

# CSC.exe



After Effectsのスキプトの**system.callSystem()**から呼び出すコマンドです。

今までいっぱい作ってましたがいっぱいあると管理が面倒だったので、一つにまとめてしまおうと思い作ったものです。

## Usage

```
[csc.exe] system.callSystem command
csc -FromClipboard          or -clip      // クリップボードのテキストを出力。
csc -ToClipboardFromFile or -clipFile // テキストファイル(UTF-8)をクリップボードへ
csc -ToClipboard           or -toclip    // 引数を全部クリップボードへ
csc -FolderDialog          or -fd        // フォルダ選択ダイアログ
csc -DropFolder            or -dd        // D&Dでフォルダを選択
csc -MousePos              or -mp        // マウスカーソルの座標を出力。toSource形式
({x:100,y:100})
csc -MousePosJson          or -mpj       // マウスカーソルの座標を出力。JSON形式
{"x":100,"y":100}
csc -WindowMax             or -wmax      // AEのウィンドウを最大化
csc -WindowMin             or -wmin      // AEのウィンドウを最小化
csc -WindowNormal          or -wn        // AEのウィンドウを通常化
csc -AEProcessList         or -aps       // 起動しているAEのプロセス情報。配列の
toSource形式
csc -LineEdit              or -le        // 1行のテキスト編集
csc -TextFile              or -tf        // テキストファイル編集 csc -tf ファイルパス
<タイトル名>
csc -PCInfo                or -pi        // PC情報
csc -Alert                 or -at        // アラート表示
csc -Calc                  or -cc        // 計算機
csc -JavaScript            or -js        // JavaScript解析
csc -CPPL                  or -cp        // CPP1サイズ byteサイズ文字列が返り値
csc -CPPLFix              or -cpf       // CPP1削除 : csc -cpf "[元Path]" "[変換後
Path]"
csc -KillAerender          or -kr        // aerender強制終了 : csc -kr
```

基本的に、第一引数が機能のセパレータで、その後に細かいパラメータが指定できます。

## csc.exe setup



インストーラを用意しましたので、それでインストールしてください。

デフォルトで以下にインストールされます。

```
C:\Program Files\bry-ful\csc\csc.exe
```

発行元の情報を獲得していないので、インストール時にその警告出ますが、気にしないでください。

## csc.exeの実行

system.callSystemで呼び出す時はWindows形式のパスで呼び出す必要があります。ただし、引数でcscに与えるパスはjs形式でもWindows形式でも構いません。

```
// ダブルクオートでくる
function WS(s){return "\"" + s + "\";"}
// 半角スペースを返す
function SP(){return " ";}

var cscPath = "C:\Program Files\bry-ful\csc\csc.exe";
var sepa = "-clipFile";
var arg1 = "/c/test.txt"; // "C:\\test.txt" でもいい
var cmd = WS(cscPath) + SP() + sepa + SP() + WS(arg1);
system.callSystem(cmd);
```

上記のコードがcsc.exeの呼び出しになります。

---

## csc.exeの返り値

各コマンド毎に必ず文字列を標準出力へ返します。空の文字列(Empty)が返された時はエラーとなります。

"[csc.exe] system.callSystem command"で始まる文字列の場合はヘルプ表示になります。

詳細なドキュメントは完成したら作るつもりです。

## command References

---

### csc -FromClipboard or -clip

クリップボードにあるテキストデータを出力します。

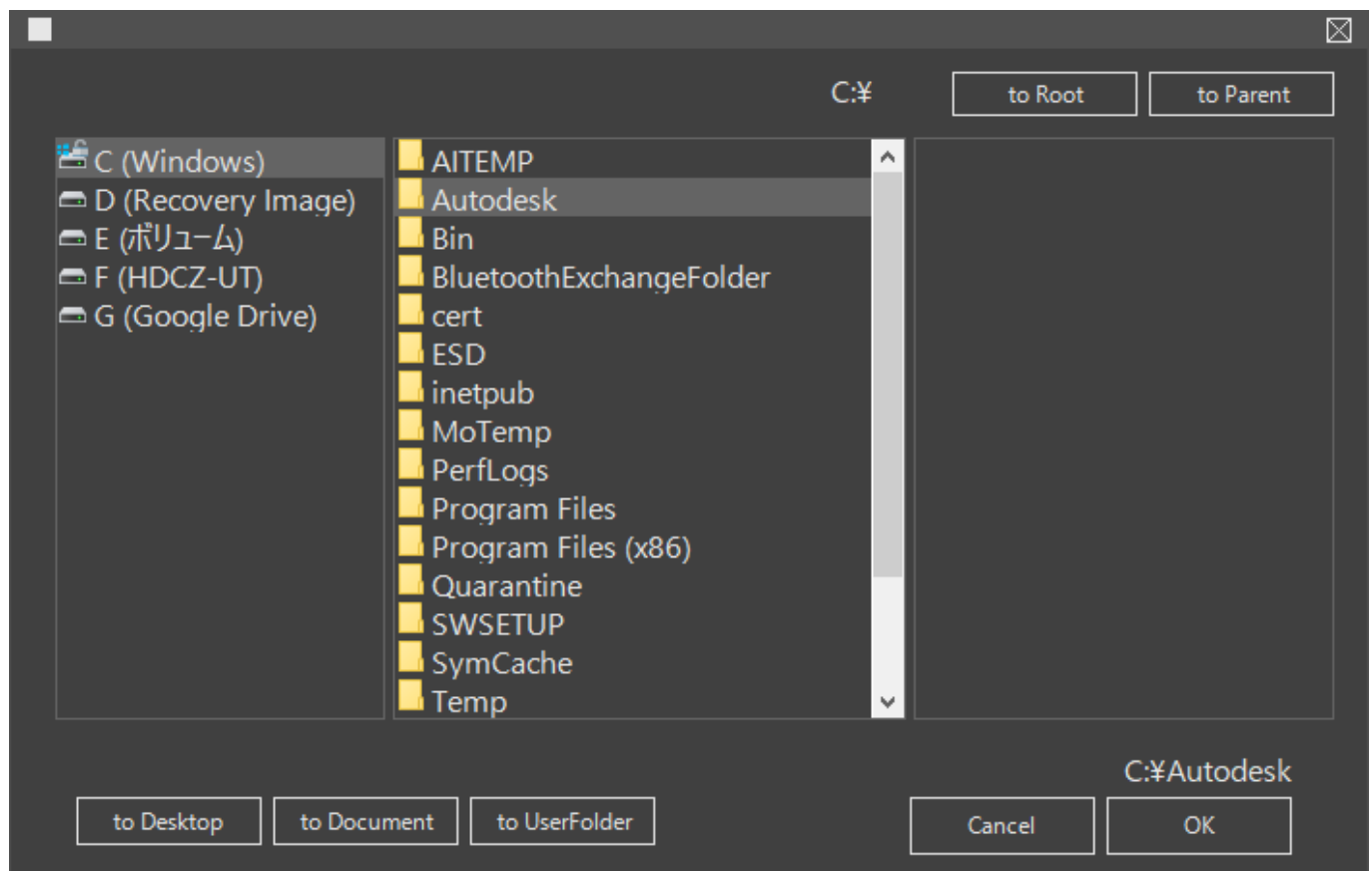
```
var csc = "C:\\Program Files\\bry-ful\\csc\\csc.exe";  
var result = system.callSystem(csc+ " -clip");  
alert(result); // クリップボードの内容
```

## csc -ToClipboardFromFile or -clipFile

指定されたテキストファイルをクリップボードへ取り込みます。

```
system.callSystem(csc+ " -clipFile /c/test.txt");
```

## csc -FolderDialog or -fd



フォルダ選択ダイアログを表示します。

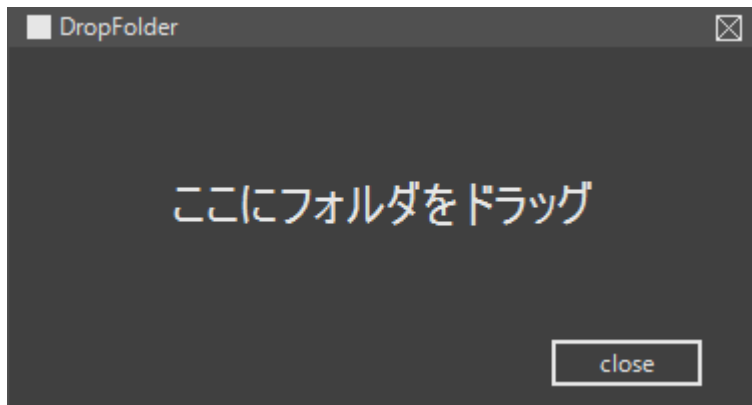
引数にデフォルトのパスを指定できます。

右のファイルリストはAfter Effectsに関係ありそうなファイルしか表示しません。

```
var defpath = "/e/TestFolder";  
var result = system.callSystem(csc + " -fd \"" + defpath + "\"");  
alert(result); // フォルダのパス文字列
```

---

## csc -DropFolder or -dd



ドラッグ&ドロップでフォルダのパスを得るダイアログを表示します。

```
var result = system.callSystem(csc + " -dd");  
alert(result); //フォルダのパス文字列
```

---

## csc -MousePos or -mp

現在のマウス座標をtoSource形式で返します。evalでオブジェクト化して扱えます。

```
var result = system.callSystem(csc + " -mp");  
alert(result); //現在のマウス座標  ({x:100,y:100})
```

MousePosJson は返り値がJSON形式になります。

---

## csc -WindowMax or -wmax

After Effectsのウィンドウを最大化します。

- -WindowMax  
After Effectsのウィンドウを最大化します。
- -WindowMin  
After Effectsのウィンドウを最小化します。
- -WindowNormal  
After Effectsのウィンドウを通常化します。

```
system.callSystem(csc + " -wmax");
```

---

## csc -AEProcessList or -aps

現在起動中のAfter Effectsの状態を取得します。

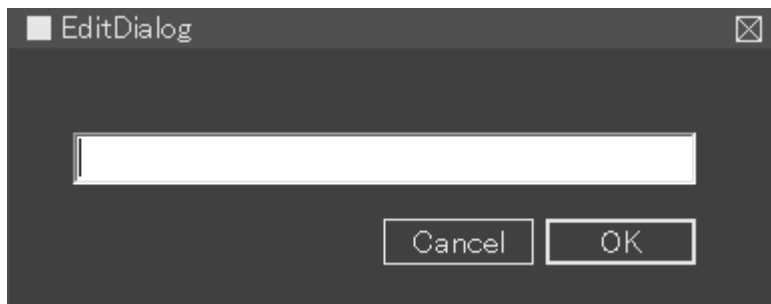
```
var result = system.callSystem(csc + " -aps");  
alert(result);
```

返り値は複数起動された状態に対応するためにtoSource形式の配列になります。

```
[  
  (  
    id:6232,  
    mainWindowTitle : "Adobe After Effects 2021 - 名称未設定プロジェクト.aep",  
    processName : "AfterFX",  
    fileName : "C:\Program Files\Adobe\Adobe After Effects 2021\Support  
Files\AfterFX.exe"  
  )  
]
```

---

### csc -LineEdit or -le

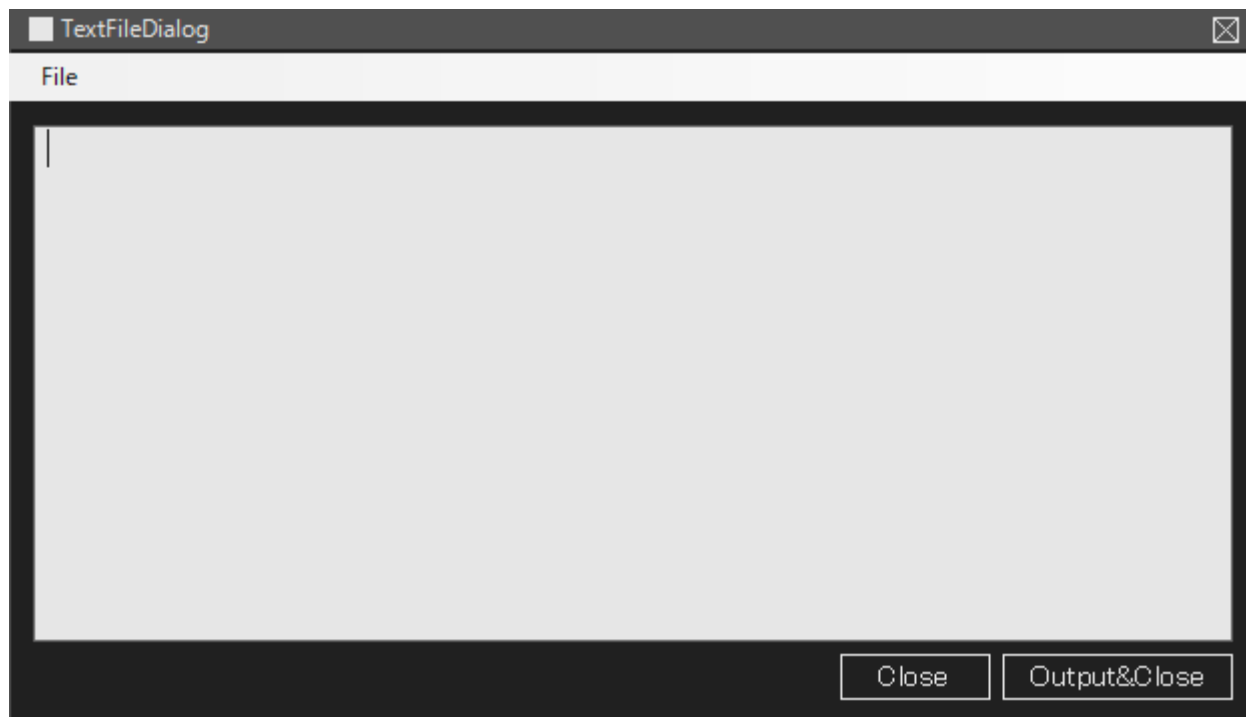


1行テキストエディタダイアログが表示されます。  
文字入力用です。

```
var result = system.callSystem(csc + " -lt");  
alert(result);
```

---

### csc -TextFile or -tf



テキストファイルを読み込んで編集します。引数入れなくても起動します。  
保存等はメニューで行います。

close&Outputで返回值にテキストを返します。

```
var txFile = "/c/aaa.txt";  
var result = system.callSystem(csc + " -lt " + txFile + " テキスト編集");  
alert(result);
```

---

## csc -PCInfo or -pi

PC情報を返します。

```
var result = system.callSystem(csc + " -pi");  
alert(result);
```

返回值はtoSource形式です。

```
(  
  OSName:"Windows 10 Enterprise",  
  OSVersion:"Microsoft Windows NT 10.0.19045.0",  
  PCName:"XXXXXX",  
  UserName:"XXXXX"  
)
```

## csc -Alert or -at



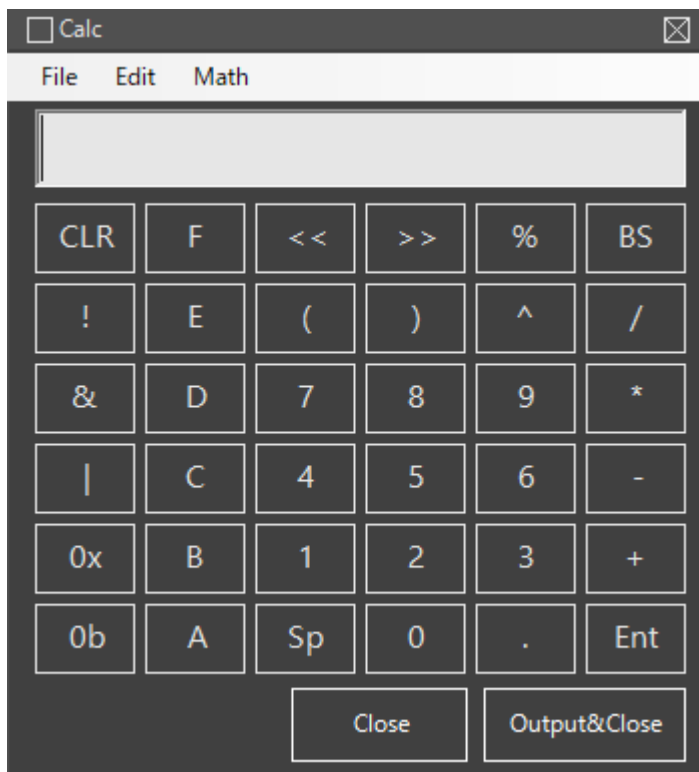
alert表示です。

引数でhtmlファイルを渡せばそれを表示します。普通のテキストファイルでも構いません。  
引数がファイルでなかったらそのまま表示します。

```
var html = "/c/ddd/aaa.html";  
system.callSystem(csc + " -at " + html);
```

---

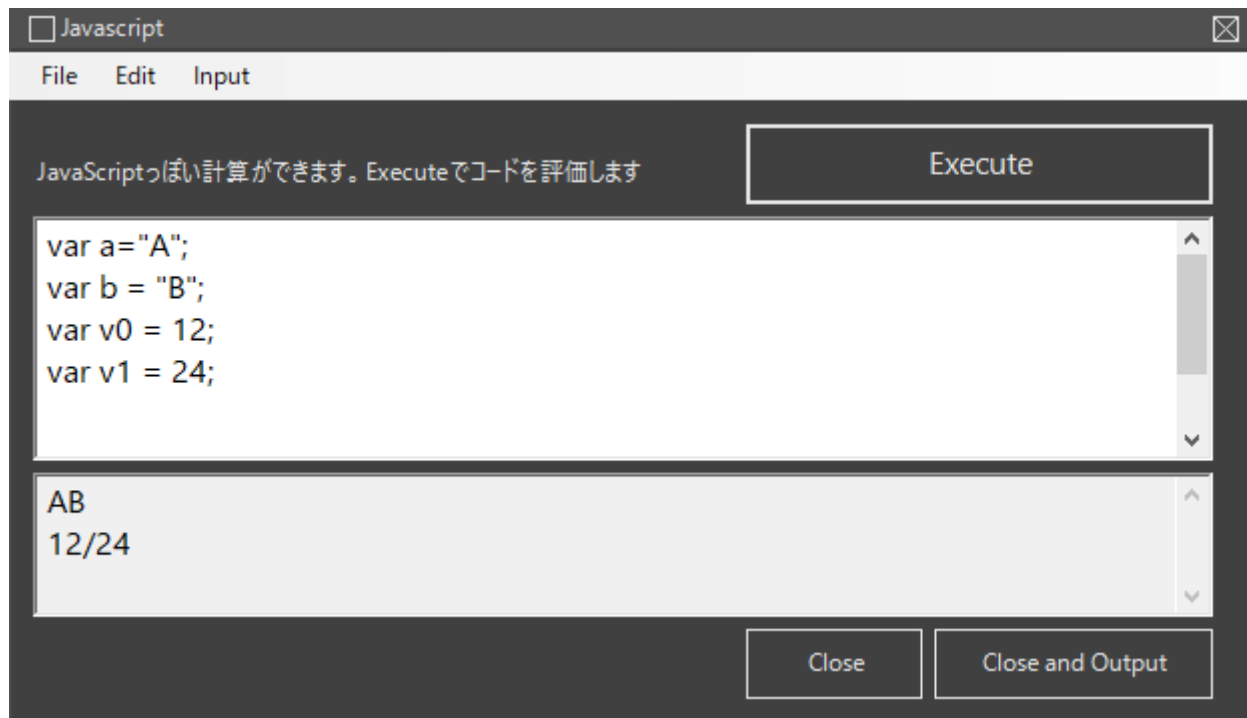
## csc -Calc or -cc



電卓を表示します。

```
var result = system.callSystem(csc + " -cc");  
alert(result);
```

## csc -JavaScript or -js



Javascriptコンソールが出ます。

上記の電卓をClearScriptで実装しなおしたので、おまけでつけてます。

使い道を作者も思いつきません。

```
var result = system.callSystem(csc + " -cc");  
alert(result);
```

## csc -CPPI or -cp

aepファイル内のCPPIタグのサイズを調べます。

通常なら900kbyte前後ですが、それを超えると読み込みが信じられない位重くなります。

aepをインポートするスクリプトを作った時非常に困ったので、スクリプト上から確認するために実装しました。

```
var aepFile = "/c/aep/aaa.aep";  
var result = system.callSystem(csc + " -cp " + aepFile);
```



```
alert(result);
```

## csc -CPPIFix or -cpf

指定したaepファイルのCPPIタグを消去します。  
目的が目的なので、使用には注意してください。

```
var aepFile1 = "/c/aep/aaa.aep";  
var aepFile2 = "/c/aep/aaa_fix.aep";  
var result = system.callSystem(csc + " -cpt " + aepFile1 + " " + aepfile2);
```

---

## csc -KillAerender or -kr

aerenderを強制的に停止させます。

たまにaerenderのプロセスが残ってしまい起動できなくなるので作ったものです。

```
system.callSystem(csc + " -kr");
```

---

## License

This software is released under the MIT License, see LICENSE.

## Authors

bry-ful(Hiroshi Furuhashi)

twitter:[bryful] (<https://twitter.com/bryful>)

bryful@gmail.com

## References

---