Ver00.75

EC-CUBE 2.12.0
プラグイン仕様書

作成：株式会社ロックオン　吉本

改訂履歴

日付	履歴	更新者	更新箇所
2012/3/8	Ver00.01	吉本	作成
2012/3/8	Ver00.02	吉本	文言を修正
2012/3/9	Ver00.03	吉本	文言を修正・テーブル定義を追加
2012/4/11	Ver00.04	吉本	β 版の仕様に沿って修正
2012/4/26	Ver00.50	吉本	正式版の仕様に沿って修正
2012/5/15	Ver00.60	梶原	プラグインのライセンスに関する事項を追記
2012/5/19	Ver00.70	吉本	追加仕様を追加
2012/5/26	Ver00.75	吉本	アーカイブ作成方法についてを追加

目次

1. 参考資料
2. プラグインで出来る事
    1. 処理の介入
    2. テンプレートの変更
    3. トランスフォームの使い方
    4. プラグインのライセンスについて
3. プラグインの作り方
    1. 作り方概要
    2. プラグインファイル構成
    3. アーカイブ作成方法
    4. 本体ディレクトリ構成
    5. ライセンスの表記
    6. 命名規約
    7. スーパーフックポイント
    8. ローカルフックポイント
    9. SC_FormParamのフック
    10. SC_系クラスのフックポイント
    11. テンプレートの変更
    12. ヘッダーにタグ追加

目次

4. リファレンス
    1. SC_Helper_Transform
    2. テーブル定義
    3. 定数一覧

1. 参考資料

1-1　参考資料

・プラグイン関連情報

http://svn.ec-cube.net/open_trac/ticket/1783

・サンプルプラグイン

カテゴリコンテンツ

http://svn.ec-cube.net/open_trac/attachment/ticket/1692/CategoryContents.tar.gz

パンくず

http://svn.ec-cube.net/open_trac/attachment/ticket/1692/TopicPath.tar.gz

2. プラグインで出来ること

2-1　プラグインで出来る事

- **EC-CUBE本体処理へ介入して処理・結果を書き換える。**
  - フックポイント機能を使って実現
  - すべてのPageクラス・SCクラスで介入可能

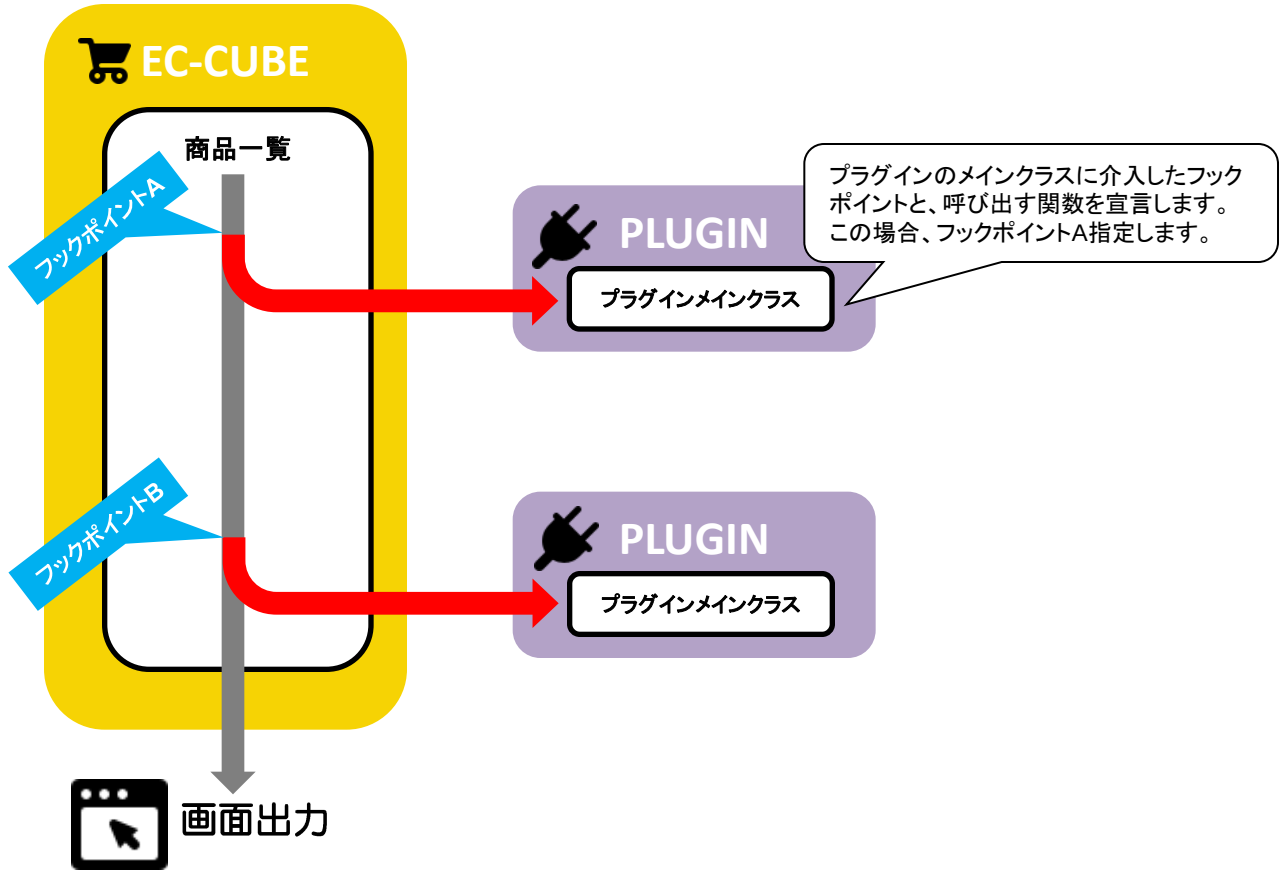
- **EC-CUBE本体テンプレートを変更する。**
  - SmartyのFilter機能をフックする事で実現
  - SC_Helper_Transformというインターフェイスで簡単に変更可能

2-2　処理の介入

●処理の介入

EC-CUBE本体処理へ介入して処理・結果を書き換える。

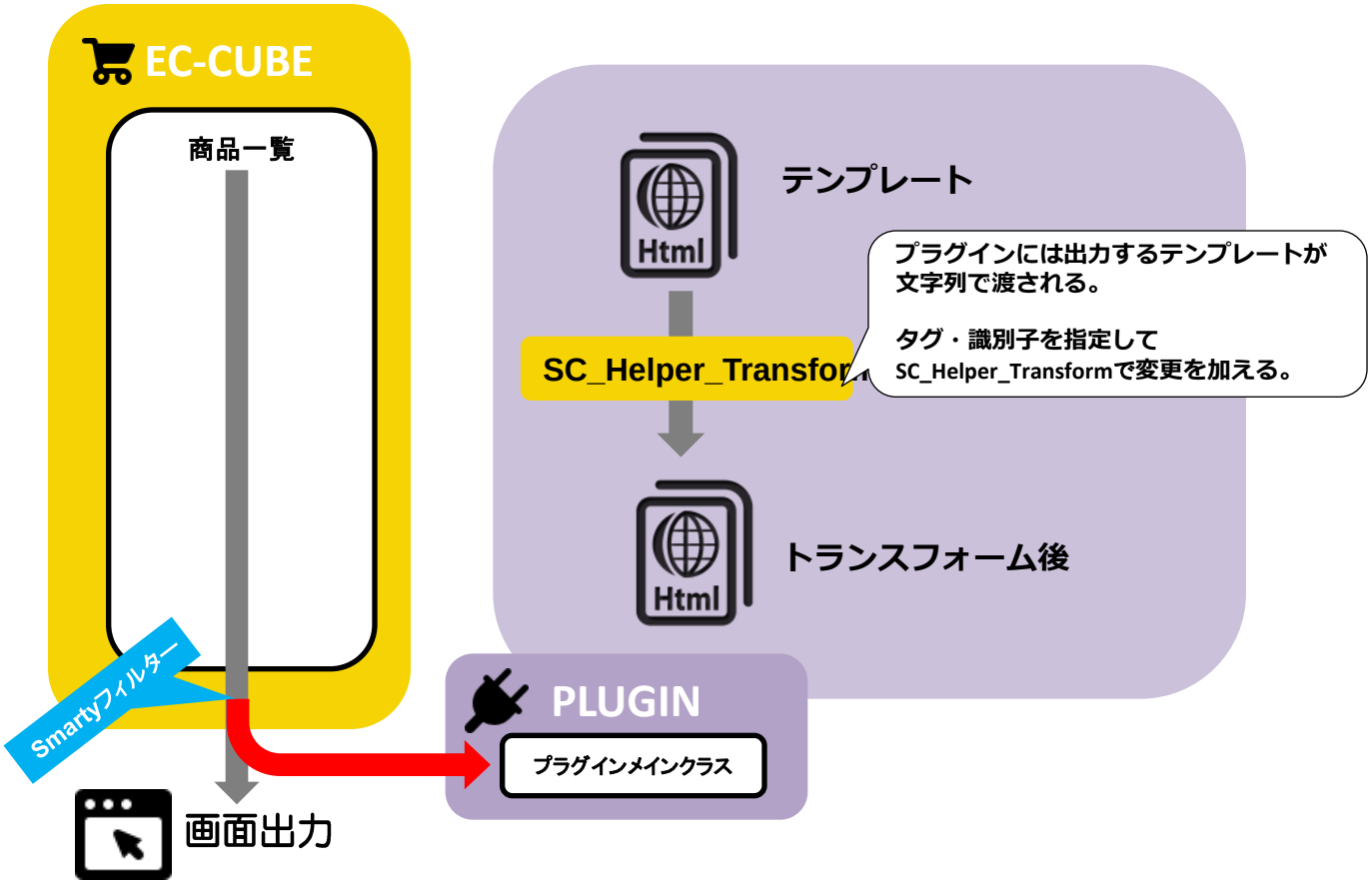
・EC-CUBE本体処理にプラグインで介入する事が出来ます。
・**フックポイント（介入箇所）** と、**フックポイント**通過時に実行する関数をプラグインに宣言する事で
本体処理の**フックポイント**通過時、宣言した関数が実行されます。



2-3　テンプレートの変更

●テンプレートの変更
　　EC-CUBE本体テンプレートを変更する。

・EC-CUBE本体テンプレートに変更を加える事が出来ます。
・Smartyがコンパイルファイルを作る処理にフックします。
　この時、プラグイン側にはテンプレートのソースが文字列で渡されます。
　テンプレートに対してSC_Helper_Transformを使い変更を加えます。



2-4　トランスフォームの使い方

●テンプレートの変更
EC-CUBE本体テンプレートを変更する。

・テンプレートの変更はSC_Helper_Transformを使う事で、簡単に変更する事が出来ます。
SC_Helper_Transformで変更するソースを読み込み、対象とする要素を絞り込んで、変更処理を加えます。

Smartyテンプレート

```
<div id="products-category-right">
    <div class="now_dir">
        <input type="text" name="category_name" value" />
        <a ><span class="btn-next">登録</span></a>
    </div>

    <h2><!--{* jQuery で挿入される *}--></h2>
    <table class="list" id="categoryTable">
        <col width="5%" />
        <col width="60%" />
        <col width="10%" />
        <tr class="nodrop nodrag">
            <th>ID</th>
            <th>カテゴリ名</th>
            <th class="edit">編集</th>
        </tr>
        <!--{section name=cnt loop=$arrList}-->
        <tr >
            <td class="center"><!--{$arrList[cnt].category_id}--></td>
            <td><a href="?"/></td>
            <td class="center">編集中</td>
        </tr>
        <!--{/section}-->
    </table>
</div>
```

この部分変更したい

タグと識別子をjQuaryの指定方式に沿って表記

左のテンプレートのdivタブ内を全て変更したい場合は以下の様に指定します。

```
// カテゴリ登録画面
$objTransform = new SC_Helper_Transform( Smartyテンプレートの文字列 );

$objTransform->select('div.now_dir')->replaceElement(★置換後の文字列));
```

プラグイン側で用意したファイルを
file_get_contentsなで呼び出す

select()でテンプレートの変更したい場所を指定し、下記の関数で操作します。

insertBefore	要素の前にHTMLを挿入
insertAfter	要素の後にHTMLを挿入
appendFirst	要素の先頭にHTMLを挿入
appendChild	要素の末尾にHTMLを挿入
replaceElement	要素を指定したHTMLに置換
removeElement	要素を削除する

2-5　プラグインのライセンス

プラグインのライセンスは基本的に**自由**です。
（ただし、プラグインの作成方法によってはEC-CUBE本体のライセンスに抵触する場合（その場合、強制的にGPLライセンスになります）がありますので、基本ルールにのっとり作成してください。**参照：3-5 ライセンスの指定方法**）

また、EC-CUBEの商用ライセンスに矛盾するライセンス形態をとった場合、商用ライセンスご購入サイトにプラグインが導入できなくなりますので、注意が必要です。

以下の点をご参照いただき、作成者の判断にてプラグインのライセンスを選択してください。

①**推奨**：プラグインを無料で配布する場合（EC-CUBEオフィシャルサイトで配布する場合）

プラグインのライセンスは商用ライセンスに矛盾しない**LGPLライセンス** もしくは、**LGPLライセンスに矛盾しないライセンス（BSDライセンス、MITライセンス等）**を推奨します。
参照：LGPLについて（ http://ja.wikipedia.org/wiki/GNU_Lesser_General_Public_License ）

※本マニュアルでは例として、LGPLライセンスでの作成方法を記載します。**（参照：3-5　ライセンスの指定方法）**

② プラグインを有料配布する等、再配布不可なプラグインにしたい場合

プラグインのライセンスは**自由に**決定することが可能です。
ただし、商用ライセンスに矛盾するライセンス形態にした場合、商用ライセンスご購入サイトにて使用することができませんので、ご注意いただき、作成者の判断にて作成してください。

③プラグインのライセンスを指定しない場合

ライセンスを指定しない場合、プラグインのライセンスは**EC-CUBE本体のライセンスを継承し自動的にGPLライセンス**になります。
プラグインのライセンスがGPLライセンスの場合、**商用ライセンスと矛盾**してしまうため、商用ライセンスご購入サイトにプラグインを導入することができなくなります。
（EC-CUBEをGPLライセンスのまま使用される場合はプラグインがGPLライセンスでも問題なく導入できます。）

3. プラグインの作り方

3-1　作り方概要

① 命名規則に沿ってプラグイン名・ファイル名を決定

・ルール・機能に則したプラグイン名・プラグインコード名の徹底。
・命名規則については後述。

http://svn.ec-cube.net/open_trac/wiki/EC-CUBE%E6%A8%99%E6%BA%96%E8%A6%8F%E7%B4%84

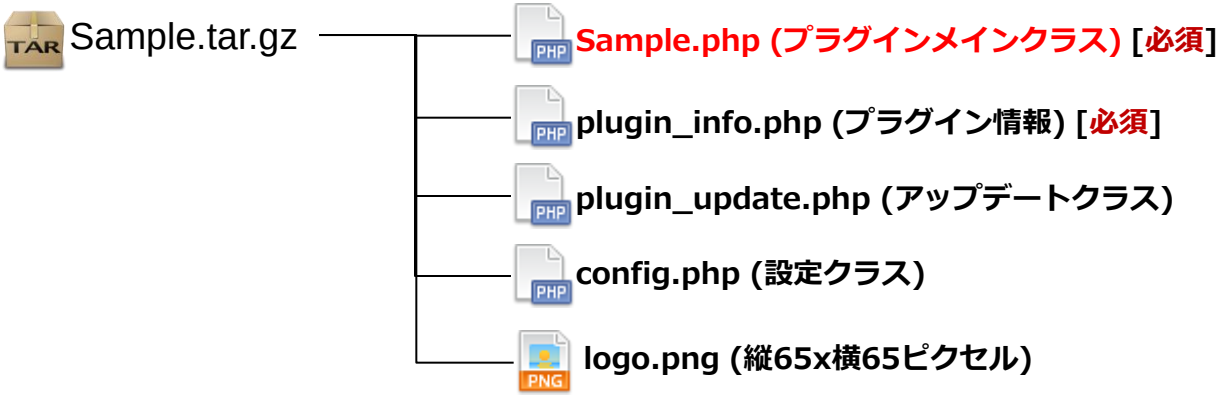
② 必須となる関数・ファイルに注意してプラグインを作成

・plugin_infoにプラグインに関連情報を定義
　　→インストール時にDBに登録されます。
・フックポイントで処理に介入　（フックポイント一覧は別紙参照）
・トランスフォームでテンプレートの変更（使用方法は後述）

③ 必要なファイルを用意して圧縮する。

・必要なファイルが用意できたらtar.gzに圧縮

3-2-1　プラグインファイル構成

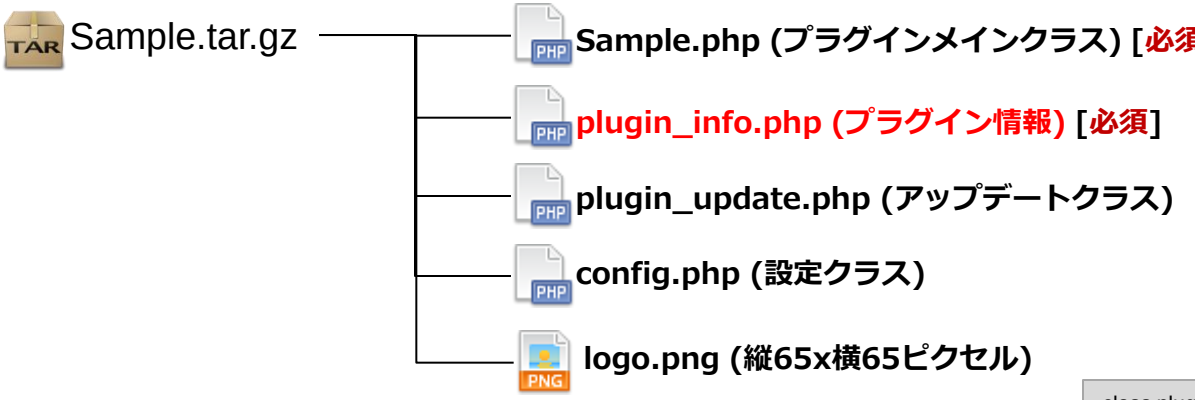


プラグインメインクラス宣言関数一覧

プラグインメインクラスには下記の関数を定義する事で、ケースに応じて実行されます。

install	必須	installはプラグインのインストール時に実行されます。 param ; プラグイン情報(dtb_plugin)の連想配列
uninstall	必須	uninstallはアンインストール時に実行されます。 param ; プラグイン情報(dtb_plugin)の連想配列
enable		enableはプラグインを有効にした際に実行されます。 param ; プラグイン情報(dtb_plugin)の連想配列
disable		disableはプラグインを無効にした際に実行されます。また、アンインストール時にプラグインが有効な場合、uninstallの前に実行されます。 param ; プラグイン情報(dtb_plugin)の連想配列
register		registerはプラグインインスタンス生成時に実行されます。フックポイントの登録はここで行います。 param ; SC_Helper_Pluginのインスタンス
preProcess		preProcessはスーパーフックポイントを使って実行されます。各Pageクラスのinit処理で実行されます。 param ; 呼び出し元のLC_Pageオブジェクト
process		preProcessはスーパーフックポイントを使って実行されます。各PageクラスのsendResponse処理で実行されます。 param ; LC_Pageオブジェクト

3-2-2　プラグインファイル構成

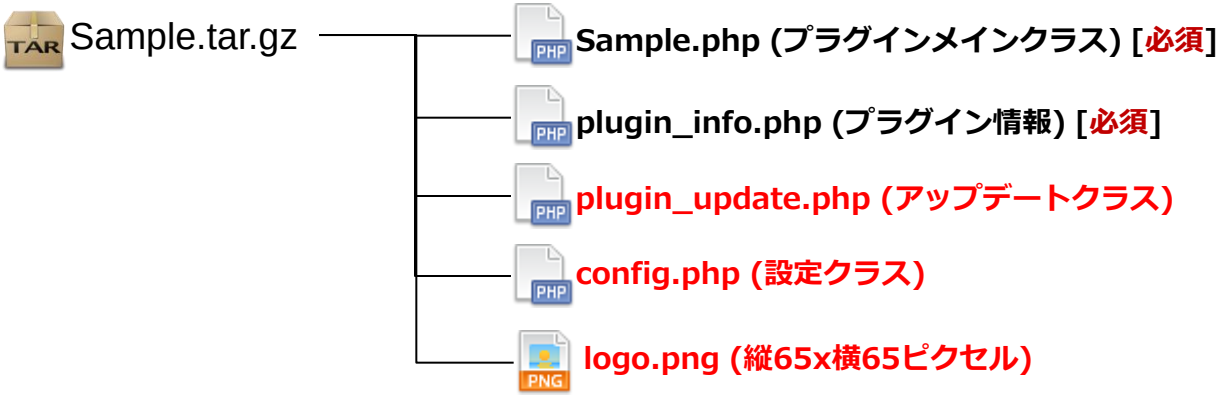



```
class plugin_info {
    static $PLUGIN_CODE       = "SampleClassHook";
    static $PLUGIN_NAME       = "SCクラス フックサンプル";
    static $PLUGIN_VERSION    = "0.1";
    static $COMPLIANT_VERSION = "2.12.0";
    static $AUTHOR            = “株式会社ロックオン";
    static $DESCRIPTION       = "SC系クラスをフックするサンプルです。";
    static $PLUGIN_SITE_URL   = "http://www.ec-cube.net/";
    static $AUTHOR_SITE_URL   = "http:// www.ec-cube.net /";
    static $CLASS_NAME        = "SampleClassHook";
    static $HOOK_POINTS       = array(
        array("LC_Page_Admin_Products_Category_action_after",
'setLogoParameter'),
        array("LC_Page_Products_List_action_after", 'replaceLogo'));
    static $LICENSE       = “LGPL";
}
```

plugin_info.php宣言定数一覧

PLUGIN_CODE	必須	システム上でのキーとなります。 プラグインコードは一意である必要があります。
PLUGIN_NAME	必須	プラグイン名。 プラグイン管理・画面出力（エラーメッセージetc）にはこの値が出力されます。
CLASS_NAME	必須	プラグインメインクラス名。 本体がプラグインを実行する際に呼ばれるクラス。拡張子は不要です。
PLUGIN_VERSION	必須	プラグインバージョン
COMPLIANT_VERSION	必須	本体対応バージョン
AUTHOR	必須	作者
DESCRIPTION	必須	説明文
PLUGIN_SITE_URL		プラグイン用のサイトURL設定されている場合はプラグイン管理画面のプラグイン名がリンクになります。
AUTHOR_SITE_URL		作者用のサイトURL。 設定されている場合はプラグイン管理画面の作者名がリンクになります。
HOOK_POINTS		使用するフックポイント・コールバック関数。 使用するフックポイントを設定すると、フックポイントが競合した際にアラートが出ます。 ここで宣言することで、インストール時にdtb_plugin_hookpointsに登録され、register関数を書かずにフックポイントでの介入が可能です。
LICENSE		プラグインのライセンスを指定します。（例：LGPL） ライセンスを指定しない場合、自動的にGPLライセンスになりますのでご注意ください。（参照：2-5　プラグインのライセンス）

3-2-3　プラグインファイル構成



plugin_update.php

プラグインをアップデートする場合はplugin_update.php::update()が実行されます。
以下の様に定義します。

```
class plugin_update{
   /**
     * アップデート
     * updateはアップデート時に実行されます.
     * 引数にはdtb_pluginのプラグイン情報が渡されます.
     *
     * @param array $arrPlugin プラグイン情報の連想配列(dtb_plugin)
     * @return void
     */
    function update($arrPlugin) {
        // nop
    }
}
```

config.php

config.php はプラグインに設定画面が必要な場合に作成します。
プラグイン管理画面はプラグインディレクトリにconfig.phpがある場合、「設定」リンクを表示します。
設定リンクを押下するとconfig.phpが実行されます。

logo.png

プラグインにロゴがある場合はlogo.pngをインストール時にhtml/plugin/プラグインディレクトリ以下に保存して下さい。

3-3　アーカイブ作成方法

プラグインファイルのアーカイブ作成方法

・プラグインの圧縮形式は tar.gz を推奨とします。（tar形式でもインストールは可能です。）

＊圧縮する際に **フォルダごと** 圧縮しない様にご注意下さい！！


```
## PluginDir というプラグインを開発した場合
$ ls -al PluginDir
drwxr-xr-x   9 user staff    306  5 17 16:00 .
drwx------+ 39 user  staff   1326  5 23 19:33 ..
-rw-r--r--   1 user  staff   8955  5 18 00:32 PluginDir.php
-rw-r--r--   1 user  staff   3359  5 18 00:03 logo.png
-rw-r--r--   1 user  staff    618  5 18 00:03 plugin_info.php
-rw-r--r--   1 user  staff   1280  5 18 00:03 plugin_update.php

## プラグイン内のディレクトリへ移動し, tar コマンドでアーカイブ作成
$ cd PluginDir
$ tar cvzf ../PluginDir.tar.gz *

## ディレクトリを含めない形で作成可能
$ tar tzvf PluginDir.tar.gz
-rw-r--r--  0 user staff    8955  5 18 00:32 PluginDir.php
-rw-r--r--  0 user staff    3359  5 18 00:03 logo.png
-rw-r--r--  0 user staff     618  5 18 00:03 plugin_info.php
-rw-r--r--  0 user staff    1280  5 18 00:03 plugin_update.php
```

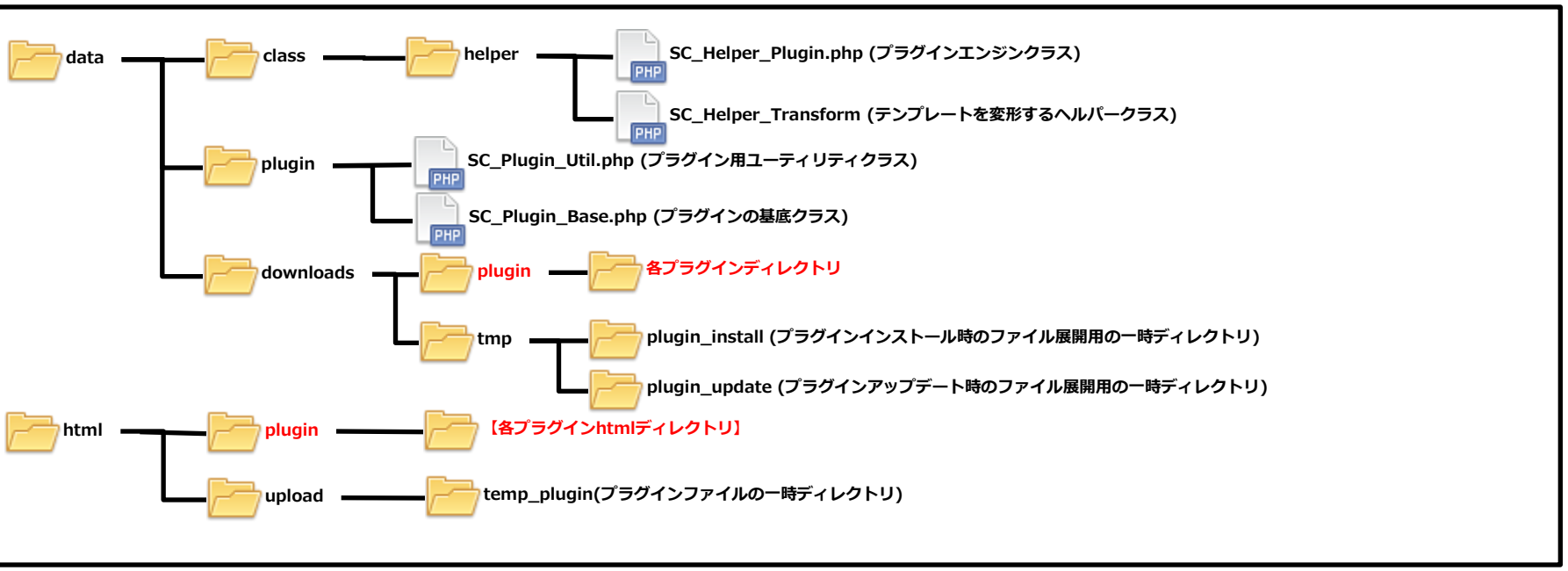
3-3　アーカイブ作成方法

Windows環境でのアーカイブ作成方法

圧縮ツール：Lhaplus 1.58

フォルダ内のファイルを選択し、圧縮する

3-4　本体ディレクトリ構成



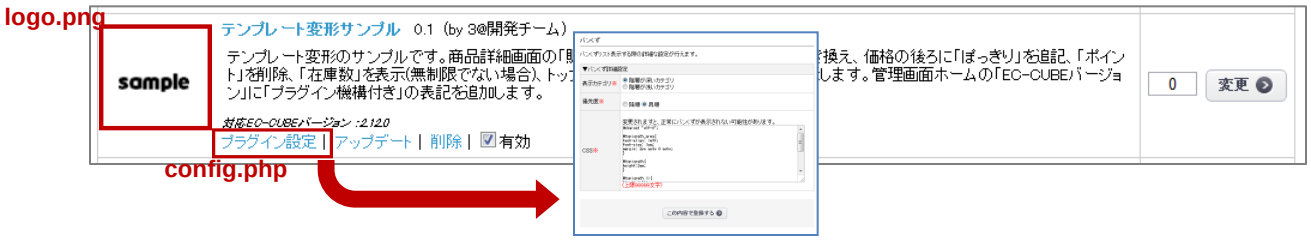
プラグインインストール後のディレクトリ

全てdata/downloads/plugin以下にプラグインコード名のディレクトリを作成し、それ以下に展開されます。

html以下のプラグイン用ディレクトリ

インストール時にhtml/plugin/以下にも同様にプラグインコード名でディレクトリが作成されます。
このディレクトリはjs、cssなどプラグインの必要に応じて使用します。

- logo.pngをこのディレクトリに配置する事でプラグイン管理画面でロゴが自動的に表示されます。
- config.phpをこのディレクトリに配置する事でプラグイン管理画面で設定のリンクがアクティブになります。



3-5　命名規約

フックポイント

フックポイントを追加する際は、以下の命名規約に沿って設置する事。

クラス名_関数名_場所

Ex)フックポイント名
LC_Page_Guide_action_begin

プラグインコード　(プラグインコードについてはプラグインファイル構成　を参照)

プラグインコードはキャメルケースでの命名とする。

Ex)プラグインコード
SamplePlugin

ファイル

・ファイル名の先頭には, プラグイン用識別子(**plg**)・プラグインコード(***PLUGIN_CODE***)を付与する事。
・追加するファイル名は、原則**EC-CUBE標準規約**に従う事。

Ex)
plg_*SamplePlugin*_hoge.tpl
plg_*SamplePlugin*_hoge.php
plg_*SamplePlugin*_LC_Page_Hoge.php
plg_*SamplePlugin*_hoge.css
plg_*SamplePlugin*_hoge.js
…etc

データ

テーブル名 ・ カラム名 ・ メンバ変数といった本体にデータを生成する際はファイル名と同様にプラグインコードから開始する事とする。

Ex) テーブルを追加する場合

```
CREATE TABLE plg_SamplePlugin_holiday (
    hoge_id int NOT NULL,
    hoge_title text NOT NULL,
    create_date timestamp NOT NULL DEFAULT
CURRENT_TIMESTAMP,
    update_date timestamp NOT NULL,
    PRIMARY KEY (holiday_id)
);
```

Ex) 本体側のインスタンス変数をセットする場合

```
$objPage-> plg_SamplePlugin_hoge="test";
```

3-6-1　ライセンスの指定方法

前提条件
（参照：3-3　本体ディレクトリ構成　）

ライセンスの指定をする**前提**として、プラグインが以下の条件を満たす形で作成してください。

以下条件にのっとらない場合はプラグインのライセンスが**自動的にGPLライセンス**になります。
プラグインがGPLライセンスになった場合は、商用ライセンス購入サイトにてプラグインが適用できなくなりますのでご注意ください。
（EC-CUBE本体を商用ライセンスではなく、GPLライセンスとしてご使用する場合は問題なくご使用いただけます。）

プラグインにてLGPLライセンスやその他GPLライセンス以外のライセンスを適用できる条件

以下2点の条件のものはGPLライセンスを適用する必要はなく、自由にライセンスの設定が可能です。

・data/downloads/plugin/内に配置されたもの
・data/downloads/plugin/内に配置されたプログラムがコピー、生成、修正したもの

ただし適用除外プラグイン又は適用除外プラグインと連携又は関連して動作するプログラムがEC-CUBEのソースコード・バイナリコードを改変（変更・上書き・削除を含む。以下同じ。）しないこと及び改変したものを包含・コピー・生成・転送しないことを条件とします。

3-6-2　ライセンスの指定方法

本項では、例として、LGPLライセンスをプラグインに適応する方法（**推奨**）を記載します。
ご参照いただき、その他ライセンスを指定する場合は独自に指定いただきますようお願いいたします。

①plugin_info.php (プラグイン情報)　の宣言にてライセンスを指定します。

（参照：3-2-2　プラグインファイル構成　）

以下、LGPLライセンスの例

```
class plugin_info {
    static $PLUGIN_CODE       = "SampleClassHook";
    static $PLUGIN_NAME       = "SCクラス フックサンプル";
    static $PLUGIN_VERSION    = "0.1";
    static $COMPLIANT_VERSION = "2.12.0";
    static $AUTHOR            = “株式会社ロックオン";
    static $DESCRIPTION       = "SC系クラスをフックするサンプルです。";
    static $PLUGIN_SITE_URL   = "http://www.ec-cube.net/";
    static $AUTHOR_SITE_URL   = "http:// www.ec-cube.net /";
    static $CLASS_NAME        = "SampleClassHook";
    static $HOOK_POINTS       = array(
        array("LC_Page_Admin_Products_Category_action_after", 'setLogoParameter'),
        array("LC_Page_Products_List_action_after", 'replaceLogo'));
    static $LICENSE       = "LGPL";
}
```

3-6-3　ライセンスの指定方法

②各、PHPファイルのヘッダ部分にてライセンスを表記します。

以下、LGPLライセンスの場合

```
<?php
/*
 * プラグインの名前、もしくは何をするのかについての簡単な説明（必須）
 * Copyright (C) 作成年（必須）、プラグイン作者名（必須）
 * 問合せ先email or URL（任意）
 *
 * This library is free software; you can redistribute it and/or
 * modify it under the terms of the GNU Lesser General Public
 * License as published by the Free Software Foundation; either
 * version 2.1 of the License, or (at your option) any later version.
 *
 * This library is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
 * Lesser General Public License for more details.
 *
 * You should have received a copy of the GNU Lesser General Public
 * License along with this library; if not, write to the Free Software
 * Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA  02111-1307  USA
 */
```

詳細なソースをご覧になりたい場合は、サンプルプラグインも併せてご参照ください。

・カテゴリコンテンツ
http://svn.ec-cube.net/open_trac/attachment/ticket/1692/CategoryContents.tar.gz

・パンくず
http://svn.ec-cube.net/open_trac/attachment/ticket/1692/TopicPath.tar.gz

3-6-4　ライセンスの指定方法

③各、テンプレートファイルのヘッダ部分にてライセンスを表記します。

以下、LGPLライセンスの場合

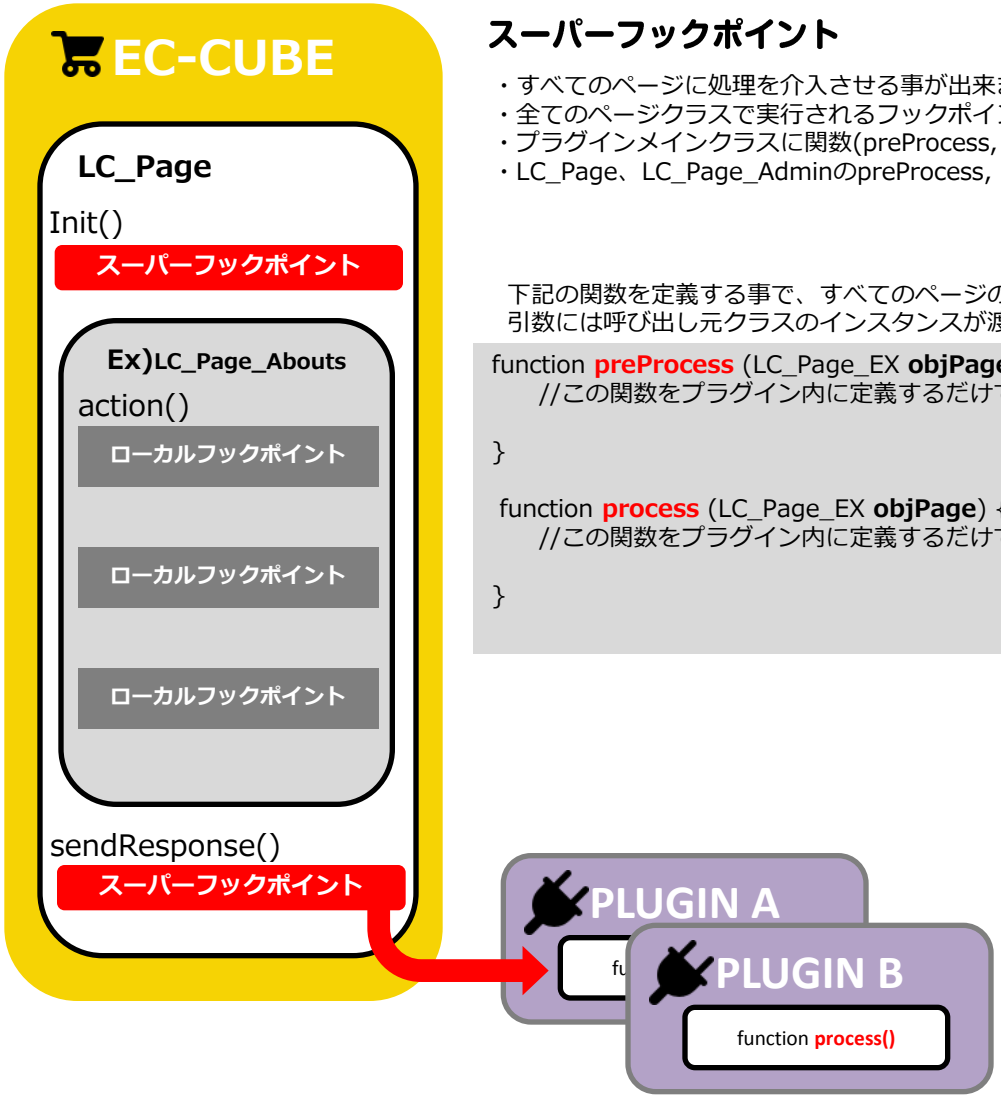
```
<!--{*
 * プラグインの名前、もしくは何をするのかについての簡単な説明（必須）
 * Copyright (C) 作成年（必須）、プラグイン作者名（必須）
 * 問合せ先email or URL（任意）
 *
 * This library is free software; you can redistribute it and/or
 * modify it under the terms of the GNU Lesser General Public
 * License as published by the Free Software Foundation; either
 * version 2.1 of the License, or (at your option) any later version.
 *
 * This library is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
 * Lesser General Public License for more details.
 *
 * You should have received a copy of the GNU Lesser General Public
 * License along with this library; if not, write to the Free Software
 * Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA  02111-1307  USA
 *}-->
```

詳細なソースをご覧になりたい場合は、サンプルプラグインも併せてご参照ください。

・カテゴリコンテンツ
http://svn.ec-cube.net/open_trac/attachment/ticket/1692/CategoryContents.tar.gz

・パンくず
http://svn.ec-cube.net/open_trac/attachment/ticket/1692/TopicPath.tar.gz

3-7　スーパーフックポイント



スーパーフックポイント

・すべてのページに処理を介入させる事が出来ます。
・全てのページクラスで実行されるフックポイントです。
・プラグインメインクラスに関数(preProcess, process)を定義する事で実行されます。
・LC_Page、LC_Page_AdminのpreProcess, process通過時に実行されます。

下記の関数を定義する事で、すべてのページの最初と最後にそれぞれプラグインに処理が渡ります。
引数には呼び出し元クラスのインスタンスが渡されます。

```
function preProcess (LC_Page_EX objPage) {
    //この関数をプラグイン内に定義するだけで実行されます。

}

 function process (LC_Page_EX objPage) {
    //この関数をプラグイン内に定義するだけで実行されます。

}
```

SamplePlugin.php

3-7　スーパーフックポイント

スーパーフックポイントをフックする

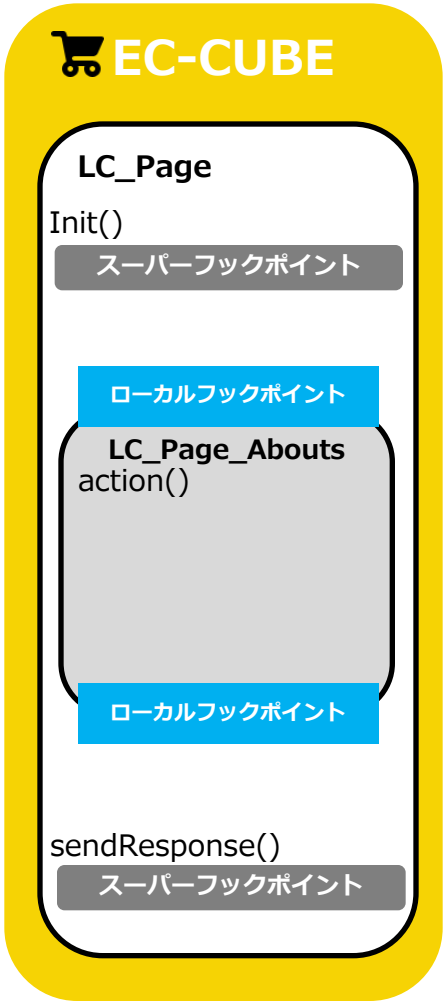
・全ページで実行したい処理はスーパーフックポイントを使います。
・スーパーフックポイントは命名規則に沿った関数を定義にするだけで実行されます。

フックポイント名	LC_Page_preProcess・LC_Page_process（フックポイントを使う必要はありません）	
パラメータ	LC_Page_EX	フックしたPageクラスのインスタンス

・スーパーフックポイントはLC_Page, LC_Page_Adminによって呼び出されます。
・全ページクラスのactionが実行される前(init)・実行された後(sendResponse)に実行されます。
　＊ブロッククラスでは実行されません

```
function preProcess (LC_Page_EX objPage) {
    //この関数をプラグイン内に定義するだけで実行されます。

}
-----------------------------------------------------------------------------------------------------------
function process (LC_Page_EX objPage) {
    //この関数をプラグイン内に定義するだけで実行されます。

}
```

3-8　ローカルフックポイント



ローカルフックポイント(指定方法)

・特定のページに処理を介入させる事が出来ます。
・特定ページの特定箇所で呼び出されるフックポイントです。
・ローカルフックポイントの定義方法は２つあります。

①plugin_info.phpに定義する。

1-1： plugin_info.php :static $HOOK_POINTSを宣言し、フックポイントとコールバック関数を定義します。

```
class plugin_info {
                ～省略～
static $HOOK_POINTS       = array(
      array(" LC_Page_Admin_Products_category_action_end", 'contents_set '),
}
```

ローカルフックポイントを指定　　コールバック関数を指定

```
--------------------------------------------------------------------------------
// フックポイント通過時に実行されるコールバック関数
function contents_set (LC_Page_EX objPage) {
}
```

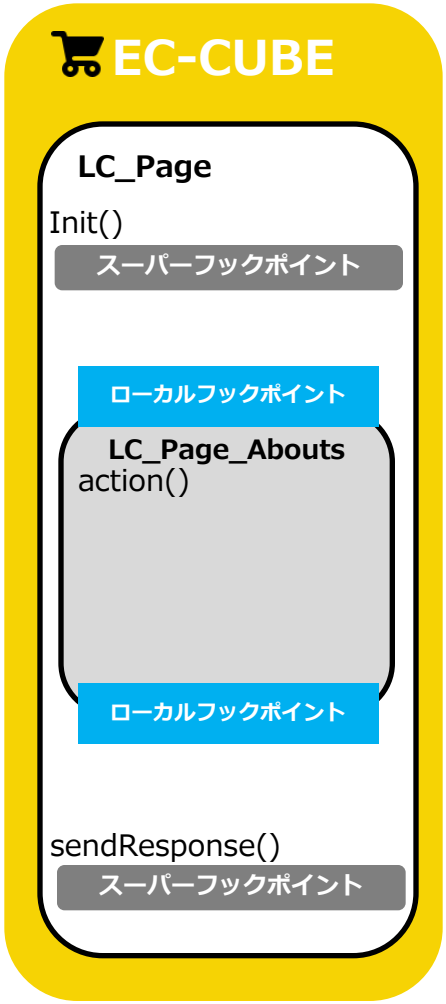
1-2：インストール時に定義に従ってdtb_plugin_hookpointに登録されます。

1-3： フックポイント通過時にDBに登録しているフックポイントに応じてコールバック関数が実行されます

＊ plugin_info.phpに定義したフックポイント以外に追加したい場合はregister関数を作成します。
その場合、以下の様に明示的に呼び出す必要があります。(registerについては後述)

```
function register($objHelperPlugin, $priority)) {
      parent::register($objHelperPlugin, $priority);
      $objHelperPlugin->addAction('LC_Page_Admin_Products_action_after', array($this, 'contents_set'));
}
```

3-8　ローカルフックポイント



ローカルフックポイント(指定方法)

・特定のページに処理を介入させる事が出来ます。
・特定ページの特定箇所で呼び出されるフックポイントです。
・ローカルフックポイントの定義方法は２つあります。

② SC_Helper_Pluginの関数addActionでセットする。

1-1：プラグインメインクラスにregister()関数を定義し、フックポイントとコールバック関数をセットします。

1-2： フックポイント通過時に指定のコールバック関数が実行されます。

```
function register(SC_Helper_Plugin $objHelperPlugin) {
$objHelperPlugin->addAction('prefilterTransform', array(&$this, 'prefilterTransform'));
}

// フックポイント通過時に実行されるコールバック関数
function prefilterTransform (LC_Page_EX objPage) {
}
```

ローカルフックポイントを指定

コールバック関数を指定

3-8　ローカルフックポイント

ローカルフックポイントをフックする

・各ページ処理にプラグインで処理を介入する場合はローカルフックポイントを使います。
・フックポイント名は実行されるページ・タイミングによって自動的に生成されます。

フックポイント名	クラス名_アクション_モード	
パラメータ	LC_Page_EX	フックしたPageクラスのインスタンス

・ローカルフックポイントはLC_Page, SC_Responseによって動的に生成されます。
・各ページクラスのaction()関数実行前・実行後されます。

クラス名　　**タイミング**
LC_Page_Admin_Contents_action_begin (or after)

・action()関数途中にSC_Response_Ex::actionExit()やSC_Response_Ex::reload()で処理を抜ける場合は
以下のルールでフックポイントが生成されます。

クラス名　　**モード名**
LC_Page_Admin_Contents_action_delete

```
static $HOOK_POINTS      = array(
      array("LC_Page_Admin_Products_category_action_end", 'contents_set'));
-----------------------------------------------------------------------------------------------------------
// フックポイント通過時に実行されるコールバック関数
function contents_set (LC_Page_EX objPage) {
}
```

3-9　SC_FormParamのフック

SC_FormParamをフックする

・プラグインでフォームに項目追加した場合、エラーチェックを入れたい場合があります。
　SC_FormParamにフックする事でパラメータの追加が簡単に出来るので、プラグイン側で入力バリデータを作成する必要がありません。

フックポイント名	SC_FormParam_construct	
パラメータ	String	SC_FormParam呼び出し元のクラス名
	SC_FormParam	SC_FormParamのインスタンス

・SC_FormParamのSC_FormParam ::__construct()にフックする事が出来ます。
・SC_FormParamのコンストラクタ生成時にフック出来るので$objFormParamへのパラメータ追加が容易になります。

```
static $HOOK_POINTS      = array(
      array(" SC_FormParam_construct ", 'addParam'));
-----------------------------------------------------------------------------------------------------------
function addParam($calss_name, $param) {
      if (strpos($calss_name, 'LC_Page_Entry') !== false) {
            $param->addParam('追加パラメータ', 'plg_param', INT_LEN, 'n', array('EXIST_CHECK', 'NUM_COUNT_CHECK', 'NUM_CHECK'));
      }
   }
```

3-10　SC_系クラスのフック

SC_系のクラスをフックする

・SC_系のクラスをプラグイン側で置き換える事が出来ます。
SC_系クラスをプラグイン側で自由にカスタマイズしたい場合はこのフックポイントを使用します。
＊他プラグインと競合する可能が高くなり、正常に動作しなくなる可能性があります。

フックポイント名	loadClassFileChange	
パラメータ	String	読み込む事を要求されたクラスの名前
	String	本来読み込む予定であるクラスファイルのパス
・PHPのオートロード機能に介入し、ロードするクラスを別クラスに変更する事が出来ます。		

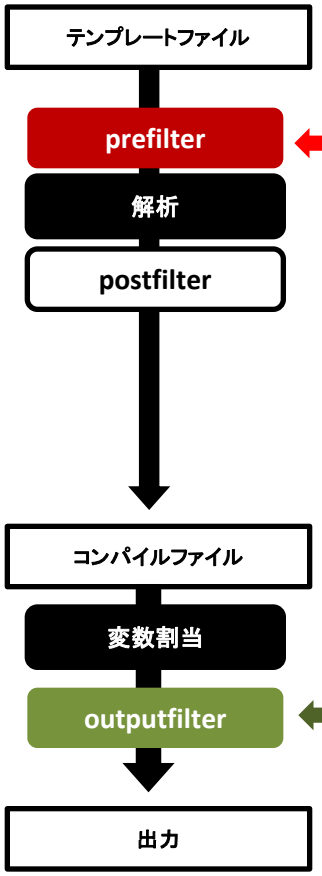
```
static $HOOK_POINTS      = array(
      array("loadClassFileChange", 'loadClassFileChange'));
-----------------------------------------------------------------------------------------------------------
function loadClassFileChange (&$classname, &$classpath) {
      if($classname == 'SC_Customer_Ex') { // 変えたいクラス名でフィルタ,*_Exにフィルタ推奨
         // 代替読み込みされるクラスファイルを用意
         $classpath = PLUGIN_UPLOAD_REALDIR . "CategoryContents/SC_MyCustomer.php";

         // 上で指定した代替読み込みされるファイル内のクラス名が、本来の読み込み先と違うクラス名の場合、$classname を変更するクラス名にする。
         $classname = 'SC_MyCustomer';
      }
   }
```

*注意
このコールバック関数では、特別な処理がされますので注意が必要です。
(1)$classname が本来読み込むクラスと別の名前を設定して関数が終わった場合
　→　自動的に、本来読み込む予定だったSC_XXXX_Exのextends を新しく指定された$classname に書き換えます。これにより、下記のようなextends関係になります。
SC_XXXX_Ex -> SC_XXXX の関係から、 SC_XXXX_Ex -> $classname (-> SC_XXXX) と変える。

$classname で新しく指定されたクラスは、_ExではないSC_をextendsしていることが推奨される。

(2)$classname が本来読み込むクラスと同名の場合
新しく指定された$classpath のみが読み込まれます。
_Exは無視されますが、_Ex無視はカスタマイズをする人から分かりにくくなるため(1)の利用方法が望ましいです。

3-11　テンプレートの変更



テンプレートファイルの変更

・テンプレートの変更はSmartyのfilter機能を使います。
・ prefilterはコンパイルファイル作成時にのみ実行されます。
（プラグインインストール時に全コンパイルファイルを削除します）

prefilter

・コンパイル時のみ呼ばれます（template_c/以下にコンパイルファイルが無い場合）
・使用するにはフックポイント同様にregister関数に定義します。

```
$objHelperPlugin->addAction('prefilterTransform', array(&$this, 'prefilterTransform'));
```

prefilter　　コールバック関数

・コールバック関数を定義します。
定義されたコールバック関数には以下のパラメータが渡されます。
$source / (string)テンプレートソース
$objPage / (LC_Page_Ex)呼び出し元のPageオブジェクト
$filename / (string)テンプレートのファイル名

```
function prefilterTransform(&$source, LC_Page_Ex  $objPage, $filename) {
    テンプレートソースの変更はSC_Helper_Transformを使用します。（後述）
}
```

outputfilter

・テンプレート出力時に毎回呼ばれます
・使用方法はprefilterと同様です。

```
$objHelperPlugin->addAction('outputfilterTransform', array(&$this, 'outputfilterTransform'));
```

outputfilter　　コールバック関数

・コールバック関数を定義します。

```
function outputfilterTransform(&$source, LC_Page_Ex $objPage, $filename) {
}
```

3-11　テンプレートの変更

テンプレートをフックする(prefilterTransform)

・テンプレートの変更も他フックポイントと同様にフックポイントでフックし、プラグイン側でデータを加工します。

フックポイント名	prefilterTransform	
パラメータ	string	テンプレートソース
	LC_Page_Ex	呼び出し元のPageオブジェクト
	string	テンプレートのファイル名

・テンプレートのコンパイル実行時に呼び出されます。
　＊一度テンプレートのコンパイルファイルが生成されると、次回からは呼び出されません。
・第一引数で渡るテンプレートソースに対して変更を加えます。

```
static $HOOK_POINTS      = array(
      array("prefilterTransform", 'prefilterTransform'));
-----------------------------------------------------------------------------------------------------------------
function prefilterTransform(&$source, LC_Page_Ex $objPage, $filename) {
      $objTransform = new SC_Helper_Transform($source);
      $template_dir = PLUGIN_UPLOAD_REALDIR . 'CategoryContents/templates/';
      switch($objPage->arrPageLayout['device_type_id']){
         case DEVICE_TYPE_MOBILE:
         case DEVICE_TYPE_SMARTPHONE:
         case DEVICE_TYPE_PC:
            // 商品一覧画面
            if (strpos($filename, 'products/list.tpl') !== false) {
               $objTransform->select('h2.title')->insertBefore(file_get_contents($template_dir . 'categorycontents_products_list_add.tpl'));
            }
            break;
         case DEVICE_TYPE_ADMIN:
         default:
            // カテゴリ登録画面
            if (strpos($filename, 'products/category.tpl') !== false) {
               $objTransform->select('div.now_dir')->replaceElement(file_get_contents($template_dir . 'categorycontents_admin_basis_category_add.tpl'));
            }
            break;
      }
      $source = $objTransform->getHTML();
   }
```

3-11　テンプレートの変更

テンプレートをフックする(outputfilter_transform)

・テンプレートの変更も他フックポイントと同様にフックポイントでフックし、プラグイン側でデータを加工します。

フックポイント名	outputfilter_transform	
パラメータ	string	テンプレートソース
	LC_Page_Ex	呼び出し元のPageオブジェクト
	string	テンプレートのファイル名

・コンパイルファイルからHTML形式に変換後、呼び出されます。
　＊既にHTMLの形になっているのでSmartyタグを挿入する事は出来ません。
　＊prefilterTransformと違い、毎回実行される処理となるので使い方によってはパフォーマンスが低下します。
・第一引数で渡るテンプレートソースに対して変更を加えます。

```
static $HOOK_POINTS      = array(
      array("prefilterTransform", 'prefilterTransform'));
-----------------------------------------------------------------------------------------------------------
function outputfilterTransform(&$source, LC_Page_Ex $objPage){
   $objTransform = new SC_Helper_Transform($source);
   switch($objPage->arrPageLayout['device_type_id']){
      case DEVICE_TYPE_MOBILE:
         break;
      case DEVICE_TYPE_SMARTPHONE:
         break;
      case DEVICE_TYPE_PC:
         // 商品詳細画面
         if (strpos($objPage->tpl_mainpage, 'products/detail.tpl') !== false) {
            $objTransform->select('#header_wrap')->insertBefore(
               '<div><p>このページは' . date('Y年n月j日 H:i:s') . 'に表示更新されました。</p></div>'
            );
         }
         break;
      case DEVICE_TYPE_ADMIN:
      default:
         break;
   }

   $source = $objTransform->getHTML();
}
```

3-11　テンプレートの変更

テンプレートをフックする

*注意
テンプレートの変更には、簡単にソースを加工出来るSC_Helper_Transformを推奨としますが、以下のケースにはSC_Helper_Transformが対応していません。
・メールテンプレート、HTMLの書式に沿っていないテンプレート（ popup_footer .tplなど）

テンプレートは文字列ですのでSC_Helper_Transformを使用することなく、好きに加工する事は可能です。
以下、 SC_Helper_Transformを使わない例

```
public function prefilterTransform(&$source, LC_Page_Ex $objPage, $filename) {
    $template_dir = PLUGIN_UPLOAD_REALDIR . $this->arrSelfInfo['plugin_code'] . '/templates/';
    switch($objPage->arrPageLayout['device_type_id']){
    case DEVICE_TYPE_MOBILE:
    case DEVICE_TYPE_SMARTPHONE:
    case DEVICE_TYPE_PC:
            // site_main.tplへタグを差し込む
            if (strpos($filename, 'site_main.tpl') !== false) {
                // タグテンプレートを取得
                $cm_tag = file_get_contents($template_dir . 'cm_tag.tpl');
                // タグテンプレートを取得
                $cv_tag = file_get_contents($template_dir . 'cv_tag.tpl');
                // bodyタグ直下へ挿入
                $body = '<body>';
                $source = str_replace(
                        $body,
                        $body . $cm_tag . $cv_tag,
                        $source);
            }
            break;
    case DEVICE_TYPE_ADMIN:
    default:
            break;
    }
}
```

3-12　ヘッダーにタグ追加

ヘッダーにタグ追加する

・テンプレートのヘッダーにプラグインからタグを追加したい場合はいくつか方法があります。

•SC_Helper_Plugin::setHeadNaviを使う

•①プラグインメインクラスのregister関数で使う事が出来ます。

```
function register(SC_Helper_Plugin $objHelperPlugin) {
        // ヘッダへの追加
        $template_dir = PLUGIN_UPLOAD_REALDIR . 'TopicPath/templates/';
        $objHelperPlugin->setHeadNavi($template_dir . 'plg_topicPath_header.tpl');
}
```

② ナビに追加したテンプレート（plg_topicPath_header.tpl）で、タグを出したいページだけに出力される様にします。

```
$arrPageLayout = $this->get_template_vars('arrPageLayout');
    switch($arrPageLayout['device_type_id']){
        case 1:
            break;
        case 2:
            break;
        case 10:
            switch($_SERVER['PHP_SELF']){
                case '/products/list.php':
                case '/products/detail.php':
                    echo('<link rel="stylesheet" href="/plugin/TopicPath/media/topicPath.css" type="text/css" media="screen" />');
                    break;
                default:
            }
            break;
        default:
            switch($_SERVER['PHP_SELF']){
                case '/admin/design/bloc.php':
                    echo('<script type="text/javascript" src="/plugin/TopicPath/media/topicPath.js"></script>');
                    break;
                default:
            }
    }
```

・ site_frameをトランスフォームする

・その他（良い方法があればご連絡ください）

4. リファレンス

4-1　SC_Helper_Transform

トランスフォーム

SC_Helper_Transformを使うことで簡単にテンプレートの変更を行う事が出来ます。
select.・find・endを使い、テンプレートの変更を加えたい箇所を指定し、指定した要素に対してhtml(文字列)を挿入・置換します。

関数名	備考
select	セレクタを用いてエレメントを選択する セレクタの指定方法は、タグ名、ID名、クラス名のみに対応しています。 セレクタの併用は可能ですが 指定する順番は、タグ名→ID名→クラス名 の順となっています。
find	セレクタを用いて、選択したエレメント内をさらに絞り込む
end	選択状態を指定数戻す
insertBefore	要素の前にHTMLを挿入
insertAfter	要素の後にHTMLを挿入
appendFirst	要素の先頭にHTMLを挿入
appendChild	要素の末尾にHTMLを挿入
replaceElement	要素を指定したHTMLに置換
removeElement	要素を削除する

変更対象のテンプレートのソース(文字列)を引数にインスタンスを生成

```
Ex)
$objTransform = new SC_Helper_Transform($source);
$objTransform->select('div.now_dir')->replaceElement(file_get_contents($template_dir . 'snip.admin_basis_category_add.tpl'));
```

** タグで識別子がid="now_dir"の要素をセレクト**

置換するソースを渡す(文字列)

4-2　テーブル定義

dtb_plugin : プラグイン情報テーブル

ﾌｨｰﾙﾄﾞ名	型	長さ	NOT NULL	ﾃﾞﾌｫﾙﾄ値	備　　考
plugin_id	integer	11	NOT NULL		PK
plugin_name	text	0	NOT NULL		表示用プラグイン名
plugin_code	text	0	NOT NULL		プラグインコード（ユニーク）
class_name	text	0	NOT NULL		メインクラス
author	text	0		NULL	作者名
author_site_url	text	0		NULL	参考サイトURL（作者）
plugin_site_url	text	0		NULL	参考サイトURL（プラグイン）
plugin_version	text	0		NULL	プラグインバージョン
compliant_version	text	0		NULL	対応バージョン
plugin_description	text	0		NULL	詳細説明
priority	integer	11	NOT NULL	0	優先順位（高い方が優先）
enable	smallint	6	NOT NULL	0	1:有効, 2:無効
free_field1	text	0		NULL	カスタムフィールド
free_field2	text	0		NULL	カスタムフィールド
free_field3	text	0		NULL	カスタムフィールド
free_field4	text	0		NULL	カスタムフィールド
create_date	timestamp	0	NOT NULL	now()	登録日時
update_date	timestamp	0	NOT NULL	now()	更新日時

dtb_plugin_hookpoint : 使用フックポイントテーブル

ﾌｨｰﾙﾄﾞ名	型	長さ	NOT NULL	ﾃﾞﾌｫﾙﾄ値	備　　考
plugin_hookpoint_id	integer	11	NOT NULL		PK
plugin_id	integer	11	NOT NULL		プラグインID
hook_point	text	0	NOT NULL		使用するフックポイント
callback	text	0			フックポイントで実行されるコールバック関数
create_date	timestamp	0	NOT NULL	now()	登録日時
update_date	timestamp	0	NOT NULL	now()	更新日時

4-3　定数一覧

定数一覧

定数名	値	備　　考
PLUGIN_UPLOAD_REALDIR	DATA_REALDIR . "downloads/plugin/"	インストールされた プラグインの保存ディレクトリ
PLUGIN_HTML_REALDIR	HTML_REALDIR . "plugin/"	インストールされた プラグインのhtmlディレクトリ
PLUGIN_TEMP_REALDIR	HTML_REALDIR . "upload/temp_plugin/"	プラグインファイル一時保存先
PLUGIN_EXTENSION	tar,tar.gz	プラグインファイル登録可能拡張子
DOWNLOADS_TEMP_PLUGIN_UPDATE_DIR	DATA_REALDIR . "downloads/tmp/plugin_update/"	プラグイン一時展開用ディレクトリ(アップデート用)
DOWNLOADS_TEMP_PLUGIN_INSTALL_DIR	DATA_REALDIR . "downloads/tmp/plugin_install/"	プラグイン一時展開用ディレクトリ(インストール用)
PLUGIN_HTML_URLPATH	ROOT_URLPATH . "plugin/"	プラグインURL
HOOK_POINT_PREPROCESS	LC_Page_preProcess	スーパーフックポイント(プレプロセス)
HOOK_POINT_PROCESS	LC_Page_process	スーパーフックポイント(プロセス)
SMARTY_FORCE_COMPILE_MODE	false	SMARTYコンパイルモード
PLUGIN_ACTIVATE_FLAG	true	プラグインのロード可否フラグ

＊SMARTY_FORCE_COMPILE_MODEをtrueにする事で 、template_c 以下のコンパイルファイル生成が毎回実行されます。開発時にお使い下さい
＊ PLUGIN_ACTIVATE_FLAGをfalseにする事で全プラグインが実行されなくなります。