

# バイナリデータ表示ツール bi の使い方

## 目次

バイナリデータ表示ツール bi の使い方.....	1
1. ライセンスキー.....	2
1-1. ライセンスキーを登録する方法(GUI).....	2
1-2. ライセンスキーを登録する方法(CUI).....	3
1-3. ライセンスキーを削除する方法(GUI).....	5
1-4. ライセンスキーを削除する方法(CUI).....	6
2. 使い方.....	7
2-1. 右クリックメニューからの表示方法.....	7
2-2. GUIでの使用方法.....	9
2-3. コマンドラインでの使用方法.....	12
3. 暗号化機能について.....	22

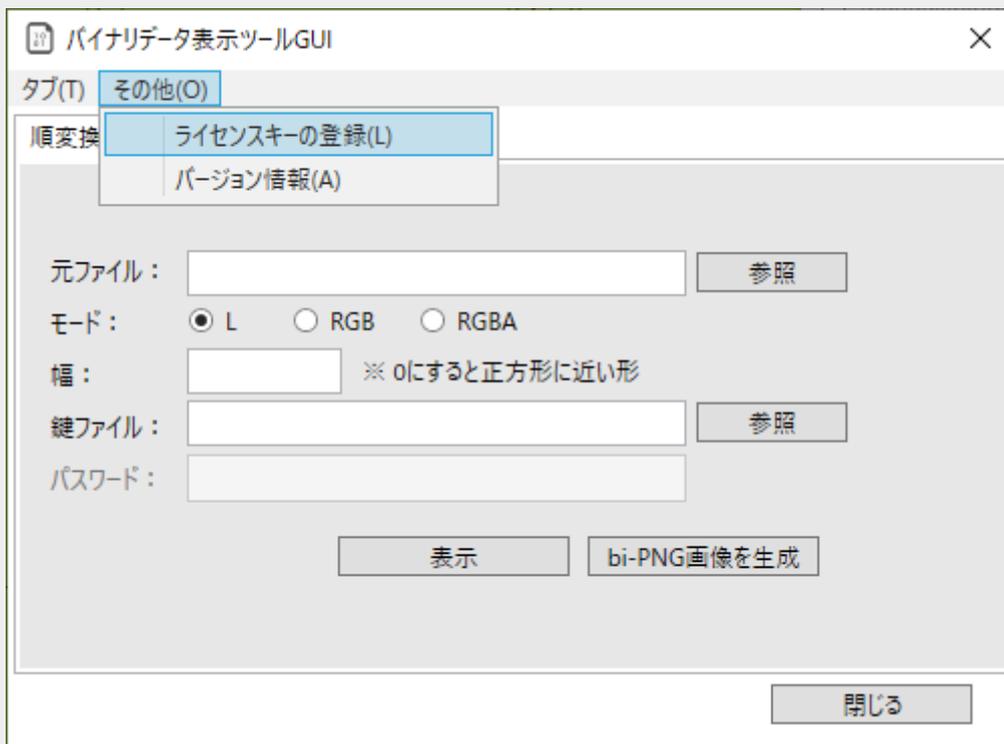
# 1. ライセンスキー

試用後、このソフトウェアを気に入った場合は、ライセンスキーを購入し、プログラムでライセンスキーを登録してください。  
ライセンスキーを登録すると、使用制限が解除されパスワード付き暗号化機能が使用できるようになります。

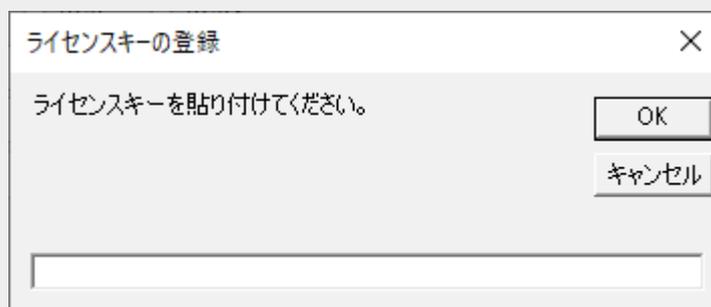
## 1-1. ライセンスキーを登録する方法(GUI)

GUI でライセンスキーを登録する方法です。

メニュー” その他” →” ライセンスキーの登録” を左クリック

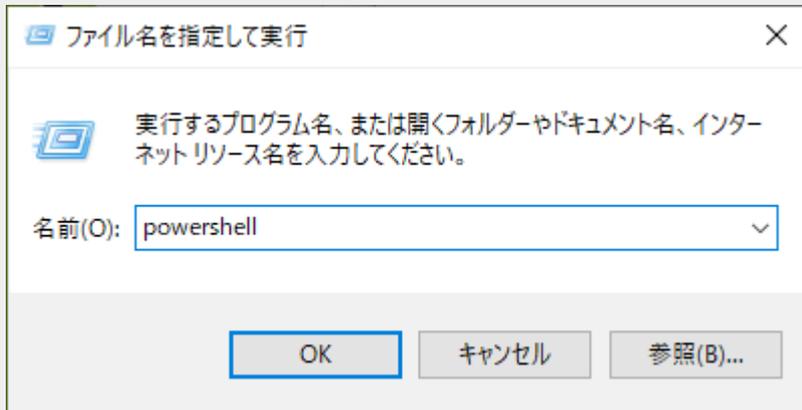


ライセンスキーを貼り付けて” OK” ボタンを押す



## 1-2. ライセンスキーを登録する方法(CUI)

コマンドラインでライセンスキーを登録する方法です。  
キーボードでWindows キー+R を押して、powershell と入力してエンターキーを押し、PowerShell を開きます。



bi -l と入力しエンターキーを押します。  
すると、ライセンスキー入力画面になるので、渡されたライセンスキーを入力してエンターキーを押します。



正しいライセンスキーが登録されても、間違ったライセンスキーが登録されても以下の“ライセンスキーを登録しました。”という文字が表示されます。

A screenshot of a Windows PowerShell terminal window. The title bar reads "Windows PowerShell". The terminal content shows the following sequence of commands and output:

```
PS C:\testdir> bi -l  
ライセンスキーを入力してください (input licensekey) > LICENSEKEY-HOEHOE-MOHEMOHE  
ライセンスキーを登録しました。  
PS C:\testdir> █
```

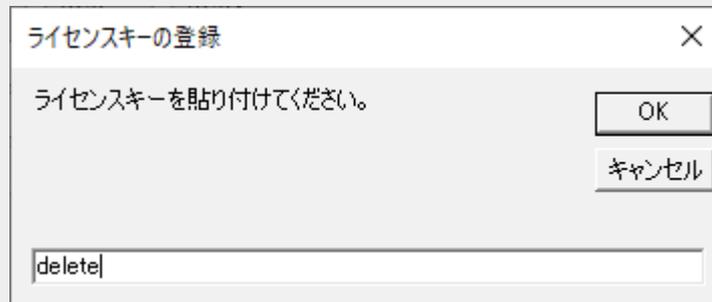
これでライセンスキーの登録は完了です。  
ただし、間違ったライセンスキーを入力した場合はプログラムは正常に動作しません。

### 1-3. ライセンスキーを削除する方法(GUI)

GUI でライセンスキーを削除する方法です。

メニュー” その他” →” ライセンスキーの登録” を左クリックした後に出てくる

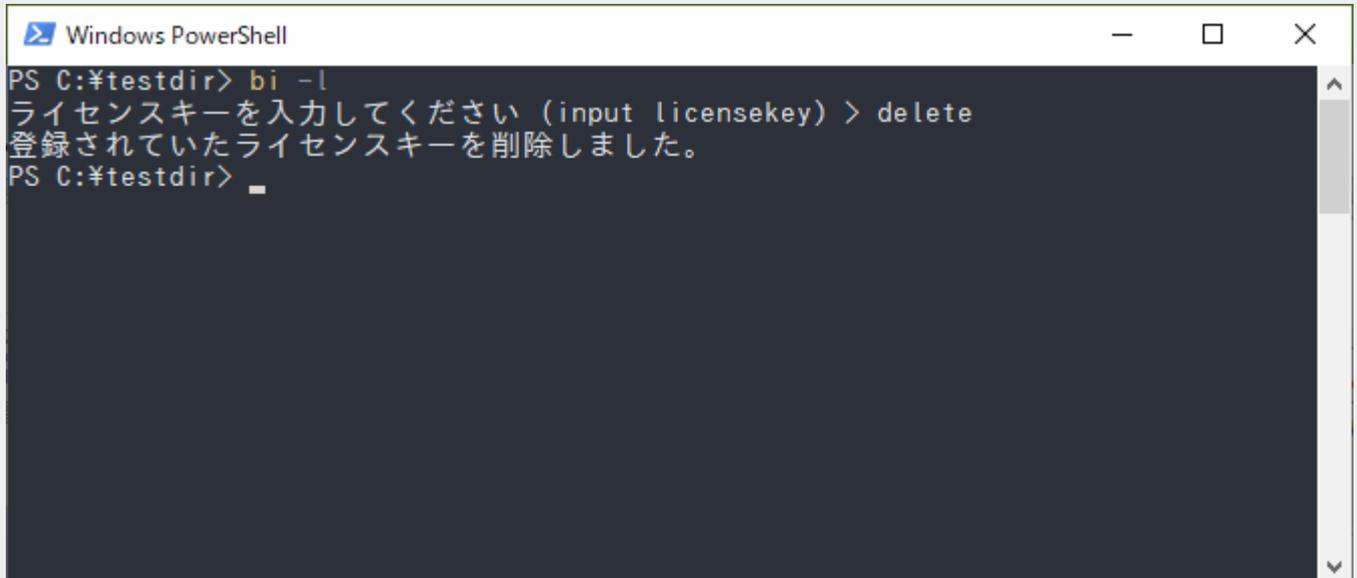
ライセンスキーの登録ダイアログでテキストボックスに、” delete” と入力し” OK” ボタンを押します。



## 1-4. ライセンスキーを削除する方法(CUI)

CUI でライセンスキーを削除する方法です。

登録したライセンスキーを削除したい場合は、ライセンスキー入力画面で delete を入力し、エンターキーを押します。



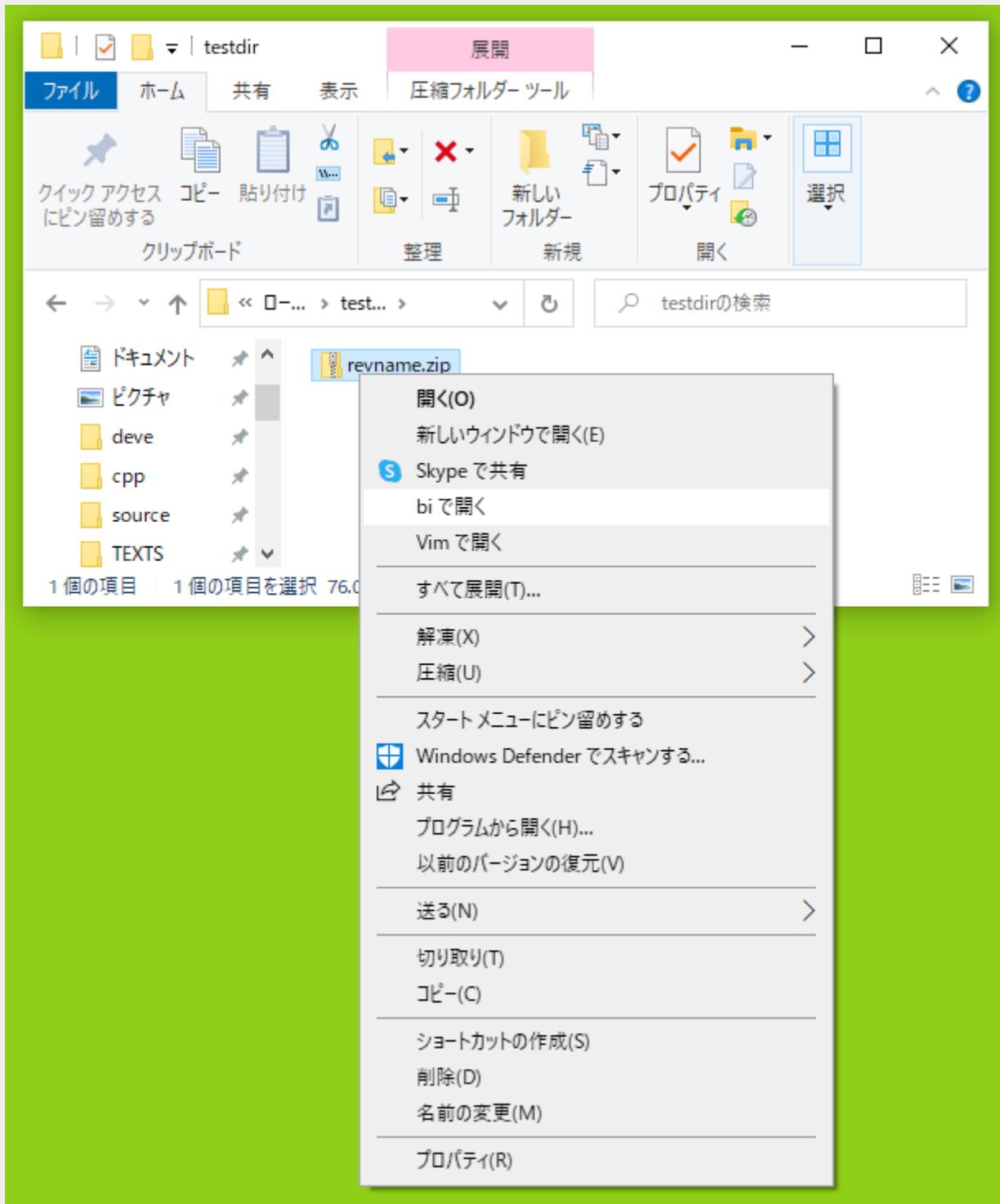
```
Windows PowerShell
PS C:\testdir> bi -l
ライセンスキーを入力してください (input licensekey) > delete
登録されていたライセンスキーを削除しました。
PS C:\testdir> █
```

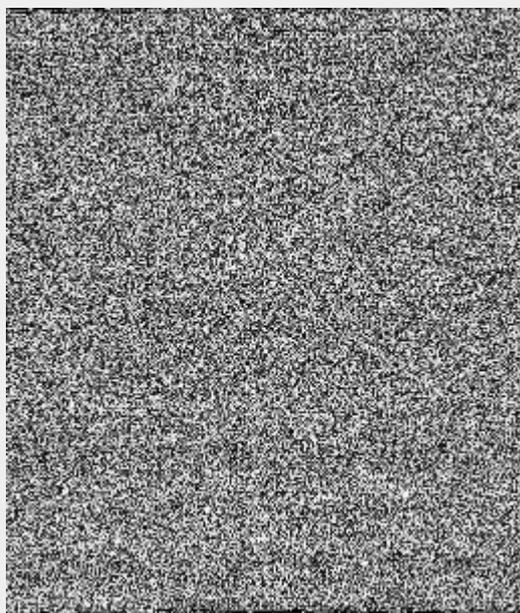
入力画面から抜け出したいときは、exit か quit か q を入力してエンターキーを押します。

## 2. 使い方

### 2-1. 右クリックメニューからの表示方法

エクスプローラーで開いているフォルダで適当なファイルを右クリックして、bi で開くを選択してみましょう。





bi-PNG 画像がデフォルトの画像ビューアで表示されたはずですが、  
ということで、エクスプローラーの右クリックメニューから、ファイルのデータ構造の確認を  
することができました。  
ただし、ビューアの制限やPNGの幅制限もあるので、あまり大きなサイズのファイルを見るこ  
とはできません。

## 2-2. GUIでの使用方法

順変換は、コンピューター上のファイルを bi-PNG 画像に変換する処理です。

逆変換は、コンピューター上の bi-PNG 画像を元のファイルに戻す処理です。



元ファイルや鍵ファイルのテキストボックスにはエクスプローラー上のファイルをドラッグアンドドロップすることができます。参照ボタンを押すと、ファイル選択ダイアログが現れるのでそちらを用いてもファイルを指定することができます。

モードの選択は、“L”，“RGB”，“RGBA”です。それぞれ、“L”は256色の無彩色、“RGB”は、赤緑青を使用した1677万色、“RGBA”は、“RGB”に透過情報を加えたものを表しています。

幅は、何も入力しないとデフォルトの256ピクセルになります。0を指定すると正方形に近い形になります。

鍵ファイルは、最低37バイトのサイズをしたファイルを指定する必要があります。何も入力しないと、暗号化は行われません。

“表示”ボタンや“bi-PNG 画像を生成”ボタンを押すと、bi-PNG 画像を生成する処理が始まります。ファイルサイズが大きいと時間がかかるので、気長に待ちましょう。

“bi-PNG 画像を生成”ボタンを押すと、元ファイルと同じフォルダに、元ファイル名.png という後ろに .png という拡張子が追加されたファイル名の bi-PNG 画像が生成されます。

パスワードは、ライセンス登録をすると使用可能になります。パスワードを指定した場合は、鍵ファイルも指定してください。使用できる文字は、英数字です。大文字と小文字の区別がありますので、CAPS LOCK に注意してください。





ファイルを指定して、元ファイルを復元ボタンを押すと、bi-PNG 画像を元のファイルに復元します。

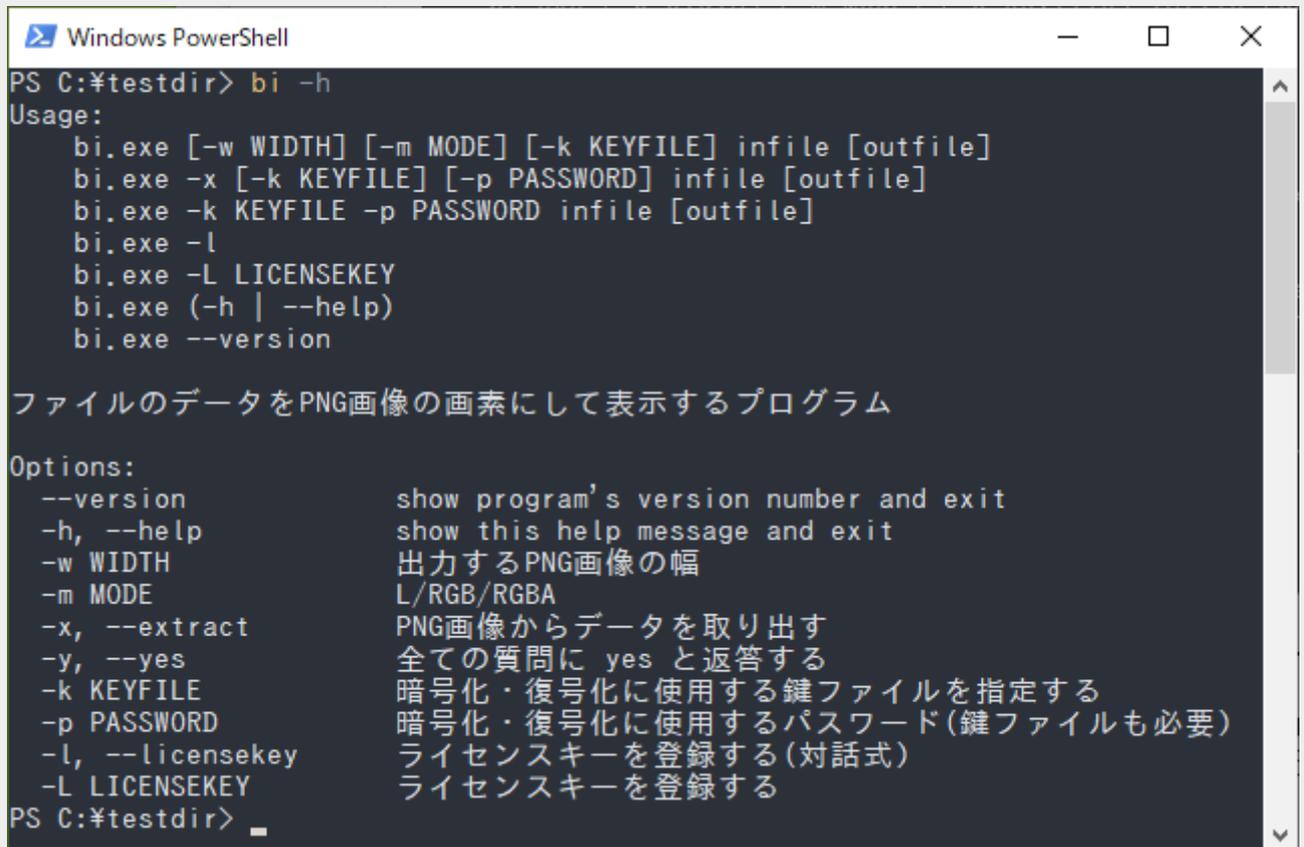
暗号化された bi-PNG 画像を指定した場合は、鍵ファイルも指定してください。

閉じるボタン、もしくは右上の×ボタンを押すと、この GUI プログラムを終了します。

## 2-3. コマンドラインでの使用方法

エクスプローラーで、目的のファイルが存在するフォルダを開いて、ファイル→Windows PowerShell を開く と選択してください。すると、PowerShell が開きます。

bi -h と入力してエンターキーを押すとコマンドラインオプションが表示されます。



```
Windows PowerShell
PS C:\testdir> bi -h
Usage:
  bi.exe [-w WIDTH] [-m MODE] [-k KEYFILE] infile [outfile]
  bi.exe -x [-k KEYFILE] [-p PASSWORD] infile [outfile]
  bi.exe -k KEYFILE -p PASSWORD infile [outfile]
  bi.exe -l
  bi.exe -L LICENSEKEY
  bi.exe (-h | --help)
  bi.exe --version

ファイルのデータをPNG画像の画素にして表示するプログラム

Options:
  --version          show program's version number and exit
  -h, --help        show this help message and exit
  -w WIDTH          出力するPNG画像の幅
  -m MODE           L/RGB/RGBA
  -x, --extract     PNG画像からデータを取り出す
  -y, --yes        全ての質問に yes と返答する
  -k KEYFILE       暗号化・復号化に使用する鍵ファイルを指定する
  -p PASSWORD      暗号化・復号化に使用するパスワード(鍵ファイルも必要)
  -l, --licensekey ライセンスキーを登録する(対話式)
  -L LICENSEKEY    ライセンスキーを登録する
PS C:\testdir>
```

[ ]で囲まれているオプションは、省略可能であることを表しています。

-w WIDTH は、出力する bi-PNG 画像 の横幅を指定するオプションです。横幅を指定しなかった場合、デフォルトでは横幅は 256(pixel)です。WIDTHに 0 を指定すると、出力する bi-PNG 画像 をできるだけ正方形に近い形にします。

-m MODE は、出力する bi-PNG 画像の種類を指定しするオプションです。Lを指定すると、グレースケール。RGBを指定すると16777216色のカラー画像。RGBAを指定すると、カラー+透明度を持った画像になります。モードを指定しなかった場合は、デフォルトのL(グレースケール)が選択されます。ファイル構造を見るためには、グレースケールが一番適しているでしょう。

-x は、bi-PNG 画像からデータを取り出すときに指定するオプションです。

-k KEYFILE は、データを埋め込むときに暗号化するための鍵を指定するオプションです。鍵ファイルは、最低37バイト必要です。私は、ファイルを暗号化するときには、約700バイト程度の歌詞を書いているテキストファイルを鍵ファイルにしています。

-p PASSWORD は、暗号化機能を使用するときにパスワードをつけるオプションです。鍵ファイルも同時に指定しないと、パスワードをつけることはできません。パスワードは4文字以上必要です。使用できる文字は英数字です(全角は使用できません)。

-l は前述のとおり、ライセンスキーを登録・削除するときに指定するオプションです。

infile は 入力ファイル名を、outfile は 出力ファイル名を表しています。

データを埋め込む時に outfile に指定するファイルの拡張子は、.png でなくてはなりません。  
データを取り出す時に infile に指定するファイルの拡張子は、.png でなくてはなりません。

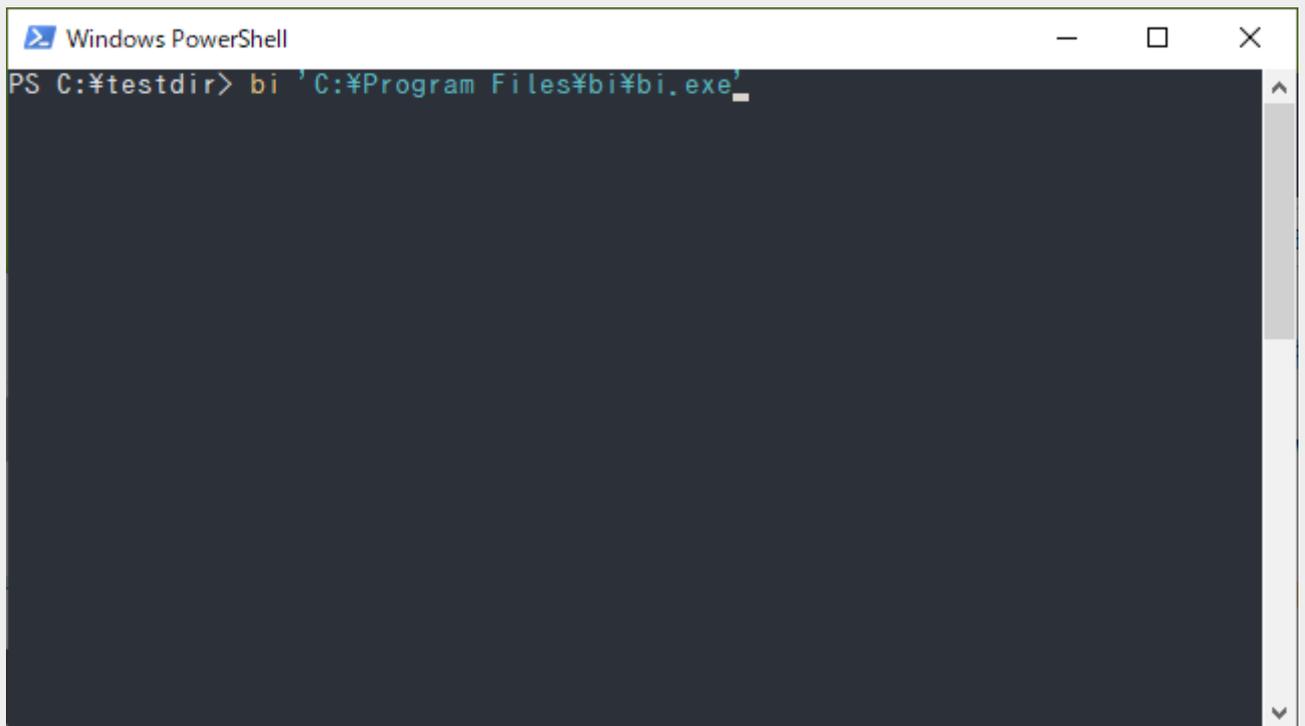
データを埋め込む時に、outfile を省略すると、デフォルトの画像ビューアが開いて、即座に bi-PNG 画像 を表示することができます。

データを取り出す時に、 outfile を省略すると、infile のファイル名から .png の拡張子を取り除いただけのファイル名のファイルを生成します。

bi には、現在のところ元のファイルのファイル名を埋め込む機能はついていないので、bi-PNG 画像 を作成する際には、 ファイル名.txt.png のように、元のファイル名の後ろに .png をつけると良いでしょう。

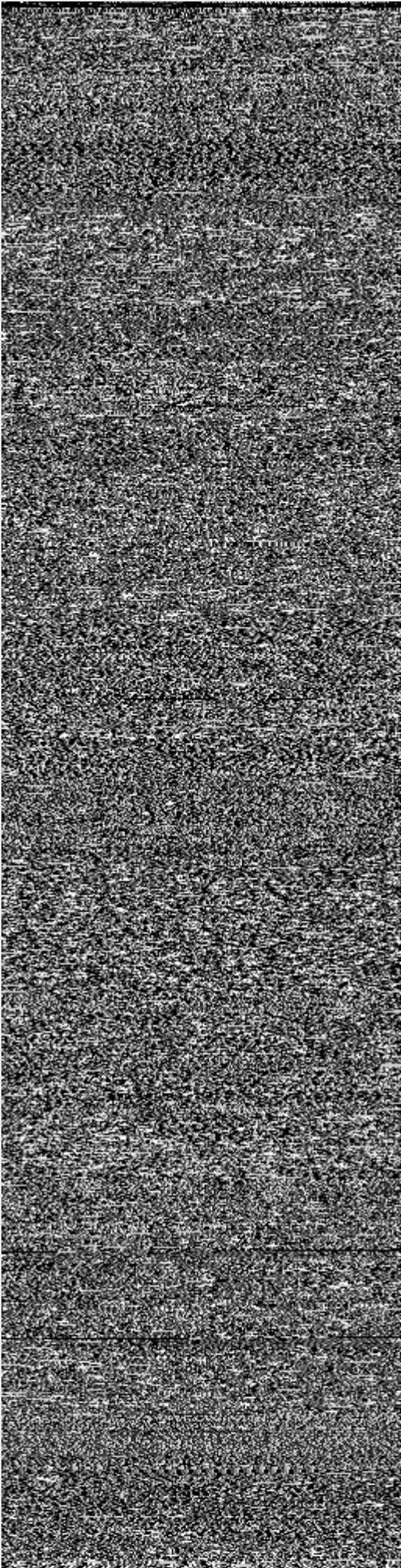
それでは、実際に bi を使用してみましょう。

bi 'C:¥Program Files¥bi¥bi.exe' と入力してエンターキーを押してください。 入力ファイルの部分は C:¥Pro と入力してタブキーを押すと補完機能が働いて入力しやすいです。



```
Windows PowerShell
PS C:¥testdir> bi 'C:¥Program Files¥bi¥bi.exe'
```

すると、以下のような画像がデフォルトの画像ビューアで表示されるはずです。

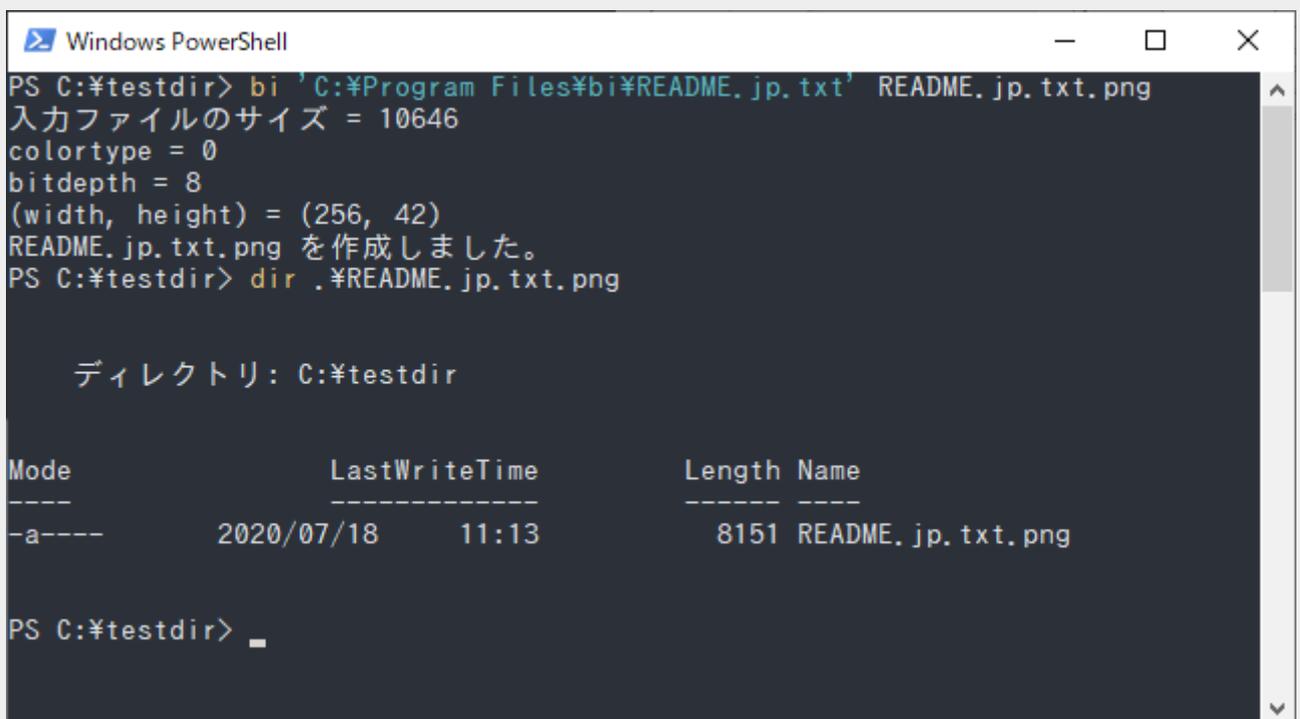


これが bi.exe のデータ構造をあらわした bi-PNG 画像です。画像ファイルの末尾に、幅調整のための白い詰め物と 360 バイトのヘッダーがついていますが、これは元ファイルの実際のデータとは関係ありません。

次は、ファイルのデータを埋め込んで bi-PNG 画像を出力してみましょう。

bi 'C:¥Program Files¥bi¥README.jp.txt' README.jp.txt.png と入力してください。

引数を 2 つにすると 2 番目の引数が出力ファイル名になります。出力ファイル名は必ず .png で終了してはなりません。現在のフォルダに、README.jp.txt.png というファイルが出来ているはずですが、出来ていない場合は、権限のないフォルダに出力しようとしている可能性があるため、権限のあるフォルダに移動してからコマンドをもう一度打ち直してください。



```
Windows PowerShell
PS C:¥testdir> bi 'C:¥Program Files¥bi¥README.jp.txt' README.jp.txt.png
入力ファイルのサイズ = 10646
colortype = 0
bitdepth = 8
(width, height) = (256, 42)
README.jp.txt.png を作成しました。
PS C:¥testdir> dir .¥README.jp.txt.png

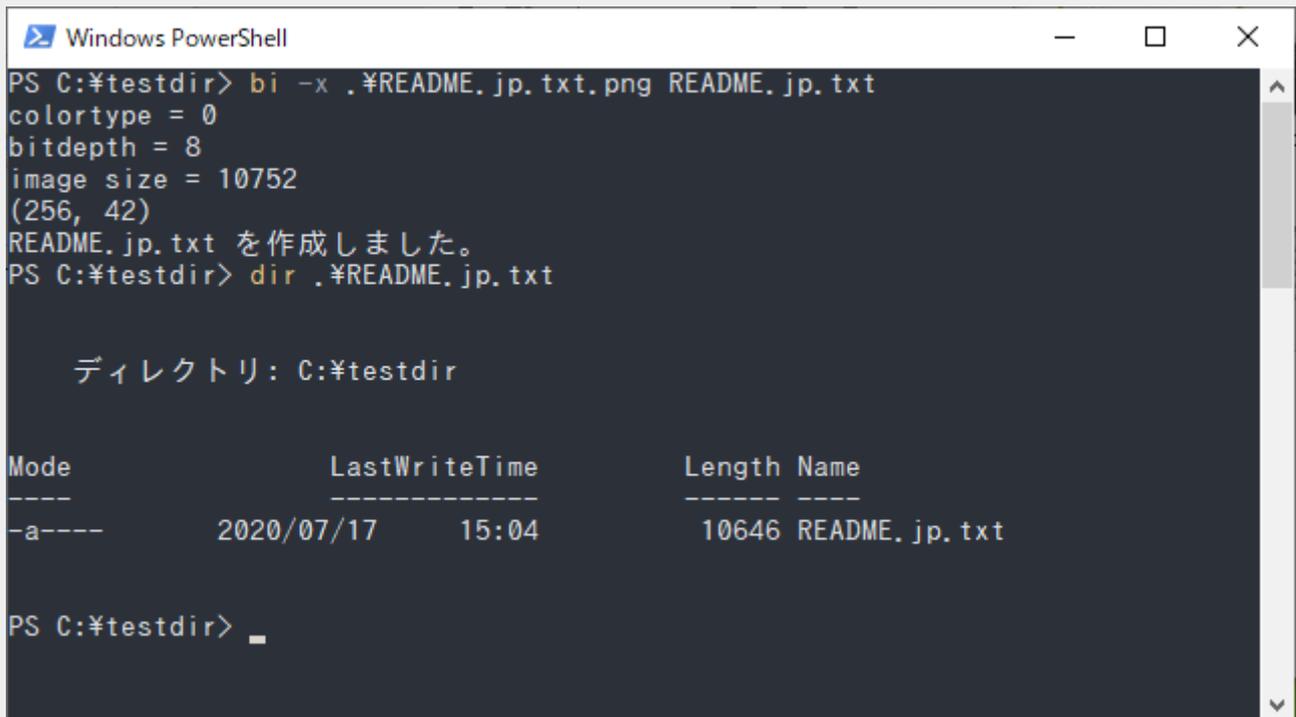
ディレクトリ: C:¥testdir

Mode                LastWriteTime         Length Name
----                -
-a-----          2020/07/18   11:13             8151 README.jp.txt.png

PS C:¥testdir> _
```

※ ファイル名の前に付いている .¥ は、現在のフォルダを表しています。

次は、bi-PNG 画像から元のファイルを取り出してみましよう。  
bi -x .¥README.jp.txt.png README.jp.txt と入力してください。現在のフォルダに、bi-PNG 画像から取り出した README.jp.txt という元のファイルと同じファイルが出力されるはずですよ。



```
Windows PowerShell
PS C:¥testdir> bi -x .¥README.jp.txt.png README.jp.txt
colortype = 0
bitdepth = 8
image size = 10752
(256, 42)
README.jp.txt を作成しました。
PS C:¥testdir> dir .¥README.jp.txt

ディレクトリ: C:¥testdir

Mode                LastWriteTime         Length Name
----                -
-a-----          2020/07/17   15:04         10646 README.jp.txt

PS C:¥testdir> _
```

diff (cat 'C:¥Program Files¥bi¥README.jp.txt') (cat '.¥README.jp.txt')

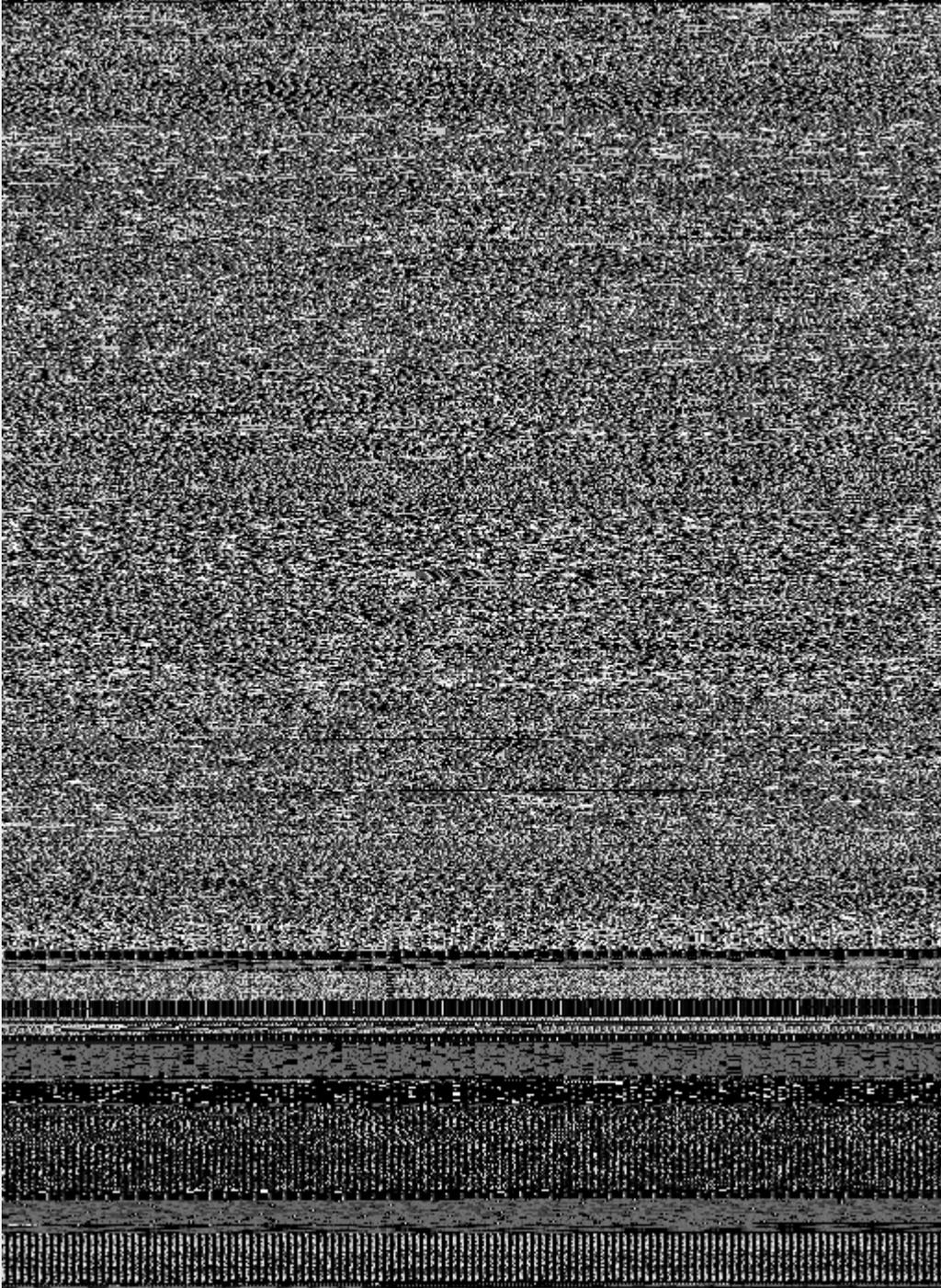
というコマンドで元のファイルと取り出したファイルが同じかどうか確認してみてください。なにも表示されなければ同じですよ。

-w オプションで幅を指定してみましょう。512 を指定してみます。



```
Windows PowerShell
PS C:\testdir> bi -w 512 'C:\Program Files\bi\bi.exe'
入力ファイルのサイズ = 360448
colortype = 0
bitdepth = 8
(width, height) = (512, 705)
C:\Users\footway\AppData\Local\Temp\bi_MEVQ3425.png を作成しました。
C:\Users\footway\AppData\Local\Temp\bi_MEVQ3425.png を削除しました。
PS C:\testdir> _
```

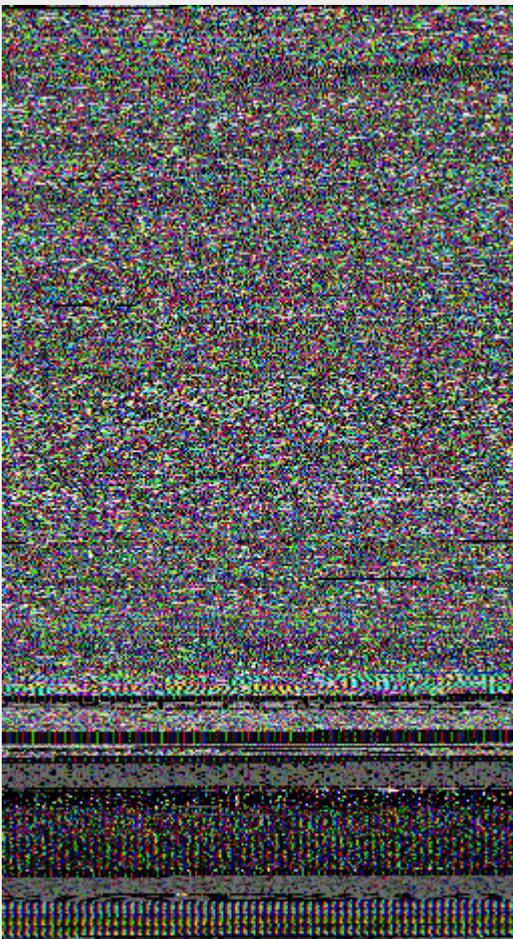
以下のように幅が 512 ピクセルの bi-PNG 画像が表示されるはずですが。



-m オプションで モードを指定してみましょう。RGB を指定してみます。

```
Windows PowerShell
PS C:\testdir> bi -m RGB 'C:\Program Files\bi\bi.exe'
入力ファイルのサイズ = 360448
colortype = 2
bitdepth = 8
(width, height) = (256, 470)
C:\Users\footway\AppData\Local\Temp\bi_LPQK1962.png を作成しました。
C:\Users\footway\AppData\Local\Temp\bi_LPQK1962.png を削除しました。
PS C:\testdir> █
```

カラーの bi-PNG 画像が表示されるはずですが、  
カラーになった分、グレースケールの時より画像の高さが減るはずですが、



以上で使い方の説明を終わります。

bi を用いて色々なタイプのファイルの構造を見てみてください。

.txt .html .jpg .png .gif .zip .mp3 .ogg .avi .mpg .exe .pdf .rtf .docx .xlsx .csv など世の中には色々な形式のファイルがあります。

### 3. 暗号化機能について

bi では、データを暗号化してから bi-PNG 画像に変換することができます。

共通鍵暗号方式で暗号化されます。共通鍵暗号とは暗号化するときも復号化するときも同じ鍵を使用する暗号化方式です。

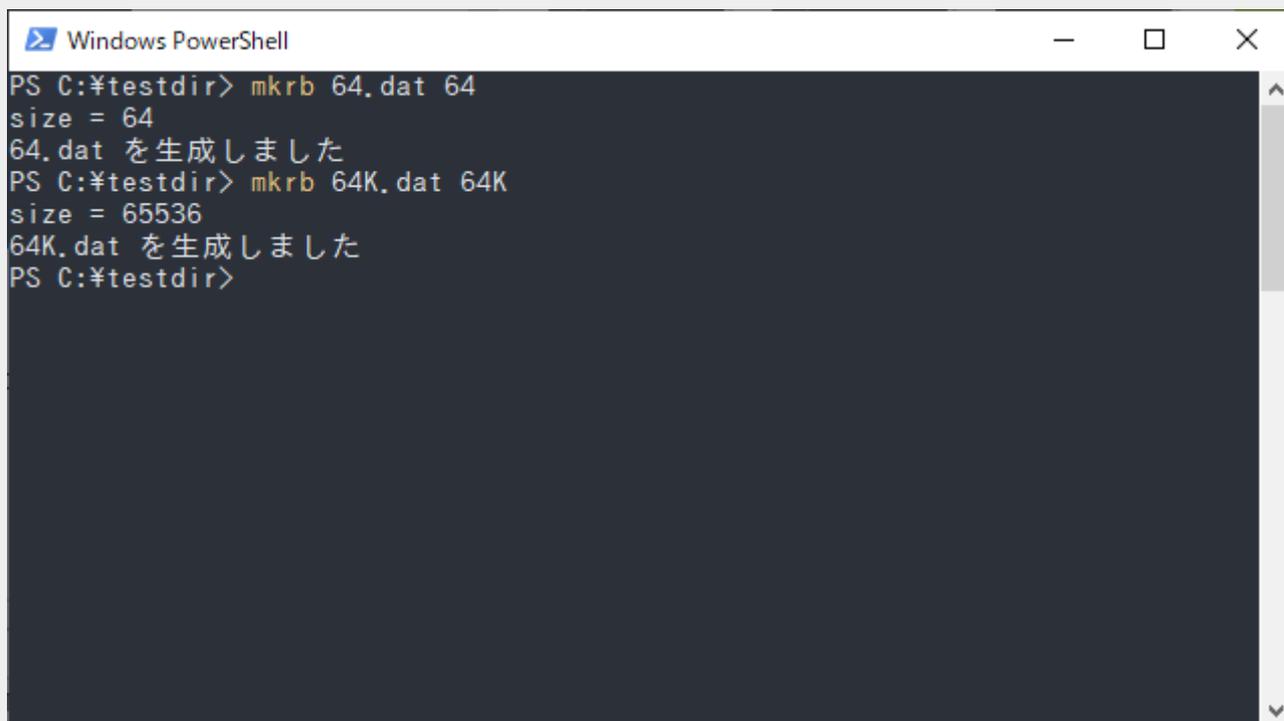
暗号化機能を使うには、`-k KEYFILE` というオプションをコマンドに追加します。KEYFILE の部分は、鍵ファイルのファイル名を指定します。

実際に暗号化機能を使う前に、同梱の `mkrb.exe` コマンドを使ってみましょう。このコマンドは、指定したファイルサイズのランダムなバイナリファイルを出力するプログラムです。

`mkrb 64.dat 64` と入力してエンターキーを押してみてください。

続いて

`mkrb 64K.dat 64K` と入力してエンターキーを押してください。



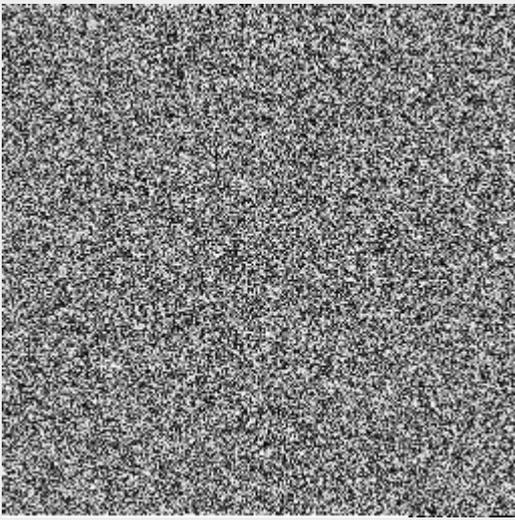
```
Windows PowerShell
PS C:\testdir> mkrb 64.dat 64
size = 64
64.dat を生成しました
PS C:\testdir> mkrb 64K.dat 64K
size = 65536
64K.dat を生成しました
PS C:\testdir>
```

現在のフォルダに `64.dat` と `64K.dat` というファイルが作成されたはずです。

画像で入力しているもう一つのコマンドを見るとわかるかもしれませんが、第2引数に `64K` もしくは `2M` もしくは `1G`などを指定するとそれぞれ、64Kbyte 2Mbyte 1Gbyteのランダムなファイルが作成されます。

それでは、ちょっと `64K.dat` を `bi` で表示してみましょう。

`bi 64K.dat`



ランダムなファイルというのは、このような砂嵐のような見た目をしています。話がそれますが、良いエンコーダーで作られた圧縮ファイルや暗号化ファイルもこの砂嵐のような見た目になります。

さっしの良い方はもうお気づきでしょうが、この `mkrb.exe` コマンドで作成したファイルを鍵ファイルとして使用するすることができます。先ほどのファイルとは別に少し小さな鍵ファイルを作ってみましょう。1キロバイトのサイズの鍵ファイルを作成します。

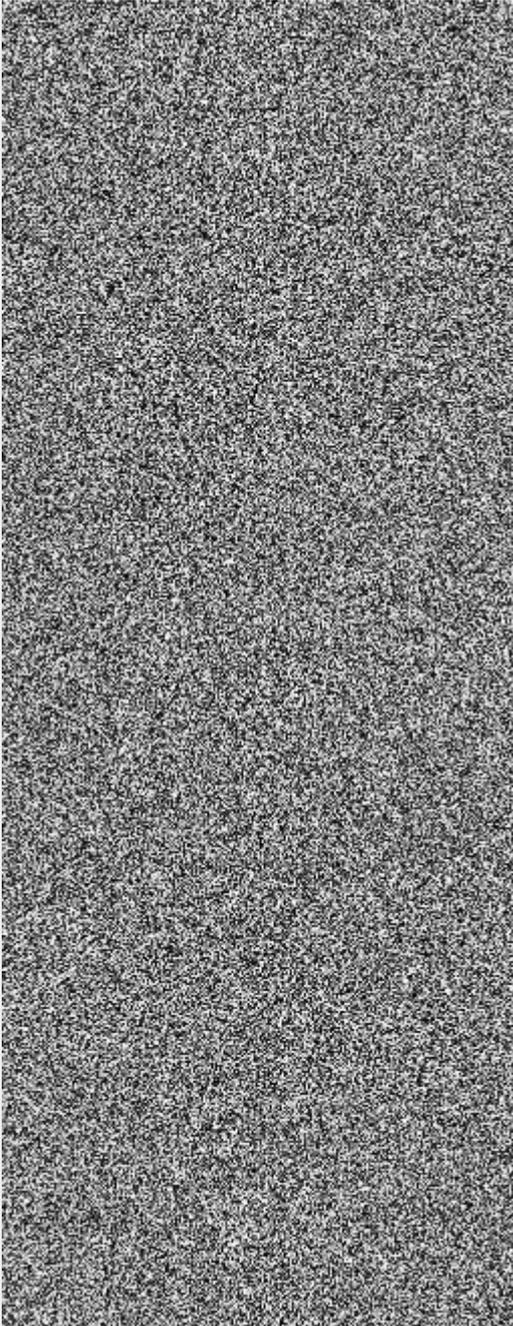
```
mkrb 1K.dat 1K
```

作成した鍵ファイルを指定して、何かのファイルを暗号化した bi-PNG 画像を作ってみてください。

```
bi -k 1K.dat revname.exe revname_enc.exe.png
```

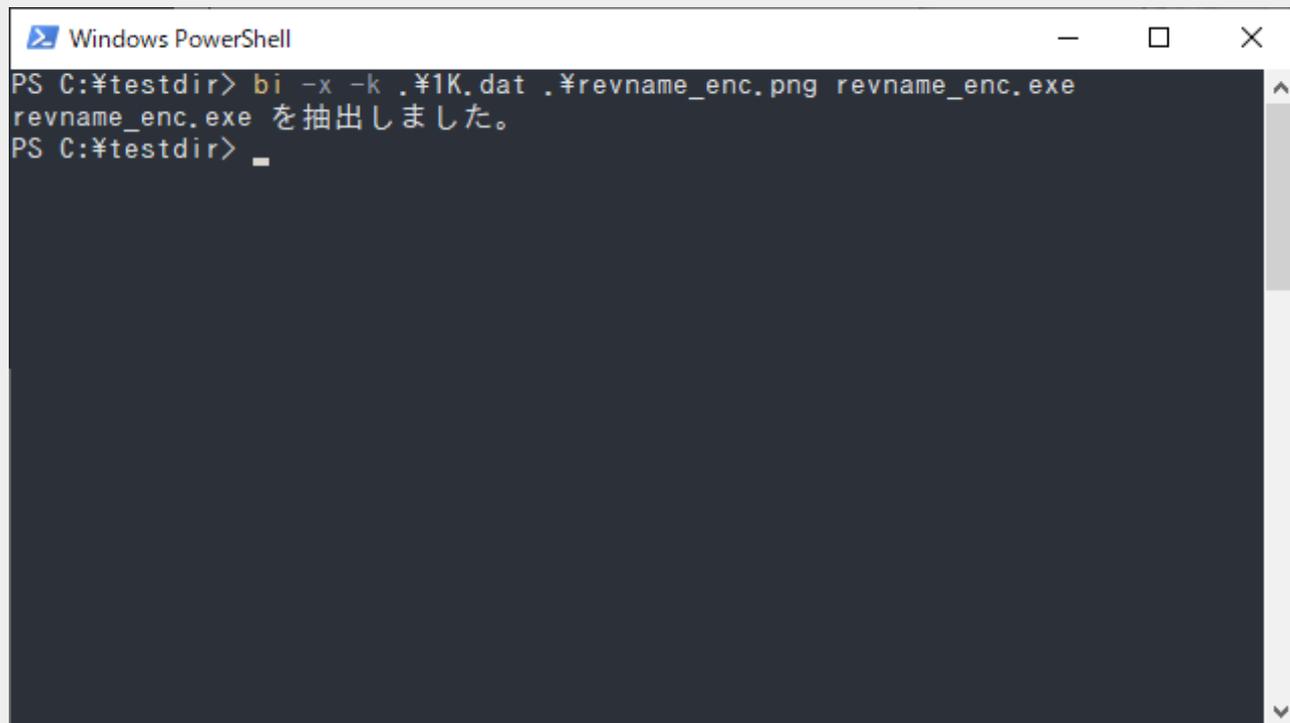
```
Windows PowerShell
PS C:\%testdir> mkrb 1K.dat 1K
size = 1024
1K.dat を生成しました
PS C:\%testdir> bi -k .\%1K.dat .\%revname.exe revname_enc.png
revname_enc.png を作成しました。
PS C:\%testdir> _
```

出来上がった bi-PNG 画像は、以下のように砂嵐に近い見た目になりました。



暗号化された bi-PNG 画像 からデータを取り出すには、`-x` オプションと `-k KEYFILE` オプションを同時につけると取り出すことができます。

```
bi -x -k 1K.dat revname_enc.exe.png revname_enc.exe
```



```
Windows PowerShell
PS C:\testdir> bi -x -k .\1K.dat .\revname_enc.png revname_enc.exe
revname_enc.exe を抽出しました。
PS C:\testdir> _
```

どうでしょうか？

元のファイルと同じ内容のファイルが出力されたはずです。

暗号化にパスワードをつけたい場合は、`-p` オプションでパスワードを指定することができます。しかし、コマンドラインで入力すると、コマンドライン履歴にパスワードが残ってしまうので、パスワード機能は、GUI で使用したほうがよいでしょう。

暗号化機能は、暗号化してから PNG 画像にして(圧縮して)いるため、圧縮の効果はありません。

この暗号化機能はどのような使い方をすればいいのでしょうか？

鍵ファイルを相手に手渡し、もしくは郵送し、暗号化した bi-PNG 画像 は、インターネットで送るという方法で、比較的安全にデータを送信することができます。

鍵ファイルは、ここで作成したようなランダムバイナリファイルでなくても構いません。私は、700 バイト程度の歌詞を書いたあるテキストファイルを暗号化の鍵ファイルにしています。サイズの小さな JPEG ファイルも鍵ファイルにするのに向いているでしょう。

注) 本プログラムは 64bit プログラムなので、理論上約 18EB のサイズのファイルを扱えることにはなりますが、現在のパーソナルコンピュータでは、その様な大きなサイズのファイルを扱うことはできません。ほとんどの場合、コンピュータの搭載しているメモリ容量内に制限されるでしょう。私のコンピュータ(Vostro3800)では、メモリを 8GB 積んでいますが、5GB のファイルを変換することが出来ました。5GB のファイルを変換するのに約 5 分かかります。