

HiroyukiSato Jul 23, 2018 5:26 AM

[Ai] 書類の文字スタイルに設定されているフォントを使った文字のサイズ変更について

This question has been **Answered**.

Illustrator でテキストオブジェクトを選択した状態で、

```
01. activeDocument.selection[0].textRange.characterAttributes.size = 12;  
01. activeDocument.selection[0].textRange.characterAttributes.size = 12;
```

とやっても 12pt にならないケースがあったので、ウェブを調べたところ、Adobe のスクリプティングフォーラムの記事がありました。

(Why won't my script resize the font to size 13? <https://forums.adobe.com/thread/2189589>)

ここで提示されている解決方法で、いったん別のフォントに変更してからサイズを変え、また元のフォントに戻す、という手順を踏むと、変えることができました。

ただ、スクリプトでこれを行う場合、「別のフォント」を何にするかが問題になります。

というのも、上の記事では書類のデフォルトフォントを使用したテキストオブジェクトの場合に起こる現象とされているようですが、

検証したところ、文字スタイルのどれかに設定されているフォントで起こるようでした。

(追記 7/23-21:24: 起こる現象 = 文字スタイルのどれかに設定されているフォントを使ったテキストオブジェクトは、文字サイズをその文字スタイルのデフォルトサイズに変更することができない)

つまり「別のフォント」が別の文字スタイルで使われていたら上の方法は使えません。

もう 1 つの問題は、「別のフォント」が欧文フォントの場合、日本語の文字に適用されないことです。

つまりそのフォントにない文字のサイズは変えられないということで、日本語フォントならば良いというわけでもありません。

いまのところ、以下のような処理をしていますが、上の問題を踏まえると、いろいろと確実性に欠ける部分があります。

なにか良い方法はないのでしょうか？

```
01. var escape_font = textFonts.getByName("KozGoPro-Regular"); // 「別のフォント」
```

```

02.
03. // (設定したいフォントサイズ(new_font_size)と違うものが、この処理に回ってくるものとする。)
04. function changeFontSize(textObj, new_font_size){
05.     var err = false;
06.     var cattr = textObj.textRange.characterAttributes;
07.
08.     cattr.size = new_font_size;
09.
10.     // 検証。ただ、ここではあくまで1文字目しか見ていない。
11.     if(cattr.size != new_font_size){ // 変わっていない場合
12.         var original_font = cattr.textFont;
13.         cattr.textFont = escape_font;
14.         cattr.size = new_font_size;
15.
16.         if(cattr.size != new_font_size){ // まだ変わっていない
17.             // エラー表示
18.             alert("フォントサイズを変更できませんでした云々");
19.             err = true;
20.         }
21.
22.         // フォントを元に戻す
23.         cattr.textFont = original_font;
24.     }
25.     return err;
26. }
27.
28. // 使用例
29. changeFontSize(activeDocument.selection[0], 12);

```

```

01. var escape_font = textFonts.getByName("KozGoPro-Regular"); // 「別のフォント」
02.
03. // (設定したいフォントサイズ(new_font_size)と違うものが、この処理に回ってくるものとする。)
04. function changeFontSize(textObj, new_font_size){
05.     var err = false;
06.     var cattr = textObj.textRange.characterAttributes;
07.
08.     cattr.size = new_font_size;
09.
10.     // 検証。ただ、ここではあくまで1文字目しか見ていない。
11.     if(cattr.size != new_font_size){ // 変わっていない場合
12.         var original_font = cattr.textFont;
13.         cattr.textFont = escape_font;
14.         cattr.size = new_font_size;
15.
16.         if(cattr.size != new_font_size){ // まだ変わっていない
17.             // エラー表示
18.             alert("フォントサイズを変更できませんでした云々");
19.             err = true;
20.         }
21.
22.         // フォントを元に戻す
23.         cattr.textFont = original_font;
24.     }
25.     return err;
26. }
27.
28. // 使用例
29. changeFontSize(activeDocument.selection[0], 12);

```



Correct Answer

by [kawamoto_a](#) on Jul 23, 2018 11:26 PM

この「文字サイズを12ptにできない」と似たような問題は、「段落を左揃えにできない」とか「打ち消し線をつけられるけど、消すことができない」とか様々な形ででてきます

が、これらは「スタイルのオーバーライドが完全になくなるような文字・段落属性の変更が無効化される」というふうに一般化できるんじゃないかと思っています。

たとえば文字サイズを12ptにできない状況でも、

「トラッキングを10にする→文字サイズを12ptにする」ならできるかと思います。

しかし今度はトラッキングを0に戻せなくなる。

この堂々巡りを回避するための方法の一つが、例外的にこのバグが効かないtextFont属性を変更するというものなのですが、他にもページアイテムの変形を使って、200%に拡大する→文字属性を変更する→50%に縮小して戻すというのを使ってるのも目にします。

私は「めったに使われない文字属性の一つを変更したままにする」というのもアリだと思います。たとえば「

割注設定のスケールを50から49に変更して文字サイズを12にする」とか、どうでしょう？

```
var textObj=app.activeDocument.selection[0].textRange;
textObj.characterAttributes.wariChuScale = 49;
textObj.characterAttributes.size = 12;
```

698 Views Tags :

3 HELPFUL

10 A Jul 23, 2018 6:55 PM (in response to HiroyukiSato)

Re: [Ai] 書類の文字スタイルに設定されているフォントを使った文字のサイズ変更について

フォントに関する問題はとても面倒な事になりがちですが、すこし考えてみました。Illustratorが扱うフォント情報はApplicationクラスにあるtextFontsかtextRange配下のcharacterAttributesクラスの2つが基本になっています。ドキュメント上で利用されているフォントを知るためにはドキュメント上のtextRangeを総当たりしてリストアップするか、既存のファイルを開いているのであればXMPメタデータを参照することでも抽出する事が可能です。

アプリケーションが利用できるフォントは単純にapp.textFontを参照する事で得られますからそのリストと以下のようなコードで抽出したリストを突き合わせてドキュメント上で利用していないフォントを指定すると件の条件は回避できそうです。

```
01. function usedFonts(){
02.     var tmp = '';
03.     var d = app.activeDocument;
04.     if (d.textFrames.length>0){
05.         var fLst = d.textFrames[0].characters[0].characterAttributes.textFont.name + '\n';
06.         for (var i=0;i<d.textFrames.length;i++){
07.             for (var j=0;j<d.textFrames[i].characters.length;j++){
08.                 tmp = d.textFrames[i].characters[j].characterAttributes.textFont.name;
09.                 eval('re = /'+tmp+';/');
10.                 if (!re.test(fLst)){
11.                     fLst += tmp + '\n';
12.                 }
13.             }
14.         }
```

```

15.     } else {
16.         return "There are no textFrames.";
17.     }
18.     return fLst.split('\n');
19.     return fLst;
20. }

01. function usedFonts(){
02.     var tmp = '';
03.     var d = app.activeDocument;
04.     if (d.textFrames.length>0){
05.         var fLst = d.textFrames[0].characters[0].characterAttributes.textFont.name + '\n';
06.         for (var i=0;i<d.textFrames.length;i++){
07.             for (var j=0;j<d.textFrames[i].characters.length;j++){
08.                 tmp = d.textFrames[i].characters[j].characterAttributes.textFont.name;
09.                 eval('re = /'+tmp+';/');
10.                 if (!re.test(fLst)){
11.                     fLst += tmp + '\n';
12.                 }
13.             }
14.         }
15.     } else {
16.         return "There are no textFrames.";
17.     }
18.     return fLst.split('\n');
19.     return fLst;
20. }

```

ここで問題になるのはフォント自体のキャラクタセットですが、Illustratorではこの類に関連するプロパティが設定されていません。これについては幾つかの逃げ道があります。1つはBridgeTalkを利用してInDesignにフォント情報を参照させる方法。もう1つがダイレクトにフォントファイルを参照する方法です。フォントファイルを参照する方法が妥当なのですが、これにも1つ問題があります。Illustratorではフォントファイル及びそのファイルパスの情報を持たない事です。

これについてはフォントフォルダのファイルとIllustratorで得られたnameプロパティを利用して比較しながら特定する以外ありません。フォントファイルが特定できれば内包するnameテーブルからフォントの持つ情報を取り出すことが出来ます。以下のURLはわたしが過去に書いたフォントファイルからランゲージIDを取り出すものです。

<https://gist.github.com/ten-A/aa69b5e4a43bad99b67bed2b4532c19b> 

また、一度保存してしまっってXMPメタデータを作成させるのも1つの手となるかもしれません。

```

01. if (ExternalObject.AdobeXMPScript == undefined) {
02.     ExternalObject.AdobeXMPScript = new ExternalObject('lib:AdobeXMPScript');
03. }
04. var fNm = File.openDialog ("select target file");
05.
06.
07. var xmp = new XMPFile(fNm.fsName, XMPConst.UNKNOWN, XMPConst.OPEN_FOR_READ);
08. obj = xmp.getXMP();
09. xmp.closeFile();
10. var num = obj.countArrayItems("http://ns.adobe.com/xap/1.0/t/pg/", "Fonts");
11. var fnts = [];
12. for (var i=1;i<=num;i++) {
13.     fnts.push(obj.getProperty("http://ns.adobe.com/xap/1.0/t/pg/", "Fonts[" + i + "]/stFnt:fontf
14. }
15.

```



```

16. ExternalObject.AdobeXMPScript.unload();
17. ExternalObject.AdobeXMPScript = undefined;
18.
19. alert(fnts);
01. if (ExternalObject.AdobeXMPScript == undefined) {
02.     ExternalObject.AdobeXMPScript = new ExternalObject('lib:AdobeXMPScript');
03. }
04. var fNm = File.openDialog ("select target file");
05.
06.
07. var xmp = new XMPFile(fNm.fsName, XMPConst.UNKNOWN, XMPConst.OPEN_FOR_READ);
08. obj = xmp.getXMP();
09. xmp.closeFile();
10. var num = obj.countArrayItems("http://ns.adobe.com/xap/1.0/t/pg/", "Fonts");
11. var fnts = [];
12. for (var i=1;i<=num;i++) {
13.     fnts.push(obj.getProperty("http://ns.adobe.com/xap/1.0/t/pg/", "Fonts[" + i + "]/stFnt:fontf
14. }
15.
16. ExternalObject.AdobeXMPScript.unload();
17. ExternalObject.AdobeXMPScript = undefined;
18.
19. alert(fnts);

```

この様にAdobeXMPScriptというExternalObjectを利用するとドキュメントフォントのリストとファイルネームが抽出できます。

と、長々と書きましたが、Illustratorの場合はフォント周りをどう扱おうがスマートに処理出来ないというのが実情です。恐らくトライ&エラーによる処理の方がコンパクトに書けると思います。

4 people found this helpful

Actions ▾

 Edit

 Delete

 Like (0)

HiroyukiSato Jul 23, 2018 8:27 PM (in response to 10 A)

Re: [Ai] 書類の文字スタイルに設定されているフォントを使った文字のサイズ変更について

Ten A さん

返信ありがとうございます。フォントの仕組みについて理解が深まりました。
ただ今回はフォントのキャラクタセットやランゲージIDを参照するところまで踏み込むと大変そうですね。
フォントを置き換える方法をとるなら、やはり1文字ずつトライ&エラーで処理結果を確認していくほうが間違いないのだろうと思いました。

私の質問の文章構成がわかりにくかったのですが、ここで問題になるのはドキュメントの「文字スタイル」で使用されているフォントで、TextFrameが1つもなくても、少なくとも「[標準文字スタイル]」は存在しますし、TextFrameで使われていないフォントでも「文字スタイル」で使用されている可能性はあります。
なので今回の場合、文字スタイルで使われていないフォントを取得するには
activeDocument.characterStyles と app.textFonts を突き合わせる形になると思います。

以下は今しがた考えた、フォントを置き換えない解決策です。

とりあえず機能しているようですが、どんな落とし穴があるかわからないので、エラー表示の処理は入れています。

機能しないケースや考慮が抜けている点があればご指摘頂けると有難いです。

また、これも無理やりな感じなので、もっとすっきり正攻法のやり方があれば知りたいです。

```
01. function changeFontSize(textObj, new_size){
02.     var err = false;
03.     var cattr = textObj.textRange.characterAttributes;
04.     cattr.size = new_size;
05.
06.
07.     if(cattr.size != new_size){
08.         //alert("変わってない");
09.         if(textObj.textRange.characterStyles.length > 0){
10.             // ↑現在のスタイルは以下では参照しないが、
11.             // おそらく以下の処理は文字スタイル絡みでのみ有効だろうという目算。
12.             try{
13.                 // いったん仮のスタイルを適用し、サイズ変更後に仮のスタイルは削除する。
14.                 // ↓実際は本当に存在しないスタイル名を決めるための処理が別途必要
15.                 var dummyStyle = activeDocument.characterStyles.add("存在しないスタイル名");
16.                 // 仮スタイルの設定は文字サイズのみ。
17.                 // 文字サイズは new_size と異なるものにする。
18.                 // （文字スタイルのデフォルトサイズへの変更はできないので）
19.                 dummyStyle.characterAttributes.size = new_size + 1;
20.                 dummyStyle.applyTo(textObj.textRange);
21.
22.
23.                 // 再定義。しないとイラレが異常終了する。
24.                 cattr = textObj.textRange.characterAttributes;
25.                 cattr.size = new_size;
26.             } finally {
27.                 dummyStyle.remove();
28.             }
29.         }
30.         if(cattr.size != new_size){
31.             // エラー表示
32.             err = true;
33.             alert("まだ変わってない");
34.         }
35.     }
36.     return err;
37. }
38. // 使用例
39. var err = changeFontSize(activeDocument.selection[0], 12);
```

```
01. function changeFontSize(textObj, new_size){
02.     var err = false;
03.     var cattr = textObj.textRange.characterAttributes;
04.     cattr.size = new_size;
05.
06.
07.     if(cattr.size != new_size){
08.         //alert("変わってない");
09.         if(textObj.textRange.characterStyles.length > 0){
10.             // ↑現在のスタイルは以下では参照しないが、
11.             // おそらく以下の処理は文字スタイル絡みでのみ有効だろうという目算。
12.             try{
13.                 // いったん仮のスタイルを適用し、サイズ変更後に仮のスタイルは削除する。
14.                 // ↓実際は本当に存在しないスタイル名を決めるための処理が別途必要
15.                 var dummyStyle = activeDocument.characterStyles.add("存在しないスタイル名");
16.                 // 仮スタイルの設定は文字サイズのみ。
17.                 // 文字サイズは new_size と異なるものにする。
```

```
18. // (文字スタイルのデフォルトサイズへの変更はできないので)
19. dummyStyle.characterAttributes.size = new_size + 1;
20. dummyStyle.applyTo(textObj.textRange);
21.
22.
23. // 再定義。しないとイラレが異常終了する。
24. cattr = textObj.textRange.characterAttributes;
25. cattr.size = new_size;
26. } finally {
27.     dummyStyle.remove();
28. }
29. }
30. if(cattr.size != new_size){
31.     // エラー表示
32.     err = true;
33.     alert("まだ変わってない");
34. }
35. }
36. return err;
37. }
38. // 使用例
39. var err = changeFontSize(activeDocument.selection[0], 12);
```

Actions ▾

 Edit

 Delete

 Report Abuse

 Liked (1)



monokano Jul 23, 2018 9:08 PM (in response to HiroyukiSato)

Re: [Ai] 書類の文字スタイルに設定されているフォントを使った文字のサイズ変更について

そのフォントにない文字のサイズは変えられない

[GitHub - adobe-fonts/adobe-notdef: Adobe NotDef](https://github.com/adobe-fonts/adobe-notdef)

こちらで公開されている「AND-Regular.otf」は、U+0001～U+10FFFDまでの文字がすべて使用できます。

(文字がすべて豆腐になりますが、文字化けではなく、全コードポイントで豆腐グリフが表示されるフォントです)

ROSがAdobe-Identity-0なので欧文フォントではありません。

試していないのですが、もしかするとこのフォントでうまくできるかもしれません。

追記： うまくできたかも。

```
01. var target = activeDocument.selection[0];
02. var preFont = target.textRange.characterAttributes.textFont;
03. target.textRange.characterAttributes.textFont = app.textFonts.getByName("AND-Regular");
04. target.textRange.characterAttributes.size = 12;
05. target.textRange.characterAttributes.textFont = preFont;

01. var target = activeDocument.selection[0];
02. var preFont = target.textRange.characterAttributes.textFont;
03. target.textRange.characterAttributes.textFont = app.textFonts.getByName("AND-Regular");
04. target.textRange.characterAttributes.size = 12;
05. target.textRange.characterAttributes.textFont = preFont;
```

4 people found this helpful

Actions ▾

 Edit

 Delete

 Report Abuse

 Liked (1)

Re: [Ai] 書類の文字スタイルに設定されているフォントを使った文字のサイズ変更について

monokanoさん、情報ありがとうございます。

そんなフォントがあったんですね！

質問文にあるコードの `escape_font` に設定して、簡単な日欧混在のテキストのサイズを変更してみたところ期待通りに機能しました。

このフォントなら別の文字スタイルに使われることもなさそうですし、あらかじめこのフォントを使える状態にしておくことができる環境であれば、フォント置き換え法に使うには最適なフォントだと思います。

Actions

Edit

Delete

Report Abuse

Like (0)

monokano Jul 23, 2018 9:27 PM (in response to HiroyukiSato)

Re: [Ai] 書類の文字スタイルに設定されているフォントを使った文字のサイズ変更について

AND-Regular.otf は SIL Open Font License なので、とくに許可を得ることなくフォントファイルをスクリプトと一緒に配布するのも可能です。

Actions

Edit

Delete

Report Abuse

Liked (2)

kawamoto_α Jul 23, 2018 11:26 PM (in response to HiroyukiSato)

Correct Answer Re: [Ai] 書類の文字スタイルに設定されているフォントを使った文字のサイズ変更について

この「文字サイズを12ptにできない」と似たような問題は、「段落を左揃えにできない」とか「打ち消し線をつけられるけど、消すことができない」とか様々な形ででてきますが、これらは「スタイルのオーバーライドが完全になくなるような文字・段落属性の変更が無効化される」というふうに一般化できるんじゃないかと思っています。

たとえば文字サイズを12ptにできない状況でも、

「トラッキングを10にする→文字サイズを12ptにする」ならできるかと思います。

しかし今度はトラッキングを0に戻せなくなる。

この堂々巡りを回避するための方法の一つが、例外的にこのバグが効かない `textFont` 属性を変更するというものなのですが、他にもページアイテムの変形を使って、200%に拡大する→文字属性を変更する→50%に縮小して戻すというのを使ってるのも目にします。

私は「めったに使われない文字属性の一つを変更したままにする」というのもアリだと思います。たとえば「

割注設定のスケールを50から49に変更して文字サイズを12にする」とか、どうでしょう？

```
var textObj=app.activeDocument.selection[0].textRange;
```

```
textObj.characterAttributes.wariChuScale = 49;
```

```
textObj.characterAttributes.size = 12;
```


**HiroyukiSato** Jul 24, 2018 2:53 AM (in response to kawamoto_α)

Re: [Ai] 書類の文字スタイルに設定されているフォントを使った文字のサイズ変更について

kawamoto_α さん、情報ありがとうございます。

この問題の仕組み(バグ)が、スタイル絡みの他の問題の仕組みとあわせて腑に落ちました。

wariChuScaleも含めて、どの属性の変更が適切なのかはユーザーごとの事情によりそうですね。

monokanoさん情報のフォントを使った方法も、属性変更の痕跡を残さずにすっきりした形で処理が書けて良いと思いますが、

この現象の本質的な部分が理解できたうえで様々な対策をとれるという意味で、kawamoto_α さんの返信を正解としたいと思います。