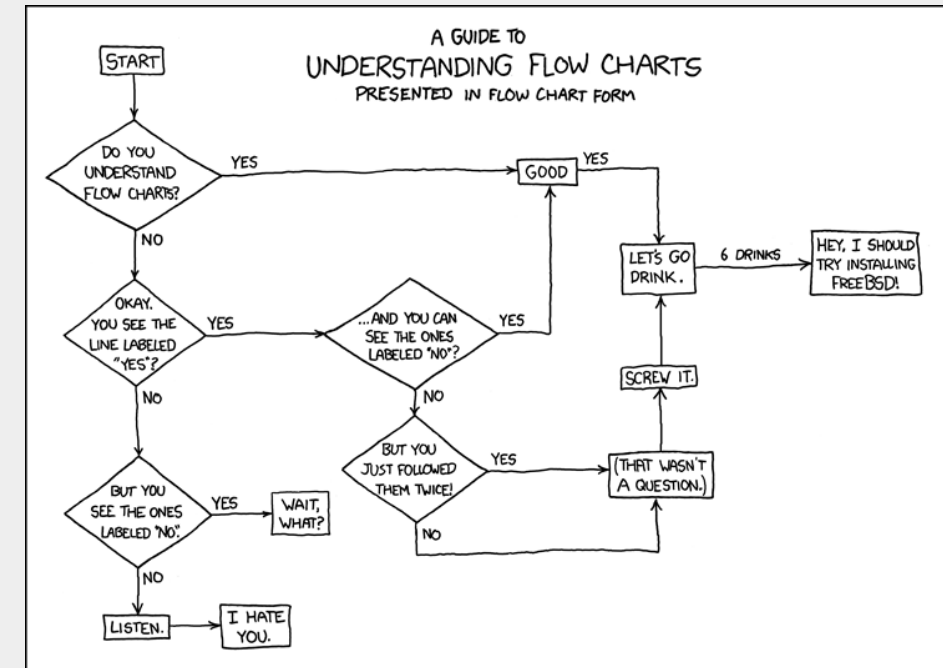


VS Code [Draw.io](#) Integrationで作成した図を画像ファイルとして取り込む

Background

- コード書いてると処理概要の整理のためにちょっとした図を描きたくなることがある



Background

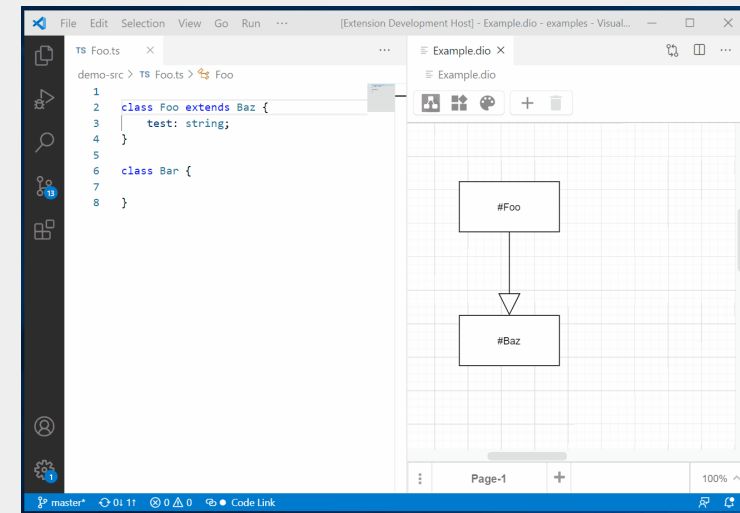
- Microsoft Office
 - PowerPoint: Canvasのサイズが決まってる / コネクタ貼りづらい
 - Excel: 表計算ソフトでは？
- [Mermaid](#)
 - テキストでMarkdownファイル内に書けるのは良い
 - 変換ツール要？ 複雑なものだと管理しきれなそう...
- [Draw.io](#)
 - コネクタ貼り易い (**重要**)
 - サーバーを立てておく必要がある
 - Webアプリなので作成した図をエクスポートして取り込んだりする手間

VS Code [Draw.io](#) Integrationで作成した図を画像ファイルとして取り込む

Draw.io Integration | VisualStudio Marketplace

大勝利 🎉

- スタンドアロンのためDraw.ioサーバー不要！
- Markdown編集からシームレスに図の編集も可能に！



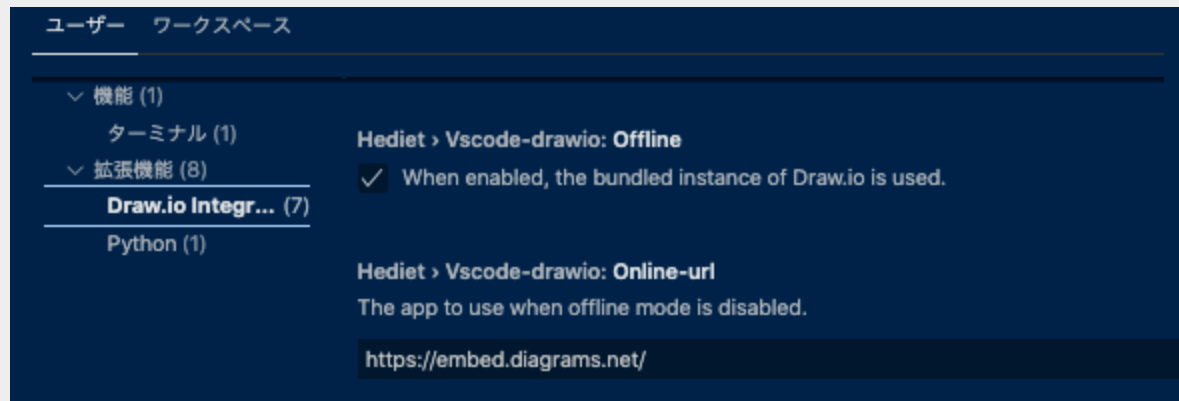
大勝利 🎉 ...なのか？

- スタンドアロンのためDraw.ioサーバー不要！
 - -> 本当に？ 「会社の資料作ってたら実は外部に転送されてた」とか無い？
- Markdown編集からシームレスに図の編集も可能に！
 - -> 結局、画像ファイルへの変換は必要になるのでは？

疑問1: 本当にスタンドアロンで起動してるの？

本当にスタンドアロンで起動してるの？ (1/4)

- VSCodeの設定を見ると、**Offline mode** と **Online Mode** があるらしい
 - **Offline mode**: バンドル済みのDraw.ioを使う
 - **Online mode**: `Online-url` に指定されたサーバーを使う(多分)
 - Draw.ioはプラグインがあったりするので、それを利用するため？



本当にスタンドアロンで起動してるの？ (2/4)

- コードを "offline" で検索するとそれっぽいメソッドが見つかる^{^1}

```
config: DiagramConfig,  
  options: DiagramOptions,  
  webview: Webview  
) : string {  
  if (config.mode.kind === "offline") {  
    return this.getOfflineHtml(config, options, webview); <---  
  } else {  
    return this.getOnlineHtml(config, config.mode.url);  
  }  
}
```

本当にスタンドアロンで起動してるの？ (3/4)

- どうやらwebviewでローカルのファイルを表示しているだけらしい ^2
- `../../drawio` はgit submoduleとして持っているDraw.ioリポジトリのパス

```
private getOfflineHtml(  
    config: DiagramConfig,  
    options: DiagramOptions,  
    webview: Webview  
): string {  
    const vsuri = webview.asWebviewUri( <---  
        Uri.file(path.join(__dirname, "../../drawio/src/main/webapp"))  
    );  
    const customPluginsPath = webview.asWebviewUri(  
        ...
```

本当にスタンドアロンで起動してるの？ (4/4)

- 本当にスタンドアロンで立ってるっぽい
 - 他にも社内でプロキシ切った状態で立ち上げてみたり、DevToolsでネットワークを見てみたりもしたけど問題なさそうだった
- (アップデートで挙動変わったりするのが少し怖い)

疑問2: 結局、画像ファイルへの変換は必要になるのでは？

結局、画像ファイルへの変換は必要になるのでは？(1/4)

<https://qiita.com/riku-shiru/items/5ab7c5aecdfea323ec4e>



@riku-shiru 2020年06月16日に更新

...

VSCodeでDraw.ioが使えるようになったらしい！

VSCode Draw.io

追記

versionによっては設定を変えないとエクスポートができないようです。

エクスポートできない方はこちらの記事をご参考に設定いただくとエクスポートできるかもしれません。

現状バージョン0.4ではこちらの設定が必要となります。

[VSCodeでDraw.io Integration使用時にエクスポートできないことがある問題への対処](#)

- v0.4でエクスポートメニュー非表示 (対処はOnline modeにすること)

結局、画像ファイルへの変換は必要になるのでは？(2/4)

- Issueを見るとPNG出力機能は開発済み？
 - 現時点(2020/07/22)での最新版v0.7.2ではエクスポートメニューは無く...

[Support PNG Export · Issue #56 · hediet/vscode-drawio](#)

It would be useful if [Draw.io](#) diagrams could be exported to PNG files.

<https://github.com/hediet/vscode-drawio/issues/56#issuecomment-633490768>

As you can see, this feature has been released to insiders:

結局、画像ファイルへの変換は必要になるのでは？ (3/4)

- PDFエクスポートについては対応する気がなさそう

[can't export file · Issue #34 · hediet/vscode-drawio](#)

Because export does not work and I don't think exporting things is a great user experience, I will rather try to hide the export options altogether (#22).

If you think it is a must have, please convince me ;) A cli should be the way to go for exports.

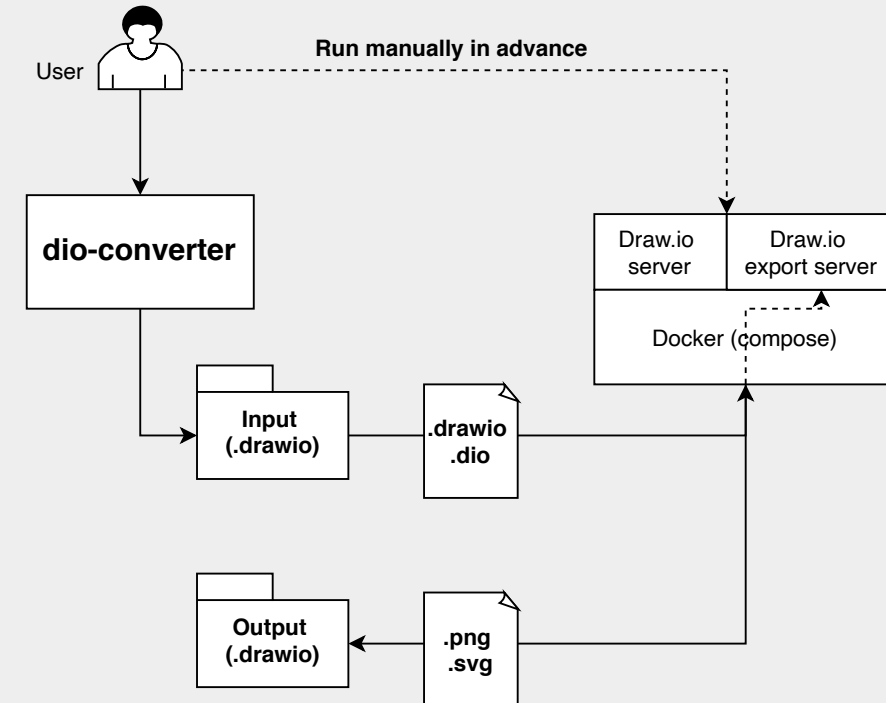
結局、画像ファイルへの変換は必要になるのでは？(4/4)

- (現時点では) 公式にサポートされる雰囲気ではなさそう
- そもそもExport機能があっても手動実行が必要になるのでは？
- 自分で作っちゃおう 🛠️

そして出来たもの

kaakaa/dio-converter

- Draw.io公式の画像エクスポートWebアプリケーションを利用
 - [jgraph/draw-image-export2](#)
 - The 2018 server-side PNG/PDF export implementation using Node, Puppeteer and Chrome headless
 - ↑の[Dockerイメージ](#) ([GitHub](#)) を使っている
 - Draw.ioの `/export3.html` を使っている
 - ドキュメント化されていない隠し機能？



しかし...

事前に `docker-compose` でアプリ立ち上げるとか面倒臭い...

結局DockerでDraw.ioサーバー立ててるのとあまり変わらないのでは...

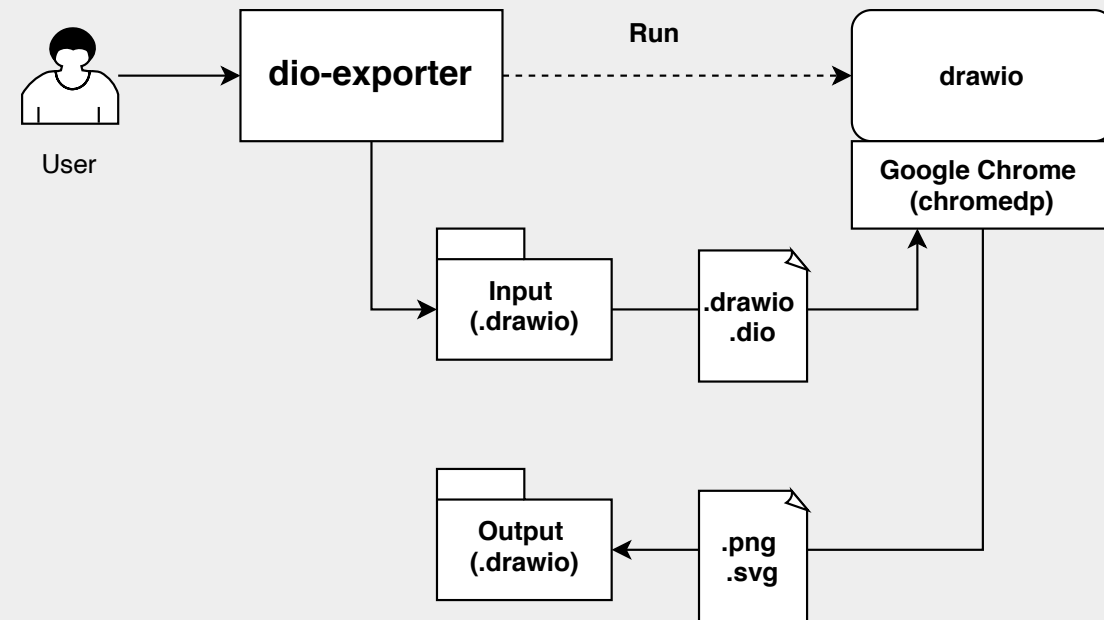
あれ？ Draw.ioはアプリ起動する必要はないのでは？

(VS Code Extensionではwebviewでファイルを開いてるだけ)

顧客が必要だったもの

kaakaa/dio-exporter

- GoでDraw.ioファイルサーバーを起動
 - [http.FileServer](#)
- GoでHeadless Chromeを操作する
[chromedp](#) を使用
 - `/export3.html` へ `*.drawio` ファイルを送信
 - 描画された図をスクリーンショットで取得



顧客が必要だったもの (開発)

kaakaa/dio-exporter

- テスト
 - テスト用 *.drawio ファイルから生成される画像の一致テスト
 - ~~pixelmatchを使用して生成されるPNGの一致チェック~~
 - 環境により日本語フォントに差異があるためSVG形式のみのテスト
 - GitHub Actionsでテスト実行
- リリース作成時にIssue, PRのタグを見てリリースノートを作成

ええもんでけた

同じ問題で困ってる人がいないかTwitterで検索...

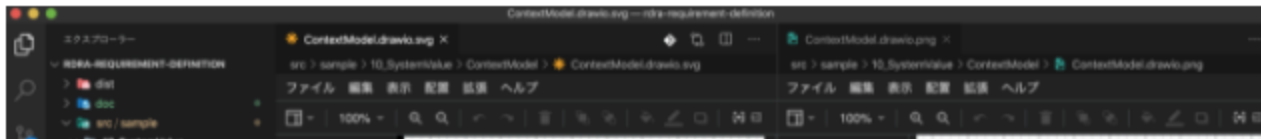
あれ？

*.drawio.svg と *.drawio.png

さてさて、上記で記載したとおり、VSCode で draw.io が使用できる拡張子には複数種類があります。

- `.drawio`
- `.dio`
- `.drawio.svg`
- `.drawio.png`

VSCode で draw.io が使えるだけで驚いていたのですが、下2つの拡張子の動作に驚きました。なんと、VSCode + draw.io でファイル作成・編集・保存すると、それぞれ、SVG や PNG のファイルとして保存されるのです。



<https://blog.anfangd.me/entry/2020/07/08/220628>

あれれ？

Markdown ファイルからの参照

上記で記載した「VSCode 上は draw.io で編集しながら、保存したファイルは SVG や PNG になる」ということは、その保存したファイルを画像参照として Markdown ファイルなどから指定してあげれば、そのまま画像として表示できることを意味します。

これ、凄くないですか？

<https://blog.anfangd.me/entry/2020/07/08/220628>

これ、凄くないですか？

これ、凄く（`dio-exporter` 必要なく）ないですか？

終 (ではない)

いや、バイナリファイル(PNG)にXML情報(drawio)埋め込めるわけないやろ！

イチャモン

*.drawio.png の中身

- hexdump for VSCode で開く
- PNGを読む - Qiita

PNGファイルではファイルの先頭に8バイトの 89 50 4E 47 0D 0A 1A 0A が存在するようです。文字列にすると \x89PNG\r\n\x1a\n こうなります。

```
test.dio.png.hexdump ×
1 | Offset: 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
2 | 00000000: 89 50 4E 47 0D 0A 1A 0A 00 00 00 0D 49 48 44 52 .PNG.....IHDR
3 | 00000010: 00 00 00 79 00 00 00 3D 08 06 00 00 00 89 33 C6 ...y...=.....3F
4 | 00000020: 33 00 00 04 BC 74 45 58 74 6D 78 66 69 6C 65 00 3...<tEXtmxfile.
5 | 00000030: 25 33 43 6D 78 66 69 6C 65 25 32 30 68 6F 73 74 %3Cmxfile%20host
6 | 00000040: 25 33 44 25 32 32 31 32 39 35 37 32 63 31 2D 37 %3D%22129572c1-7
7 | 00000050: 33 63 31 2D 34 63 35 30 2D 61 38 32 39 2D 62 38 3c1-4c50-a829-b8
8 | 00000060: 62 39 30 31 31 38 32 62 39 65 25 32 32 25 32 30 b901182b9e%22%20
9 | 00000070: 6D 6F 64 69 66 69 65 64 25 33 44 25 32 32 32 30 modified%3D%2220
10 | 00000080: 32 30 2D 30 37 2D 32 35 54 30 35 25 33 41 30 30 20-07-25T05%3A00
11 | 00000090: 25 33 41 34 37 2E 38 35 34 5A 25 32 32 25 32 30 %3A47.854Z%22%20
12 | 000000a0: 61 67 65 6E 74 25 33 44 25 32 32 35 2E 30 25 32 agent%3D%225.0%2
13 | 000000b0: 30 28 4D 61 63 69 6E 74 6F 73 68 25 33 42 25 32 0(Macintosh%3B%2
14 | 000000c0: 30 49 6E 74 65 6C 25 32 30 4D 61 63 25 32 30 4F 0Intel%20Mac%200
15 | 000000d0: 53 25 32 30 58 25 32 30 31 30 5F 31 35 5F 35 29 S%20X%2010_15_5)
16 | 000000e0: 25 32 30 41 70 70 6C 65 57 65 62 4B 69 74 25 32 %20AppleWebKit%2
17 | 000000f0: 46 35 33 37 2E 33 36 25 32 30 28 4B 48 54 4D 4C F537.36%20(KHTML
18 | 00000100: 25 32 43 25 32 30 6C 69 68 65 25 32 30 47 65 63 %2C%20like%20Gec
19 | 00000110: 68 6F 29 25 32 30 43 6F 64 65 25 32 46 31 2E 34 ko)%20Code%2F1.4
20 | 00000120: 37 2E 30 25 32 30 43 68 72 6F 6D 65 25 32 46 37 7.0%20Chrome%2F7
21 | 00000130: 38 2E 30 2E 33 39 30 34 2E 31 33 30 25 32 30 45 8.0.3904.130%20E
22 | 00000140: 6C 65 63 74 72 6F 6E 25 32 46 37 2E 33 2E 32 25 lectron%2F7.3.2%
23 | 00000150: 32 30 53 61 66 61 72 69 25 32 46 35 33 37 2E 33 20Safari%2F537.3
24 | 00000160: 36 25 32 32 25 32 30 65 74 61 67 25 33 44 25 32 6%22%20etag%3D%2
```

*.drawio.png の中身

- PNGを読む - Qiita

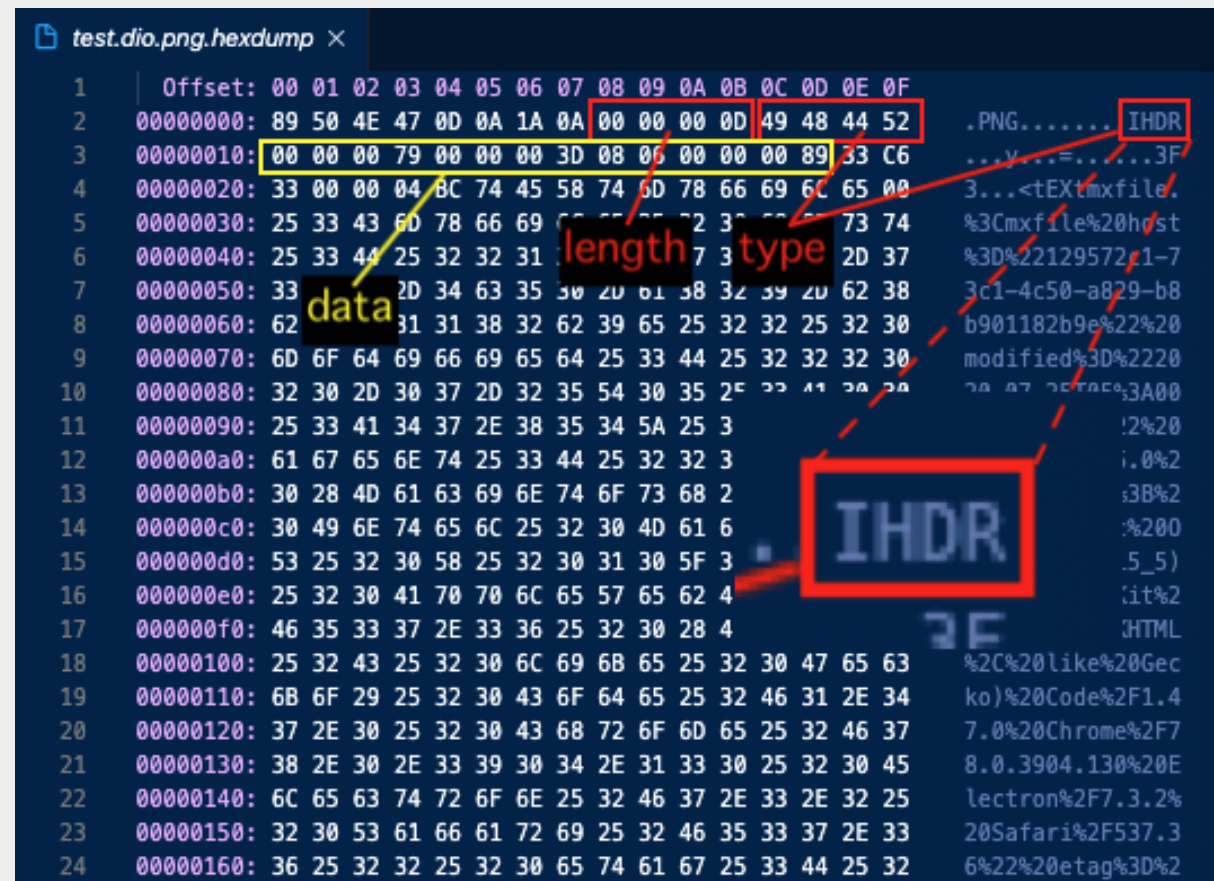
最初の8バイトのPNGシグネチャをのぞいて、チャンクというデータのまとまりが続きます。

種類	サイズ	説明
length	4 bytes	チャンクのdata部分のバイト数を表す
type	4 bytes	4文字でチャンクの種類を表す
data	可変長 bytes	lengthで指定されたbyte長のチャンクのデータ
CRC	4 bytes	typeとdataのCRCで、データの破損をチェックできる

*.drawio.png の中身

- PNGを読む - Qiita

最初の8バイトのPNGシグネチャを
のぞいて、チャンクというデータ
のまとまりが続きます。



*.drawio.png の中身

mxfile ...?

*.drawio ファイルの中身では...?

```
<mxfile <---  
  host="f1e0d6ca-e141-42aa-960f-eede6d01d428"  
  modified="2020-07-26T05:56:53.199Z"  
  agent="5.0 (Macintosh; ..."  
  etag="r0m-Pgl4D5wC1QD8uaEo"  
  version="13.1.3">  
    <diagram ...
```

The image shows a hex dump of a .drawio.png file. The first part of the dump shows the PNG header (IHDR, gAMA, etc.). The second part shows the 'tEXt' block, which contains the mxfile metadata. The text '<tEXtmxfile.' is highlighted with a red box. A red dashed line points from this box to the text '<tEXtmxfile.' in the code block on the left.

Offset	Hex	ASCII
00000000	89 50 4E 47 0D 0A 1A 0A 00 00 0D 49 48 44 52	.PNG.....IHDR
00000010	00 00 00 79 00 00 00 3D 08 06 00 00 00 89 33 C6	...y...=.....3F
00000020	33 00 00 04 BC 74 45 58 74 6D 78 66 69 6C 65 00	3...<tEXtmxfile.
00000030	25 33 43 6D 78 66 69 6C 65 25 32 30 68 6F 73 74	%3Cmxfile%20host
00000040	25 33 44 25 32 32 31 32 39 35 37 32 63 31 2D 37	%3D%22129572c1-7
00000050	33 63 31 2D 34 63 35 30 2D 61 38 32 39 2D 62 38	3c1-4c50-a829-b8
00000060	62 39 30 31 31 38 32 62 39 65 25 32 32 25 32 30	b901182b9e%22%20
00000070	6D 6F 64 69 66 69 65 64 25 33 44 25 32 32 32 30	modified%3D%2220
00000080	32 30 2D 30 37 2D 32 35 54 30 35 25 33 41 30 30	20-07-25T05%3A00
00000090	25 33 41	0
000000a0	61 67 65	:2
000000b0	30 28 4D	:2
000000c0	30 49 6E	0
000000d0	53 25 32)
000000e0	25 32 30	:2
000000f0	46 35 33	IL
00000100	25 32 43	:C
00000110	6B 6F 29 25 32 30 43 6F 64 65 25 32 46 31 2E 34	ko)%20Code%2F1.4
00000120	37 2E 30 25 32 30 43 68 72 6F 6D 65 25 32 46 37	7.0%20Chrome%2F7
00000130	38 2E 30 2E 33 39 30 34 2E 31 33 30 25 32 30 45	8.0.3904.130%20E
00000140	6C 65 63 74 72 6F 6E 25 32 46 37 2E 33 2E 32 25	lectron%2F7.3.2%
00000150	32 30 53 61 66 61 72 69 25 32 46 35 33 37 2E 33	20Safari%2F537.3
00000160	36 25 32 32 25 32 30 65 74 61 67 25 33 44 25 32	6%22%20etag%3D%2

*.drawio.png の中身

- Draw.ioのコード見る ^3

```
//PNG+XML format
if (data.xml.substring(0, 5) == 'iVBOR' || (extras != null && extras.isPng))
{
    data.xml = Editor.extractGraphModelFromPng('data:image/png;base64,' + data.xml);
}
```

extractGraphModelFromPng

あやしい...

*.drawio.png の中身

- zTXt or tEXt チャンクにDraw.ioのXMLデータ入ってるっぽい ^4

```
Editor.extractGraphModelFromPng = function(data)
...
  if (type == 'zTXt')
  {
    ...
  }
  // Uncompressed section is normally not used
  else if (type == 'tEXt')
  {
    ...
  }
}
```

*.drawio.png の中身

tEXt チャンク。なるほど。

```
Offset: 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
00000000: 89 50 4E 47 0D 0A 1A 0A 00 00 0D 49 48 44 52 .PNG.....IHDR
00000010: 00 00 00 79 00 00 00 3D 08 06 00 00 00 89 33 C6 ...y...=.....3F
00000020: 33 00 00 04 BC 74 45 58 74 6D 78 66 69 6C 65 00 3..<tEXtmxfile.
00000030: 25 33 43 6D 78 66 69 6C 65 25 32 30 68 6F 73 74 %3Cmxfile%20host
00000040: 25 33 44 25 32 32 31 32 39 35 37 32 63 31 2D 37 %3D%22129572c1-7
00000050: 33 63 31 2D 34 63 35 30 2D 61 38 32 39 2D 62 38 3c1-4c50-a829-b8
00000060: 62 39 30 31 31 38 32 62 39 65 25 32 32 25 32 30 b901182b9e%22%20
00000070: 6D 6F 64 69 66 69 65 64 25 33 44 25 32 32 32 30 modified%3D%2220
00000080: 32 30 2D 30 37 2D 32 35 54 30 35 25 33 41 30 30 20-07-25T05%3A00
00000090: 25 33 41 ..y...=.....3F 0
000000a0: 61 67 65 ..y...=.....3F :2
000000b0: 30 28 4D :2
000000c0: 30 49 6E 0
000000d0: 53 25 32 ..<tEXtmxfile. )
000000e0: 25 32 30 :2
000000f0: 46 35 33 IL
00000100: 25 32 43 Cmxfile%20host :C
00000110: 6B 6F 29 25 32 30 43 6F 64 65 25 32 46 31 2E 34 ko)%20Code%2F1.4
00000120: 37 2E 30 25 32 30 43 68 72 6F 6D 65 25 32 46 37 7.0%20Chrome%2F7
00000130: 38 2E 30 2E 33 39 30 34 2E 31 33 30 25 32 30 45 8.0.3904.130%20E
00000140: 6C 65 63 74 72 6F 6E 25 32 46 37 2E 33 2E 32 25 lectron%2F7.3.2%
00000150: 32 30 53 61 66 61 72 69 25 32 46 35 33 37 2E 33 20Safari%2F537.3
00000160: 36 25 32 32 25 32 30 65 74 61 67 25 33 44 25 32 6%22%20eta%3D%2
```

*.drawio.png の中身

```
$ go run png.go
chunk: IHDR <--- 画像のサイズやカラータイプなどのメタデータ
chunk: tEXt <--- Draw.ioデータ (URLEncoded)
chunk: IDAT <--- PNGデータ
chunk: IEND <--- 画像データの終端

tEXt chunk:
  mxfile<mxfile
  host="6b69cf35-3cdc-4e90-9822-4718dea9a24a"
  modified="2020-07-26T05:16:42.690Z"
  ...
>
<diagram id="ooyZYogMv9WS_n0PFq-w" name="ページ1">
  ...
</diagram>
</mxfile>
```

***.drawio.png** の中身

https://ja.wikipedia.org/wiki/Portable_Network_Graphics

- tEXt - ISO 8859-1形式のテキストを保持する。キーワードと対になるチャンクを複数持つことができる。テキストの圧縮は行われない。
- zTXt - tEXtチャンクと同じ制限の圧縮テキスト。

*.drawio.png の中身

- Draw.ioでは tEXt チャンクはあまり使われない？
 - エディタでの編集は保存回数が多く、毎回圧縮処理をかけるとパフォーマンスに影響がでそうだから？
 - WebアプリではExportの時に圧縮、import時に解凍すれば良い

```
if (type == 'zTXt')
{
    ...
}
// Uncompressed section is normally not used
else if (type == 'tEXt')
{
    ...
}
```

*.drawio.svg の中身

- `svg` タグの `content` 属性にDraw.ioのデータが入っている

```
<svg
  xmlns="http://www.w3.org/2000/svg"
  xmlns:xlink="http://www.w3.org/1999/xlink"
  version="1.1"
  width="463px"
  ...
  content="&lt;mxfile host=&quot;78e38da8-5c2e-49bf-9...  <---
  ...
```


顧客が本当に必要だったもの

- [VisualStudio Code](#)
- [VisualStudio Code Draw.io Integration](#)
- `*.drawio.png` or `*.drawio.svg` ファイル
 - (最後のページしか画像表示できないっぽいので、複数ページのDraw.ioファイルを反復的に画像化したい場合は [kaakaa/dio-exporter](#) を...)

VS Code [Draw.io](#) Integrationで作成した図を画像ファイルとして取り込む

終