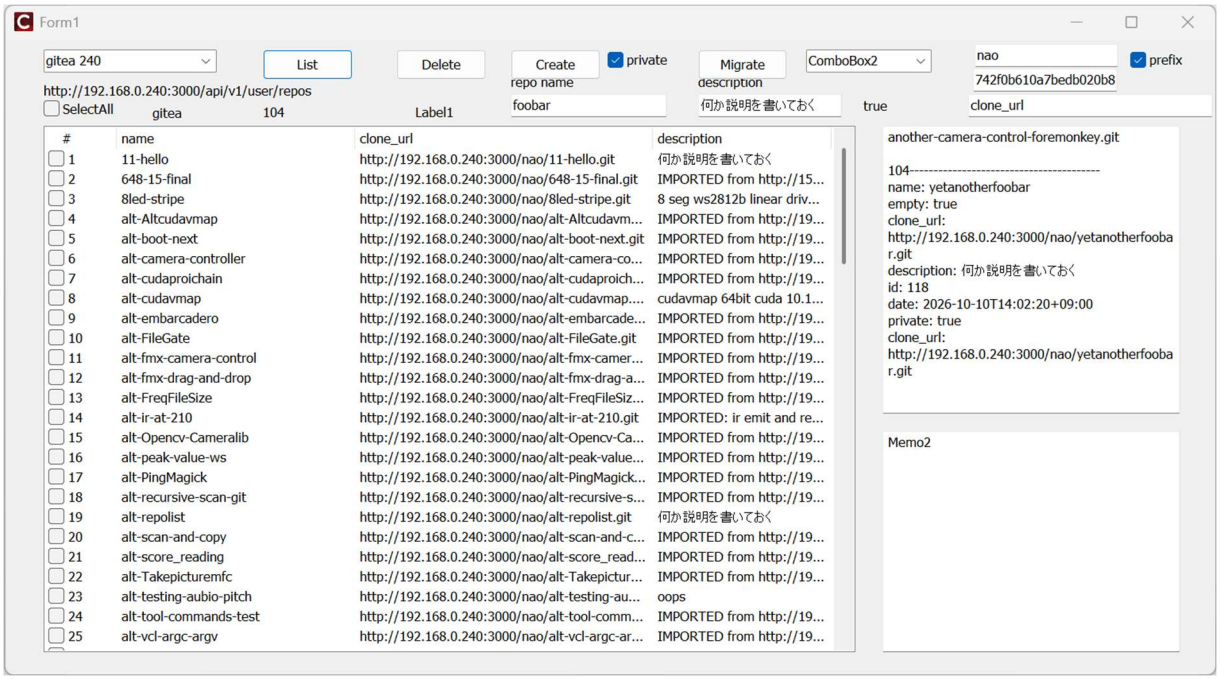
以上は最初の RRM の場合ですが、最新版 1．5 の場合は、少し構成が変わりましたので、そのあたりを付記します。

1.5 を起動すると、

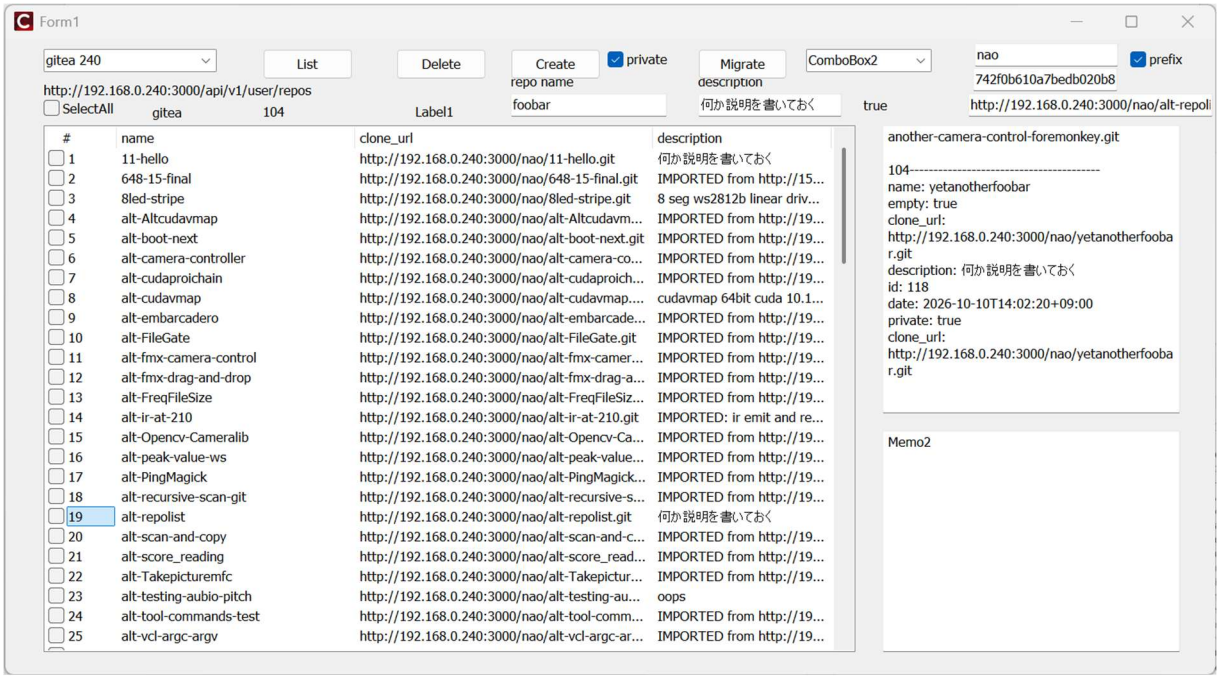


基本的な画面構成は同じですが、右上の方に付帯的な情報を表示することにしました。

また、Gitea に限定して 100 個以上のレポジトリも扱えるようにしました。要望があれば、Gitea 以外も対応するかもしれません。さて、左上のコンボボックスでサーバーを選びます。”List”ボタンをクリックすると、



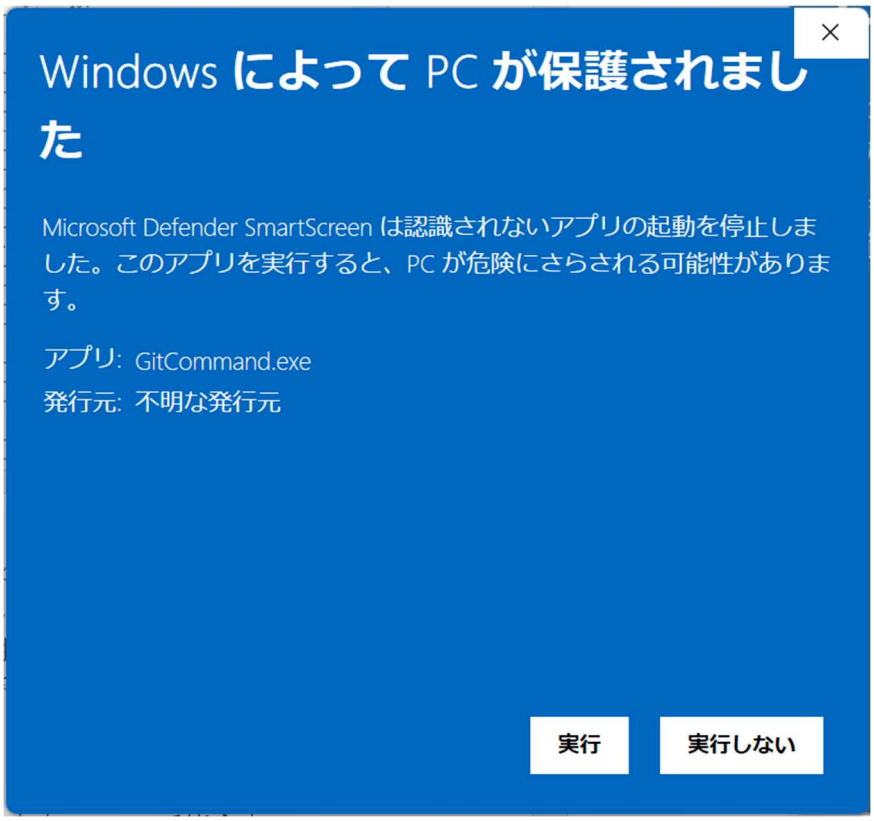
となります。ここで左のチェックボックスで選択したレポジトリの Delete も可能です。複数選択することも可能です。全選択は”SelectAll”なるチェックボックスにチェック入れてください。全 Delete とか簡単にできますので、注意して実行してください。SelectAll は”Migrate”で使いましょう。この List 状態で、項目を選択する。つまり番号のセルを選択すると、



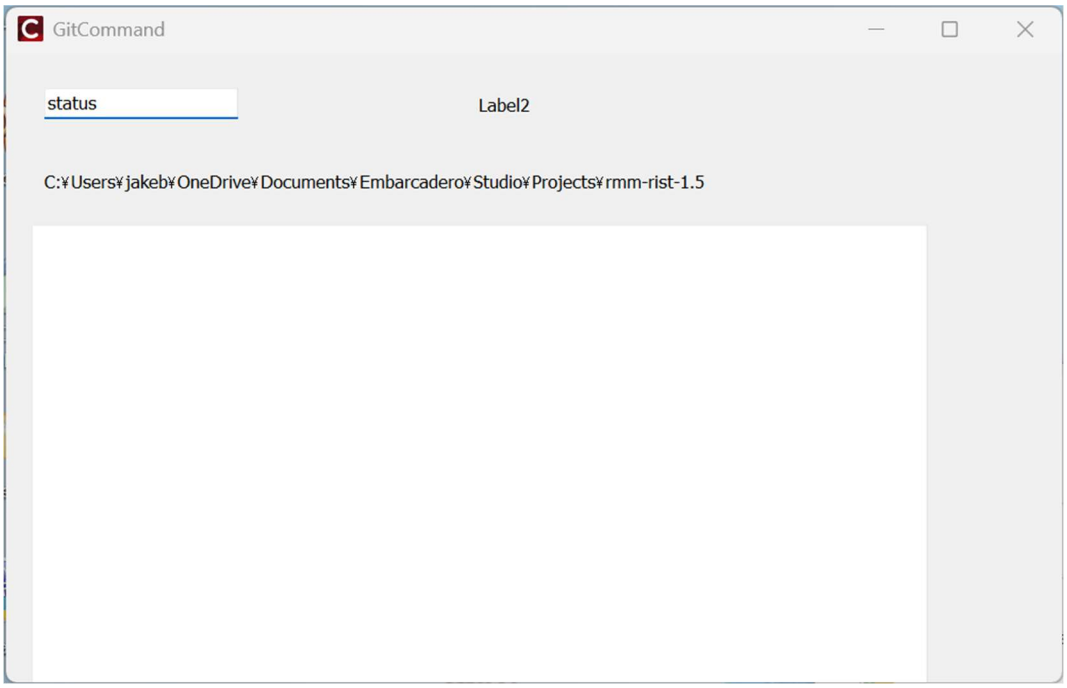
右上にこのレポジトリの clone_url が表示されていますので、この EditBox から copy and paste してもいいですし、この時点でシステムのクリップボードにも copy されていますから、単に paste(Ctrl-V)でも使えるはずです。あと、git server に認証を求められる場合には、clone_url のボックスの上に、現在選択しているサーバーの id と pat が表示されていますので、必要に応じてこれが使えます。

他のプログラムに関しては、GitCommand.exe だと、起動してそのまま return で使えます。”Doit”ボタンは廃止しました。デフォルトのコマンドは、status つまり git status ですね。起動して以下のダイアログが出たら、勇気を持って“詳細情報”をクリック。




Windows によって PC が保護されました
Microsoft Defender SmartScreen は認識されないアプリの起動を停止しました。このアプリを実行すると、PC が危険にさらされる可能性があります。
アプリ: GitCommand.exe
発行元: 不明な発行元
実行
実行しない

ここで”実行“です。


GitCommand
status
Label2
C:¥Users¥jakeb¥OneDrive¥Documents¥Embarcadero¥Studio¥Projects¥rmm-rist-1.5

最後に感想など：

Windows 版および Linux 版の curl にはお世話になりました。API を叩く例は curl の実行例で与えられていることが多く、最初の動作確認はまず curl で行うことが推奨されます。で、curl で動けば、それを C++ Builder の Indy の TIdHTTP で書くことは比較的容易で、実際にそうして開発しました。