

IBM i World 2024

IBM i コンテンツ (2024年3月版)

Visual Studioで、ILE RPG (FF RPG) がコーディングできます

- VSCodeとCode for IBM i を使ってみよう！ (応用編) -

日本アイ・ビー・エム株式会社
テクノロジー事業本部
IBM Powerテクニカルセールス



Visual Studio & Code for IBM i



Visual Studioで、ILE RPGがコーディングできます

- VSCodeとCode for IBM iを使ってみよう（応用編） -

今回は、応用編として、VSCodeとGitを活用したソースファイルのバージョンを管理する方法をご紹介します。2月の基本編のコンテンツが前提になります。併せて参照してください。

https://www.jbcc.co.jp/products/files/ibmpowercolumn_202402.pdf

目次

1. Git概説
2. VSCode とGit を使って、FFRPGを開発してみよう
 - (1) 開発環境イメージ
 - (2) PCで開発したソースをIBM iへ転送してコンパイルしてみよう
 - (3) Gitでソースのバージョン管理をしてみよう
3. 補足情報

1. Git概説 (1/2)



最初に、基礎知識として、Gitを簡単にご紹介します。

- ✓ Git (ギット) とは、Linuxで有名なリーナス・トーバルズ氏によって2005年に開発された分散バージョン管理ツールです。Linuxオペレーティングシステムの開発のために作ったようです。
- ✓ Gitとはソースコードや変更履歴を管理するために使われる、代表的な分散型バージョン管理システムです。
- ✓ Gitのようなバージョン管理システムを使えば、いつ・誰が・どのように変更したかといった変更履歴と共にファイルを保存できます。
- ✓ GitHub (ギットハブ) とは、Gitの仕組みを利用したウェブ上のバージョン管理サービスです。Gitの仕組みを採用するサービスはほかにもありますが、GitHubはそのなかでも世界中で最も利用されています。オープンソース開発のデファクトスタンダードです。

解説：

- Gitはオープンソースソフトウェアで提供されており、ライセンスは無償で利用する事ができます。
- IBM i (PASE)上でもGitは稼働します。（オープンソースなので、サポートはコミュニティにお任せです）
<https://www.ibm.com/support/pages/open-source-support-ibm-i>
- GitHubやGitLabのオンライン ツールを使用することで、使いやすいグラフィカルなユーザー インターフェースを通じて、Gitの豊富な機能を活用できます。
- GitHubは、Gitのリポジトリをホスティングしてくれるサービスです。
MicroSoft参加のGitHub社が運営しています。資料の後半で実際に使ってみます。
- GitLabは、Gitリポジトリをホスティングするソフトウェアです。IBMでも下記のURLにあるように販売しています。SaaSで利用することもできます。<https://www.ibm.com/jp-ja/products/gitlab-ultimate>

☆当資料ではGitのバージョン管理方法などの基本的な使用法は、解説していません。
詳しく知りたい方は、下記のマニュアルなどで学習してください。

<https://git-scm.com/book/ja/v2>

1. Git概説 (2/2)



- ✓ 世界のアプリケーション開発者の大半が、バージョン管理では、Gitの使用を軸にしています。
- ✓ バージョン管理にGitを採用することにより、他のプラットフォームでの開発向けに使用されているのと同じツールでIBM iアプリケーションを管理できるようになります。
- ✓ Gitは、コードのコンパイルや作成されたオブジェクトのバージョン管理は行いません。ソースコード管理のみです。
- ✓ VSCodeは、Gitと連携してソースのバージョン管理を簡単に操作できます。IBM iのFFRPGのソース管理をGitとGitHubを使ってみましょう。

解説：

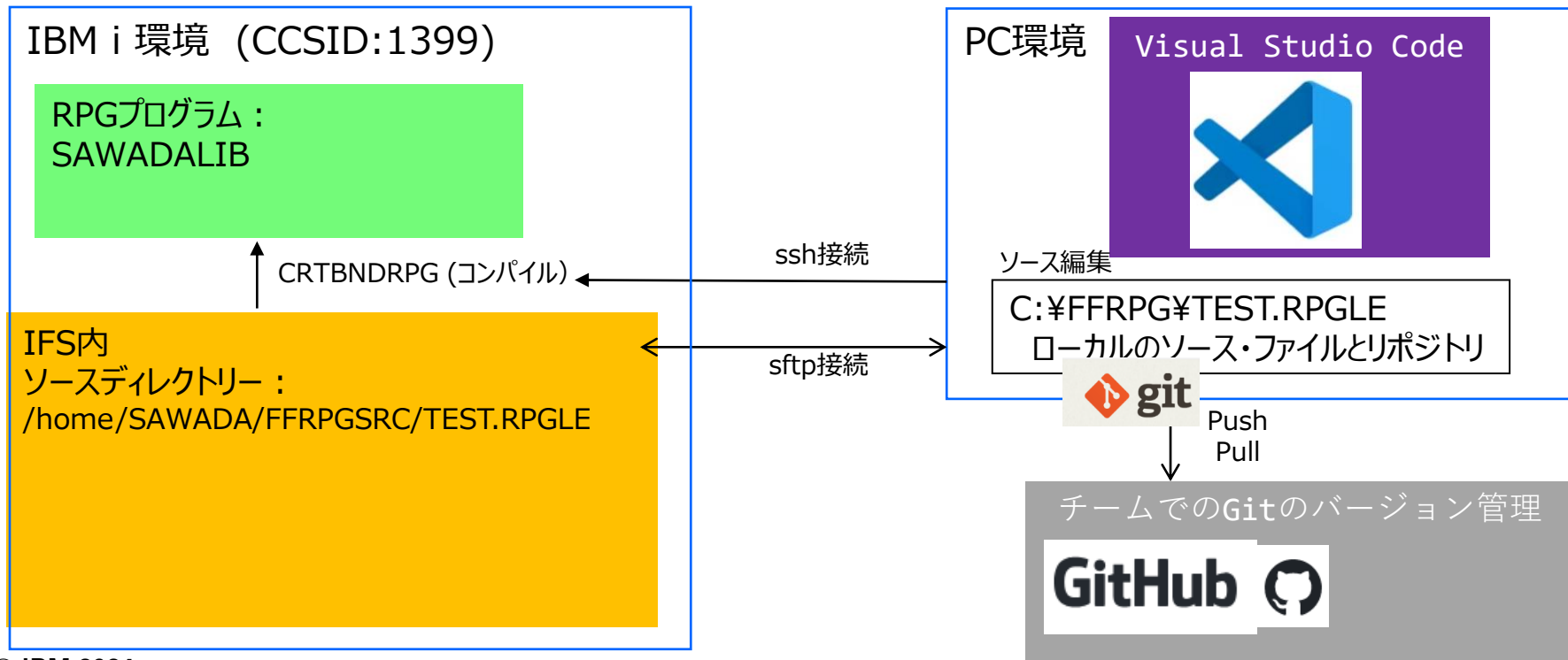
- GitはOSSなので、使用は自己責任になります。
IBMサポートがあるソースファイルのバージョン管理のしくみなどを利用したい方には、IBM i Merlinをお勧めします。
- IBM i Merlinは、バージョン管理以外に、次世代のIBM iアプリを構築およびデプロイして、ハイブリッドクラウドとマルチプラットフォームCI/CD実装の価値を最大限に引き出すことができる次世代の開発環境を提供します。
ぜひご検討ください。

<https://www.ibm.com/jp-ja/products/ibm-i-merlin>

2. VSCodeとGitを使って、FFRPGを開発してみよう

(1) 開発環境イメージ

ここからの開発環境は下記のようになります。



解説：

- ・VSCodeによるFFRPGの開発イメージです。
ここからは、実際にGitを使って、ソースのバージョン管理できる環境を作ります。
環境の作成は、下記の順です。事前にVSCodeとCode for IBM iに接続できる環境があることを前提にしています（基本編で構築済みという前提です）
まずはPCのローカルでソースファイルを管理する環境を作ります。
 - ①IBM iのIFSにソースを置くためのフォルダーを作成します。
 - ②VSCodeにSFTPの拡張機能を導入して設定します。
 - ③PCのフォルダーにサンプルコードを作成します。
 - ④IBM iのIFS上にファイルが転送されるので、コンパイルします。
 - ⑤ コンパイルがOKであれば稼働テストします。

2. VSCodeとGitを使って、FFRPGを開発してみよう

(2) PCで開発したソースをIBM iへ転送してコンパイルしてみよう

Gitによるソースのバージョン管理の前に、まず、IFS上にフォルダーを作成し、VSCodeのsftpの拡張機能を導入して、PCで作成したソースファイルをIFSに転送して、コンパイルできる環境を構築します。

①IBM iのIFS上にフォルダー作成

MD(ディレクトリー作成)
コマンドで適当に作成してください。右記は例です。

```

ディレクトリーの作成 (MD)

選択項目を入力して、実行キーを押してください。

ディレクトリー          : /home/SAWADA/FFRPGSRC
データの共通権限      : *INDIR      名前      : *INDIR, *RWX, *RW...
オブジェクトの共通権限 : *INDIR      *INDIR, *NONE, *ALL...
  値の続きは+
オブジェクトの値の監査 : *SYSVAL     *SYSVAL, *NONE, *USRPRF...
OBJ のオプションをスキャン : *PARENT     *PARENT, *YES, *NO, *CHGONLY
名前変更とリンク解除の制限 : *NO        *NO, *YES
  
```

下記のように作成できました。

```

オブジェクト・リンクの処理

ディレクトリー      : /home/SAWADA/FFRPGSRC

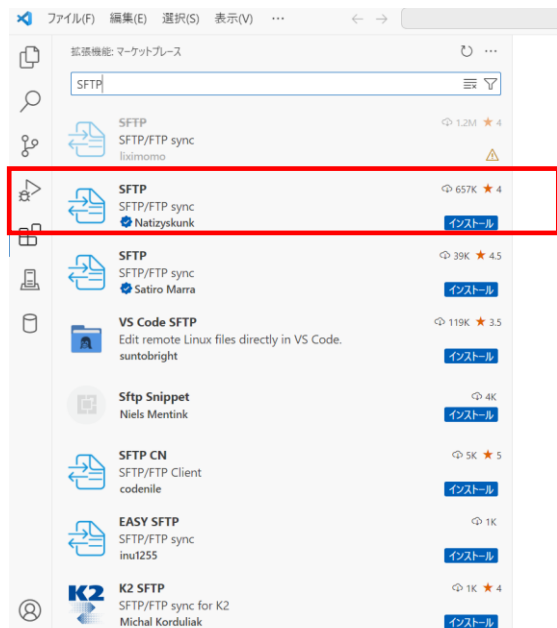
オプションを入力して、実行キーを押してください。
2= 編集    3= コピー    4= 除去    5= 表示    7= 名前の変更    8= 属性の表示
11= 現行ディレクトリーの変更 ...
  
```

2. VSCodeとGitを使って、FFRPGを開発してみよう

(2) PCで開発したソースをIBM iへ転送してコンパイルしてみよう

②VSCodeのsftp拡張機能の導入 (1/4)

VSCodeを起動して、拡張機能の欄にSFTPを入力し、下記を「インストール」する。



The screenshot shows the details page for the 'SFTP' extension by Natzyskunk. The extension version is v1.16.3, with 657,291 downloads and a 4.5-star rating. The extension is described as 'SFTP/FTP sync'. There are buttons for '無効にする' (Disable) and 'アンインストール' (Uninstall). A note states: 'この拡張機能はグローバルに有効化されています。' (This extension is globally enabled). The page also includes a description of the extension's functionality, a list of links for more information, and a category of 'Other'.

SFTP v1.16.3
Natzyskunk [natan-fourie.fr](#) | 657,291 | ★★★★★ (4.5)
SFTP/FTP sync
無効にする アンインストール ⚙️
この拡張機能はグローバルに有効化されています。

詳細 機能のコントリビューション 変更ログ

sftp sync extension for VS Code

Maintained and updated version by @Natzyskunk 🥰
(Forked from the no longer maintained [liximomo's SFTP plugin](#))

- VS Code marketplace : <https://marketplace.visualstudio.com/items?itemName=Natzyskunk.sftp>
- VSIX release : <https://github.com/Natzyskunk/vscode-sftp/releases/>

VSCode-SFTP enables you to add, edit or delete files within a local directory and have it sync to a remote server directory using different transfer protocols like FTP or SSH. The most basic setup requires only a few lines of configuration with a wide array of specific settings also available to meet the r...

カテゴリ
Other

拡張機能リソース
マーケットプレース
問題
レポートリ
ライセンス
Natzyskunk

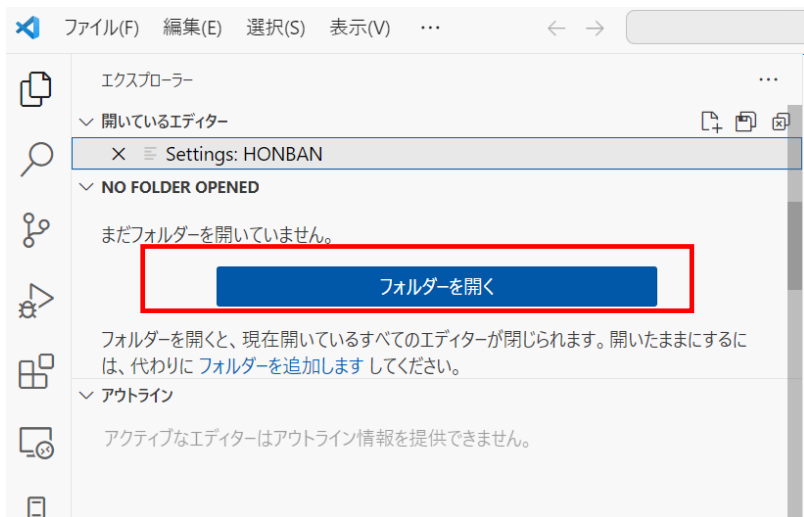
詳細情報
公開済み 2021-07-06, 12:38:18
前回のリリース日 2023-06-16, 11:29:37
識別子 natzyskunk.sftp

2. VSCodeとGitを使って、FFRPGを開発してみよう

(2) PCで開発したソースをIBM iへ転送してコンパイルしてみよう

②VSCodeのsftp拡張機能導入（2/4）

次にSFTPの設定をします。フォルダーを開くを選択先に、C:¥FFRPGを作成しておき、ここで選択する。



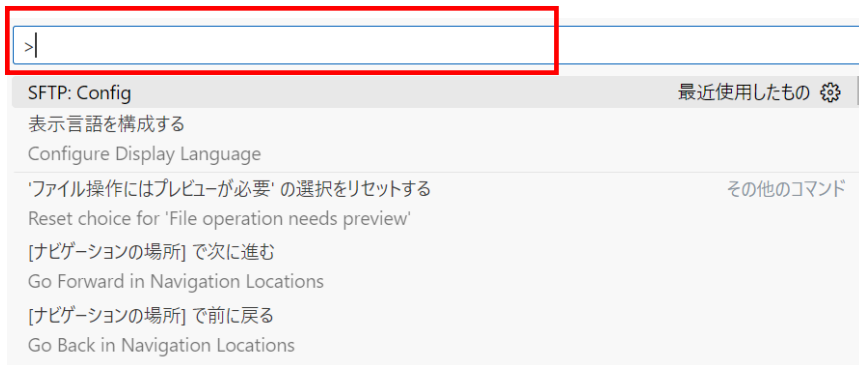
2. VSCodeとGitを使って、FFRPGを開発してみよう

(2) PCで開発したソースをIBM iへ転送してコンパイルしてみよう

②VSCodeのsftp拡張機能導入 (3/4)

次に、SFTPの設定を行います。

CTL+SHIFT+Pで、コマンドパレットの表示



sftp:configを入力すると下記が開きます。

.vscode > {} sftp.json > ...

```
1 {
2   "name": Hostname or IP address of the server.
3   "host": "localhost",
4   "protocol": "sftp",
5   "port": 22,
6   "username": "username",
7   "remotePath": "/",
8   "uploadOnSave": false,
9   "useTempFile": false,
10  "openSsh": false
11 }
12
```

2. VSCodeとGitを使って、FFRPGを開発してみよう

(2) PCで開発したソースをIBM iへ転送してコンパイルしてみよう

②VSCodeのsftp拡張機能導入 (4/4)

次に、SFTPの設定を行います。これで準備完了です。Ctrl+Sで保管します。
この後、VSCodeをIBM iへ接続しておきます。

```
.vscode > {} sftp.json > ...
```

```
1 {
2   "name": "My Server",
3   "host": "XXX.XXX.XXX.XXX",
4   "protocol": "sftp",
5   "port": 22,
6   "username": "username",
7   "password": "password",
8   "remotePath": "/home/SAWADA/FFRPGSRC",
9   "uploadOnSave": true,
10  "useTempFile": false,
11  "openSsh": false
12 }
```

☆主なパラメータの意味は、下記を参照してください。

name: サーバーの名前を決めます (任意の名前)

host: サーバーのホストです。 (ホスト名またはサーバーのIPアドレスなど)

protocol: "ftp"、または"sftp"を指定します。

port: 接続するポートを指定します。 (ftpは"21"、**sftpは"22"**が多いです)

username: ftp/sftpのユーザー名

password: ユーザーのパスワード

remotePath: ワークスペースのルートディレクトリと一致するサーバーのディレクトリパスを指定します。 (IFS上のパスを設定してください)

uploadOnSave : trueにすると、フォルダに保存するとIFSへ自動転送される。

その他の詳細オプション設定は下記に記述あります。

<https://github.com/Natizyskunk/vscode-sftp/wiki/Common-Configuration>

2. VSCodeとGitを使って、FFRPGを開発してみよう

(2) PCで開発したソースをIBM iへ転送してコンパイルしてみよう

③PCに、サンプルプログラムを作成（1/3）

ここでC:¥FFRPG¥TEST.RPGLE

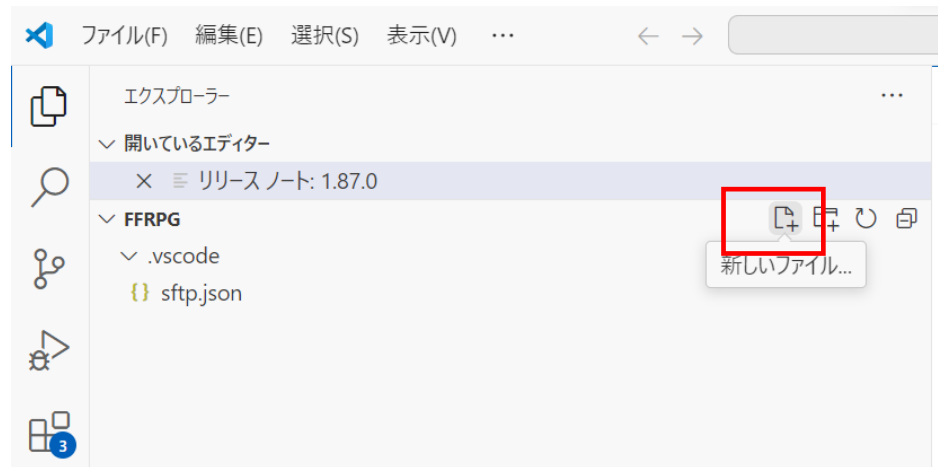
という名前のプログラムを作成します。

FFRPGのソースコードは、下記のコードを参照

(単に画面にHello FFRPG・・・を表示するだけです)

```
**free
ctl-opt dftname(TEST);
ctl-opt dftactgrp(*no) main(main);
dcl-proc main;
  dsply 'Hello FFRPG IBM i World2024';
end-proc;
```

こVSCodeのエクスプローラーで、FFRPGの横の新しいファイルのアイコンをクリック

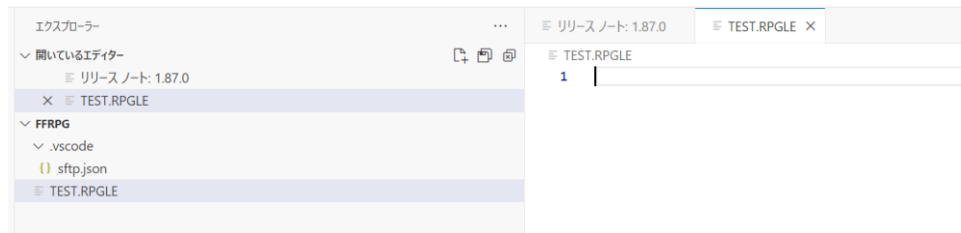


2. VSCodeとGitを使って、FFRPGを開発してみよう

(2) PCで開発したソースをIBM iへ転送してコンパイルしてみよう

③PCに、サンプルプログラムを作成 (2/3)

ファイル名に、TEST.RPGLE を入力



下記のように入力して、Ctrl + S で保存する

≡ TEST.RPGLE > ...

```
1  **free
2  ctl-opt dftname(TEST);
3  ctl-opt dftactgrp(*no) main(main);
4  dcl-proc main;
5      dsply 'Hello FFRPG IBM i World2024';
6  end-proc;
7
```

2. VSCodeとGitを使って、FFRPGを開発してみよう

(2) PCで開発したソースをIBM iへ転送してコンパイルしてみよう

③PCに、サンプルプログラムを作成 (3/3)

PC上で保存すると、同時に、SFTPでIBM iのIFSに転送されます。

(sftp.jsonで指定した、**remotePathに保存されます。**)

IFS上に同じファイルがある確認してください。

下記のように、/home/xxxxxx/FFRPGSRCに、TEST.RPGLE
が転送されています。

下記のように見えます。



home > SAWADA > FFRPGSRC > TEST.RPGLE > ...

```
1  **free
2  ctl-opt dftname(TEST);
3  ctl-opt dftactgrp(*no) main(main);
4  dcl-proc main;
5  |   dsply 'Hello FFRPG IBM i World2024';
6  end-proc;
7
```

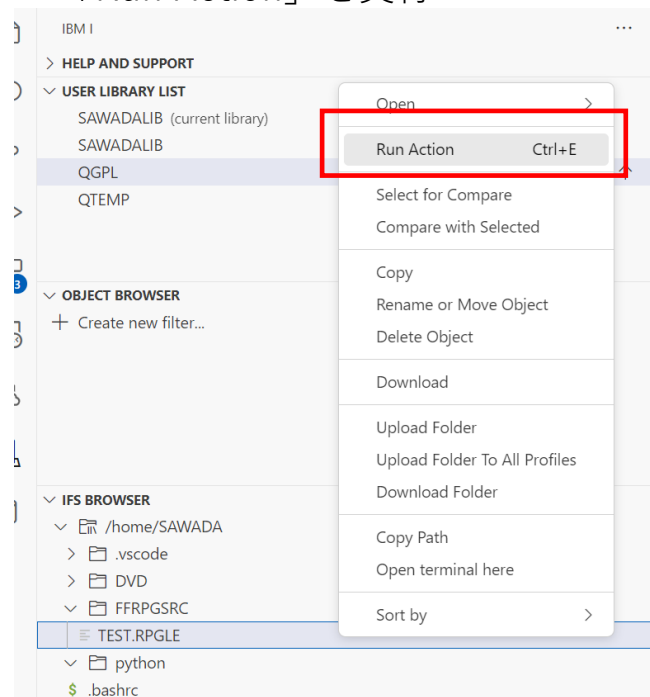
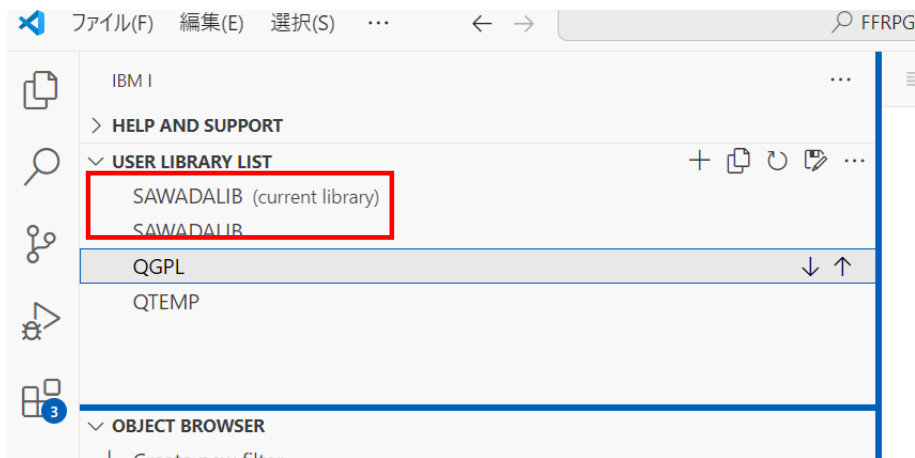

2. VSCodeとGitを使って、FFRPGを開発してみよう

(2) PCで開発したソースをIBM iへ転送してコンパイルしてみよう

④ソースをコンパイルします (1/4)

ライブラリーリストにプログラムを置くライブラリーを追加して、現行ライブラリーにします。
(方法は、基本編を参照)

IFS上のTEST.RPGLEのソースを右クリックして、「Run Action」を実行



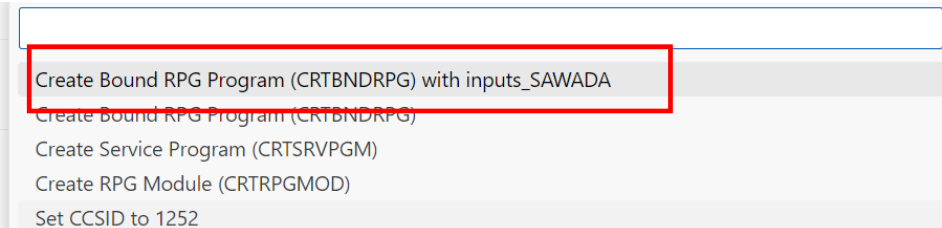
2. VSCodeとGitを使って、FFRPGを開発してみよう

(2) PCで開発したソースをIBM iへ転送してコンパイルしてみよう

④ ソースをコンパイルします (2/4)

デフォルトで用意されたコンパイルのコマンドが表示される。

一番上の[CRTBNDRPG]を選択する。



Create Bound RPG Program (CRTBNDRPG) with inputs_SAWADA

Create Bound RPG Program (CRTBNDRPG)

Create Service Program (CRTSRVPGM)

Create RPG Module (CRTRPGMOD)

Set CCSID to 1252

コンパイル先のライブラリー、プログラム名、IFS上のソースパスを確認して、[Execute]をクリックするとコンパイルされます。

Create Bound RPG Program (CRTBNDRPG) with inputs_SAWADA

Build library

SAWADALIB

Object Name

TEST

Source path

/home/SAWADA/FFRPGSRC/TEST.RPGLE

Execute

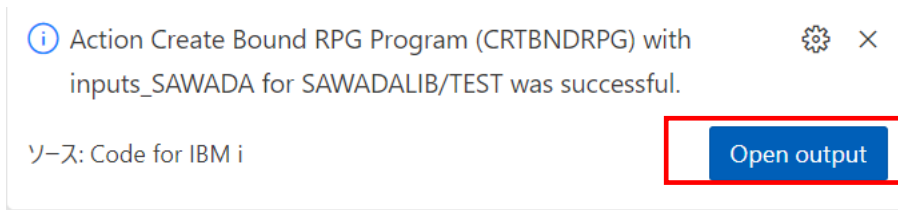
Cancel

2. VSCodeとGitを使って、FFRPGを開発してみよう

(2) PCで開発したソースをIBM iへ転送してコンパイルしてみよう

④ソースをコンパイルします (3/4)

下記のようなポップアップが出るので、
[Open output]
をクリックして、コンパイルリストを確認します。



コンパイル・リストが表示されます。エラーなしでコンパイルできていることを確認してください。

```

CRTBNDRPG PGM(SAWADALIB/TEST) SRCSTMF('/home/SAWADA/FFRPGSRC/TEST.RPGL') OPTION(*EVENTF) DE
OPD4090: 印刷装置PRT01が見つからない。出力待ち行列がライブラリー-QSPLのQPRINTに変更されました。
CP12119: AUTおよびUSRPRFパラメーター値は無視されました。
CP12121: 置き換えられたオブジェクトTEST(QSPL/QSPL)に移動されました。
RNS8304: プログラムTESTがライブラリーSAWADALIBに入れられました。最高の重大度は00。24/02/29の16:25:0
CPC0904: データRECORDがライブラリーQTEMPに作成された。
5770WS V7R6M0 220415 RN IBM ILE RPG SAWADALIB/TEST HONBAN 24/02
コマンド . . . . . CRTBNDRPG
組み元 . . . . . SAWADA
プログラム . . . . . TEST
ライブラリー . . . . . SAWADALIB
テキスト記述 . . . . . *SRCMBRTXT
ソース・ストリーム・ファイル . . . . . /home/SAWADA/FFRPGSRC/TEST.RPGL
OS/2 . . . . . 1208
ターゲットOS/2 . . . . . 5035
テキスト記述 . . . . .
最後の変更 . . . . . 24/02/29 16:11:09
生成重大度レベル . . . . . 10
制御の活動化グループ . . . . . *YES
コンパイラ・オプション . . . . . *XREF *GEN *NCSOCLVL *SHOWCOPY
*EXPDOS *EXT *NCSHOWSKP *NCSRCSTMT
*DEBUGIO *UNREF *EVENTF
デバッグ・ビュー . . . . . *SOURCE
デバッグ暗号鍵 . . . . . *NONE
出力 . . . . . *PRINT
最適化レベル . . . . . *NONE
ソース・リストの字下げ . . . . . *NONE
タイプ変換オプション . . . . . *NONE
レコード . . . . . 6
仕様 . . . . . 3
データ・レコード . . . . . 0
注記 . . . . . 0
***** 最終の要約の終わり *****
プログラムTESTがライブラリーSAWADALIBに入れられました。最高の重大度は00。24/02/29の16:25:06に
作成されました。
***** コンパイルの終わり *****
Fetching errors for SAWADALIB/TEST.
ターミナルはタスクで再利用されます。閉じるには任意のキーを押してください。

```

2. VSCodeとGitを使って、FFRPGを開発してみよう

(2) PCで開発したソースをIBM iへ転送してコンパイルしてみよう

⑤IBM i上でプログラムを実行してみます。

- ・ライブラリーリストに、コンパイル済みのプログラムを入れたライブラリー（例.SAWADALIB）を追加
- ・「CALL TEST」で、下記のようにメッセージが表示されればOKです。

```
ADDLIBLE SAWADALIB  
ライブラリー SAWADALIB がライブラリー・リストに追加された。  
CALL TEST  
DSPLY Hello FFRPG IBM i World2024
```

2. VSCodeとGitを使って、FFRPGを開発してみよう

(3) Gitでソースのバージョン管理をしてみよう

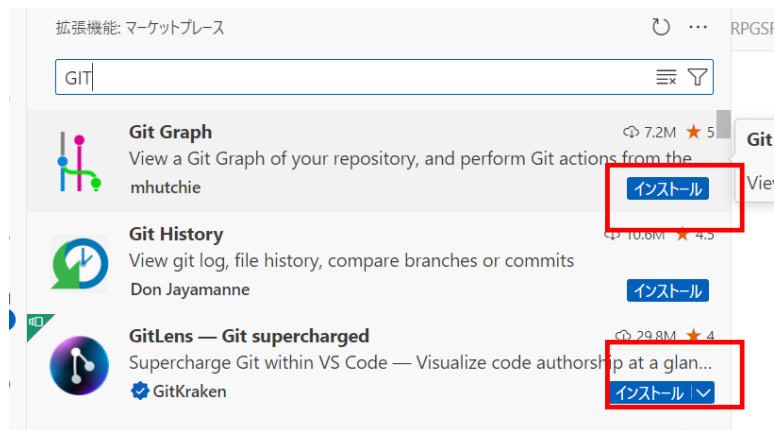
ここからは、このPCで開発したソースファイルをGitでバージョン管理してみます。

①VSCodeに、Gitに必要な拡張機能の導入します。

Gitを使うのに必須なVSCodeの拡張機能（GitlensとGit Graph）を導入します。

VSCodeの拡張機能の欄に、[Git] と入力すると、下記のように多数の拡張機能が表示されます。

そこから、GitlensとGit Graphをインストールしてください。



解説：

- ・ここからは、PCへのGitの導入と設定を行います。下記の順に導入していきます。

- ①VSCode + Gitに必要拡張機能（GitlensとGitGraph）の導入

- ②GitのWindows版の導入

- ③ローカルPCのGitでバージョン管理してみよう（ローカルリポジトリの利用）

- ④GitHubでバージョン管理してみよう

- Gitlens :誰がこのコードを書いたのか、このコミットは誰により行われたのか、コードはどのように変わっていったのかなどを、VS Codeの内部で簡単に参照できるツール。

- GitGraph: Gitのコミットの履歴を確認したり、ブランチの操作をしたりすることができるツール。

2. VSCodeとGitを使って、FFRPGを開発してみよう

(3) Gitでソースのバージョン管理をしてみよう

PCで開発したソースファイルをGitでバージョン管理してみます。

②GitのWindows版の導入 (GitはOSSなので、無償) (1/2)

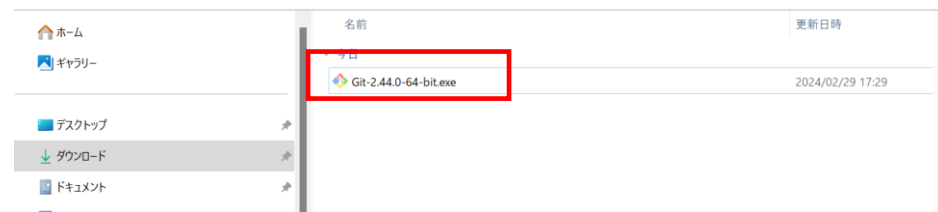
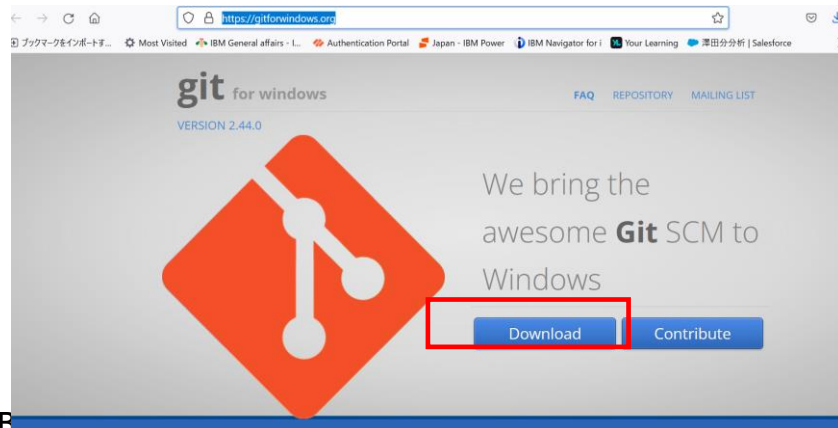
細な導入手順は、下記のurlを参照してください。

<https://qiita.com/T-H9703EnAc/items/4fbe6593d42f9a844b1c>

ここでは、概要を記述します。下記のurlでdownloadを選択

<https://gitforwindows.org/>

ダウンロードフォルダーにあるGitファイルをクリックして実行してください。

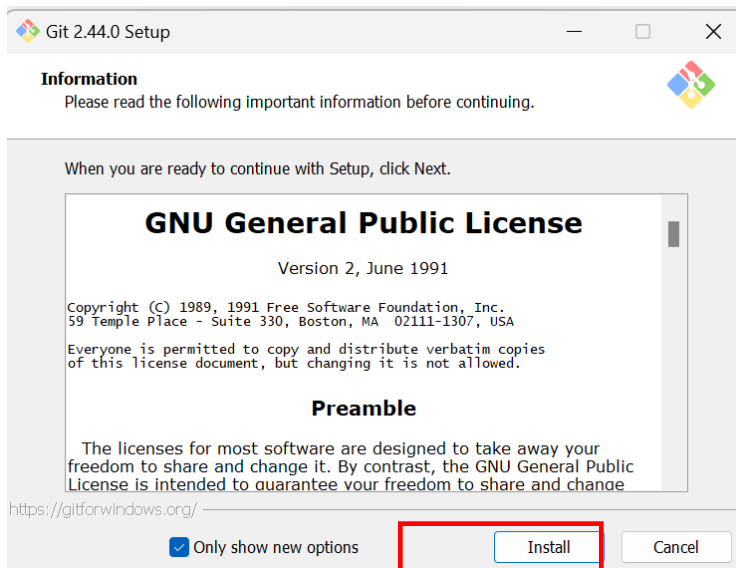


2. VSCodeとGitを使って、FFRPGを開発してみよう

(3) Gitでソースのバージョン管理をしてみよう

②GitのWindows版の導入 (2/2)

下記のライセンス確認に同意して、installをクリック(後の工程は省略)



2. VSCodeとGitを使って、FFRPGを開発してみよう

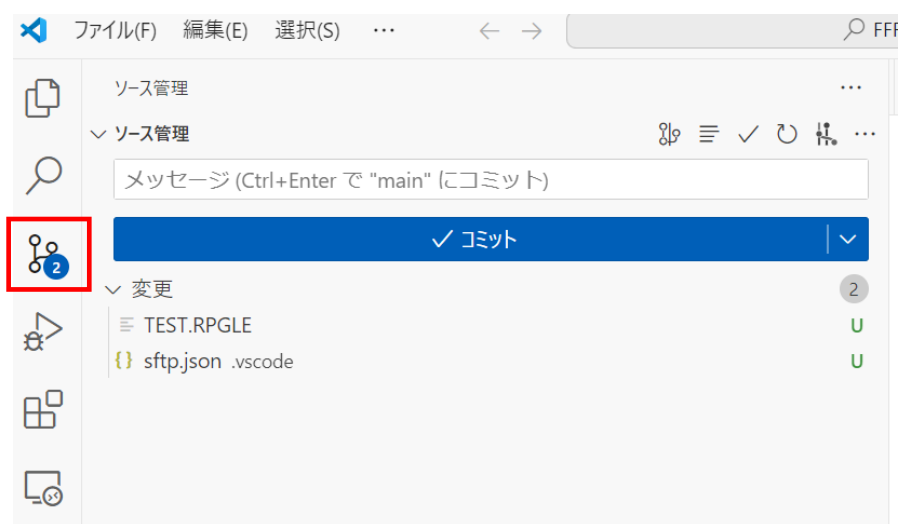
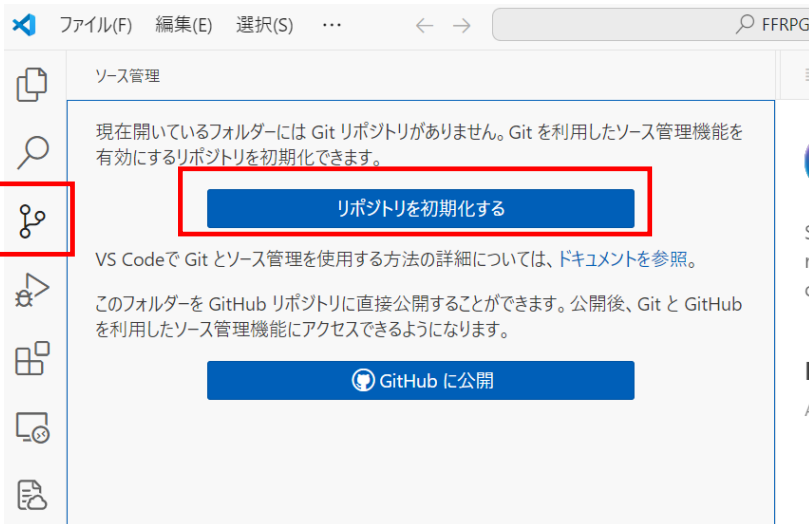
(3) Gitでソースのバージョン管理をしてみよう

③ ローカルPCのGitでバージョン管理をしてみよう (1/10)

ローカルリポジトリの作成

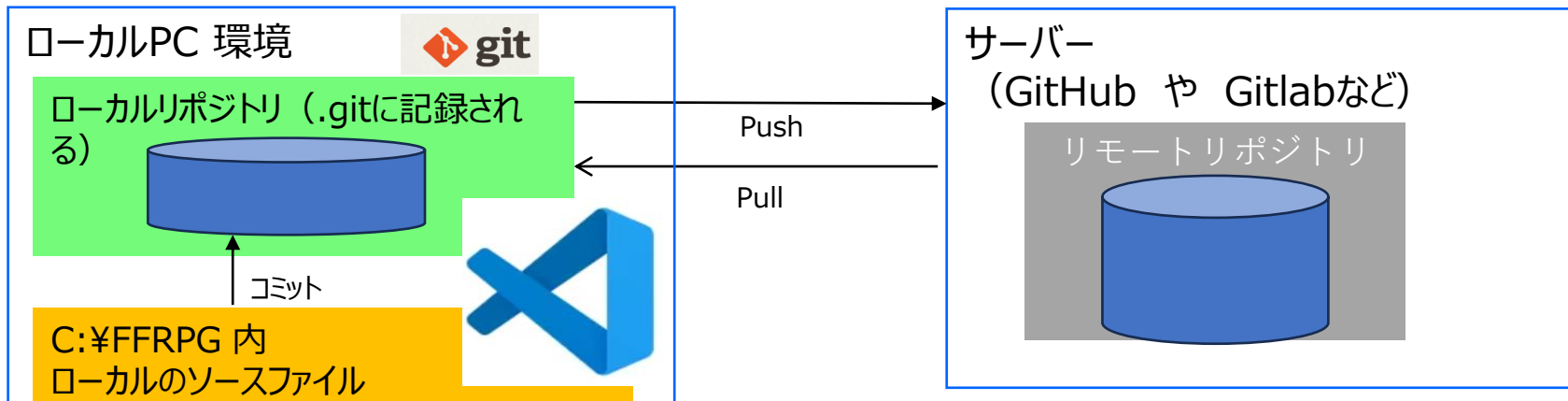
下記のVscodeのGitのアイコンをクリックし、
[リポジトリを初期化する]、をクリックする。
FFRPGフォルダーに、ローカルリポジトリが作成されます。

ソース管理欄に、まだリポジトリに登録されていないファイルが表示されるようになります。
Gitアイコンには、コミットされていないファイル数が表示されます。(ここでは下記のように2と表示されます)



解説：

- ・ Gitのソース管理のしくみは、下記のようなイメージになります。
リポジトリがソースの履歴管理データです。
- ✓ リポジトリは、ソースファイルやディレクトリの状態を保存する場所です。変更履歴を管理したいディレクトリ（フォルダ）などをリポジトリの管理下に置くことで、そのディレクトリ内のファイルなどの変更履歴を記録することができます。
- ✓ リポジトリは自分のPC内にある「ローカルリポジトリ」とサーバなどネットワーク上にある「リモートリポジトリ」の2箇所にあります。基本的にローカルリポジトリで作業を行い、その作業内容をリモートリポジトリへプッシュする流れで行います。
- ✓ コミットすると、PC内のローカルリポジトリに変更履歴が書き込まれます。サーバにPushすると、リモートリポジトリに履歴が同期されます。



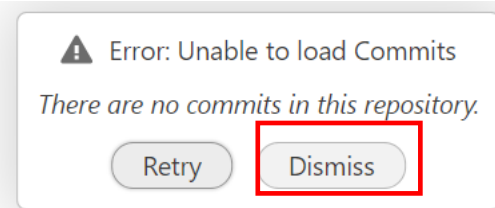
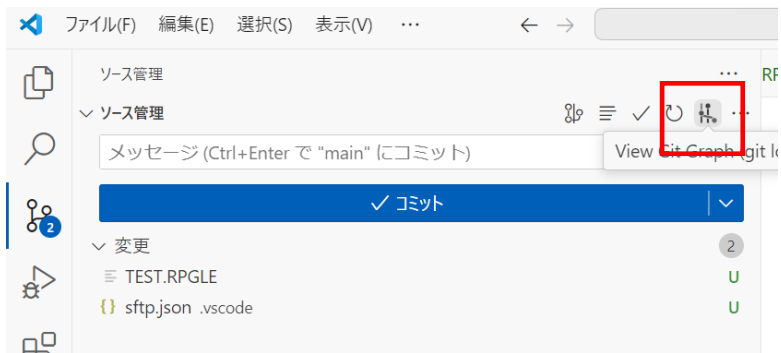
2. VSCodeとGitを使って、FFRPGを開発してみよう

(3) Gitでソースのバージョン管理をしてみよう

③ ローカルPCのGitでバージョン管理をしてみよう (2/10)

次に、コミットするユーザーを登録する。
下記のVscodeのGit Graphのアイコンをクリックする

下記の画面が表示されたら、
「Dismiss」を選択

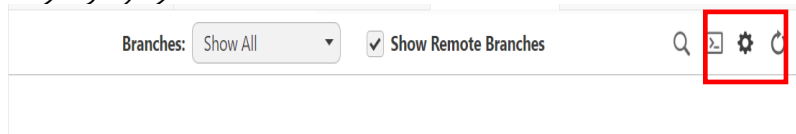


2. VSCodeとGitを使って、FFRPGを開発してみよう

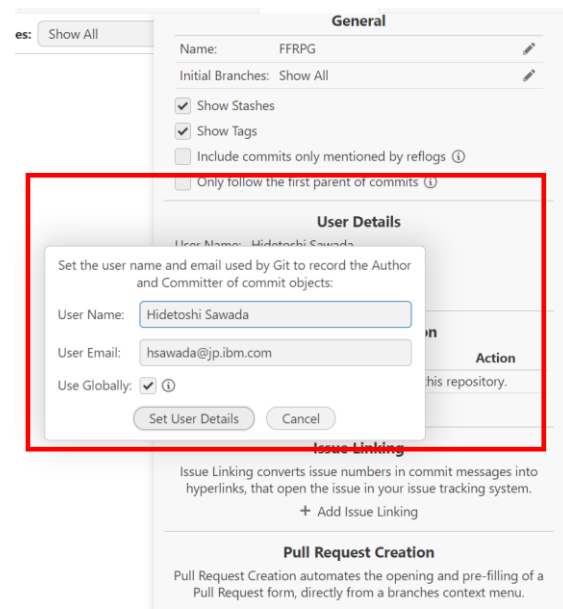
(3) Gitでソースのバージョン管理をしてみよう

③ ローカルPCのGitでバージョン管理をしてみよう (3/10)

下記画面で、Repository Setting (⚙) をクリック



下記の画面が表示されるので、「User Name」と「User Email」を入力し、Set User Detailsをクリック(下記は例です)



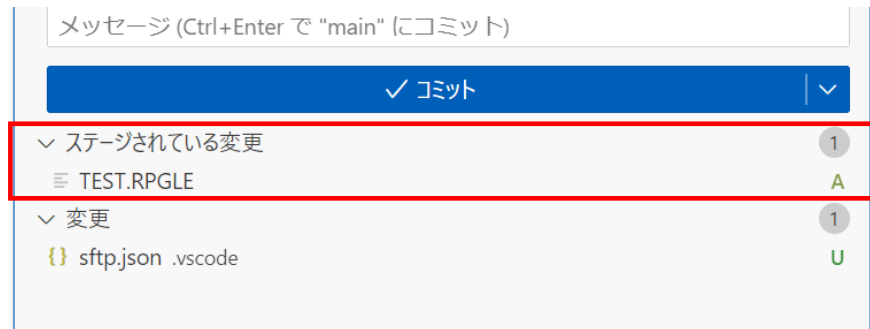
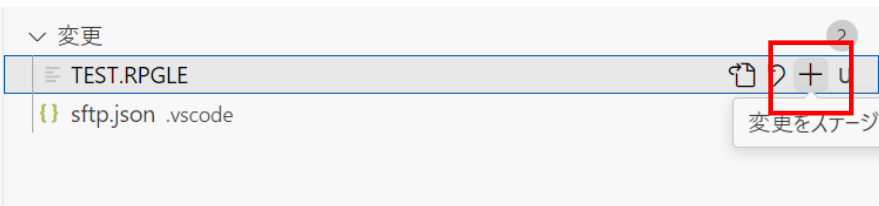
2. VSCodeとGitを使って、FFRPGを開発してみよう

(3) Gitでソースのバージョン管理をしてみよう

③ローカルPCのGitでバージョン管理してみよう (4/10)

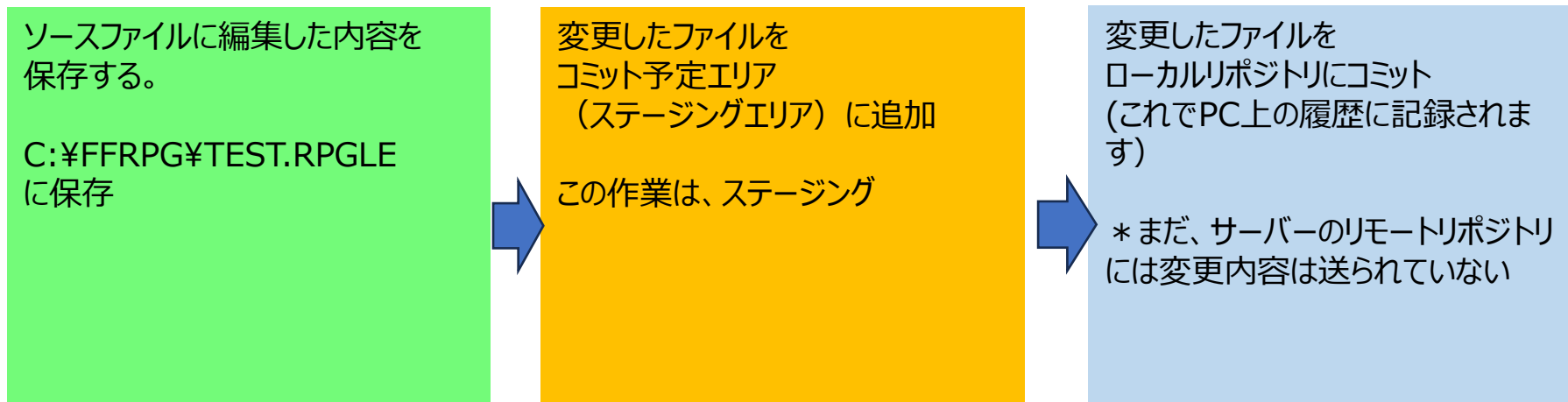
次に、TEST.RPGLEのソースをステージングエリアに追加します。ファイルの右にある+ボタンをクリック。

ステージされている変更に入りました。



解説：

- ・ Gitの基本（ローカルリポジトリへのコミットの流れ）
保存したファイルを直接コミットするのではなく、コミット予定エリアである「ステージングエリア」に追加する作業があります。
このステージングは、コミットしたい変更を選別する作業になります。




詳しくは、下記を参照してください。

<https://git-scm.com/book/ja/v2/%E4%BD%BF%E3%81%84%E5%A7%8B%E3%82%81%E3%82%8B-Git%E3%81%AE%E5%9F%BA%E6%9C%AC>

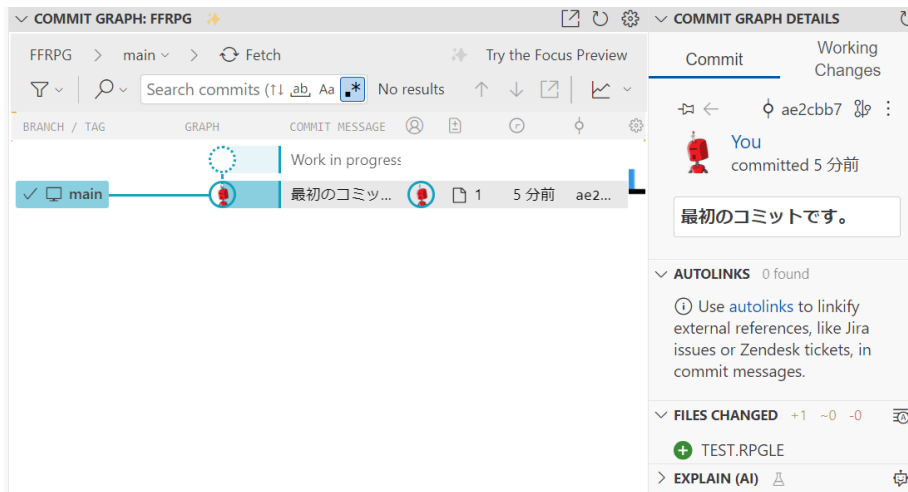
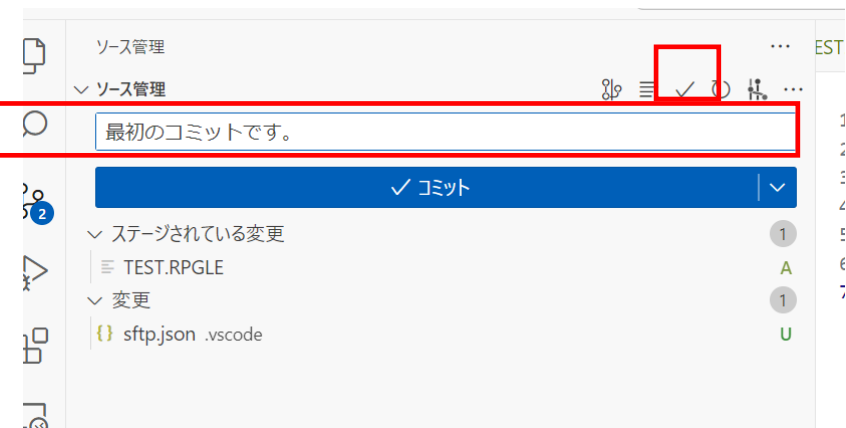
2. VSCodeとGitを使って、FFRPGを開発してみよう

(3) Gitでソースのバージョン管理をしてみよう

③ ローカルPCのGitでバージョン管理をしてみよう (5/10)

次にコミットします。
メッセージを入力して、コミット・ボタン
をクリック。
コミット

Git Graphアイコン (Graph) をクリックすると
コミット状態を確認できます。これでPC上の
ローカルリポジトリに変更が記録されました。



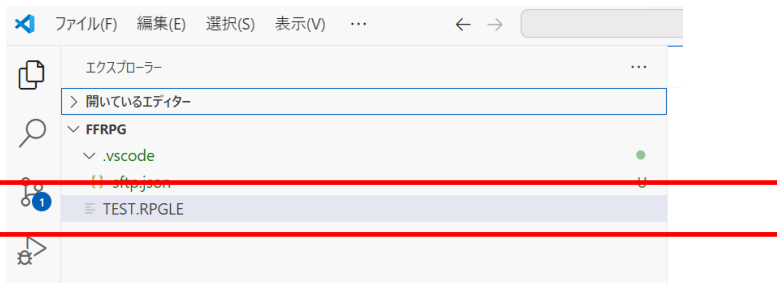
2. VSCodeとGitを使って、FFRPGを開発してみよう

(3) Gitでソースのバージョン管理をしてみよう

③ローカルPCのGitでバージョン管理してみよう（6/10）

ソースを修正して、ソースのバージョン管理できる
しくみを見てみましょう。

VSCodeのエクスプローラーにあるFFRPGを
表示し、TEST.RPGLE をクリックする



下記のように、ソースを変更する。
(dsplyにVERSION2を追加しただけです。)
変更後に、保管 (Ctrl+S)する。

```
TEST.RPGLE > main
1  **free
2  ctl-opt dftname(TEST);
3  ctl-opt dftactgrp(*no) main(main);
4  dcl-proc main;
5  dsply 'Hello FFRPG IBM i World2024 VERSION2';
6  end-proc;
7
```

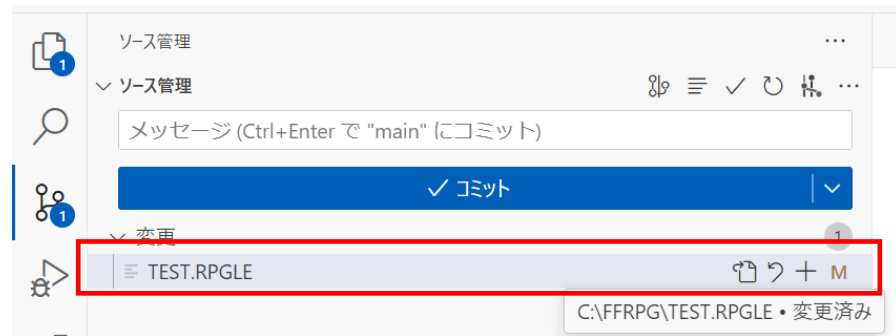
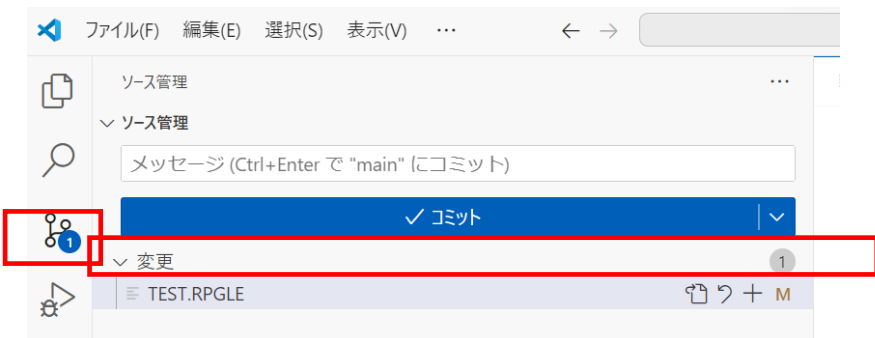

2. VSCodeとGitを使って、FFRPGを開発してみよう

(3) Gitでソースのバージョン管理をしてみよう

③ローカルPCのGitでバージョン管理をしてみよう (7/10)

ソースの保存が完了すると、ソース管理に、下記のように、変更に1が表示されます。これはコミットされていないファイルが1つあることを表します。

次に、TEST.RPGLEをクリックします。

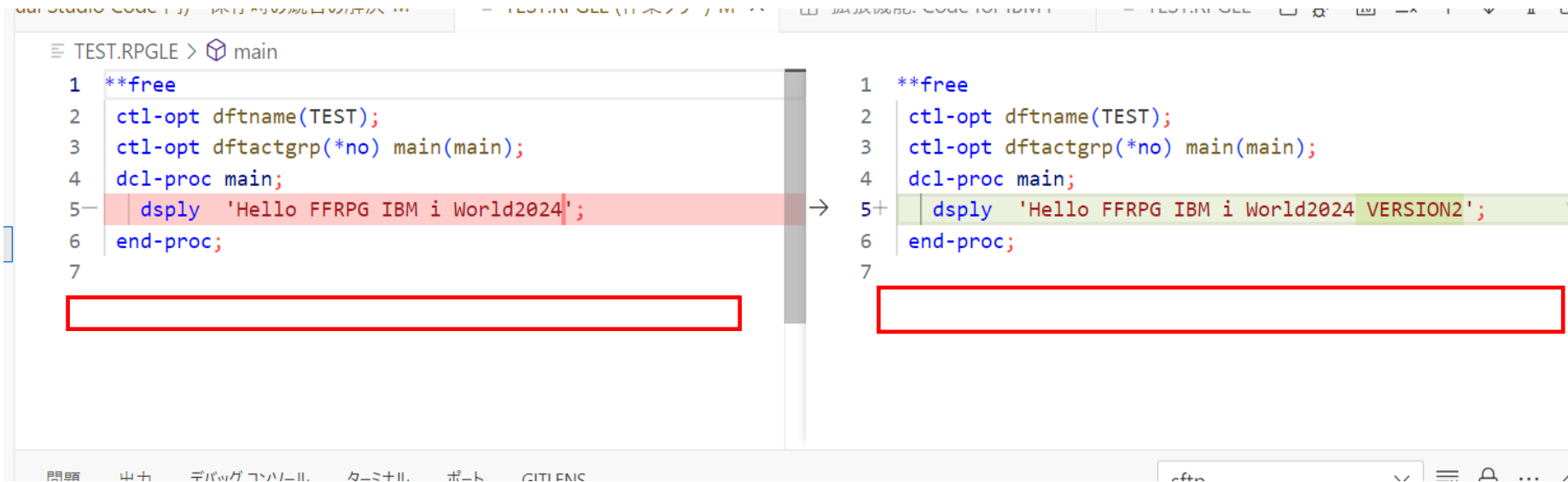


2. VSCodeとGitを使って、FFRPGを開発してみよう

(3) Gitでソースのバージョン管理をしてみよう

③ローカルPCのGitでバージョン管理をしてみよう (8/10)

下記のように、修正前のコード（左）と修正後のコード（右）が比較して確認できます。
このように、ソースのファイルのどの部分が変更されたを自動的に管理でき、確認できます。



```
TEST.RPGLE > main
1 **free
2   ctl-opt dftname(TEST);
3   ctl-opt dftactgrp(*no) main(main);
4   dcl-proc main;
5-  dsply 'Hello FFRPG IBM i World2024';
6   end-proc;
7

1 **free
2   ctl-opt dftname(TEST);
3   ctl-opt dftactgrp(*no) main(main);
4   dcl-proc main;
5+  dsply 'Hello FFRPG IBM i World2024 VERSION2';
6   end-proc;
7
```

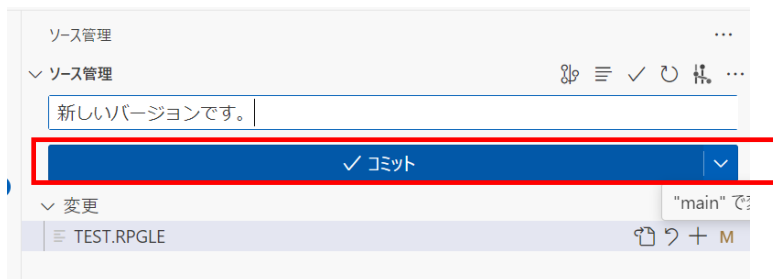
2. VSCodeとGitを使って、FFRPGを開発してみよう

(3) Gitでソースのバージョン管理をしてみよう

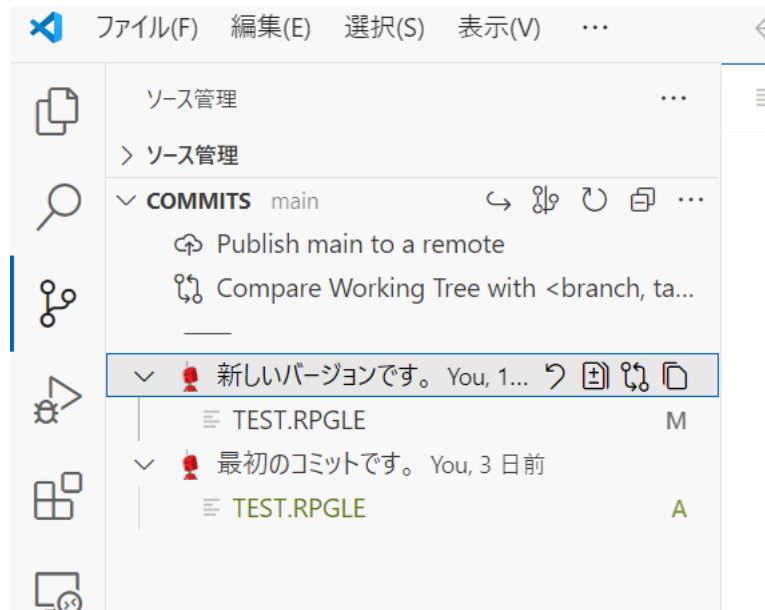
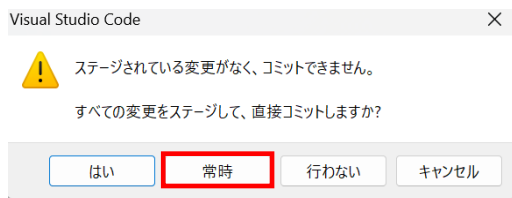
③ ローカルPCのGitでバージョン管理をしてみよう (9/10)

それではコミットして、ローカルリポジトリを更新してみましょう。下記のように、適当にコメントを入力して、コミットをクリックします。

ソース管理の下の「COMMITTS」欄を見ると、いつ、誰か、どのソースをコミットしたのかわかります。



下記のメッセージが表示されます。ここれ常時をクリックすると、ステージングをパスできます。常時をクリックしてください。これでコミット完了です。



2. VSCodeとGitを使って、FFRPGを開発してみよう

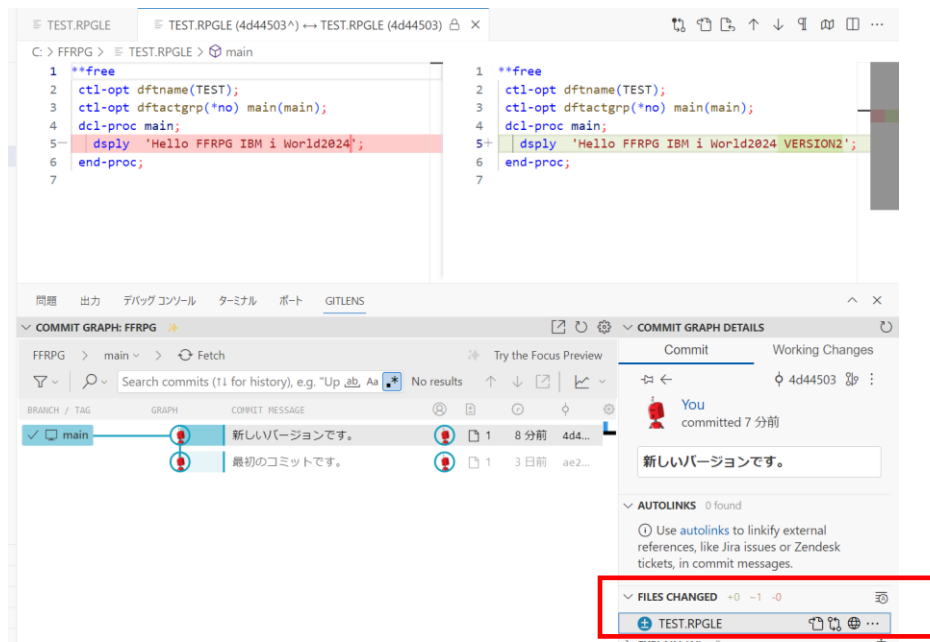
(3) Gitでソースのバージョン管理をしてみよう

③ ローカルPCのGitでバージョン管理をしてみよう (10/10)

Git Graphのアイコンをクリックするとより詳細な情報を得ることができます。



例えば、下記のようなソースの差分情報はいつでも簡単に入手できます。自分1人だけでソースを管理するのであれば、これで完了です。



2. VSCodeとGitを使って、FFRPGを開発してみよう

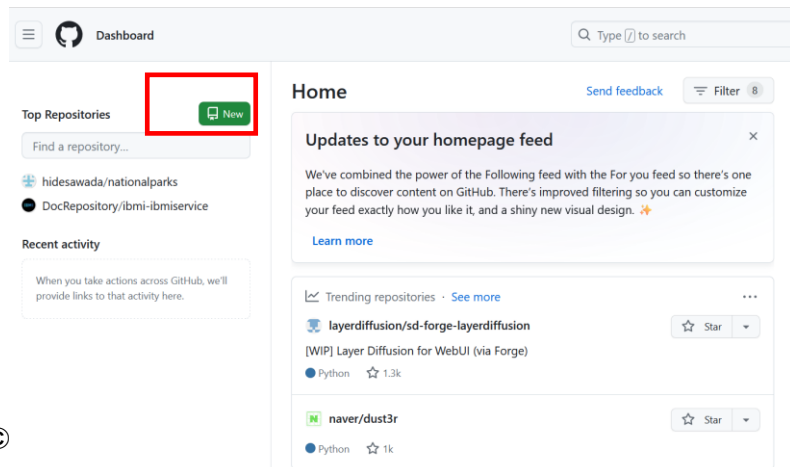
(3) Gitでソースのバージョン管理をしてみよう

④GitHubでバージョン管理をしてみよう (1/5)

基幹システムの開発の場合は、チームで開発する機会が多いので、GitHubを使ってリモトリポジトリでバージョン管理をしてみましょう。

GitHubへのユーザー登録はここでは省略します。下記のサインオン後の画面から、まずリモトリポジトリを作成するところから開始します。「New」をクリックします。

下記のように、リポジトリ名をFFRPGで、Publicにします。一番下の「create repository」をクリック。
(注意：*Publicにすると一般公開されます)



The screenshot shows the GitHub Dashboard. In the top-left corner, there is a 'New' button with a plus icon, which is highlighted with a red box. Below it, the 'Top Repositories' section lists 'hidesawada/nationalparks' and 'DocRepository/ibmi-ibmiservice'. The 'Recent activity' section shows a message about activity links. The main content area is titled 'Home' and features an 'Updates to your homepage feed' section with a 'Learn more' link. Below that, there are 'Trending repositories' including 'layerdiffusion/sd-forge-layerdiffusion' and 'naver/dust3r'.

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Required fields are marked with an asterisk (*).

Owner * / repository name

FFRPG is available.

Great repository names are short and memorable. Need inspiration? How about [special-spork](#) ?

Description (optional)

- Public**
Anyone on the internet can see this repository. You choose who can commit.
- Private**
You choose who can see and commit to this repository.

Initialize this repository with:

Add a README file
This is where you can write a long description for your project. [Learn more about READMEs.](#)

Add .gitignore

Create repository

解説：

- GitHubの使用条件が、貴社のITセキュリティー等貴社の規則と合致するかをアセスしたうえでのご利用をお勧めいたします。

- GitHubへのユーザー登録方法は下記を参考にしてください。

<https://docs.github.com/ja/get-started/start-your-journey/creating-an-account>



[thub](https://docs.github.com/ja/get-started/start-your-journey/creating-an-account)

- GitHub社のサイトには右記のような価格体系になっています。個人で使う場合には、プライベート環境でも無償で利用できます。

料金などで分からないことがある場合は下記を参照してください。

<https://docs.github.com/ja/billing>

プランご紹介
支払い体系を選んでください

Free
個人および小規模ビジネス向けの無償のフリープラン

Team
個人より小規模ビジネス向けの無償のフリープラン
\$48 / 44 / ユーザーあたり
Freeプランを始める

Enterprise
セキュリティ、コンプライアンス、最新技術
\$252 / 231 / ユーザーあたり
Enterpriseプランを始める

Freeプランの特徴

- 無制限のプライベートリポジトリ
- ホスティングとバックアップの無制限
- CI/CD: 1,000回/月
- 200MB GitHub Packages ストレージ
- 個人/チームプロフィール、公開ページ
- GitHub Copilot Access

Teamプランの特徴

- Freeプランに含まれているすべての機能
- GitHub Collectionsへのアクセス
- 企業向けプラン
- Pull Requestに署名のセキュリティ
- Pull Requestのワークフロー
- コード所有者
- 秘密レビュー機能
- GitHub Pages: 2GB
- CI/CD: 最大 1,000回/月
- 200MB GitHub Packages ストレージ
- ワークフローの無制限
- ワークフローのマネージメント機能
- 署名機能、サードパーティーからの署名
- 200MB GitHub Packages ストレージ
- 200MB GitHub Packages ストレージ

Enterpriseプランの特徴

- Freeプランに含まれているすべての機能
- Enterprise Managed Users
- セキュリティ強化されたユーザープロフィール
- 組織のOrganizational Policyを管理するEnterprise Account
- 秘密保護から一歩とワークフロー
- 署名の強化
- SOX、SOC2のComplianceレポート
- FullBSP Trusted Authority to Operate (ATO)
- SAMLシングルサインオン
- GitHub Connect
- CI/CD: 最大 10,000回/月
- 200MB GitHub Packages ストレージ
- 200MB GitHub Packages ストレージ

2. VSCodeとGitを使って、FFRPGを開発してみよう

(3) Gitでソースのバージョン管理をしてみよう

④GitHubでバージョン管理をしてみよう（2/5）

リモートリポジトリが作成され、下記の画面が表示されます。

The screenshot shows two main options for setting up the repository:

- Set up GitHub Copilot:** Use GitHub's AI pair programmer to autocomplete suggestions as you code. A button labeled "Get started with GitHub Copilot" is visible.
- Add collaborators to this repository:** Search for people using their GitHub username or email address. A button labeled "Invite collaborators" is visible.

ローカルリポジトリとリモートリポジトリを紐づけしていきます。Quick setupの右側にあるコピーボタンをクリックします。

The screenshot shows the "Quick setup" section with the following details:

- Quick setup — if you've done this kind of thing before**
- Buttons for "Set up in Desktop", "HTTPS", and "SSH". The URL is `https://github.com/hidesawada/FFRPG.git`.
- Text: "Get started by [creating a new file](#) or [uploading an existing file](#). We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#)."
- ...or create a new repository on the command line**
- Terminal commands:


```
echo "# FFRPG" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/hidesawada/FFRPG.git
git push -u origin main
```
- ...or push an existing repository from the command line**

This is a close-up of the "Quick setup" section. A red box highlights the text "Quick setup — if you've done this kind of thing before" and the URL `https://github.com/hidesawada/FFRPG.git`. Another red box highlights the copy icon (two overlapping squares) to the right of the URL. A blue arrow points from the text in the previous block to this copy icon.

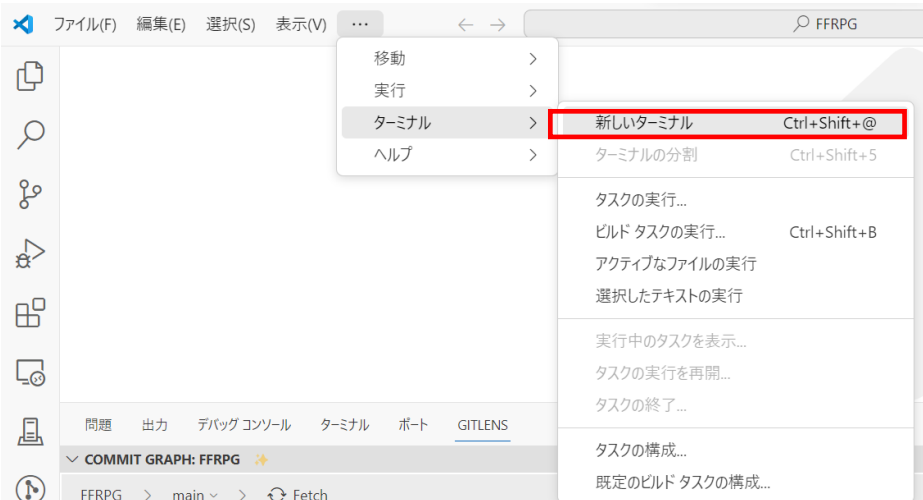
2. VSCodeとGitを使って、FFRPGを開発してみよう

(3) Gitでソースのバージョン管理をしてみよう

④GitHubでバージョン管理をしてみよう (3/5)

VSCodeを起動して、上部のターミナルの「新しいターミナル」をクリックします。

下記の画面になります。



2. VSCodeとGitを使って、FFRPGを開発してみよう

(3) Gitでソースのバージョン管理をしてみよう

④GitHubでバージョン管理をしてみよう (4/5)

下記のコマンドを入力します。

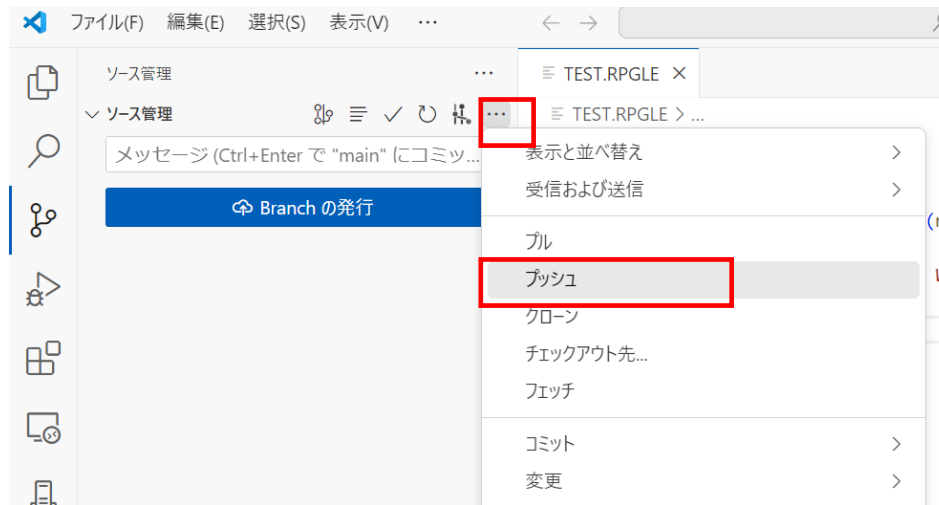
```
git remote add origin 「(2/5) でコピーした内容」
```

例. 下記の画面のようになります。これで実行キーすると、ローカルリポジトリとリモートリポジトリの紐づけは完了です。



```
問題 出力 デバッグコンソール ターミナル ポート GITLENS
PS C:\FFRPG> git remote add origin https://github.com/hidesawada/FFRPG.git
```

VSCodeで、リモートリポジトリへpushしてみましよう。ソース管理で、「…」をクリックし、プッシュを選択します。

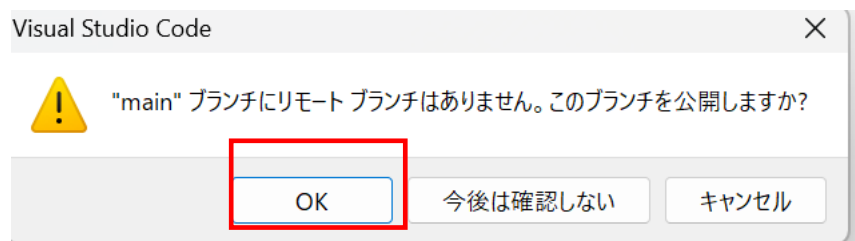


2. VSCodeとGitを使って、FFRPGを開発してみよう

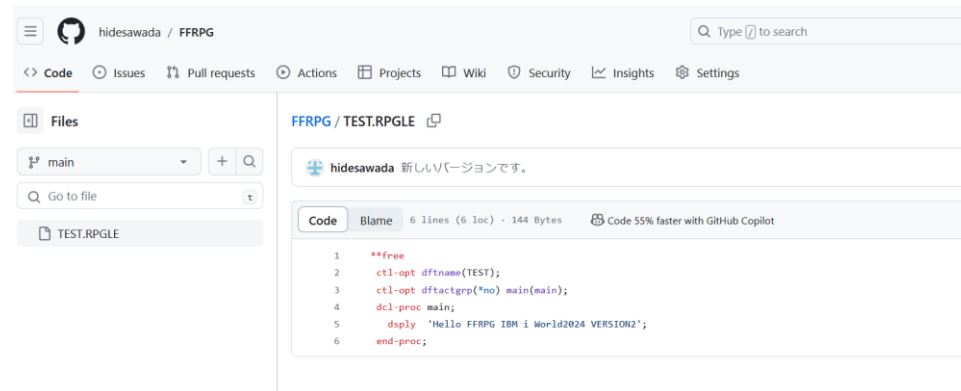
(3) Gitでソースのバージョン管理をしてみよう

④GitHubでバージョン管理をしてみよう (5/5)

下記のメッセージが表示された場合は、OKを選択



これでpushが完了です。リモートリポジトリにコピーされて、GitHubでソースの履歴管理がされます。



以上で、VSCode + Gitによるバージョン管理は完了です。

Gitはソースファイルのバージョン管理のスタンダードなので、ぜひご利用ください。

解説：

- ・ GitHubの使用法の詳細についてはこの資料では割愛します。

詳細については、GitHubのマニュアルを参照してください。

<https://docs.github.com/ja>

あるいは下記のような初心者向けのガイドが分かり易いです。

<https://qiita.com/renesisu727/items/248cb9468a402c622003>

<https://www.kagoya.jp/howto/it-glossary/develop/github/>

- ・ GitHubは、IBM iの様々なツールも公開されています。ぜひ使ってみてください。

下記は、SQLのサンプルを公開しています。

https://www.i-cafe.info/column/product/kokoi_github

下記はプログラムのサンプルを公開しています。

<https://github.com/topics/ibmi>

OSSのリンク集も管理されています。

<https://ibm.github.io/ibmi-oss-resources/>

3. 補足情報

1. iMagazine社 **IBM iユーザーに捧げるVisual Studio Code入門**
<https://www.imagazine.co.jp/feat-visual-studio-code-part1/>
<https://www.imagazine.co.jp/feat-visual-studio-code-part2/>
<https://www.imagazine.co.jp/feat-visual-studio-code-part3/>
<https://www.imagazine.co.jp/feat-visual-studio-code-part4/>
<https://www.imagazine.co.jp/feat-visual-studio-code-part5/>
2. FF RPG(フリーフォームRPG)サンプル
<https://qiita.com/gomAnomalocaris/items/0322634830ccec49e794>
3. Code for IBM iのチュートリアル動画
<https://www.youtube.com/playlist?list=PLNI31cqBafCp-ml8WqPeriHWLD1bkg7KL>
4. Qiita記事 VSCodeでFFRPGをシュッと始める
<https://qiita.com/KOINULIB/items/b7259d93ec772effdef7>
5. Qiita記事 VSCodeとGitの連携、GitHubへのpushまで (初心者向け)
<https://qiita.com/midori-game/items/0abf0013a70790518738>

IBM i 関連情報

IBM i ポータル・サイト

<https://ibm.biz/ibmijapan>

i Magazine (IBM i 専門誌。春夏秋冬の年4回発刊)

<https://www.imagazine.co.jp/IBMi/>

IBM i World 2023 オンデマンド・セミナー

<https://ibm.biz/ibmiworld2023>

IBM i World 2022 オンデマンド・セミナー

<https://video.ibm.com/recorded/132423205>

月イチIBM Power情報セミナー「IBM Power Salon」

<https://ibm.biz/power-salon>

IBM i 関連セミナー・イベント

<https://ibm.biz/powerevents-i>

IBM i Club (日本のIBM i ユーザー様のコミュニティー)

<https://ibm.biz/ibmiclubjapan>

IBM i 研修サービス (i-ラーニング社提供)

<https://www.i-learning.jp/service/it/iseriess.html>

IBM Power Systems Virtual Server 情報

<https://ibm.biz/pvsjapan>

IBM i 情報サイト iWorld

<https://ibm.biz/iworldweb>

IBM i サポートロードマップ

<https://public.dhe.ibm.com/systems/support/planning/transfer/IBM.i.Support.Roadmap.pdf>

IBM i 7.5 技術資料

<https://www.ibm.com/docs/ja/i/7.5>

IBM Power ソフトウェアのダウンロードサイト (ESS)

<https://ibm.biz/powerdownload>

Fix Central (HW・SWのFix情報提供)

<https://www.ibm.com/support/fixcentral/>

IBM My Notifications (IBM IDの登録 [無償] が必要)

「IBM i」 「9009-41G」 などPTF情報の必要な製品を選択して登録できます。

<https://www.ibm.com/support/mynotifications>

IBM i 各バージョンのライフサイクル

<https://www.ibm.com/support/pages/release-life-cycle>

IBM i 以外のSWのライフサイクル (個別検索)

<https://www.ibm.com/support/pages/lifecycle/>



ワークショップ、セッション、および資料は、IBMによって準備され、IBM独自の見解を反映したものです。それらは情報提供の目的のみで提供されており、いかなる読者に対しても法律的またはその他の指導や助言を意図したのではなく、またそのような結果を生むものでもありません。本資料に含まれている情報については、完全性と正確性を期するよう努力しましたが、「現状のまま」提供され、明示または暗示にかかわらずいかなる保証も伴わないものとします。本資料またはその他の資料の使用によって、あるいはその他の関連によって、いかなる損害が生じた場合も、IBMは責任を負わないものとします。本資料に含まれている内容は、IBMまたはそのサプライヤーやライセンス交付者からいかなる保証または表明を引き出すことを意図したもので、IBMソフトウェアの使用を規定する適用ライセンス契約の条項を変更することを意図したものでなく、またそのような結果を生むものでもありません。

本資料でIBM製品、プログラム、またはサービスに言及していても、IBMが営業活動を行っているすべての国でそれらが使用可能であることを暗示するものではありません。本資料で言及している製品リリース日付や製品機能は、市場機会またはその他の要因に基づいてIBM独自の決定権をもっていつでも変更できるものとし、いかなる方法においても将来の製品または機能が使用可能になると確約することを意図したものではありません。本資料に含まれている内容は、読者が開始する活動によって特定の販売、売上高の向上、またはその他の結果が生じると述べる、または暗示することを意図したもので、またそのような結果を生むものでもありません。パフォーマンスは、管理された環境において標準的なIBMベンチマークを使用した測定と予測に基づいています。ユーザーが経験する実際のスループットやパフォーマンスは、ユーザーのジョブ・ストリームにおけるマルチプログラミングの量、入出力構成、ストレージ構成、および処理されるワークロードなどの考慮事項を含む、数多くの要因に応じて変化します。したがって、個々のユーザーがここで述べられているものと同様の結果を得られると確約するものではありません。

記述されているすべてのお客様事例は、それらのお客様がどのようにIBM製品を使用したか、またそれらのお客様が達成した結果の実例として示されたものです。実際の環境コストおよびパフォーマンス特性は、お客様ごとに異なる場合があります。

IBM、IBM ロゴ、ibm.com、Db2、Rational、Power、POWER8、POWER9、AIXは、世界の多くの国で登録されたInternational Business Machines Corporationの商標です。

他の製品名およびサービス名等は、それぞれIBMまたは各社の商標である場合があります。

現時点での IBM の商標リストについては、www.ibm.com/legal/copytrade.shtml をご覧ください。

インテル、Intel、Intel ロゴ、Intel Inside、Intel Inside ロゴ、Centrino、Intel Centrino ロゴ、Celeron、Xeon、Intel SpeedStep、Itanium、およびPentium は Intel Corporation または子会社の米国およびその他の国における商標または登録商標です。

Linuxは、Linus Torvaldsの米国およびその他の国における登録商標です。

Microsoft、Windows、Windows NT および Windows ロゴは Microsoft Corporationの米国およびその他の国における商標です。

ITILはAXELOS Limitedの登録商標です。

UNIXはThe Open Groupの米国およびその他の国における登録商標です。

JavaおよびすべてのJava関連の商標およびロゴは Oracleやその関連会社の米国およびその他の国における商標または登録商標です。