[AI]Javascriptの機能を拡張する方法

Version 1

Created by 10 A on Jan 5, 2018 7:43 PM. Last modified by 10 A on Jan 5, 2018 7:43 PM.

イラストレータでのスクリプティング処理時に処理したいプロパティが存在せず手詰まりになる事と言うのは多々あります。ExtendScriptではExternalObjectという仕組み が存在し、**C++**でライブラリを書くことによりスクリプティング機能を拡張する事が可能なのですが、これはアプリケーションDOMへ直接アクセスする手段としては使えません。しかし、イラストレータでは**SendScriptMessage**という仕組みが存在し**JavaScriptから直接プラグインへアクセスする事が可能** です。
今回はこのプラグインを利用したスクリプト環境の機能拡張方法を解説します。

Illustrator SDKの入手

以下のリンクからプラグイン開発用のSDKが入手可能になっています。また、ドキュメント類も一通り揃っていますのでざっとでも目を通していただければと思います。

Adobe I/O Console

SDKサンプルの中には今回のテーマである**SendScriptMessage**に関するサンプルプロジェクトが含まれています。しかしこのプロジェクトはCSXS（エクステンション）用のインターフェース等も含まれる為少し複雑なものとなり、少し敷居が高く感じるかもし れません。今回は内容を出来るだけ簡素化するためにSendScriptMessageの解説に特化したプロジェクトを用意しました。
このサンプルプロジェクトは以下のgithubから入手可能です。Mac版のみですが、ソースコードは単純なのでWindowsへの転用もそう難しくないと思います。

GitHub - ten-A/Bleed

Bleedプラグイン

イラストレータのスクリプティングDOMにはドキュメントの塗り足しに関する項目が欠落しています。これ、使えると結構便利なはずです。これを**SendScriptMessage**インターフェースを利用してセット出来るようにします。
先のプロジェクトのソースを確認して頂けるとわかると思いますが、全部合わせても200行いってません。そこらへんで配布されている小洒落たスクリプトの方が複雑ですね。大半が必要な機能を持 つsuitesをメモリ上に展開するための設定とか何かの定義を行っています。この辺は自分で何かを作る場合SDKのsamplecodeの自分が欲しい機能と似たものを参考に記述（コピペ） するとよろしいでしょう。
さて、今回のものではコアとなる重要な処理は以下の30行足らずの関数となります。と言ってもこの関数以外にさしたるコードはありませんが。

```
01. ASErr BleedPlugin::Message(char *caller, char *selector, void *message)
02. {
03.   ASErr error = kNoErr;
04.   if (strcmp(caller, kCallerAIScriptMessage ) == 0)
05.   {
06.   error = kNoErr;
```

```
  AIScriptMessage *msg = (AIScriptMessage*) message;
        if (ai::UnicodeString(selector) == ai::UnicodeString("Bleed"))
        {
            //sAIUser->MessageAlert(ai::UnicodeString(msg->inParam));
            ai::NumberFormat numFormat;
            Float32 bleedOffset;
            error = sAIStringFormatUtils->StringToReal(numFormat, msg->inParam, bleedOffse
            if (bleedOffset<0||bleedOffset>72)
            {
                msg->outParam = ai::UnicodeString("false");
                return error;
            }
            AIRealRect tgRect;
            tgRect.right = bleedOffset;
            tgRect.left = bleedOffset;
            tgRect.top = bleedOffset;
            tgRect.bottom = bleedOffset;
            error = sAIDocument->SetDocumentBleeds(tgRect);
            msg->outParam = ai::UnicodeString("true");
            return error;
        }

  }
  return error;
}
```

```
ASErr BleedPlugin::Message(char *caller, char *selector, void *message)
{
  ASErr error = kNoErr;
  if (strcmp(caller, kCallerAIScriptMessage ) == 0)
  {
  error = kNoErr;
  AIScriptMessage *msg = (AIScriptMessage*) message;
        if (ai::UnicodeString(selector) == ai::UnicodeString("Bleed"))
        {
            //sAIUser->MessageAlert(ai::UnicodeString(msg->inParam));
            ai::NumberFormat numFormat;
            Float32 bleedOffset;
            error = sAIStringFormatUtils->StringToReal(numFormat, msg->inParam, bleedOffse
            if (bleedOffset<0||bleedOffset>72)
            {
                msg->outParam = ai::UnicodeString("false");
                return error;
            }
            AIRealRect tgRect;
            tgRect.right = bleedOffset;
            tgRect.left = bleedOffset;
            tgRect.top = bleedOffset;
            tgRect.bottom = bleedOffset;
            error = sAIDocument->SetDocumentBleeds(tgRect);
            msg->outParam = ai::UnicodeString("true");
            return error;
        }

  }
  return error;
}
```

少し抜き出して解説をしてみましょう。

```
if (strcmp(caller, kCallerAIScriptMessage ) == 0)
{...
```

```
if (strcmp(caller, kCallerAIScriptMessage ) == 0)
{...
```


このif文による振り分けは**strcmp**関数が0になる場合（引数同士が同一の文字列の場合）にスクリプトよりコールされたメッセージだと判断するためのものです。
ExtendScript側の**sendScriptMessage**関数では３つの引数が必要で、第１引数がコールするプラグイン名、第２引数がメッセージ・セレクターと呼ばれる文字列でプラグインの機能が複数ある場合にどの処理をするためのメッセージなのか を判断するためのものです。第３引数が引き渡すパラメータとなります。
関数に戻りますが、ES側から送られてきたデータを受け取るための処理が続きます。

```
AIScriptMessage *msg = (AIScriptMessage*) message;
```

```
AIScriptMessage *msg = (AIScriptMessage*) message;
```


AIScriptMessageの構造はそう複雑なものではありませんからSDKドキュメントの方で確認してください。
続いてセレクタによる振り分けが続きます。

```
if (ai::UnicodeString(selector) == ai::UnicodeString("Bleed"))
{...
```

```
if (ai::UnicodeString(selector) == ai::UnicodeString("Bleed"))
{...
```


ai::UnicodeStringはIAIUnicodeString.cppで定義されているものでstring等をUnicodeStringとして利用する場合に重宝する関数です。
続く部分が処理の中心になります。

```
ai::NumberFormat numFormat;
Float32 bleedOffset;
error = sAIStringFormatUtils->StringToReal(numFormat, msg->inParam, bleedOffset);
if (bleedOffset<0||bleedOffset>72)
{
    msg->outParam = ai::UnicodeString("false");
    return error;
}
AIRealRect tgRect;
tgRect.right = bleedOffset;
tgRect.left = bleedOffset;
tgRect.top = bleedOffset;
tgRect.bottom = bleedOffset;
error = sAIDocument->SetDocumentBleeds(tgRect);
msg->outParam = ai::UnicodeString("true");
```

```
ai::NumberFormat numFormat;
Float32 bleedOffset;
error = sAIStringFormatUtils->StringToReal(numFormat, msg->inParam, bleedOffset);
if (bleedOffset<0||bleedOffset>72)
{
    msg->outParam = ai::UnicodeString("false");
    return error;
}
AIRealRect tgRect;
tgRect.right = bleedOffset;
tgRect.left = bleedOffset;
```

```
12.  tgRect.top = bleedOffset;
13.  tgRect.bottom = bleedOffset;
14.  error = sAIDocument->SetDocumentBleeds(tgRect);
15.  msg->outParam = ai::UnicodeString("true");
```

受け取ったパラメータは文字列となります。**Bleedというのは0.0〜72.0（pt)の範囲の浮動小数点で表現される値を取る4つのプロパティを持つAIRealRectという構造**を取ります。このために受け取ったパラメータは文字列から浮動小数点に変換する必要があります。
これを行う為に**AIStringFormatUtils**に含まれる**StringToReal**関数を使います。また、Bleedは上記のレンジ以外の数値を受け取るとエラーを返しますから予め値をチェックしてエラー処理を行う必要があります。
ここまでの処理を通ってきたデータを**AIRealRectのright・left・top・bottom**の各プロパティーに設定し**AIDocument**クラスに含まれる**SetDocumentBleeds**関数を利用して先程のAIRealRectの値を反映します。
また、AIScriptMessageはoutParamを利用する事で返り値をES側に返す事ができます。今回の例では受け取った数値が0〜72のレンジから外れている場合にfalseを正 常にBleedを設定できた場合にtrueを返しています。ひとつ注意が必要なのは返す値はあくまでも文字列でありbooleanでは無いことです。これはES側で対処する必要があります。

Illustrator ver.21以降用コンパイル済みプラグインは以下のリンクよりダウンロード可能です。
Bleed.aip.zip

ExtendScript側の処理ですが、以下のサンプルを参考にしてください。

```
01.  alert(setBleed(36));//unit:pt
02.
03.  function setBleed(n){
04.     var result = app.sendScriptMessage('Bleed', 'Bleed', n);
05.     return result;
06.     }
```

```
01.  alert(setBleed(36));//unit:pt
02.
03.  function setBleed(n){
04.     var result = app.sendScriptMessage('Bleed', 'Bleed', n);
05.     return result;
06.     }
```

先も書きましたが引数の１つ目がプラグイン名称、２つ目がセレクタ、3つ目が引き渡す値となります。
Bleedsのレンジは0〜72の範囲ですからsetBleedの引数はこの範囲に収まるようにして下さい。

817 Views Categories: Tags : illustrator, plugin, extendscript, illustrator sdk

0 Comments