



# Adobe Common Extensibility

Version 2

## Platform（CEPエクステンション）について

Created by 10 A on Nov 9, 2017 10:35 PM. Last modified by 10 A on Nov 12, 2017 3:32 PM.

### Web開発スキルを活用してAdobe CC アプリケーションを拡張する

Adobe Common Extensibility Platform（CEP）は、HTML5、CSS3、およびJavaScriptの標準的なWeb技術を使用して、**Adobe CCアプリケーションを拡張するエクステンション**を開発することが出来ます。もう、APIにアクセスするためにC++やプラットフォーム固有のスクリプティングツールを学ぶ必要はありません。また、CEPと統合されたAdobe Exchange（<https://exchange.adobe.com/addons>）ポータルサイトを利用して、CCアプリケーション用エクステンションを販売し、配布することも可能です。

CEF（Chrome Embedded Framework）とNode.jsが組み込まれているため、CEPでは一般的なWebプラットフォームを活用して、各アプリのスクリプティング・オブジェクトモデルにアクセスし、CCアプリケーション用エクステンションを構築できます。詳しい情報及びサンプル・エクステンションは以下の公式gitを参照して下さい。

Adobe CEP公式リポジトリ（英語）

[Adobe CEP・GitHub](#)

CEP概要（英語）

[Creative Cloud Extension SDK - CC extension resources | Adobe I/O](#)

また、以下のフォーラム（英語）も役に立つでしょう。

[Extensions / Add-ons Development](#)

### CEPエクステンションの特徴

まずCEPエクステンションの特徴を見えます。

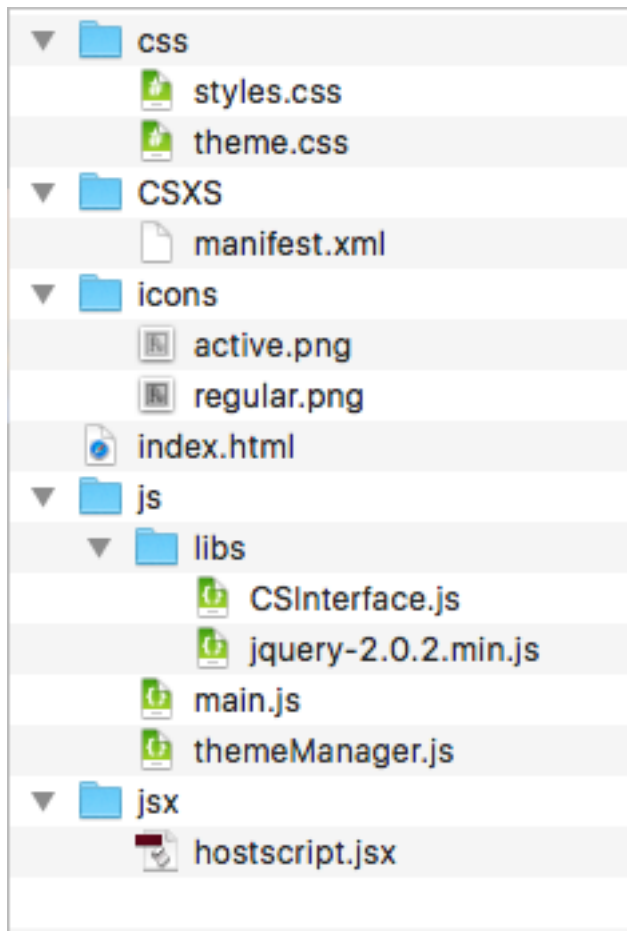
- HTML5+CSS3+Javascriptでユーザーインターフェース（エクステンションパネル）を構成できる。
- UIにはCEFが利用されている為、jQueryやAngularJS等の各種JavaScriptフレームワークに加え、ブラウザー側の実装であるCookieやLocal Storageといったものも利用できる。
- CC 2014以降のバージョンではnode.jsが搭載されている為、NPMモジュールが利用出来る。
- アプリケーションのオブジェクトモデルを直接操作するのはExtendscriptで、2つの異なるJavascriptエンジンをつなぐのはCSInterfaceクラスに内蔵されたevalScriptメソッドだけである。

### エクステンションの種類

一般的な形態はパネルと呼ばれるもので、CCアプリケーション上で、他のパネルとドッキングしたりアイコン化して収納しておける機能を有します。その他モーダルダイアログ、モードレスダイアログ、インビジブル、エンベデッドがあります。各CCアプリケーションによってサポートする種類が異なりますので前述の公式リソース文書にて確認して下さい。

## エクステンションの構造

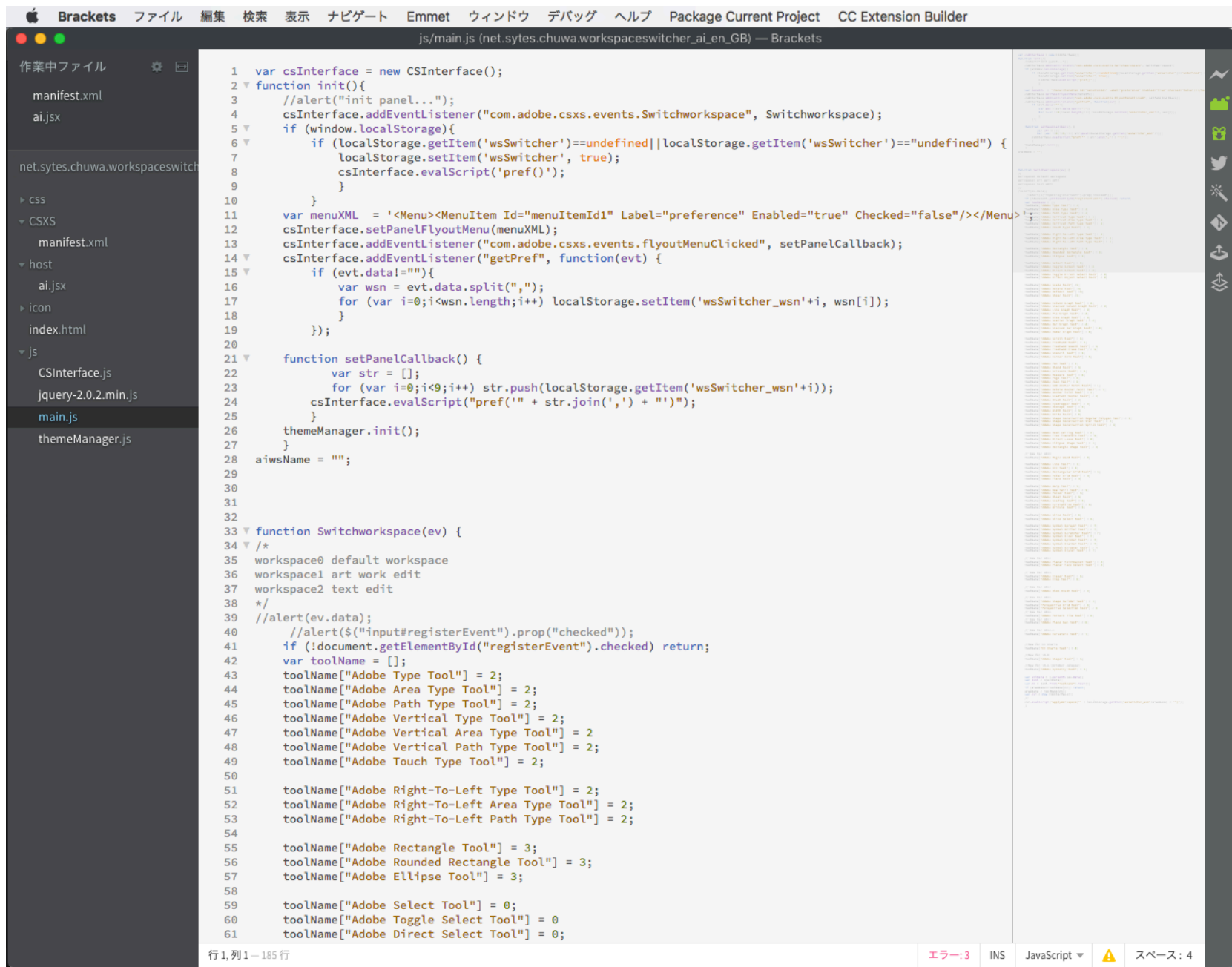
以下に示すのはテンプレートプロジェクトのファイルツリーです。このテンプレートプロジェクトは必要最小限の要素で構成されています。



パネルのUIを構成するのは**index.html**であり、このファイルがmainPathとして**CSXS**フォルダ内の**manifest.xml**内で指定されます。**index.html**は通常のHTMLファイルの構成と同様で、CEPインフラ上では内蔵するCEFがHTMLのレンダリングを担当します。利用されるJavascriptエンジンはV8であり動作的にはChromeだと考えて差し支えありません。通常のブラウザで利用可能なCookieやLocal Storage、そしてIndexed DB等も正常に動作します。また、**CSXSインターフェースクラス** にいくつかの**イベント**が設定されています。これらで幾つかのアプリケーションの状態を監視することが出来ます。しかし、基本的にCEP側のインフラからは直接アプリケーション・オブジェクトモデルへのアクセスは出来ません。CEPインフラから各アプリケーション・オブジェクトモデルへのアクセスは**evalScript**という**CSInterface**クラスに用意されたメソッドを介してのアクセスのみが利用可能だという事に注意が必要です。多くのCCアプリケーションでは、アプリケーション・オブジェクトモデル側は**ExtendScript**を利用しますので、従来から利用されていたスクリプト関連のリソースをそのまま利用することが出来ます。また、プラグイン経由でイベントを発行させるなど、ネイティブ系からダイレクトにCEPへとアクセスさせることも可能です。

## 開発環境

CEPエクステンションの開発は一般的なWeb開発に準じたものなので様々な開発ツールが利用可能ですが、CEPエクステンション開発に便利な機能拡張が提供されている**Brackets**がよく利用されています。CC Extension Builder for BracketsとPackage Current Projectというエクステンション開発用機能拡張を組み込んだ例を以下に示します。



## デバッグモード

CCアプリケーションでは通常、署名されていないエクステンションの動作は許可されていません。しかし、エクステンション開発時のデバッグ作業の為にこの制限を回避することが出来ます。これは PlayerDebugMode と呼ばれるものです。

## 設定方法

●Macの場合／ターミナルから以下のコマンドを実行（CC2018リリースの場合）

**defaults write ~/Library/Preferences/com.adobe.CSXS.8.plist PlayerDebugMode 1**

●Windowsの場合／コマンドプロンプトから以下のコマンドを実行（CC2018リリースの場合）

**reg add HKEY\_CURRENT\_USER\SOFTWARE\Adobe\CSXS.8 /t REG\_SZ /v PlayerDebugMode /d 1 /f**

※設定する必要のある値はターゲットとするCCアプリケーションのバージョンによって異なります。必要なCSXSバージョンを下に示すリンクを参照し適宜書き換えて実行して下さい。

[Quick reference of CSSDK and CEP supported application version](#)

2515 Views Categories: Tags : cep, エクステンション, adobe exchange, playerdebugmode, adobe addons

