

SYMETRIC



GENECODE

ジーンコード ver 2

ユーザマニュアル

---

サーバーインストールマニュアル

第 5 版

最終更新日 2018/12/20

## 目次

第 1 章	ライブラリのインストール.....	6
第 2 章	Apache のインストール.....	8
2-1	パッケージインストール.....	8
2-2	ソースビルド(2.4 系).....	8
2-3	ソースビルド(2.2 系).....	8
第 3 章	ジーンコードのインストール.....	10
第 4 章	ImageMagick のインストール.....	13
4-1	パッケージインストール(OS 提供パッケージ利用).....	13
4-2	パッケージインストール(同梱パッケージ利用).....	13
4-3	ソースビルド.....	14
第 5 章	Apache の設定.....	16
第 6 章	ジーンコード設定ファイルの編集.....	19
第 7 章	Apache を起動.....	20
第 8 章	動作確認コンテンツをインストール.....	22
8-1	変換元 Web サーバーの設定.....	22
(1)	任意の Web サーバーを利用する場合.....	22
(2)	簡易ローカル Web サーバーを利用する場合.....	22
8-2	ジーンコードの設定.....	22
8-3	動作確認手順.....	23

## 改訂履歴

版数	発行日	改訂内容
第 1 版	2014 年 11 月 28 日	・初版発行
第 2 版	2016 年 1 月 27 日	<ul style="list-style-type: none"> <li>・第 8 章のチェックコンテンツの URL を全て <a href="http://check2.genecode.jp/">http://check2.genecode.jp/</a> に統一</li> <li>・第 2 章に Apache 2.4 を使ったインストールを追加</li> <li>・第 4 章に OS 付属の ImageMagick の対応を追加</li> <li>・第 5 章にプロセスの上限についてを追記</li> </ul>
第 3 版	2016 年 5 月 16 日	<ul style="list-style-type: none"> <li>・第 8 章のチェックコンテンツの URL を全て <a href="http://check2.genecode.jp/">http://check2.genecode.jp/</a> に統一</li> </ul>
第 4 版	2017 年 12 月 26 日	<ul style="list-style-type: none"> <li>・第 1 章 libpng12 のインストールを追加</li> <li>・第 2 章 Apache 2.4 用の設定ファイルについて記述を追加 Apache 2.4 のソースビルド例を変更</li> <li>・第 4 章 yum による ImageMagick インストール手順を追加</li> <li>・第 5 章 mod_filter の読み込み設定を追加 共有ライブラリパスの設定内容を変更</li> </ul>
第 5 版	2018 年 12 月 20 日	<ul style="list-style-type: none"> <li>・標準インストール手順をパッケージ版 Apache と ImageMagick を使用する方式に変更</li> <li>・第 3 章 システムライブラリパスを変更する手順を削除</li> <li>・第 8 章 変換元 Web サーバーとして簡易ローカル Web サーバーを利用する手順を追加</li> </ul>

## インストール前の注意事項

ジーンコードサーバーのインストールを行うには以下 3 つのパッケージが必要となりますので、事前にご準備ください。

- ジーンコード ランタイムパッケージ(ex. genecode-runtime-standard-3.0.0\_x86\_64-72.zip)
- ジーンコード ライブラリパッケージ(ex. genecode-library-2.3.0-42.zip)

※使用するバージョンによって、ファイル名の一部が異なります

このインストール手順では、以下のディレクトリにインストールする例を記載しています。下記以外にインストールをする場合は、設定ファイルの編集が必要になります。

- Apache: パッケージ版の利用
- ImageMagick: パッケージ版の利用
- アプリケーション: /home/genecode

また、ジーンコードランタイムでは以下のライブラリを使用し、システムライブラリパスに設定しています。これらのライブラリの別バージョンをジーンコード導入環境で使用する場合は、以下のライブラリをシステムライブラリのパスに設定しないようにし、別途設定を行う必要があります。詳しくは、第 3 章 ジーンコードのインストール、第 5 章 Apache の設定を参照してください

- Mozilla Firefox 3.5.19
- Google V8 JavaScript Engine 3.17.16
- Apache Xerces-C++ 3.1.1

※SELinux が有効な場合、無効化するか SELinux の適切な設定が必要です。

※インストール作業はサーバー上の root 権限で行ってください。

FAQ や Tips については、GeneCode Developer Connection (<http://developer.genecode.jp/>)にて公開していますので、必要に応じて参照してください。

## 第1章 ライブラリのインストール

---

ジーンコードランタイムの動作には以下の各種ライブラリが必要になります。

※Red Hat Enterprise Linux/Cent OS のインストールメディアに含まれています。もしくは、yum コマンドを利用してインターネットから取得・更新/インストールが可能です。

- alsa-lib
- atk
- cairo
- fontconfig
- freetype
- gtk2
- libICE
- libSM
- libX11
- libXau
- libXrender
- libXt
- libjpeg-turbo (※Red Hat Enterprise Linux 5/Cent OS 5 では libjpeg)
- libpng
- libpng12 (※Red Hat Enterprise Linux 6/Cent OS 6 では不要)
- nss
- openssl098e (※Red Hat Enterprise Linux 5/Cent OS 5 では openssl)
- pango

- `zlib-devel`

1. 以下のコマンドで各ライブラリのインストールを行います。

■ Red Hat Enterprise Linux 7/CentOS 7 の場合

```
# yum install alsa-lib libXt libXrender freetype fontconfig pango gtk2 libjpeg-turbo libpng libpng12  
openssl098e zlib-devel
```

■ Red Hat Enterprise Linux 6/CentOS 6 の場合

```
# yum install alsa-lib libXt libXrender freetype fontconfig pango gtk2 libjpeg-turbo libpng  
openssl098e zlib-devel
```

■ Red Hat Enterprise Linux 5/CentOS 5 の場合

```
# yum install alsa-lib libXt libXrender freetype fontconfig pango gtk2 libjpeg libpng zlib-devel
```

2. (ImageMagick をソースからビルドする場合のみ) 以下のライブラリをインストールします。

• `libjpeg-turbo-devel` (Red Hat Enterprise Linux 5/CentOS 5 の場合は `libjpeg-devel`)

• `libpng12-devel` (Red Hat Enterprise Linux 6/CentOS 6 の場合は `libpng-devel`)

3. (Apache をソースからビルドする場合のみ) 以下のライブラリをインストールします。

• `openssl-devel`

## 第2章 Apache のインストール

---

Apache 2.4/2.2 を prefork MPM でインストールします(※worker には対応していません)。

### 2-1 パッケージインストール

---

Apache をパッケージからインストールする場合は、Apache モジュール mod\_ssl(※1)も含めてインストールします。

yum コマンド例:

```
# yum install httpd mod_ssl
```

※1 mod\_ssl は SSL を有効化する場合に必要です。https コンテンツを変換する場合も必要です。

### 2-2 ソースビルド(2.4 系)

---

Apache 2.4 系をソースからビルドする場合は、apr のバージョン 1.4 系、apr-util のバージョン 1.5 系を使用するようにビルドします。

以下は Apache 内に apr、apr-util を組み込む場合の configure オプションの例となります。

--with-included-apr オプションを使用する際は、「Apache ソースディレクトリ/src/lib」ディレクトリ内に apr、apr-util のソースを配置する必要があります。

configure オプション例:

```
# ./configure --enable-so --enable-mods-shared=all --with-mpm=prefork --with-included-apr  
--prefix=/usr/local/apache
```

### 2-3 ソースビルド(2.2 系)

---

Apache 2.2 系をソースからビルドする際、以下の configure オプションを必ず指定してください。

- --enable-so
- --enable-proxy
- --enable-proxy-http



- --enable-headers
- --enable-rewrite
- --enable-deflate
- --enable-ssl(※1)

※1 SSL を使用する場合のみ必要です。

configure オプション例:

```
# ./configure --enable-so --enable-proxy --enable-proxy-http --enable-headers --enable-rewrite  
--enable-deflate --enable-ssl --prefix=/usr/local/apache
```

## 第3章 ジーンコードのインストール

1. ジーンコードのインストールを行う前に Apache が停止していることを確認します。
2. runtime パッケージ内の `product/apache/modules/<APACHE バージョン>` ディレクトリにある以下の `so` ファイルを `/etc/httpd/modules` ディレクトリにコピーします。
  - `mod_gc_css.so` ... スタイルシート変換モジュール
  - `mod_gc_html.so` ... 言語変換モジュール
  - `mod_gc_img.so` ... 画像変換モジュール
  - `mod_gc_js.so` ... JavaScript 変換モジュール

※Apache のバージョンを取り違えにご注意ください。  
 ※Red Hat Enterprise Linux 7/CentOS 7 のパッケージ版は Apache 2.4、Red Hat Enterprise Linux 6 以下/CentOS 7 以下のパッケージ版は Apache 2.2 です。
3. runtime パッケージ内の `product/apache/conf/<APACHE バージョン>` ディレクトリにある以下のファイルを `/etc/httpd/conf` ディレクトリにコピーします。
  - `httpd-genecode.conf` ... ジーンコード用 Apache 設定ファイル
  - `httpd-genecode-vhost.conf` ... ジーンコード用 Apache 設定ファイル (virtualhost 用)

※Apache のバージョンを取り違えにご注意ください。  
 ※Red Hat Enterprise Linux 7/CentOS 7 のパッケージ版は Apache 2.4、Red Hat Enterprise Linux 6 以下/CentOS 7 以下のパッケージ版は Apache 2.2 です。
4. runtime パッケージ内に同梱されている `product/var/genecode.tgz` を `/var` ディレクトリに展開します。  
 展開例:  

```
# tar -C /var -zxvf genecode.tgz
```
5. 展開後、下記のディレクトリとファイルが配置されていることを確認します。
  - `/var/genecode/cache` ... キャッシュディレクトリ (ディレクトリのみ)
  - `/var/genecode/sysconf`
    - `/css-conversion.csv`
    - `/group.gcl`
    - `/specific-iplist.csv`
    - `/terminfo-script-keys.conf`
  - `/var/genecode/sysjs` ... ジーンコードランタイム用 JS ディレクトリ
    - `/gcparts.js.bin` ... GC パーツランタイム

/jquery-1.7.2.gc.js ... ジーンコードランタイム用 jQuery

/terminfo.js ... 端末情報クライアント JS

/terminfo ... 端末情報拡張用スクリプトディレクトリ

- /var/genecode/lib ... ライブラリディレクトリ

/firefox-3.5.19-gc

/v8-3.17.16

/xerces-c-3.1.1

- /var/genecode/temp ... 一時ディレクトリ(ディレクトリのみ)

- /var/genecode/work ... 作業ディレクトリ(ディレクトリのみ)

※ 各種インストールを手順で指定された場所以外に配置した場合は設定ファイルの編集が必要です。

※ /var/genecode/lib ディレクトリの場所を変更できません。

6. 別途提供されたライセンスファイル(gc.license)を /var/genecode/sysconf ディレクトリにコピーします。

7. genecode ユーザを作成し、権限を設定します。

```
# useradd genecode
```

```
# passwd genecode ...パスワードを設定します
```

```
# chmod 755 /home/genecode
```

8. 以下の操作は genecode ユーザーに変更して行います。

9. runtime パッケージ内の product/apl ディレクトリにある以下の各ディレクトリを /home/genecode ディレクトリにコピーします。

• conf

• appjs

• template

コピー例:

```
$ cp -R product/apl/* /home/genecode/
```

10. library パッケージ内の appjs/gcutils.js ファイルを /home/genecode/appjs/ 以下にコピーします。

```
$ cp appjs/gcutils.js /home/genecode/appjs/
```

11. library パッケージ内の gcparts\_server ディレクトリ内にある tgz ファイルを /home/genecode/appjs/gcparts/ に展開します。

展開例:

```
$ find gcparts_server -name "*.tgz" -exec tar -C /home/genecode/appjs/gcparts -zxvf {} \;
```

12. library パッケージ内の template ディレクトリ内にある gcparts ディレクトリを /home/genecode/template/ 以下にコピーします。

```
$ cp -R template/gcparts/ /home/genecode/template/
```

13. library パッケージ内の css ディレクトリを/home/genecode/template/以下にコピーします

```
$cp -R template/css/ /home/genecode/template/
```

## 第4章 ImageMagick のインストール

---

ImageMagick を RPM パッケージもしくはソースからインストールします。

### 4-1 パッケージインストール(OS 提供パッケージ利用)

---

OS のインストールパッケージを利用する場合は、yum コマンドでインストールします。

インストールコマンド例:

```
# yum install ImageMagick
```

### 4-2 パッケージインストール(同梱パッケージ利用)

---

runtime パッケージに同梱されているパッケージからインストールする場合は、以下の rpm をインストールします。

- ImageMagick-6.3.9-10.roundabout.x86\_64.rpm

1. 既に ImageMagick が RPM でインストールされている場合、削除します。

確認例:

```
# rpm -qa | grep ImageMagick
```

削除例:

```
# rpm -e `rpm -qa | grep ImageMagick`
```

2. RPM により ImageMagick 6.3.9-10 をインストールします。

インストールコマンド例:

```
# rpm -ivh ImageMagick-6.3.9-10.roundabout.x86_64.rpm
```

※環境によっては rpm でインストールを行うと正常動作しない場合がございます。その場合は、ソースからビルドしてインストールしてください。

## 4-3 ソースビルド

ImageMagick をソースからビルドしてインストールします。RPM によるインストールが完了している場合は、この手順は不要です。

1. 既に ImageMagick がパッケージインストールされている場合、ImageMagick をアンインストールします。

※インストールされている ImageMagick を別用途でご利用中の場合は、ImageMagick を削除する必要はありません。

確認例:

```
# rpm -qa | grep ImageMagick
```

削除例:

```
# rpm -e `rpm -qa | grep ImageMagick`
```

2. 下記ライブラリをインストールします。

- libjpeg-turbo: 1.2.1 以降
- libpng: 1.2.7 以降
- libjpeg-turbo-devel: 1.2.1 以降(※1)
- libpng-devel: 1.2.7 以降(※2)

インストールコマンド例:

```
# yum install libjpeg-turbo libpng libjpeg-turbo-devel libpng-devel
```

※1 libjpeg-turbo-devel は libjpeg-turbo と同一バージョンをインストールしてください。

※2 libpng-devel は libpng と同一バージョンをインストールしてください。

3. ImageMagick 6.3.9-10 のソースをダウンロードします。

※弊社が安定動作を確認している ImageMagick のバージョンは、6.3.9-10 と 6.5.6-10、OS 付属の ImageMagick です。ただし、より優れたパフォーマンス動作を確認している 6.3.9-10 の使用を推奨しております。

4. ソースアーカイブを任意のディレクトリで解凍します。

解凍例:

```
# cd /usr/local/src
```

```
# tar -zxvf ImageMagick-6.3.9-10.tar.gz
```

5. ソースビルドを行います。

ビルド例:

```
# cd /usr/local/src/ImageMagick
# ./configure --prefix=/usr/local/ImageMagick --without-magick-plus-plus --without-bzlib
--without-dps --without-fpx --without-gslib --without-jbig --without-tiff --without-wmf
--without-fontconfig --without-freetype --without-lcms --without-gvc --without-openexr
--without-rsvg --without-xml --with-perl=no --with-x=no
```

※ImageMagick をソースからビルドしてインストールする場合、httpd-genecode.conf の変更が必要になります。詳細は 0 を参照ください。

## 第5章 Apache の設定

### 【既存の Apache を使用する場合の注意事項】

既にセットアップされた Apache にジーンコードをインストールする場合、PHP やフィルタ、その他モジュールを必ず無効化してください。設定内容によっては、ジーンコードが動作しない可能性があります。

1. mod\_proxy、mod\_proxy\_http、mod\_filter(※1)、mod\_deflate、mod\_headers、mod\_rewrite、mod\_ssl モジュールをロードする記述を行います。

記述例:

```
LoadModule filter_module modules/mod_filter.so
LoadModule deflate_module modules/mod_deflate.so
LoadModule headers_module modules/mod_headers.so
LoadModule rewrite_module modules/mod_rewrite.so
LoadModule proxy_module modules/mod_proxy.so
LoadModule proxy_http_module modules/mod_proxy_http.so
LoadModule ssl_module modules/mod_ssl.so
```

※1 mod\_filter は Apache 2.4 のみ必要です。

※2 パッケージ版 Apache では全て初期設定で有効化されています。

2. /etc/httpd/conf/httpd.conf ファイルの末尾に、ジーンコード設定ファイル (httpd-gencode.conf) をインクルードする記述を追加します。

記述例:

```
Include conf/httpd-gencode.conf
```

※設定ファイル httpd-gencode.conf は<VirtualHost>内では Include できません。

3. ジーンコードを有効にする<VirtualHost>ディレクティブ内で httpd-gencode-vhost.conf ファイルをインクルードします。

設定例:

```
<VirtualHost *:80>
    ServerName sp.example.com
    Include conf/httpd-gencode-vhost.conf
</VirtualHost>
```

※ httpd-gencode-vhost.conf を Include するより前に httpd-gencode.conf を Include してください。httpd-gencode-vhost.conf を Include した後で httpd-gencode.conf を Include した場合、設定エラーとなります。



4. MaxRequestPerChild ディレクティブの値を 4000 (推奨値) 以下に設定します。

■パッケージ版 Apache 2.4 の場合

/etc/httpd/conf.d/mpm.conf を作成します。

設定例:

```
<IfModule mpm_prefork_module>
    StartServers      5
    MinSpareServers   5
    MaxSpareServers   10
    ServerLimit       150
    MaxClients        150
    MaxRequestsPerChild 4000
</IfModule>
```

■ソースビルド版 Apache の場合

<APACHE\_HOME>/conf/extra/httpd-mpm.conf を開き、MaxRequestPerChild ディレクティブの値を変更します。また、<APACHE\_HOME>/conf/httpd.conf で httpd-mpm.conf を Include します。

設定例:

```
<IfModule mpm_prefork_module>
    StartServers      5
    MinSpareServers   5
    MaxSpareServers   10
    MaxClients        150
    MaxRequestsPerChild 4000
</IfModule>
```

※MaxRequestPerChild を 0 に設定することは避けてください。

※ジーンコードでは、1 プロセスあたり 11 個のスレッドを使用します。想定する最大プロセス数によっては、Apache ユーザーのプロセス・スレッド数の上限に達し、子プロセスが生成できなくなります。その場合は、Apache ユーザーのプロセス・スレッド数の上限を上げてください。

5. 共有ライブラリと環境変数の設定を行います。

■パッケージ版 Apache の場合

/etc/sysconfig/httpd で以下のように LD\_LIBRARY\_PATH と HOME 環境変数を設定します。

```
LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/lib64:/var/genecode/lib/firefox-3.5.19-gc/lib/firefox-  
3.5.19/./var/genecode/lib/v8-3.17.16/./var/genecode/lib/xerces-c-3.1.1/lib/  
export LD_LIBRARY_PATH  
HOME=/root  
export HOME
```

※LD\_LIBRARY\_PATH 行は改行せず記述してください。

## ■ソースビルド版 Apache の場合

<APACHE\_HOME>/bin/envvars で以下のように LD\_LIBRARY\_PATH を設定します。

```
LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/lib64:/var/genecode/lib/firefox-3.5.19-gc/lib/firefox-  
3.5.19/./var/genecode/lib/v8-3.17.16/./var/genecode/lib/xerces-c-3.1.1/lib/  
.....  
export LD_LIBRARY_PATH  
HOME=/root  
export HOME
```

※LD\_LIBRARY\_PATH 行は改行せず記述してください。

## 第6章 ジーンコード設定ファイルの編集

---

1. 変換元サイトへの Proxy 設定を行います。/etc/httpd/conf/httpd-gencode-vhost.conf を開き、ProxyPass ディレクティブのコメントを外し、Proxy 先のホスト名を指定します。

設定例: Proxy 先の URL が http://www.example.com/ の場合

```
ProxyPass / http://www.example.com/
```

**※変換元サイトへの接続が HTTPS プロトコルの場合、SSLProxyEngine On も指定する必要があります。**

2. (ImageMagick をソースビルドした場合のみ) ImageMagick のインストールディレクトリを /etc/httpd/conf/httpd-gencode.conf で指定します。
3. (アプリケーションディレクトリを変更した場合のみ) /home/gencode ディレクトリ以外にアプリケーションディレクトリを配置した場合は、/etc/httpd/conf/httpd-gencode-vhost.conf を修正します。適切なディレクトリ、ファイルパスに設定し直してください。

## 第7章 Apache を起動

1. 設定を確認するため、`configtest` コマンドを実行します。

確認例:

```
# apachectl configtest
```

2. “Syntax OK” と出力されることを確認します。

※ “undefined symbol: sqlite3\_initialize” のようなエラーメッセージが表示される場合は、`httpd-gencode.conf` の先頭に以下を追加してください。

```
LoadFile /var/genencode/lib/firefox-3.5.19-gc/lib/firefox-3.5.19/libsqlite3.so
```

※ “Invalid command 'GCHtmlTemplate'” のようなエラーメッセージが表示される場合は、`httpd-gencode-vhost.conf` を Include する前に `httpd-gencode.conf` を Include しているかどうかを確認してください。

※ “Invalid command 'ProxyPass'” のようなエラーメッセージが表示される場合は、`mod_proxy` をロードしているかどうかを確認してください(第 5 章の手順 1)。

※ “undefined symbol:

```
_ZN2v816FunctionTemplate3NewEPFNS_6HandleINS_5ValueEEERKNS_9ArgumentsEES3_NS1_INS_9SignatureEEEEi” のようなエラーメッセージが表示される場合は、httpd-gencode.conf の先頭に以下を追加してください。
```

```
LoadFile /var/genencode/lib/v8-3.17.16/libv8.so
```

3. Apache を起動します。

起動例:

```
# systemctl start httpd
```

※ ジーンコードのモジュール新規追加時、更新時は `restart` での再起動は行わないでください。

※ `service` コマンドによる Apache 起動が正しく行えない場合、環境変数 `HOME` が正しく設定されているかどうかを確認してください(0 の手順 5)。

4. Apache の `/var/log/httpd/error_log` に以下のようなログが出力されていることを確認します。

```
[Thu Nov 08 11:39:33 2012] [notice] gc_html:mod_gc_html standard edition ver 2.0.0 build 99
```

```
[Thu Nov 08 11:39:33 2012] [notice] gc_html:----- license information -----
```

```
[Thu Nov 08 11:39:33 2012] [notice] gc_html:license id : GC2S-0000-0000-0000
```

```
[Thu Nov 08 11:39:33 2012] [notice] gc_html:product edition: GeneCode standard edition
[Thu Nov 08 11:39:33 2012] [notice] gc_html:product type      : standard
[Thu Nov 08 11:39:33 2012] [notice] gc_html:-----
[Thu Nov 08 11:39:33 2012] [notice] gc_html:configuration end
[Thu Nov 08 11:39:33 2012] [notice] gc_img:mod_gc_img standard edition ver 2.0.0 build 99
[Thu Nov 08 11:39:33 2012] [notice] gc_img:----- license  information -----
[Thu Nov 08 11:39:33 2012] [notice] gc_img:license id        : GC2S-0000-0000-0000
[Thu Nov 08 11:39:33 2012] [notice] gc_img:product edition: GeneCode standard edition
[Thu Nov 08 11:39:33 2012] [notice] gc_img:product type      : standard
[Thu Nov 08 11:39:33 2012] [notice] gc_img:-----
[Thu Nov 08 11:39:33 2012] [notice] gc_img:configuration end
[Thu Nov 08 11:39:33 2012] [notice] gc_css:mod_gc_css standard edition ver 2.0.0 build 99
[Thu Nov 08 11:39:33 2012] [notice] gc_css:----- license  information -----
[Thu Nov 08 11:39:33 2012] [notice] gc_css:license id        : GC2S-0000-0000-0000
[Thu Nov 08 11:39:33 2012] [notice] gc_css:product edition: GeneCode standard edition
[Thu Nov 08 11:39:33 2012] [notice] gc_css:product type      : standard
[Thu Nov 08 11:39:33 2012] [notice] gc_css:-----
[Thu Nov 08 11:39:33 2012] [notice] gc_css:configuration end
[Thu Nov 08 11:39:33 2012] [notice] gc_js:mod_gc_js standard edition ver 2.0.0 build 99
[Thu Nov 08 11:39:33 2012] [notice] gc_js:----- license  information -----
[Thu Nov 08 11:39:33 2012] [notice] gc_js:license id        : GC2S-0000-0000-0000
[Thu Nov 08 11:39:33 2012] [notice] gc_js:product edition: GeneCode standard edition
[Thu Nov 08 11:39:33 2012] [notice] gc_js:product type      : standard
[Thu Nov 08 11:39:33 2012] [notice] gc_js:-----
[Thu Nov 08 11:39:33 2012] [notice] gc_js:configuration end
[Thu Nov 08 11:39:33 2012] [notice] Apache/2.2.23 (Unix) mod_ssl/2.2.23 OpenSSL/1.0.0-fips
mod_gc_html/1.0.0 mod_gc_img/1.0.0 mod_gc_css/1.0.0 mod_gc_js/1.0.0 configured -- resuming
normal operations
```

※バージョン情報やライセンス情報はご利用のモジュールやライセンスによって異なります。

※サーバーが起動しない場合の対処方法については、GeneCode Developer Connection  
(<http://developer.genecode.jp/>)内の FAQ を参照してください。

## 第8章 動作確認コンテンツをインストール

---

### 8-1 変換元 Web サーバーの設定

---

動作確認用の変換元コンテンツは `http://check2.genecode.jp/check/` に配置されています。

ファイアウォールなどにより、Apache から上記 URL への外部 HTTP 通信ができない場合は、任意の Web サーバーに動作確認用の変換元コンテンツを配置します。

#### (1) 任意の Web サーバーを利用する場合

---

runtime パッケージで `check/htdocs` ディレクトリ内のファイルを、Web サーバーのドキュメントルートに配置してください。

コピー例:

```
# cp -pR check/htdocs/* /var/www/html/
```

#### (2) 簡易ローカル Web サーバーを利用する場合

---

外部 Web サーバーの準備ができない場合は、PHP や Python の簡易 Web サーバーを利用する方法があります。`check/htdocs` ディレクトリに移動し、以下のコマンドを実行すると、ローカルで Web サーバーが起動します。

PHP コマンド例:

```
$ php -S localhost:8080
```

Python コマンド例:

```
$ python -m SimpleHTTPServer 8080
```

### 8-2 ジーンコードの設定

---

1. ジーンコードの設定を動作確認コンテンツの設定に置き換えます。runtime パッケージの `check/conf` ディレクトリ、`check/appjs` ディレクトリ、`check/template` ディレクトリをアプリケーションディレクトリにコピーします。

コピー例:

```
$ cp -pR check/conf/* /home/genecode/conf/
```

```
$ cp -pR check/appjs/* /home/genecode/appjs/  
$ cp -pR check/template/* /home/genecode/template/
```

2. ジーンコードサーバー上で、Apache の Proxy 設定を変更します。変換元コンテンツを配信する Web サーバーへと Proxy されるように/etc/httpd/conf/httpd-genecode-vhost.conf を変更します。

設定例:

```
ProxyPass / http://check2.genecode.jp/
```

```
ProxyPassReverse / http://check2.genecode.jp/
```

※簡易ローカル Web サーバーの場合、URL は http://localhost:8080/ となります。

3. Apache を再起動します。

コマンド例:

```
# systemctl restart httpd
```

## 8-3 動作確認手順

---

ブラウザで以下の URL にアクセスして動作を確認します。

http://<ジーンコードサーバーのホスト名>/check/

※サーバーが起動しない場合や正しく動作しない場合は、Apache のアクセスログ (/var/log/httpd/access\_log) および Apache のエラーログ (/var/log/httpd/error\_log) を参照してください。

※変換が行われない場合は、ブラウザキャッシュを削除してから再度アクセスしてみてください。

※トラブルシューティングについては、GeneCode Developer Connection (http://developer.genecode.jp/) 内の FAQ を参照してください。