



GENECODE

テンプレート開発・運用
ガイド

第 1 版

最終更新日 2016/03/31

目次

第 1 章	テンプレート開発	5
1-1	テンプレート書式	5
1-2	サンプルテンプレート	5
1-3	変換元コンテンツの利用	7
1-4	パーツを使用する	7
(1)	パーツの種類	7
(2)	共通オプションと個別オプション	8
1-5	<gc-script>を使用する	9
1-6	テンプレートマッピングとコントローラ	10
(1)	テンプレートマッピング	10
(2)	コントローラ	11
1-7	PostJS	11
1-8	画像変換	12
1-9	CSS 変換	12
(1)	ブラウザによる出し分け	12
(2)	ベンダープレフィックス変換	12
1-10	端末グループ	12
第 2 章	検証	15
2-1	単体テスト	15
(1)	単体テストの概要	15
(2)	ジーンコード IDE によるテスト方法	15
(3)	実機によるテスト方法	15

(4)	単体テストの観点.....	15
2-2	結合テスト.....	16
(1)	PC ブラウザによるテスト.....	16
(2)	実機によるテスト.....	17
(3)	結合テストの観点.....	17
2-3	デバッグ方法.....	17
(1)	問題切り分け.....	17
(2)	ジーンコードサーバーのログ.....	18
(3)	PC ブラウザによるデバッグ.....	19
(4)	テンプレートやコントローラのデバッグ.....	19
(5)	よくある問題.....	19
第3章	運用.....	20
3-1	アプリケーションのリリース.....	20
3-2	テンプレート修正の必要性について.....	21
3-3	PC サイトのマーキング.....	21
3-4	バージョン管理システムを利用する.....	22

改訂履歴

版数	発行日	改訂内容
第 1 版	2016 年 3 月 31 日	・初版発行

第1章 テンプレート開発

テンプレート開発は、ジーンコード IDE を使用して行います。

1-1 テンプレート書式

テンプレートは HTML 拡張子のテキストファイルです。

テンプレートは HTML タグと「gc-」から始まるジーンコードタグとで構成されます。「gc-」から始まるタグがジーンコードサーバーによって処理されます。

以下が主な<gc-*>タグです。

<code><gc-parts>...</gc-parts></code>	パーツによる変換結果を出力します。
<code><gc-script>...</gc-script></code>	サーバーサイド JavaScript を実行します。プログラミングによる高度な変換処理を実行します。jQuery による DOM アクセスも可能です。
<code><gc-include /></code>	別テンプレートを読み込みます。読み込んだファイル内にも<gc-*>タグを記述できます。
<code><gc-each>...</gc-each></code>	繰り返し処理を実行します。
<code><%-- --%></code>	コメント。<gc-*>タグをコメントアウトできます。

1-2 サンプルテンプレート

```

<!DOCTYPE html>
<gc-script>gcruntime.loadJs('/gcutils.js');</gc-script>
<html>
<head>
  <%-- metaタグ --%>
  <meta charset="utf-8">
  <gc-script>gcutil.writeViewport();</gc-script>
  <gc-script>
    $('meta[name=viewport]').remove();
    gcutil.toHtml($('meta[name]'));
  </gc-script>
  <%-- titleタグ --%>
  <gc-script>gcutil.toHtml($("#title"));</gc-script>
  <%-- linkタグ --%>
  <link href="/template/css/common/gcreset.css" rel="stylesheet" type="text/css" />
  <link href="/template/css/common/gccommon.css" rel="stylesheet" type="text/css" />
  <link href="/template/gcparts/css/gcparts-2.2.0.css" rel="stylesheet" type="text/css" />
  <%-- scriptタグ --%>
  <gc-script>
    gcutil.toHtml($('head script'));
  </gc-script>
  <script src="/template/gcparts/js/jquery-1.8.0.min.js" type="text/javascript"></script>
  <gc-postJsHead />
</head>
<body>
  <gc-script>
    var $script = $("body script");
    $script.remove();
  </gc-script>
  <gc-include path="inc/header.html" />
  <gc-script>
    gcutil.out($("#h1").html());
  </gc-script>
  <gc-parts gcid="xxx" name="xxx" view="001">
    <gc-paramlist>
      <gc-param name="selector">#foo</gc-param>
    </gc-paramlist>
  </gc-parts>
  <gc-include path="inc/footer.html" />
  <%-- scriptタグ --%>
  <gc-script>
    gcutil.toHtml($script);
  </gc-script>
  <script src="/template/gcparts/js/gcparts-2.2.0.js" type="text/javascript"></script>
  <gc-postJsBody />
</body>
</html>

```

HTMLタグは直接記述します。

```

<link href="/template/css/common/gcreset.css" rel="stylesheet" type="text/css" />
<link href="/template/css/common/gccommon.css" rel="stylesheet" type="text/css" />
<link href="/template/gcparts/css/gcparts-2.2.0.css" rel="stylesheet" type="text/css" />

```

<gc-include>タグ。
別のテンプレートファイルを読み込みます。
ヘッダ・フッタなど共通化したい部分に使えます。

```

<gc-include path="inc/header.html" />

```

<gc-script>タグ。
サーバー側でJavaScriptコードが実行されます。
PCサイトのコンテンツをjQueryで参照したり、
HTMLタグを取得したりできます。

```

<gc-script>
  gcutil.out($("#h1").html());
</gc-script>

```

<gc-parts>タグ。
PCサイトのコンテンツをパーツを使用
して変換した結果に置き換わります。

```

<gc-parts gcid="xxx" name="xxx" view="001">
  <gc-paramlist>
    <gc-param name="selector">#foo</gc-param>
  </gc-paramlist>
</gc-parts>

```

1-3 変換元コンテンツの利用

変換元コンテンツをテンプレート内から参照・利用する方法としては、以下の 2 通りの方法があります。

(A) パーツを使用する

変換元ページのうち、セレクトで指定した HTML ブロックを指定したパーツにより変換し、出力します。パーツはスマートフォン向けの UI を部品化したコンポーネントであり、ジーンコード IDE での入力・編集がサポートされているため、最も迅速な開発ができます。


(B) <gc-script>を使用する

変換元ページのうち、セレクトで指定した HTML ブロックを JavaScript で参照したり、変換・出力します。通常の JavaScript と同様、条件分岐処理や関数化、DOM(変換元コンテンツ)へのアクセスも可能です。

1-4 パーツを使用する

(1) パーツの種類

ジーンコードで提供しているパーツは 40 個ほどあります。そのうちの一部を紹介します。

パーツ名	説明	サンプルイメージ
パネル	HTML タグをそのまま出力します。	
テキストリンクリスト	リンク一覧をタッチしやすい UI に変換して出力します。	

<p>画像付リンクリスト</p>	<p>サムネイル画像を含むリンク一覧をタッチしやすいUIに変換して出力します。</p>	
<p>カルーセル</p>	<p>スワイプ操作で複数の画像を閲覧できるUIに変換して出力します。</p>	
<p>テーブルフォーム</p>	<p>table タグで構成された入力フォームを見出しとフォームフィールドが縦に並んだUIに変換します。th タグを見出し、td タグをフォームフィールドとみなします。</p>	
<p>ページャ</p>	<p>ページ切り替え部分をタッチしやすくコンパクトなUIに変換して出力します。</p>	

その他、Bootstrap や jQuery Mobile の書式で出力するパーツもあります。

ジーンコードパーツ一覧

<http://developer.genecode.jp/manuals/gcparts/>

(2) 共通オプションと個別オプション

パーツにはパーツの動作を個別に制御するためのオプション機能があります。オプション機能には、パーツ毎に機能が異なる個別オプションと、どのパーツでも共通して使用できる共通オプションの2種類が存在します。共通オプションには以下のようなものがあります。

オプション名	説明
--------	----

削除する画像	削除対象となる img タグの src 属性を指定します。ワイルドカードも使用できます。
テキスト化する画像	alt 属性でのテキスト化対象となる img タグの src 属性を指定します。ワイルドカードも使用できます。
画像の URL 置換	画像 URL を置換します。画像を差し替える場合に使用します。
テキスト置換ルール	テキストをキーワードで置換します。
class 属性への要素名の追加	class 属性に自身のタグ名を追加します。

1-5 <gc-script>を使用する

<gc-script>はサーバーサイドで実行される JavaScript コードブロックです。通常の JavaScript 構文に加え、変換元コンテンツへの DOM API/jQuery による参照、POST データの参照、レスポンスの出力等が可能です。以下に主な API を列挙します。

- 変換元コンテンツの取得
 - セレクタによる取得: \$(selector)
- リクエスト情報の取得
 - リクエストメソッドの取得: gcutil.getMethod()
 - リクエストヘッダの参照: gcutil.getHeader(name)
 - リクエスト URI の参照: gcutil.getUri()
 - クエリー文字列の参照: gcutil.getQueryString()
 - POST データの参照: gcutil.getPostParameter(name)
- レスポンスの設定
 - レスポンスヘッダの設定: gcutil.addResponseHeader(name, value)
 - テキストの出力: gcutil.out(text)
 - Query オブジェクトの出力: gcutil.toHtml(\$element)

- 端末グループ
 - グループ判定: `gcutil.isGroup(name)`
- ユーティリティ
 - HTML エスケープ: `gcutil.escapeHtml(text)`
- ログ
 - デバッグログ出力: `gclog.debug(text)`
 - エラーログ出力: `gclog.error(text)`
- ランタイム
 - 外部 JS ファイル実行: `gcruntime.loadJs(file)`

※`<gc-script>`はサーバーサイドで動作するため、`window` や `navigator` 等のブラウザオブジェクトやイベント(`addEventListener` 等)は利用できません。詳しくはテンプレート構文リファレンスを参照してください。

1-6 テンプレートマッピングとコントローラ

(1) テンプレートマッピング

URL と変換に使用するテンプレートとの割り当てをテンプレートマッピングと呼びます。テンプレートマッピングを使用すると、複数の URL を同一のテンプレートで変換させることができます。

テンプレートマッピングでは、ホスト名およびクエリー文字列を除いたリクエスト URI とテンプレートファイル名をペアで指定します。リクエスト URI 部分には「/」以外にマッチするワイルドカード(*)を使用することができます。

リクエスト URI	テンプレートパス	マッチする URL の例
<code>/news/</code>	<code>news.html</code>	http://www.example.com/news/ http://www.example.com/news/?from=top
<code>/category/*</code>	<code>category.html</code>	http://www.example.com/category/aaaaaa http://www.example.com/category/bbbbbbb
<code>/search/*/*/*</code>	<code>search.html</code>	http://www.example.com/search/ccc/ddd/eee

※その他、「/」にもマッチするワイルドカード(**)を使用することもできます。詳しくは設定ファイルリファレンスを参照してください。

(2) コントローラ

クエリー文字列や POST データ、コンテンツ内容を使用してテンプレート選定を行う場合は、コントローラを使用します。コントローラは JavaScript コードにより、変換に使用するテンプレートを決定する方法で、高度なマッピングが可能です。

controller.js

```
var param = gcutil.getParameter("from");  
  
if (param != null && param == "top") {  
  gcruntime.setTemplate("news-top.html");  
} else {  
  gcruntime.setTemplate("news.html");  
}
```

変換に使用するテンプレートを gcruntime.setTemplate関数で指定します。

なお、コントローラを使用する場合は、テンプレートマッピングにコントローラを使用する URL とコントローラのファイル名を指定します。

1-7 PostJS

クライアント JavaScript によってドキュメントの生成や更新が行われた後に、変換を行う場合に使用します。たとえば、JavaScript 型のリコメンドエンジンによって生成されたランキングやおすすめ商品の HTML ブロックをスマートフォン向けに変換するといった用途に使用できます。変換処理はクライアント JavaScript で記述します。

PostJS の編集は、ジーンコード IDE のブラウザ変換ステージで行うことができます。

詳しくは「PostJS マニュアル」を参照してください。

1-8 画像変換

ジーンコードには 2 つの画像変換機能が含まれています。

- 画像圧縮: 3G/LTE 回線の場合に画像圧縮を行って配信します。JPEG 画像のみが対象です。デフォルト設定では 10KB 以上の JPEG 画像に対して適用されます。
- リサイズ・クリッピング: 画像パーツを使用すると、画像のリサイズやクリッピングを行うことができます。

1-9 CSS 変換

スマートフォン向けに配置した CSS を特定ブラウザ向けに変換する機能もあります。

(1) ブラウザによる出し分け

CSS ファイル内で特定ブラウザのみに適用するルールを記述できます。



(2) ベンダープレフィックス変換

ブラウザ毎に異なる CSS のベンダープレフィックスを一律に変換できます。変換ルールは CSS 変換シート(css-conversion.csv)で指定します。標準設定では、-webkit-から始まるプロパティ及び-webkit-から始まる値が変換対象となります。

1-10 端末グループ

端末グループは、アクセスしたデバイスをグループ化したもので、テンプレートや CSS 変換ルールで利用できます。標準設定では、以下の端末グループが定義されています。

カテゴリ	端末グループ名	説明

デバイス	pc	PC 端末を表します。
	tablet	タブレット端末を表します。
	smartphone	スマートフォン端末を表します。
	mobile	フィーチャーフォン端末を表します。
OS	ios	iOS を表します。
	android	Android OS を表します。
	windowsphone	Windows Phone OS を表します。
	blackberry	BlackBerry OS を表します。
	android2x	Android OS ver2 系を表します。
ブラウザ	chrome	Chrome を表します。
	firefox	Firefox を表します。
	opera	Opera を表します。
	iemobile	Internet Explorer Mobile を表します。
	ie	Internet Explorer を表します。
	edge	Edge を表します。
	safari	Safari を表します。

その他	unsupport	非対応グループを表します。標準設定ではスマートフォン以外が非対応端末グループに当たります。
-----	-----------	---

第2章 検証

2-1 単体テスト

(1) 単体テストの概要

単体テストはジーンコード IDE を使用して行います。

(2) ジーンコード IDE によるテスト方法

テストを行うページを表示し、プレビューボタンをクリックすると変換結果が画面に表示されます(※)。

(3) 実機によるテスト方法

スマートフォン実機での動作確認を行う場合は、まずスマートフォン側のネットワーク設定を次のように変更します。

- プロキシのホスト名:ジーンコード IDE をインストールした PC のホスト名/IP アドレスを指定します
- プロキシポート番号:8080

ジーンコード IDE でテストを行うページを表示し、QR コードボタンをクリックします。実機で QR コードを読み込ませると、変換後の結果が表示されます(※)。

※変換したページから別ページへの遷移はできません。ページをまたいだテストを行う場合は、結合テストの章をご覧ください。

(4) 単体テストの観点

以下の2つの観点でテストを行います。

- デザインのチェック:スマートフォン画面がスマートフォン向けの UI/デザインになっているかどうかを調べます。

- PC サイトの情報・動作の連動性: 変換によって、情報や機能が失われていないかどうかを確認します。具体的には PC サイトとスマートフォンサイトとを見比べ、表示されている項目に抜けや重複がないかを調べます。また、フォームの選択項目によって表示される項目が変更される等、元々の PC サイトと同じ動作をしているかどうかを調べます (JavaScript の動作確認)。

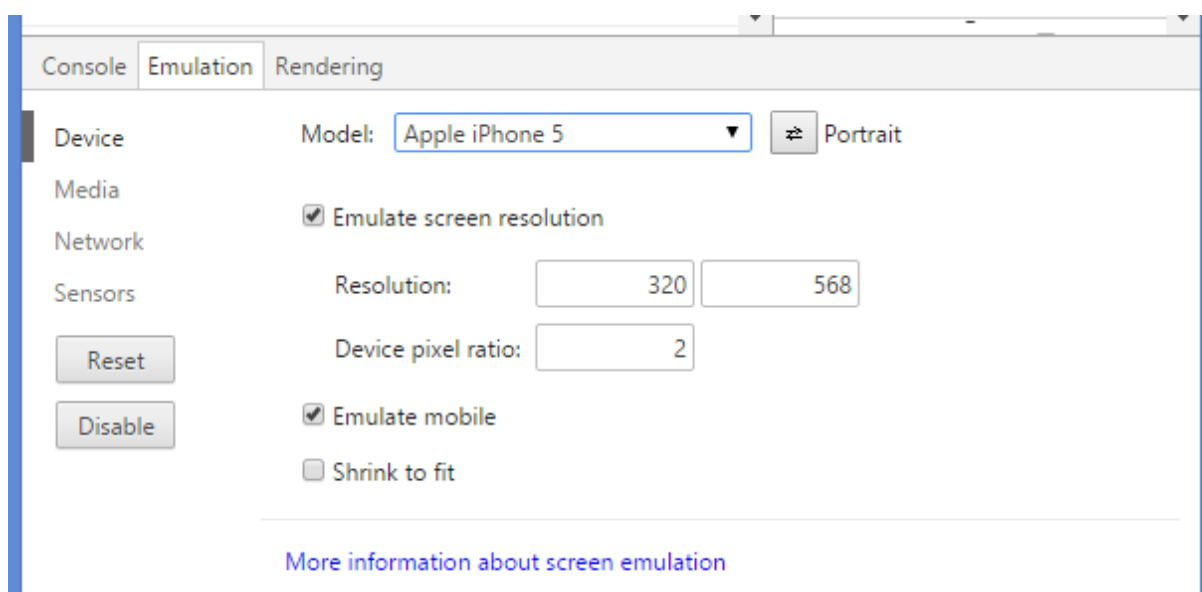
2-2 結合テスト

結合テストはジーンコード IDE ではなく、ジーンコードサーバーを使用した環境で行います。

(1) PC ブラウザによるテスト

Chrome ブラウザを使用し、スマートフォンをエミュレーションしてテストします。

Chrome で対象ページを開いた後、[F12]キーをクリックして Developer Tools を開きます。Emulation タブの Device メニューで、テストを行う機種を選択します。



(2) 実機によるテスト

実機テストでは機種選定がポイントとなります。スマートフォンは、主に以下のパラメータによって表示内容や動作が変化します。対象機種とブラウザシェアを考慮し、テスト機種を選定します。

- ウィンドウ幅: 320px (iPhone 5/5S)、360px (Android 4 系)、375px (iPhone 6)
- ブラウザ種別: Safari (iOS)、Android 標準ブラウザ、Chrome (iOS/Android)
- OS バージョン: iOS 7/8/9、Android 4/5/6

(3) 結合テストの観点

以下の 2 つの観点で複数の実機を使用してテストを行います。

- 遷移/フォームの動作: リンクやフォーム送信などの画面遷移が PC サイトと同様に行えるかをチェックします。
- 様々なバリエーション下での連動性: ログインユーザーの種別や商品カテゴリなど、ページの表示バリエーションを変動させたとき、PC サイトからの連動箇所が正しく表示されているかをチェックします。

2-3 デバッグ方法

(1) 問題切り分け

デバッグのポイントは、その事象がジーンコードに関連した問題なのかどうかを切り分けることにあります。

まず、事象が発生した機種を使用して PC サイトにアクセスした場合、同様の問題が発生するかどうかを調べます。問題が発生する場合は、PC サイト自体にその問題が存在するか、機種依存による問題が発生しています。

一方、PC サイト側では問題が発生せず、ジーンコードを経由した場合のみ発生する問題であれば、ジーンコードサーバーの設定やテンプレートが影響しています。

(2) ジーンコードサーバーのログ

ジーンコードサーバーが出力するログは以下の通りです。

ログ種別	内容
アクセスログ	<p>Apache のアクセスログです。ジーンコードサーバーでは Apache を使用するため、ジーンコードサーバーへのアクセスログは全て Apache のアクセスログに記録されます。</p> <p>パッケージ版 Apache の標準設定では <code>/var/log/httpd/access_log</code> に、ビルド版 Apache の標準設定では <code>/usr/local/apache2/logs/access_log</code> に出力されます。また、IDE では <code>C:%genecode%programs%genecode-ide-x.x.x%apache%logs%access_log</code> に出力されます。</p>
エラーログ	<p>ジーンコードが出力するログのうち、システムレベル／サーバーレベルのエラーは Apache のエラーログに出力されます。例えば、Apache 起動時にジーンコードで発生したエラーは、Apache エラーログに出力されます。</p> <p>パッケージ版 Apache の標準設定では <code>/var/log/httpd/error_log</code> に、ビルド版 Apache の標準設定では <code>/usr/local/apache2/logs/error_log</code> に出力されます。また、IDE では <code>C:%genecode%programs%genecode-ide-x.x.x%apache%logs%error_log</code> に出力されます。</p>
アプリケーションログ (app_log)	<p>ジーンコードが出力するログのうち、アプリケーションレベルのエラーはアプリケーションログ (app_log) に出力されます。例えば、テンプレートの構文エラーや実行時エラーは、アプリケーションログに出力されます。</p> <p>また、テンプレートやコントローラで <code>gclog.debug()</code> コードを記述すると、アプリケーションログに任意のログを出力できます。</p> <p>パッケージ版 Apache の場合、ジーンコードの出荷時設定では <code>/var/log/httpd/app_log</code> に、ビルド版 Apache の標準設定では</p>

	<p>/usr/local/apache2/logs/app_log に出力されます。また、IDE では C:¥genecode¥programs¥genecode-ide-x.x.x¥apache¥logs¥app_log に出力されま す。</p>
--	--

(3) PC ブラウザによるデバッグ

Chrome ブラウザの Developer Tools を使用すると、JavaScript や CSS のデバッグが可能です。

(4) テンプレートやコントローラのデバッグ

ログ出力コードを埋め込んだり、変換処理を単純化して試します。

(5) よくある問題

事象	主な原因
変換されずに PC サイトが表示されます	<ul style="list-style-type: none"> ● 実行時にテンプレートで実行時エラーが発生した ● テンプレートマッピングやコントローラで変換に使用するテンプレートが指定されていない ● Apache 設定で変換が無効になっている
パーツを使用しても何も出力されません	<ul style="list-style-type: none"> ● セレクタで指定した要素が PC サイト上に存在しない ● パーツの仕様で変換対象外の HTML ブロックを変換しようとした ● パーツのオプションパラメータの値が不適切

第3章 運用

3-1 アプリケーションのリリース

ジーンコード IDE で開発したテンプレート一式をジーンコードサーバーに反映するには、以下の 3 つのディレクトリをサーバーにアップロードします。

- [プロジェクトルート]¥[プロジェクト名]¥appjs
- [プロジェクトルート]¥[プロジェクト名]¥conf
- [プロジェクトルート]¥[プロジェクト名]¥template

ジーンコードサーバー上のアプリケーションデータは、/home/genecode ディレクトリに集約して配置されます。これらのファイルを更新すると、即時に反映されます。

なお、ジーンコードサーバー上でのアプリケーションの構成ファイル、初期設定では以下の状態になっています。

テンプレートマッピング	/home/genecode/conf/template-mapping.csv
JS 変換ルール	/home/genecode/conf/js/*.gcl
パーツ	/home/genecode/appjs/gcparts
コントローラ	/home/genecode/appjs/controller.js
エラーコントローラ	/home/genecode/appjs/error-controller.js
JavaScript ユーティリティ	/home/genecode/appjs/gcutils.js
その他のサーバー用 JS	/home/genecode/appjs/*.js
テンプレート	/home/genecode/template/*.html
postJS ファイル	/home/genecode/template/*.gcl
画像ファイル	/home/genecode/template/img
CSS ファイル	/home/genecode/template/css
JS ファイル	/home/genecode/template/js
パーツ	/home/genecode/template/gcparts

3-2 テンプレート修正の必要性について

PC サイトが変更された場合、テンプレート修正が必要なケースと不要なケースの 2 通りあります。基本的な判断基準としては、PC サイトの構造もしくは連動性が変更された場合はテンプレート修正が必要で、PC サイトのデータが変更された場合はテンプレート修正は不要です。

例えば、次のような場合はデータの変更に当たるため、通常テンプレート修正は不要です。

- 文言の変更
- 画像の変更
- リンク先の変更
- ニュース一覧でのニュースの追加
- セレクトボックスやラジオボタンの選択項目の追加

一方、次のような場合は HTML 構造の変更に当たるため、テンプレート修正が必要になることがあります。

- 従来はロコミが表示されないことがないページだったが、ロコミを表示するエリアを追加した
- 従来は table タグでレイアウトしていたが、ul/li タグでレイアウトしなおした

また、次のように連動性に関係する部分の変更に当たる場合も、テンプレート修正が必要になることがあります。

- URL を変更した
- id 属性や class 属性の値を変更した
- id 属性を別の要素に移動した

3-3 PC サイトのマーキング

PC サイト側にジーンコード専用の属性 (data-gc-cmd 属性) を追加することで、簡単な変換を行うことができるため、テンプレート修正の代わりにマーキングで代用することも可能です。なお、マーキングによる変換はテンプレート処理よりも前に実行されます。

属性	説明
data-gc-cmd="removeall"	属性が付与された要素を子要素も含めて削除します。

data-gc-cmd="rename:タグ名”	タグ名を変更します。
--------------------------	------------

詳しくは、マーキングリファレンスをご覧ください。

3-4 バージョン管理システムを利用する

Subversion などのバージョン管理システムを使用し、プロジェクトを管理する場合は以下の 5 つのディレクトリをバージョン管理の対象にします。

- <プロジェクトルート>%appjs
- <プロジェクトルート>%conf
- <プロジェクトルート>%pattern
- <プロジェクトルート>%snippets
- <プロジェクトルート>%template

なお、<プロジェクトルート>%reuse ディレクトリには、ジーンコード IDE の URL リストに登録された情報が記録されておりますが、ファイルサイズが非常に大きくなります。したがって reuse ディレクトリはバージョン管理の対象にせず、別途チーム間でファイル共有することをお勧めします。