



# GENECODE

ジーンコード ver 3

ユーザマニュアル

---

設定ファイルリファレンス

第 1 版

最終更新日 2018/12/20

## 目次

第 1 章	設定ファイル一覧.....	5
第 2 章	変換テンプレートマッピングファイル(template-mapping.csv).....	6
2-1	概要.....	6
2-2	書式.....	6
(1)	requestUri 列: リクエスト URI.....	6
(2)	templatePath 列: テンプレート・コントローラパス.....	7
(3)	attribute 列: 属性.....	8
(4)	templateMode 列: テンプレートモード.....	8
第 3 章	変換テンプレート設定コントローラ(controller.js).....	10
3-1	概要.....	10
3-2	書式.....	10
第 4 章	エラー用変換テンプレート設定コントローラ(error-controller.js).....	12
4-1	概要.....	12
4-2	書式.....	12
第 5 章	CSS 変換シート(css-conversion.csv).....	13
5-1	概要.....	13
5-2	書式.....	13
(1)	selector 列.....	13
(2)	property_name 列.....	13
(3)	property_value 列.....	14
(4)	グループ A 以降の列.....	14
5-3	変換ルールの種類.....	14
5-4	ワイルドカードと後方参照.....	14

5-5	ルールの適用順序	15
第 6 章	端末グループ設定(group.gcl)	16
6-1	概要	16
6-2	書式	16
6-3	条件式	17
(1)	完全一致	17
(2)	範囲指定	17
(3)	OR	17
(4)	否定	18
第 7 章	端末情報項目設定(terminfo.js)	19
7-1	概要	19
7-2	書式	19
7-3	注意事項	19
第 8 章	IP アドレス帯域リスト(specific-iplist.csv)	21
8-1	概要	21
8-2	書式	21
8-3	更新データについて	21
第 9 章	端末情報収集スクリプト(terminfo.js)	22
9-1	概要	22
9-2	書式	22
9-3	注意事項	22

**改訂履歴**

版数	発行日	改訂内容
第 1 版	2018 年 12 月 20 日	・初版発行

## 第1章 設定ファイル一覧

---

- 変換テンプレートマッピングファイル(template-mapping.csv)
- 変換テンプレート設定コントローラ(controller.js)
- エラーステータス用変換テンプレート設定コントローラ(error-controller.js)
- CSS 変換シート(css-conversion.csv)
- 端末グループ設定(group.gcl)
- 端末情報項目設定(terminfo-script-keys.csv)
- IP アドレス帯域リスト(specific-iplist.csv)
- 端末情報収集スクリプト(terminfo.js)

## 第2章 変換テンプレートマッピングファイル (template-mapping.csv)

---

### 2-1 概要

---

リクエスト情報のうち、リクエスト URI をベースとしたマッピングを行うためのファイルです。

このファイルでは、テンプレートを直接マッピングさせる事以外にコントローラにマッピングさせる事もできます。このファイルではリクエスト URI でのみのマッピングとなるので、それ以外の情報を元にテンプレートを決めたい場合は、一旦コントローラへマッピングさせコントローラにてマッピングを行います。

どのマッピングにも当てはまらない場合は、変換が行われず PC サイトの内容がそのまま出力されます。

このファイルは変更後、即座に反映されます。

初期状態では/home/genecode/conf/template-mapping.csv にあります。

### 2-2 書式

---

テンプレートマッピングファイルは、カンマ区切りの CSV フォーマットとなります。1 行目はコメント行、2 行目はヘッダ行となるため、データは 3 行目以降に記述します。

---

(1) requestUri 列: リクエスト URI

---

スキーム、ホスト名・ポート番号、クエリ文字列を除いた URI のパス部分を記述します。

ex. リクエスト URI の指定

URL が http://example.jp:8080/ の場合: /

URL が http://example.jp:8080/index.html の場合: /index.html

※/と/index.html のように同一ページを表示する場合でも、URL の表記が異なる場合はそれぞれ設定する必要があります。

リクエスト URI では任意のパスセグメントを表すワイルドカード「\*」が使用できます。このワイルドカードは「/」にはマッチしません。また、「\*」の前後には「/」しか記述できません。

ex. ワイルドカードの使用例

`/path1/*` : `/path1/`や`/path1/index.html` など、`/path1/`以下の任意の URL にマッチします。ただし、`/path1/path2/`のように、`/path1/`以下に「/」を含む URL にはマッチしません。

`/path1/*/path2*/index.html` : `/path1/foo/path2/bar/index.html` などにマッチします。

1 つのパスセグメントではなく、1 つ以上のパスセグメントを表す場合はワイルドカードを 2 つ組み合わせます。「\*\*」はリクエスト URI の末尾にのみ、指定できます。

ex. ワイルドカードの使用例

`/path1/**` : `/path1/foo/bar` のように`/path1/`で始まる任意の URI

`/path1*/path2/**` : `path1` と `path2` の間は任意で、`path2` 以下も任意の URI

`/path1/**/path2/index.html` : パスの途中で\*\*は出現できないので使用不可

---

## (2) templatePath 列: テンプレート・コントローラパス

---

GeneCode 設定ディレクティブ `GCHtmlTemplate` で設定したディレクトリをルートとした相対パスでテンプレートの位置を指定します。先頭の/を省略した場合は自動で補完されます。

テンプレートパスは、存在するテンプレートパスを設定してください。

ex. `GCHtmlTemplate` で設定したディレクトリが`/home/genecode/template` の場合

`/index.tmpl.html` -> `/home/genecode/template/index.tmpl.html`

`index.tmpl.html` -> `/home/genecode/template/index.tmpl.html`

3 列目の属性で 1 を設定する場合は、この列はコントローラの指定とみなされ、GeneCode 設定ディレクトリタイプ GCHtmlJsDirectory で設定したディレクトリをルートとした相対パスでコントローラパスを設定します。

先頭の / を省略した場合は自動で補完されます。

ex. GCHtmlJsDirectory で設定したディレクトリが /home/genecode/appjs の場合

/controller.js → /home/genecode/appjs/controller.js

foo/controller2.js → /home/genecode/appjs/foo/controller2.js

得意のリクエストだけ、変換したくない場合は、リクエスト URI に対するテンプレートパスに none と指定します。

ex. /pc/以下は全て変換したくない場合

/pc/\*\*, none, 0

---

### (3) attribute 列:属性

---

2 列目の設定が、テンプレートに対するパスなのか、コントローラに対するパスなのかを設定します。

- 0:通常のテンプレートファイル
- 1:コントローラ用 JS ファイル

0・1 以外の値を記述した場合は、0 とみなされます。

---

### (4) templateMode 列:テンプレートモード

---

元コンテンツのうち、マークが付与された要素のみで DOM を構築する場合に使用します。

templateMode 列を使用する場合は、1 行目のバージョンを 1.1.0 に設定する必要があります。

- default: 元コンテンツの全ての要素を使用して DOM を構築します。通常は default を使用します。
- mark: 元コンテンツのうち、マーク (data-gc-mark 属性) が付与された要素のみで DOM を構築します。マークが付与されていない要素は DOM 化されないため、メモリ使用量を削減できます。元コ



コンテンツがマーキングされ、かつマークされた要素のみを使用するテンプレートの場合に使用します。

なお上記以外の値を記述した場合は、default とみなされます。

## 第3章 変換テンプレート設定コントローラ (controller.js)

---

### 3-1 概要

---

リクエスト URI だけでなく、リクエストヘッダやレスポンスの内容からテンプレートを決定したい場合は、テンプレートマッピングではなく、コントローラにて行います。

初期状態では/home/genecode/appjs/controller.js にあります。

### 3-2 書式

---

コントローラは、サーバ上で動作する JavaScript として記述し、リクエスト時に GeneCode ランタイムから実行されます。テンプレート中の<gc-script>タグ内と同じ構文の JavaScript で記述でき、GeneCodeAPI や PC コンテンツの jQuery オブジェクトも参照可能となっています。

キャラクタエンコーディングは UTF-8 固定です。

JavaScript で判定処理を行った結果は、以下の API でテンプレートを GeneCode ランタイムに設定します。

```
void gcruntime.setTemplate(string templatePath)
```

ex.コンテンツ内に id が error となっている要素が存在する場合はエラー用のテンプレートを設定する

```
if($("#error").size() >= 1) {  
    gcruntime.setTemplate("/error.tmpl.html");  
    return;  
}  
gcruntime.setTemplate("/common.tmpl.html");
```

※`gcruntime.setTemplate` 関数で設定するテンプレートパスは、変換テンプレートマッピングで使用できるテンプレートパスと同じルールになります

`gcruntime.setTemplate` 関数が呼ばれなかった場合は変換が行われず、PC サイトの内容がそのまま出力されます。

## 第4章 エラー用変換テンプレート設定コントローラ (error-controller.js)

---

### 4-1 概要

---

PC サーバからのレスポンス時のステータスコードが 400、500 番台のエラーステータスの時でもコンテンツ変換を行いたい場合には、このエラーコントローラで変換テンプレートとのマッピングを行います。

初期状態では/home/genecode/appjs/error-controller.js にあります。エラーコントローラのファイル名は error-controller.js 固定となります。

### 4-2 書式

---

コントローラ中に記述する書式は、変換テンプレート設定コントローラと同じで、JavaScript で記述します。キャラクタエンコーディングは UTF-8 固定です。

PC サーバからのステータスコードは以下の API で取得します。

```
int gcruntime.getStatusCode();
```

エラーコントローラ内で setTemplate 関数が一度も実行しなかった場合は、HTML 変換が行われず、Apache のエラー設定に従ったエラー処理が行われます。

## 第5章 CSS 変換シート(css-conversion.csv)

---

### 5-1 概要

---

CSS 変換を行うためのルールを指定する設定ファイルです。

初期状態では/var/genecode/sysconf/css-conversion.csv にあります。

### 5-2 書式

---

フォーマットは CSV 形式です。

1 行目はコメント行で、2 行目が項目名を表す行です。3 行目以降に変換ルールを記述します。

変換ルールは 1 行で 1 つの変換ルールを表します。selector 列, property\_name 列, property\_value 列の 3 列で CSS 変換のマッチング条件となるプロパティを指定し、4 列目以降(以下の例ではグループ A,グループ B,グループ C)ではグループ毎に変換後の値を指定します。

```
#1.0.0,,,,
```

```
selector,property_name,property_value,グループ A,グループ B,グループ C  
,border-radius,1em,-webkit-border-radius: 1em, -moz-border-radius: 1em,
```

---

#### (1) selector 列

---

変換の対象となるセレクタを記述します。何も指定しない場合は、すべてのセレクタが変換の対象となります。

---

#### (2) property\_name 列

---

変換の対象となるプロパティ名を記述します。プロパティ追加を行う場合を除き、必須です。プロパティ追加を行う場合は何も指定しません。

---

### (3) property\_value 列

---

変換の対象となるプロパティ名を記述します。何も指定しない場合は、プロパティ値がどんな値であっても変換の対象となります。

---

### (4) グループ A 以降の列

---

対象のグループに対する変換後の値を記述します。どのような変換を行うかによって、以下の 4 種類の記述方法に分けられています。

- (1) プロパティ値のみを変換: 「1em」のように変換後のプロパティ値だけを指定します。
- (2) プロパティ名のみを変換: 「border-radius:」のように変換後のプロパティ名だけを指定します。
- (3) プロパティ名と値の両方を変換: 「border-radius: 1em」のようにプロパティ名と値を「:」区切りで指定します。
- (4) プロパティ名と値の両方を複数のプロパティに変換: 「-webkit-border-radius: 1em; border-radius: 1em」のようにプロパティを「;」区切りで指定します。

何も指定しない場合はそのグループに対しては無変換になります。

---

## 5-3 変換ルールの種類

---

変換ルールにはプロパティ追加・プロパティ削除・プロパティ更新の 3 種類があります。変換ルールの種類は、property\_name 列やグループ列の記述方法により判別されます。

- (1) property\_name 列と property\_value 列の両方が空欄の場合、宣言ブロックに対するプロパティ追加と判別されます。
- (2) グループ列の値が「/」の場合、プロパティ削除と判別されます。
- (3) それ以外の場合は、プロパティ更新と判別されます。

---

## 5-4 ワイルドカードと後方参照

---

property\_name 列と property\_value 列にはワイルドカードを表す「?」を使用できます。ワイルドカードは、スペース・「,」「(」「)」「{」「}」「:」「;」以外の文字列にマッチします。

「?」にマッチした部分は、グループ列で後方参照できます。property\_name 列と property\_value 列とを合わせた出現順に¥1,¥2,¥3・・・という書式で参照できます。

※property\_name に「?」を書く場合は、「-webkit-?」のように末尾に 1 回しか記述できません。

※ワイルドカード「?」の上限個数は、1 行につき property\_name 列と property\_value 列とで合わせて 32 個までです。

## 5-5 ルールの適用順序

---

CSV ファイル内でのルールの記述順に沿って一番上から順にマッチング評価されます。

プロパティ追加・削除・更新された結果が、連鎖的に再びマッチング条件で評価されることはありません。

## 第6章 端末グループ設定(group.gcl)

---

### 6-1 概要

---

クライアント(ブラウザ)を端末情報によってグループ分けするための設定ファイルです。

初期状態では/var/genecode/sysconf/group.gcl にあります。

### 6-2 書式

---

基本フォーマットはjson形式です。キャラクタエンコーディングはUTF-8固定です。ただし/\* ~ \*/を使用したコメントを許可しているので、コメントを付ける事ができます。

json プロパティのキーがグループ名で、値が「端末情報の条件リスト」を表します。「端末情報の条件リスト」は「端末情報の条件」を表すオブジェクトの配列となっています。「端末情報の条件」はプロパティのキーが端末情報の項目名で、値が条件式になります。

```
/* 2.0.0 */
{
  "グループ A": [
    {
      "端末情報項目 1": "条件式 1"
      "端末情報項目 2": "条件式 2"
    }
  ],
  "グループ B": [
    {
      "端末情報項目 3": "条件式 3"
    },
    {
      "端末情報項目 4": "条件式 4"
    }
  ],
}
```



```

“グループ C”:[
  {
    “userAgent”: “正規表現文字列”
  }
]
}

```

上記の例では、端末情報項目 1 が条件式 1 に合致し、かつ端末情報項目 2 が条件式 2 に合致するクライアントをグループ A として定義しています。また、端末情報項目 3 が条件式 3 に合致するか、または端末情報項目 4 が条件式 4 に合致するクライアントをグループ B として定義しています。

また、グループ C は端末情報項目には無い特殊なキー「userAgent」を使って定義したグループです。この特殊キー「userAgent」を使って構成されたグループは、キャッシュ機能を使う際に使用するキャッシュグループとして扱う事ができるようになります。そのため、キャッシュ機能を使用する際のグループ定義は必ず「userAgent」だけで決定できるようにし、なおかつアクセスした端末が複数のグループに属さないようにする必要があります。キャッシュグループ指定は別紙「Apache 設定リファレンス」も参照してください。

## 6-3 条件式

条件式では完全一致、範囲指定、OR、否定を指定できます。

### (1) 完全一致

条件式に「値」のみを指定した場合、完全一致を表します。

### (2) 範囲指定

条件式を「開始値-終了値」形式で指定した場合、指定した範囲内(開始値と終了値を含みます)に含まれるかどうかを表します。開始値および終了値には 0~99999999 までの整数もしくは小数点以下第二位までの小数が指定できます。

### (3) OR

条件式を「値 1|値 2|値 3」のように「|」区切りで指定した場合、各項目のいずれかに一致するかどうかを表します。

---

#### (4) 否定

---

条件式の1文字目が「!」の場合、否定条件を表します。否定条件は条件式の先頭に1箇所のみ記述でき、条件式全体にかかります。ただし特殊キーuserAgentには否定は使用できません。

## 第7章 端末情報項目設定(terminfo.js)

---

### 7-1 概要

---

端末情報収集スクリプト(terminfo.js)で収集される項目を指定する設定ファイルです。この設定ファイルに記述された項目がクライアントから送信され、端末情報キャッシュに保存されます。

初期状態では/var/genecode/sysconf/terminfo-script-keys.csv にあります。

### 7-2 書式

---

フォーマットは CSV 形式です。

1 行目はコメント行で、2 行目以降に端末情報項目名を記述します。

#1.0.0

端末情報項目 1

端末情報項目 2

端末情報項目 3

:

端末情報項目: 端末情報取得スクリプト(terminfo.js)で取得する端末情報項目のキーを指定します。端末情報項目名に使用可能な文字は半角英数字で、記号は使用できません。端末情報項目名は 32 文字以内で指定します。

### 7-3 注意事項

---

端末情報項目設定の変更は、項目の追加のみがサポートされています。また、項目の追加はリストの一番最後に追加する必要があります。

既に設定済みの項目を削除したり、順序を変更したり、項目名を変更したりすることはできません。

## 第8章 IP アドレス帯域リスト(specific-iplist.csv)

---

### 8-1 概要

---

IP アドレス帯域をリストアップしたファイルです。このファイルは画像変換処理で最適画像を出力するための判定条件に使用されます。

初期状態では/var/genecode/sysconf/specific-iplist.csv にあります。

### 8-2 書式

---

フォーマットは CSV 形式です。

1 行目はコメント行で、2 行目が項目名を表す行です。3 行目以降に IP アドレス帯域を記述します。

```
#1.0.0,  
network,signature  
IP アドレス帯域,ネットワーク名称  
:
```

### 8-3 更新データについて

---

本ファイルの更新データは、genecode の開発者サイトよりダウンロードができます。

## 第9章 端末情報収集スクリプト(terminfo.js)

---

端末情報を収集するための JavaScript です。このスクリプトはブラウザ上で実行されます。

初期状態では/var/genecode/sysjs/terminfo.js にあります。

### 9-1 概要

---

端末情報スクリプトはテンプレートによるコンテンツ変換の処理中に HTML 内へインラインで埋め込まれます。端末情報収集スクリプトを読み込む script タグをテンプレート内に記述する必要はありません。

### 9-2 書式

---

フォーマットは JavaScript コードです。キャラクタエンコーディングは UTF-8 固定です。

JavaScript コード内では、取得した端末情報をオブジェクト形式で返す関数 `gcterm.createTerminfo` を定義します。オブジェクトのプロパティ名には端末情報項目名を設定します。

```
/*1.0.0*/  
gcterm.createTerminfo = function() {  
    var browserType = "android";  
  
    var t = new Object();  
    t['browserType'] = browserType;  
  
    return t;  
};
```

### 9-3 注意事項

---

端末情報項目名に使用可能な文字は半角英数字のみで、記号は使用できません。端末情報項目名は 32 文字以内で指定します。

terminfo.js はクライアント上で実行される JavaScript のため、gcutil や gclog などの関数を使用することはできません。