



noellabo Dec 10, 2017 10:59 PM

MVP

ExtendScriptを書くのに適した良いエディタ・開発環境はありますか？

This question has been **Answered**.

Creative Cloudデスクトップアプリの環境設定で「古いアプリケーションを表示」するように設定しないとリストに表示すらされなくなってしまったESTKですが、今後はどうなるのでしょうか？ 公開されている情報などありましたら教えて下さい。

(CCDAの環境設定変更の他、ここからダウンロードするという手もあるようです)

[JavaScript開発ツールキット | Adobe ExtendScript Toolkit CCをダウンロード](#)

また、ExtendScriptを記述するために、ESTK以外で、良いエディタ・開発環境はありますか？

お薦めエディタ・拡張、開発環境などがありましたら紹介いただけると助かります。



Correct Answer

by 10 A on Dec 15, 2017 7:47 PM

ESTKが風前の灯火になってきた感が出てきてちょっとアレなんですけど、スクリプト周りを一新してモダンな感じにする布石だったりすると良いなと思います
(超ポジティブ

ちなみにわたしのイチオシはBracketsです。

```
15
16 var targetStrike = 6;
17 var cmap = readCmap(0xc4c);
18 var uni = decodeSurrogatePair(app.selection[0].contents);
19 var tgGID = cmap[uni];
20
21 tg.seek(sbxOffset);
22 var info = [];
23 var strikes = [];
24 var strikeOffsets = [];
25 var version = readUShort(tg);
26 var flags = readUShort(tg);
27 var strikeLen = readULong(tg);
28 for (var i=0; i<strikeLen; i++) strikeOffsets.push(readULong(tg)+sbxOffset);
29 tg.seek(strikeOffsets[targetStrike]);
30
31 //read glyph table
32 var ppem = readUShort(tg);
33 var resolution = readUShort(tg);
34 var glyphOffsets = [];
35 for (i=0; i<glyphLen; i++) glyphOffsets.push(readULong(tg));
36 tg.seek(strikeOffsets[targetStrike] + glyphOffsets[0]);
37
38 var orignOffsets = [];
39 var type = "", a = "";
40 var f, chr;
41 tg.seek(strikeOffsets[targetStrike] + glyphOffsets[tgGID]);
42 orignOffsets.push([readUShort(tg), readUShort(tg)]);
43 for (j=0; j<3; j++) type += tg.readch();
44 a = tg.readch();
45 if (a=="f") type += a;
46 //$.writeln("orignOffset: " + orignOffsets[i] + " type: "+type);
47 f = new File("~/Desktop/colorGlyph_" + tgGID + "." + type);
48 f.encoding = 'BINARY';
49 f.open("w");
50 //$.writeln(tg.tell());
51 for (var k=0; k<glyphOffsets[tgGID+1]-glyphOffsets[tgGID]; k++){
52   chr = tg.readch().charCodeAt(0);
53   f.write(String.fromCharCode(chr));
54 }
55 f.close();
56 tg.close();
57 var tgt = app.selection[0];
58 var rct = tgt.insertionPoints[0].rectangles.add();
59 rct.strokeColor = app.activeDocument.swatches.item(0);
60 rct.fillColor = app.activeDocument.swatches.item(0);
61 rct.place(f);
62 rct.fit(FitOptions.PROPORTIONALLY);
63 rct = rct.anchoredObjectSettings.anchoredPosition = AnchorPosition.INLINE_POSITION;
64 app.selection[0].contents = "";
65
66 fo
67 focus fn()
68 for keyword
69 FormData fn()
70 Float32Array fn(size: number)
71 Float64Array fn(size: number)
72 Function fn(body: string): fn()
73 function keyword
74 parseFloat fn(string: string): number
75
76 (lo - 0xDC00);
```

エクステンションのカスタマイズが自分には合っているのでESTKと併用で使っています。

話は飛びますが、以下ではTypeScriptを利用した開発環境なども模索されています。型定義ファイルの整備等で中々追いつかないようですが...Adobe社自体がこういったものに協力してくれればとてもおもしろくなるのですが...

TypeScript Enabled CEP Development

2521 Views Tags :

3 HELPFUL

MOST LIKED

10 A Dec 15, 2017 7:47 PM (in response to noellabo)

Correct Answer Re: ExtendScriptを書くのに適した良いエディタ・開発環境はありますか？

ESTKが風前の灯火になってきた感が出てきてちょっとアレなんですけど、スクリプト周りを一新してモダンな感じにする布石だったりすると良いなって思います（超ポジティブ

ちなみにわたしのイチオシはBracketsです。

```
15
16 var targetStrike = 6;
17 var cmap = readCmap(0xc4c);
18 var uni = decodeSurrogatePair(app.selection[0].contents);
19 var tgGID = cmap[uni];
20
21 tg.seek(sbxOffset);
22 var info = [];
23 var strikes = [];
24 var strikeOffsets = [];
25 var version = readUShort(tg);
26 var flags = readUShort(tg);
27 var strikeLen = readULong(tg);
28 for (var i=0; i<strikeLen; i++) strikeOffsets.push(readULong(tg)+sbxOffset);
29 tg.seek(strikeOffsets[targetStrike]);
30
31 //read glyph table
32 var ppem = readUShort(tg);
33 var resolution = readUShort(tg);
34 var glyphOffsets = [];
35 for (i=0; i<glyphLen; i++) glyphOffsets.push(readULong(tg));
36 tg.seek(strikeOffsets[targetStrike] + glyphOffsets[0]);
37
38 var orignOffsets = [];
39 var type = "", a = "";
40 var f, chr;
41 tg.seek(strikeOffsets[targetStrike] + glyphOffsets[tgGID]);
42 orignOffsets.push([readUShort(tg), readUShort(tg)]);
43 for (j=0; j<3; j++) type += tg.readch();
44 a = tg.readch();
45 if (a=="f") type += a;
46 //$.writeln("orignOffset: " + orignOffsets[i] + " type: "+type);
47 f = new File("~/Desktop/colorGlyph_" + tgGID + "." + type);
48 f.encoding = 'BINARY';
49 f.open("w");
50 //$.writeln(tg.tell());
51 for (var k=0; k<glyphOffsets[tgGID+1]-glyphOffsets[tgGID]; k++){
52   chr = tg.readch().charCodeAt(0);
53   f.write (String.fromCharCode(chr));
54 }
55 f.close();
56 tg.close();
57 var tgt = app.selection[0];
58 var rct = tgt.insertionPoints[0].rectangles.add();
59 rct.strokeColor = app.activeDocument.swatches.item(0);
60 rct.fillColor = app.activeDocument.swatches.item(0);
61 rct.place(f);
62 rct.fit(FitOptions.PROPORTIONALLY);
63 rct = rct.anchoredObjectSettings.anchoredPosition = AnchorPosition.INLINE_POSITION;
64 app.selection[0].contents = "";
65
66 fo
67 focus fn()
68 for keyword
69 FormData fn()
70 Float32Array fn(size: number)
71 Float64Array fn(size: number)
72 Function fn(body: string): fn()
73 function keyword
74 parseFloat fn(string: string): number
75
76 (lo - 0xDC00);
```

エクステンションのカスタマイズが自分には合っているのでESTKと併用で使っています。

話は飛びますが、以下ではTypeScriptを利用した開発環境なども模索されています。型定義ファイルの整備等で中々追いつかないようですが...Adobe社自体がこういったものに協力してくれればとてもおもしろくなるのですが...

TypeScript Enabled CEP Development

2 people found this helpful

Actions

Edit

Delete

Like (4)

macneko.ayu Dec 11, 2017 1:26 AM (in response to 10 A)

Re: ExtendScriptを書くのに適した良いエディタ・開発環境はありますか？

私はBracketsを推してます。

ESTKと違い、軽いし、基本的な補完も効くので、ESTKはデバッグのみ使用しています。

Visual Studio Codeでもいいと思います。

Actions ▾

 Edit

 Delete

 Report Abuse

 Like (2)



noellabo Dec 11, 2017 5:28 PM (in response to noellabo)

Re: ExtendScriptを書くのに適した良いエディタ・開発環境はありますか？

お二方、ご回答ありがとうございます。

ESTK、デバッグについていえばコレを使うしかないですし、私も「パワーアップして帰ってくる！」という方向で捉えておこうと思います(笑)

Brackets良さそうですね。

AtomとVisual Studio Codeは軽く試してみましたが、Bracketsはまだです。

- スクリプトの実行をエディタからキックする（何らかの方法でアプリを指定しておければヨシ）
- ESTKとショートカットなどで切り替える（普段はBracketsでやって、デバッグ時に切り替える・開いてるファイルを引き継ぐ）

あたりのところを自動化しておけばストレスなくいけるかな.....。

Adobe側で色々環境を整えてくれれば最高ですね！

Actions ▾

 Edit

 Delete

 Report Abuse

 Like (0)

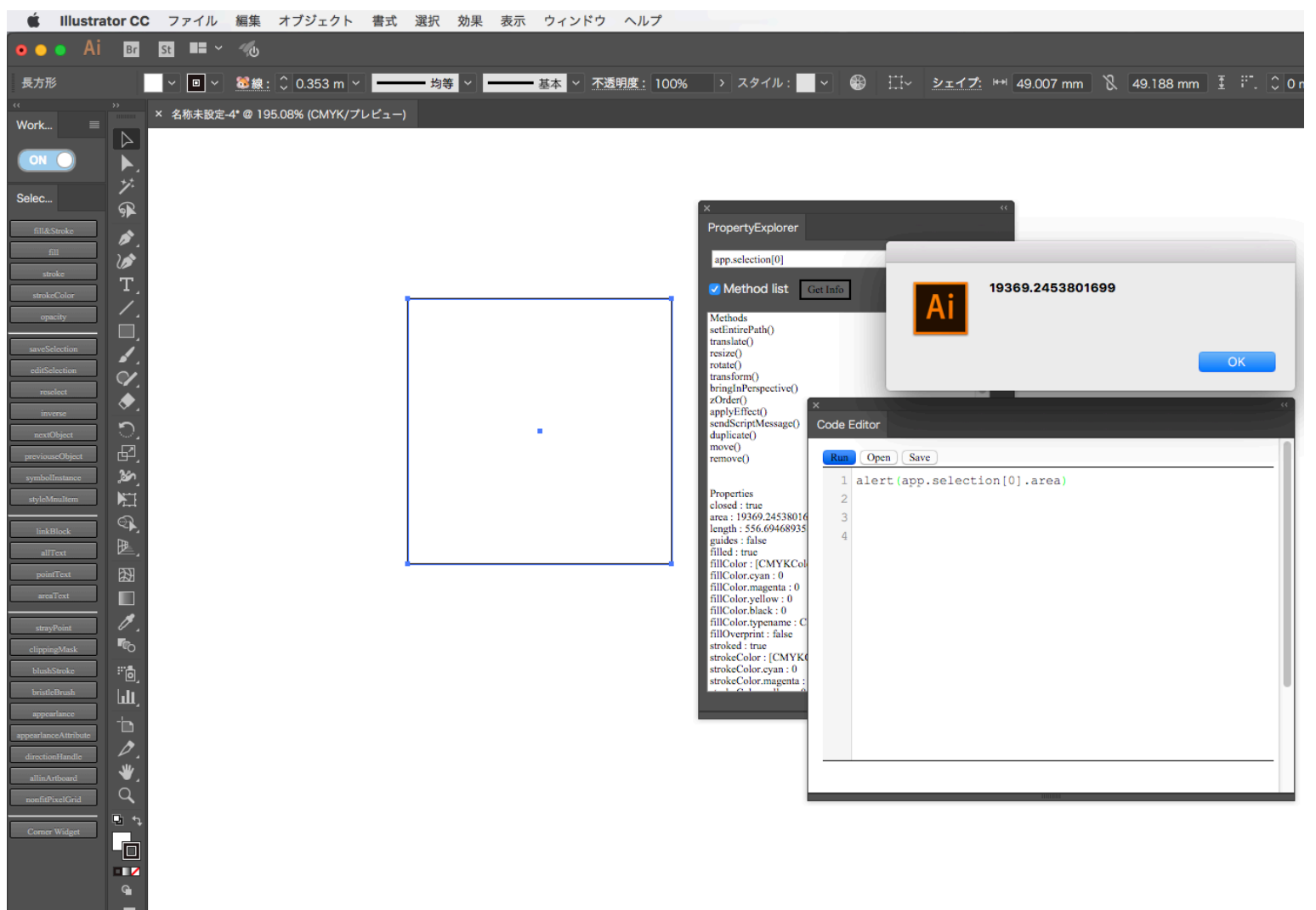


10 A Dec 11, 2017 6:53 PM (in response to noellabo)

Re: ExtendScriptを書くのに適した良いエディタ・開発環境はありますか？

とは言うものの、あっちこっち行ったり来たりは面倒なのも事実です。ので、わたしはスニペットテストやちょっとした書き換え等はエクステンションを利用してアプリケーション上でデバッグします。

ACP

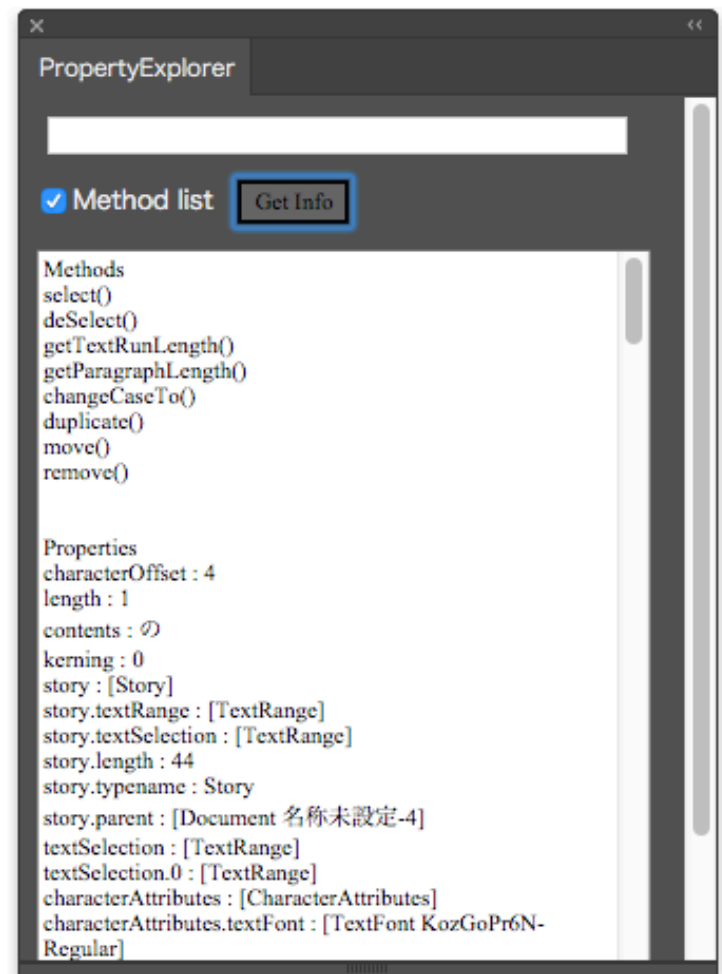
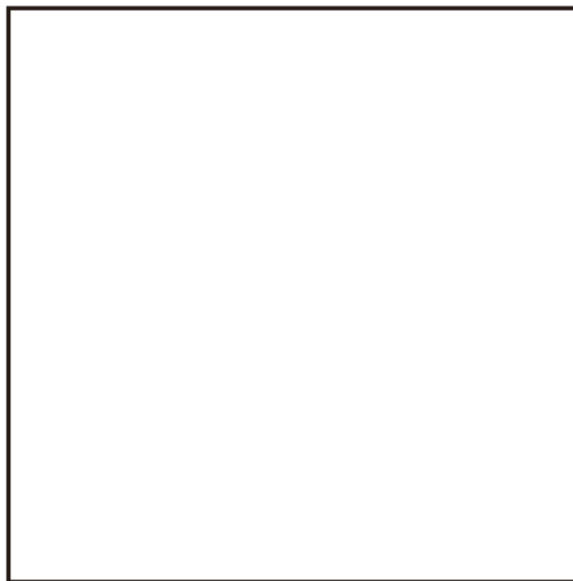


サンプルキャプチャはIllustratorCC2017上でエクステンションによるプロパティ調査及びスクリプトの編集・実行です。

エディタの方はCodemirrorが組み込まれていてJavaScriptのコード補完も実装されています。

プロパティやメソッドの調査にはPropertyExplorerが役に立ちます。上のサンプルでは対象オブジェクトをテキストボックスに入力して各プロパティを取得していますが、下のようにブランクのままだと選択オブジェクトのプロパティを収集します。

私はその人を常に先生と呼んでいた。だからここでもただ先生、



PropertyExplorerはAdobe公式のAddOnsから直接導入可能です。

<https://exchange.adobe.com/addons/products/1170#.Wi87dGWJVZ4>

CodeEditorは以下のリンクより入手可能となっています。

[CodeMirrorを組み込んだJavascriptエディタパネル - 手抜きLab@DTPの現場](#)



4 people found this helpful

Actions ▾

Edit

Delete

Like (3)



noellabo Mar 29, 2018 12:58 PM (in response to noellabo)

Re: ExtendScriptを書くのに適した良いエディタ・開発環境はありますか？

結局、ESTKとVSCodeで書いてます。

Adobe Script Runnnerを利用して、Cmd+Rで実行できるようにしました。
これだけで、十分快適に書けるようになりました。

MVP

VSCode用 [Adobe Script Runner - Visual Studio Marketplace](#)

Atom用 [adobe-script-runner](#)

いろいろありがとうございました。

2 people found this helpful

Actions ▾

 Edit

 Delete

 Report Abuse

 Liked (2)



ACP

10 A Mar 29, 2018 3:17 PM (in response to noellabo)

Re: ExtendScriptを書くのに適した良いエディタ・開発環境はありますか？

Adobe Script Runner良いですね！

Actions ▾

 Edit

 Delete

 Like (0)