

ELIXIR MOBILE PRODUCTS



エリクサー・ユニバーサル

ユーザーズ・マニュアル

Revision: 1.06

ノイアンドコンピューティング株式会社

目次

1 はじめに	6
2 インストールの必要条件	7
3 インストールの手順	9
4 インストール後の設定	10
4.1 実行ユーザ作成.....	10
4.2 各設定格納用ディレクトリ作成.....	10
4.3 アクセスログ用ディレクトリ作成.....	10
4.4 elixir.conf コピー.....	10
4.5 ライセンスコード入力.....	10
4.6 シリアルナンバー入力.....	10
4.7 起動スクリプトの設定.....	11
5 確認	12
5.1 ディレクトリの確認.....	12
5.2 パッケージの確認.....	12
5.3 インストール確認.....	12
5.4 elx ユーザと elx グループの確認.....	12
5.5 elixir.conf の確認.....	12
5.6 起動スクリプト.....	13
6 サイト設定手順	14
6.1 nginx.conf ファイル(基本設定)を作成.....	14
6.2 バーチャルホスト名.vconf ファイルを作成.....	14
6.2.1 通常サイトの場合(SSL を使用しないサイトの場合).....	14
6.2.2 SSL サイトの場合.....	14
6.2.3 補足.....	14
6.3 バーチャルホスト名.hconf ファイルを作成.....	15
6.4 SSL 秘密鍵、証明書の設定(SSL サイトの場合のみ).....	15
6.5 サイトを追加する場合.....	17
6.6 作成ファイルの確認.....	17
7 その他の設定	18
7.1 設定の基本.....	18
7.2 設定例.....	19
7.3 HTTP ヘッダの編集.....	19
8 起動	21
9 停止	22

10	自動アップデート	23
10.1	設定方法.....	23
10.2	アップデート結果確認方法.....	23
11	手動アップデート	25
11.1	手順.....	25
11.2	アップデート結果確認方法.....	25
12	ダウングレード	26
12.1	バックアップ手順.....	26
12.2	リストア手順.....	26
13	ログローテーションの設定	29
13.1	ログの説明.....	29
13.2	設定方法.....	29
14	変換対象コンテンツの仕様	30
14.1	マークアップ言語.....	30
14.2	文字コード.....	30
14.3	絵文字.....	30
14.4	CSS.....	30
14.5	画像.....	31
15	画像変換パラメータ	32
15.1	出力自動調整方法.....	32
15.2	パラメータ指定内容.....	32
15.3	画像の横幅指定.....	33
15.4	画像のファイルサイズ指定.....	34
15.5	待ち受け画像指定.....	34
15.6	待ち受け画像の余白背景色指定.....	34
15.7	画像の無変換指定.....	35
15.8	パラメータの名前指定.....	35
15.9	パラメータを付加しない指定.....	36
16	画像コピー転送防止	37
16.1	転送防止の指定方法.....	37
17	端末スペック通知	38
17.1	通知する項目.....	38
17.2	スペックの名称変更.....	38
18	キャッシュ制御	39
18.1	キャッシュ機構.....	39
18.2	キャッシュを行いたい場合.....	39

18.3	キャッシュをさせない場合	39
18.4	キャッシュ時間を指定する場合	39
18.4.1	変換後コンテンツ(メモリキャッシュ)のキャッシュ時間指定	40
18.4.2	変換元コンテンツ(ディスクキャッシュ)のキャッシュ時間指定	40
18.5	キャッシュの削除	41
19	未対応端末の動作	42
19.1	対応している端末	42
19.2	対応していない端末	42
19.3	未知の端末 (登録されていない端末)	42
19.4	携帯電話向け検索エンジンのクローラ	42
19.5	PC (その他の端末)	42
20	PC ブラウザでの動作確認方法	43
20.1	確認ツール	43
20.2	変換後コンテンツの確認	43
21	その他コンテンツ作成時の注意事項	45
21.1	POST データの文字コード	45
21.2	CSS パースエラー時の対応	45
22	独自の変換ルール	47
22.1	変換ルールの作成方法	47
22.2	変換ルールの設定方法	47
23	レイアウト変換	48
23.1	ブロック指定方法の種類	48
23.2	特定の属性へのメソッド指定	48
23.3	メソッドごとの XPath 指定	49
23.4	ベーシック変換機能 OFF	50
23.4.1	テーブル変換、object タグ変換 OFF	50
23.4.2	その他のベーシック変換機能 OFF	51
23.5	ベーシック変換の内容	51
23.6	レイアウト変換の内容	55
24	ページ分割	65
24.1	詳細内容	65
24.2	各種設定	65
25	出力切替	67
25.1	出力切替の記述方法	67
25.2	条件式の規則	67
25.3	条件式の例	68

付録 A. 独自の用語.....	70
付録 B. 参考資料.....	71
変更履歴	73

1 はじめに

この度は、弊社のエリクサーをお使い頂きまして誠にありがとうございます。

エリクサーはベース部分のプロキシサーバに nginx 、変換後コンテンツのキャッシュに memcached を使用しています。

この nginx と memcached がデーモンとして動作することで、コンテンツ変換、キャッシュ機能を実現しています。

※nginx の利用可能なモジュールは [よくあるご質問](#) を参照して下さい。

2 インストールの必要条件

必要機器

Linux サーバ、推奨ハード性能を満たしている物

必要ディレクトリ

/usr/local

インストールディレクトリのルートです。

/usr/local/etc/elx/nginx

nginx の設定ファイル用のディレクトリです。

/usr/local/etc/elx/conf

elixir の設定ファイル用のディレクトリです。

/var/elixir/cache

ディスクキャッシュ用のディレクトリです。

コンテンツサーバから取得した変換元コンテンツのキャッシュディレクトリです。

/var/log/httpd

アクセスログ用のディレクトリです。

エリクサーのアクセスログ、エラーログファイルのディレクトリです。

必要ユーザ ※

エリクサーの実行ユーザ elx をグループ elx で作成してください。

※後ほど [\[インストール後の設定\]](#) の中で作成します。

必要ディスク容量

50Mbyte

インストールのみに必要な容量です。

キャッシュ、ログ領域の容量とは別です。

キャッシュ、ログ領域の容量はお客様の使用状況によって異なります。

必要パッケージ

libgomp

通常インストールされた場合、上記パッケージがインストールされていないと思われるので、インストールを行ってください。パッケージは OS のインストールディスクに含まれています。(RedHat, CentOS の場合)

例 : CentOS 5.3 の場合

```
rpm -ivh libgomp-4.1.2-42.el5.i386.rpm
```


3 インストールの手順

CD-ROM をセットしてマウントしてください。

```
mount /dev/cdrom /mnt/cdrom
```

※デバイス名 (/dev/cdrom) やマウントポイント (/mnt/cdrom) はお使いの環境によって異なります。適宜、環境に合わせて変更してください。

マウントした CD-ROM のルートディレクトリに移動し、`elx_install.sh` を実行してください。

```
cd /mnt/cdrom
./elx_install.sh
```

インストール先は `/usr/local/elx` です。`/usr/local` に書き込み権限があるユーザでインストールを実行してください。(通常環境ではユーザ `root` での実行です。)

“install was completed” といったメッセージが出力されれば、インストールは完了です。(通常、数秒程度で終了します。)

`/usr/local/elx` はインストールしたエリクサーのパッケージを展開したディレクトリへのシンボリックリンクになります。

例 : `elx-1.000.pkg` をインストールした場合

```
/usr/local/elx -> /usr/local/elx-1.000
/usr/local/elx-1.000
```

4 インストール後の設定

4.1 実行ユーザ作成

エリクサーの実行ユーザ `elx` をグループ `elx` で作成してください。

例 : CentOS、RedHat の場合

```
/usr/sbin/useradd -s /bin/false elx
```

4.2 各設定格納用ディレクトリ作成

以下の各設定格納用ディレクトリをユーザ `root` で作成してください。

```
/usr/local/etc/elx/conf  
/usr/local/etc/elx/nginx  
/usr/local/etc/elx/nginx/vhosts
```

4.3 アクセスログ用ディレクトリ作成

以下のコンテンツキャッシュ、アクセスログ用ディレクトリをユーザ `root` で作成してください。(`/var/log/httpd` はすでに存在する場合があります。)

```
/var/log/httpd  
/var/elixir/cache
```

4.4 `elixir.conf` コピー

CD-ROM のルートディレクトリにある、`elixir.conf` を `/usr/local/etc/elx/conf` にコピーしてください。(通常的环境下ではユーザ `root` での実行です。)

4.5 ライセンスコード入力

ライセンスコードをコピーした `elixir.conf` に入力してください。

例 : ライセンスコードが、1234-5678-9ABC-DEFG-HIJK-LMNO の場合

```
LICENSE_CODE      1234-5678-9ABC-DEFG-HIJK-LMNO
```

※区切り文字は必ず<TAB> (Horizontal TAB) 1文字です。

4.6 シリアルナンバー入力

シリアルナンバーをコピーした `elixir.conf` に入力してください。

例：シリアルナンバーが 0123456789 の場合

```
SERIAL_NUMBER          0123456789
```

※区切り文字は必ず<TAB> (Horizontal TAB) 1文字です。

※シリアルナンバー、ライセンスコードが正しく入力されていない場合、エリクサー(nginx)の起動ができません。また、自動アップデートも行われません。

4.7 起動スクリプトの設定

お客様の環境に合わせて、/usr/local/elx/examples/start-scriptsにある起動スクリプトを参考に設定してください。

起動スクリプトのサンプルを使用する場合は、そのままコピーしてお使いください。

例：CentOS、RedHat 起動スクリプトのサンプルを使用する場合

```
cd /usr/local/elx/examples/start-scripts
cp nginx /etc/rc.d/init.d/
cp memcached /etc/rc.d/init.d/

chkconfig --add nginx
chkconfig --add memcached
```

5 確認

チェックリストでインストール作業の確認を行ってください。

5.1 ディレクトリの確認

下記ディレクトリが作成されているかどうか確認してください。

- (1) /usr/local/etc/elx/conf : elixir.conf を格納します。
- (2) /usr/local/etc/elx/nginx : nginx の設定ファイルを格納します。
- (3) /usr/local/etc/elx/nginx/vhosts : nginx のバーチャルホスト毎の設定ファイルを格納します。
- (4) /var/log/httpd : アクセスログを格納します。
- (5) /var/elixir/cache : コンテンツサーバから取得した変換元コンテンツをキャッシュするためのディレクトリです。

5.2 パッケージの確認

以下のパッケージがインストールされているかどうかを確認してください。

libgomp

5.3 インストール確認

"/usr/local/"ディレクトリに以下のようにエリクサーが展開されているか確認してください。

```
/usr/local/elx -> /usr/local/elx-X.XXX  
/usr/local/elx-X.XXX
```

※XXX はパッケージのバージョンです。

5.4 elx ユーザと elx グループの確認

以下の項目を確認してください。

- (1) "elx"グループ
- (2) グループ "elx" で作成された "elx" ユーザ

5.5 elixir.conf の確認

下記項目の確認を行ってください。

- (1) ライセンスコード

LICENSE_CODE 1234-5678-9ABC-DEFG-HIJK-LMNO

(2) シリアルナンバー

SERIAL_NUMBER 0123456789

5.6 起動スクリプト

お客様が設定された場所に “nginx” と “memcached” の起動スクリプトがあるかを確認ください。

続いて[{サイト設定手順}](#)を参照して、バーチャルHOST設定を行ってください。

※バーチャルHOST設定が行われていないと nginx は起動しません。

6 サイト設定手順

サイト設定には `/usr/local/etc/nginx/` の下にある `*.example` ファイルを元に設定ファイルを作成します。以下の手順に従ってサイトの設定を行ってください。

6.1 nginx.conf ファイル(基本設定)を作成

`nginx.conf.example` ファイル内の `example.jp` を設定したいバーチャルホスト名に変更し、`nginx.conf` というファイル名で `/usr/local/etc/nginx/` の下に置く。

6.2 バーチャルホスト名.vconf ファイルを作成

バーチャルホスト名.vconf(バーチャルホスト毎の設定ファイル)を作成します。

6.2.1 通常サイトの場合(SSLを使用しないサイトの場合)

(1) `vconf.example` ファイル内の `www.example.jp` を設定したいバックエンドのコンテンツサーバに変更する。

(2) `vconf.example` ファイル内の `example.jp` を設定したいバーチャルホスト名に変更する。

(3) (1), (2)の変更を行った`vconf.example`ファイルを`バーチャルホスト名.vconf`というファイル名で`/usr/local/etc/nginx/vhosts/`の下に置く。

6.2.2 SSL サイトの場合

(1) `ssl.vconf.example` ファイル内の `www.example.jp` を設定したいバックエンドサーバに変更する。

(2) `ssl.vconf.example` ファイル内の `example.jp` を設定したいバーチャルホスト名に変更する。

(3) `192.168.1.10`をSSLサイトが受けるIPアドレスに変更する。

(4) (1)~(3)の変更を行った`ssl.vconf.example`ファイルを`バーチャルホスト名.vconf`というファイル名で`/usr/local/etc/nginx/vhosts/`の下に置く。

6.2.3 補足

(1) コンテンツサーバへの接続先ポートを変更したい場合

コンテンツサーバ名の後に`:ポート番号`を記述してください。ポート番号を8080に設定したい場合は以下ようになります。

```
proxy_pass http://www.example.jp:8080;
```

(2) コンテンツサーバとの通信プロトコルを HTTPS に変更したい場合

以下のようにコンテンツサーバ名の通信プロトコルを“https://~”で記述してください。

```
proxy_pass https://www.example.jp;
```

6.3 バーチャルホスト名.hconf ファイルを作成

エリクサー独自の HTTP ヘッダの設定ファイル(バーチャルホスト名.hconf ファイル)を作成します。“hconf.example” ファイルを “バーチャルホスト名.hconf” というファイル名で“/usr/local/etc/elx/nginx/vhosts/”の下に置く。

6.4 SSL 秘密鍵、証明書の設定 (SSL サイトの場合のみ)

(1) “/usr/local/etc/elx/nginx/vhosts/バーチャルホスト名/”というディレクトリを作成する。

(2) “server.key”(秘密鍵), “server.cert”(証明書)を(1)で作成したディレクトリに置く。

※ファイルのオーナーは“root”、パーミッションは0400か0600に設定してください。

※サーバー秘密鍵は、パスフレーズを解除したのものをご使用ください。

※サーバ秘密鍵とサーバ証明書をテキスト形式で入力します。中間認証局の証明書が必要な場合はその証明書をサーバ証明書に連続してテキスト形式で入力してください。

※ ベリサインのサーバ証明書等、4階層の証明書をインストールする場合

1. サーバー証明書
 2. 3階層目中間CA局証明書(ベリサインでは中間CA証明書)
 3. 2階層目中間CA局証明書(ベリサインではクロスルート設定用証明書)
- の順番で記述してください。

例：サーバ秘密鍵(server.key)の場合

```
-----BEGIN RSA PRIVATE KEY-----
```

```
MIIBOQIBAAJBANUrwEVkvoFDJedmYA6C/FUgIXa8AvAbIDajBPCr+M9PD7eN9Kio
```

.

.
.
SyK/h9n6kgsVLcQaUAaDiVD8V+baNr95Y4ECAwEAAQJAL9f69IgvHanpdu8WPgu0
-----END RSA PRIVATE KEY-----

例：サーバ証明書 (server.cert) の場合

-----BEGIN CERTIFICATE-----
MIIE5TCCBE6gAwIBAgIQDrTmHgd2vNJzsUKnb66a3jANBgkqhkiG9w0BAQUFADCB
.
.
.
xnoGn7TdiqMTUnpRvRYoGea77Z/GbanvIYFEGPnQAXeqzSpkFXi+F8SN6TewaStb
RXYRIfKOQXv+
-----END CERTIFICATE-----

例：サーバ証明書 (server.cert) 4 階層の場合

-----BEGIN CERTIFICATE-----
MIIE5TCCBE6gAwIBAgIQDrTmHgd2vNJzsUKnb66a3jANBgkqhkiG9w0BAQUFADCB
.
[サーバ証明書]
.
xnoGn7TdiqMTUnpRvRYoGea77Z/GbanvIYFEGPnQAXeqzSpkFXi+F8SN6TewaStb
RXYRIfKOQXv+
-----END CERTIFICATE-----

-----BEGIN CERTIFICATE-----
MIIE0DCCBDmgAwIBAgIQJQzo4DBhLp8r ifcFTXz4/TANBgkqhkiG9w0BAQUFADBf
.
[3 階層目中間 CA 局証明書]
.

DDPLq9ZVTGr0SzeK0saz6r1we2uIFjxfleLuUqZ87NMwwq14IWAyMfs77o0ghZ
t0xFnfeKW/9mz1Cvxm1XjRI4t7mi0VfqH5pLr7rJjhJ+xr3/
-----END CERTIFICATE-----

-----BEGIN CERTIFICATE-----
MIIF7DCCBNSgAwIBAgIQbsx6pacDIAm4zrz06VLUKTANBgkqhkiG9w0BAQUFADCB

[2 階層目中間 CA 局証明書]

```
W+yzf5VK+wPIrSbb5mZ4EkrZnOL74ZjmQo0bj49nJOhhGbXdzbULJgW0w27EyHW4
Rs/iGAZeqa6ogZpHFt4MKGwIJ7net4RYxh84HqTEy2Y=
-----END CERTIFICATE-----
```

6.5 サイトを追加する場合

(1) nginx.conf の “proxy_cache_path” を追加する。

proxy_cache_path の行をコピーし、example.jp の部分を新たに追加するバーチャルホスト名に変更してください。

(2) 上記 [{バーチャルホスト名.vconf ファイルを作成}](#) から [{バーチャルホスト名.hconf ファイルを作成}](#) (SSL の場合 [{SSL 秘密鍵、証明書の設定\(SSL サイトの場合のみ\)}](#)) の手順を繰り返してください。

※ <バーチャルホスト名> は重複できません。一意の名前を付けてください。

※ 設定を反映させるためには nginx の再起動を行う必要があります。

6.6 作成ファイルの確認

上記 [{バーチャルホスト名.vconf ファイルを作成}](#) から [{バーチャルホスト名.hconf ファイルを作成}](#) (SSL の場合 [{SSL 秘密鍵、証明書の設定\(SSL サイトの場合のみ\)}](#)) の手順を行った場合のファイル構成は以下のようになります。

```
/usr/local/etc/elx/nginx/nginx.conf
/usr/local/etc/elx/nginx/vhosts/バーチャルホスト名.vconf
/usr/local/etc/elx/nginx/vhosts/バーチャルホスト名.hconf
--以下 SSL サイトのみ--
/usr/local/etc/elx/nginx/vhosts/バーチャルホスト名/server.key
/usr/local/etc/elx/nginx/vhosts/バーチャルホスト名/server.cert
```

以上で必要最低限のバーチャルホストの設定は終わりです。エリクサーを起動するための準備は整いましたので、ここで起動を試したい場合は [{起動}](#) を参照して下さい。次章 [{その他の設定}](#) では、キャッシュの時間指定など、より詳細な設定を説明します。

7 その他の設定

エリクサーではキャッシュ時間の変更、各パラメータ名の変更などをバーチャルホスト毎に設定することができます。また、アクセスした端末のスペック(仕様)を HTTP ヘッダに付加してバックエンドに送出する機能(端末スペック通知)がありますので、そこで使われる HTTP ヘッダを編集することができます。

7.1 設定の基本

設定は、設定項目名 (Key) と値 (Value) のペアを書いていくことです。これらの区切り文字として半角スペースを挿入します。また Apache の設定を行ったことのある方でしたら、比較的理解しやすいかと思います。

```
設定項目名 [space] 値;
```

設定項目はディレクティブと呼ばれます。値には複数のパラメータを受け付けるものもあります。

```
設定項目名 [space] 値 [space] パラメータ 1 [space] パラメータ 2;
```

設定の途中で改行を入れる方法もあります。

```
設定項目名 [space] 値
    パラメータ 1
    パラメータ 2;
```

行頭にシャープを入れるとコメント行になります。

```
# この行はコメントになります
```

設定の基本は エリクサーがベースとする nginx の設定を行うことです。一度 [{参考資料}](#) の "nginx" や "nginx の設定情報" をざっと見ておくことをお勧めします。特に設定 (Configuration) や各モジュール (Modules) のページからは、より詳しい設定情報を得ることが可能です

nginx の基本的な (HTTP コンテキストレベルの) 設定は "nginx.conf" に、バーチャルホスト毎の設定は "バーチャルホスト名.vconf" ファイルに行います。

7.2 設定例

具体例を示します。

例1 画像の横幅指定のパラメータ名を “elx_w” から “image_width” に変更する

```
location @fallback{
    ...
    elx_image_filter_width_param_name elx_w;
        ↓ 変更 ↓
    elx_image_filter_width_param_name image_width;
    ...
}
```

例2 PC に対しても絵文字の変換を行う

PC に対しても絵文字の変換を行う場合は以下の一行を追加します。追加する場所(コンテキストと呼ばれます)は “location @fallback” 内です。

```
location @fallback{
    ...
    elx_emoji_filter_exclude_carrier '' ;
    ...
}
```

その他のエリクサーに関する設定項目については[{参考資料}](#)の弊社ウェブサイト“設定項目一覧”を参照して下さい。

nginx の設定についてより詳しく知りたい方は[{参考資料}](#)の“nginx の設定情報”の設定情報を参照して下さい。

7.3 HTTP ヘッダの編集

HTTP ヘッダの編集は “バーチャルホスト名.hconf” ファイルにて行います。このファイルには端末スペック通知で使用する HTTP ヘッダの設定が記述されています。

以下は端末のキャリア情報の設定を抜粋したものです。

```
proxy_set_header X-ELX-CARRIER $elx_carrier;
```

proxy_set_header : ヘッダをセットする設定項目名
X-ELX-CARRIER : HTTP ヘッダ名
\$elx_carrier : HTTP ヘッダの値を保持する変数名

HTTP ヘッダ名である “X-ELX-CARRIER” の文字列は変更可能です。また、特定の HTTP ヘッダをオリジンサーバ(バックエンド)に送りたくない場合は行を削除するかまたは、行の先頭にシャープを追記しコメントアウトしてください。

```
# proxy_set_header X-ELX-CARRIER $elx_carrier;
```

※HTTP ヘッダ名は RFC 2616 section 4.2 に準拠してください。

※端末スペック通知を行う項目については[{参考資料}](#)の弊社ウェブサイト“変数名・端末スペック通知一覧”を参照して下さい。

8 起動

nginx 及び memcached デーモンをユーザ root で起動してください。起動後にユーザ elx で動作します。

例：起動スクリプトにより起動する場合

```
/etc/rc.d/init.d/memcached start  
/etc/rc.d/init.d/nginx start
```

※nginx の起動スクリプトのオプションに“configtest”を指定することで設定ファイルの構文チェック等のテストを行うことができます。

```
/etc/rc.d/init.d/nginx configtest
```

9 停止

nginx と memcached デーモンをユーザ root で停止してください。

例：起動スクリプトにより停止する場合

```
/etc/rc.d/init.d/nginx stop  
/etc/rc.d/init.d/memcached stop
```

※お客様のサーバ環境や起動スクリプトの設定により適宜変更してください。

10 自動アップデート

自動アップデートは、弊社アップデートサーバに HTTPS で通信を行い、自動的にアップデートを行います。自動アップデートを起動する前に以下の項目を確認してください。

(1) `elixir.conf` にシリアルナンバーとライセンスコードの値が設定されていることを確認ください。

```
SERIAL_NUMBER      0123456789
LICENSE_CODE       1234-5678-9ABC-DEFG-HIJK-LMNO
```

(2) エリクサーがインストールされたサーバから HTTPS (tcp ポート 443) で弊社アップデートサーバへアクセスができる必要があります。そのため、必要があれば、お客様のネットワーク環境(ファイアーウォール, DNS 等)の設定を行ってください。

(3) `syslog` の設定(必要があれば)

アップデートの結果を `local1.info` で `syslogd` にメッセージをおくります。`syslog` にメッセージを残したい場合は `syslog.conf` で `local1.info` が `syslog` に書き出される設定をする必要があります。`syslog` のファシリティは `local1` です。

10.1 設定方法

`/usr/local/elix/tools/` にある `elix_update.sh` を `cron` 等により定期的に行うことができるように設定してください。アップデートの即実行を行いたい場合は直接 `elix_update.sh` を実行してください。

自動アップデートではインストールディレクトリの書き換えや、`elixir.conf` の書き換えを行いますので、`elix_update.sh` は `elixir.conf` に対して書き込み権限を持つユーザで `cron` 等から起動されるように設定してください。

例 : `cron` を使用してユーザ `root` で、毎日午前 5:00 に自動アップデートさせる場合 `/etc/crontab` に以下を一行で追記

```
0 5 * * * root /usr/local/elix/tools/elix_update.sh
```

10.2 アップデート結果確認方法

アップデートが成功した場合は標準出力に `update was completed` と表示されます。エラーが発生した場合はそのメッセージが標準出力に表示されます。またアップデー

トの結果は“local1.info”で syslogd にエラーメッセージを送ります。

= 注意 =

アップデート成功後、必要がある場合には nginx、memcached が再起動します。再起動には数秒かかります。

11 手動アップデート

手動アップデートは、弊社アップデートサーバと通信せずソフトウェアを更新できます。

11.1 手順

(1) ソフトウェア保守サービス内容確認書の

「最新ソフトウェア ダウンロードサイト」よりアップデートファイルをダウンロードします（事前にお知らせしております ID、PASS が必要です）

(2) エリクサーをインストールしているサーバへアップデートファイルを

をコピーします（コピー先ディレクトリの例: /tmp/elx-1.0XX.pkg）

(3) 下記のコマンドを実行します

```
/usr/local/elx/tools/elx_update.sh -u /tmp/elx-1.0XX.pkg
```

※手動アップデート時には nginx, memcached の再起動を行うためサービスの瞬断が発生いたします。

11.2 アップデート結果確認方法

(1) 下記のメッセージが表示されることを確認します

```
update was completed
```

(2) ディレクトリのバージョンが変わっていることを確認します

```
/usr/local/elx-1.0XX
```

(3) elixir.conf のバージョンが変わっていることを確認します

・ ファイル名

```
/usr/local/etc/elx/conf/elixir.conf
```

・ 確認する項目

```
ELIXIR_VERSION          1.0XX
```

以上で手動アップデート完了です。

12 ダウングレード

バックアップを取り、そのバックアップデータをリストアすることで、緊急時にソフトウェアをバックアップ時のバージョンに戻すことが可能です。なお、エリクサーでは最新バージョンのソフトウェアのみサポート対象とさせていただいております。下記の手順は緊急時のみご利用ください。

下記に一例としてソフトウェアを Ver. 1.135 から Ver. 1.134 に戻す場合を例に手順を説明します。※root 権限で操作してください。

12.1 バックアップ手順

(1) カレントディレクトリを変更

```
cd /usr/local
```

(2) elx-1.134 の付くディレクトリを確認

```
ls -l
```

(3) 必要なディレクトリを tar+gzip、確認

```
tar -cvf - elx-1.134 etc/elx | gzip > elx-1.134.tar.gz
```

```
ls -l
```

(4) ident の結果を保存

```
ident /usr/local/elx/sbin/nginx
```

※ident コマンドが無い場合は実施する必要ございません。

12.2 リストア手順

※(1)～(9)のエリクサー停止中は、アクセスを受け付けませんのでご注意ください。

(1) エリクサーを停止

```
/etc/init.d/nginx stop
```

```
/etc/init.d/memcached stop
```

(2) カレントディレクトリを変更

```
cd /usr/local
```

(3) シンボリックリンクを削除、確認

```
rm -f elx
ls -l
```

(4) 設定ファイルのディレクトリ名を変更、確認

```
mv etc/elx etc/elx-1.135
ls -l etc/
```

(5) tar を展開、確認

```
zcat elx-1.134.tar.gz | tar -xvpf -
ls -l
ls -l etc/
```

(6) シンボリックリンクを作成

```
ln -s elx-1.134 elx
ls -l
```

(7) ident の結果をバックアップ時の結果と比較

```
ident /usr/local/elx/sbin/nginx
```

※ident コマンドが無い場合は、下記を確認

```
ls -l /usr/local/elx/sbin/
```

下記の2つが存在するか確認

```
nginx -> nginx_x86_64
nginx_x86_64
```

(8) キャッシュ削除、確認

```
rm -rf /var/elixir/cache/*
ls -l /var/elixir/cache/
```

(9) エリクサーを起動

```
/etc/init.d/memcached start
/etc/init.d/nginx start
```

(10) 登録サイトへアクセスして動作を確認

(11) リストア前のバージョンを削除、確認

```
rm -rf elx-1.135 etc/elx-1.135
```

```
ls -l
```

```
ls -l etc/
```

13 ログローテーションの設定

エリクサー(nginx)はログローテーションを行いませんので、logrotate 等を利用してログローテーションの設定を行う必要があります。

13.1 ログの説明

エリクサーではアクセスログとエラーログを以下のファイル名(デフォルト)で出力します。(お客様が.vconf に設定されたバーチャルホスト名のついたファイル)

アクセスログ : /var/log/httpd/access-<バーチャルホスト名>.log

エラーログ : /var/log/httpd/error-<バーチャルホスト名>.log

13.2 設定方法

エリクサーでは以下に logrotate のサンプル設定を用意しています。サンプル設定を使用する場合は/etc/logrotate.d/ などにこの設定ファイルを置き、お客様の環境に合わせた設定を行ってください。

/usr/local/elx/examples/etc/logrotate.example

14 変換対象コンテンツの仕様

エリクサーはパソコンや携帯向けに作成された Web コンテンツを、様々な端末に合わせて変換します。

また、主な変換対象コンテンツとして画像コンテンツと XHTML/HTML コンテンツ（以下「HTML コンテンツ」とします）があります。広い意味で CSS も HTML コンテンツに含まれます。

※ 端末の不具合により正常に表示されないケースがございます。

※ {付録 B. 参考資料} に各種資料や出典を記載しています。

こちら合わせてご覧下さい。

14.1 マークアップ言語

PC 用コンテンツ

XHTML1.0 Transitional、HTML4.01 Transitional

フレーム (XHTML1.0 Frameset、HTML4.01 Frameset) には対応していません。

携帯用コンテンツ

i モード対応 XHTML2.3、i モード対応 HTML (いわゆる CHTML)。

これらが非整形形式のコンテンツであっても、HTML Tidy が整形形式に変換します。

参考: HTML Tidy の処理例

<http://tidy.sourceforge.net/docs/Overview.html>

言語の将来性を考慮し、コンテンツ新規作成時には XHTML をお勧めします。

14.2 文字コード

UTF-8、Shift_JIS、EUC-JP

14.3 絵文字

docomo の絵文字に対応します。

- ・ Unicode のテキスト入力
- ・ Shift_JIS のテキスト入力
- ・ Shift_JIS のバイナリコード入力

詳細は NTT ドコモの「作ろう i モードコンテンツ」を参照して下さい。

14.4 CSS

以下の記述に準拠しています。

- (1) CSS level 2.1
- (2) i-CSS (i モード対応 XHTML 用の CSS)

- ・インライン (style 属性) 記述方式
- ・内部参照 (疑似クラスのみ) 記述方式

(3) 外部参照 (外部 CSS) ファイル読み込み

外部 CSS (@import や<link type="text/css" href="...">での読み込み) 非対応機種の場合エリクサーが自動的に読み込みます。

※ CSS の media 記述が無いか、media が all またはアクセスしたデバイスが対応している media の場合のみ処理を行います。その他の media が指定された場合は削除されます。

※ 全ての機種において、CSS をインライン(style="")に展開します。

14.5 画像

GIF (アニメーション GIF を含む)、JPEG、PNG

15 画像変換パラメータ

エリクサーは、内部に持つ端末仕様のデータベース（以下「端末プロファイル」とします）とHTMLコンテンツ内の画像ファイルを元に各端末に合った画像を出力します。また、HTTP URL クエリのパラメータ（以下「URI パラメータ」または単に「パラメータ」とします）も指定できます。

15.1 出力自動調整方法

以下の情報を元に自動調整を行います。

- ・ 端末でサポートされている画像形式
- ・ 端末でサポートされている画像形式が元画像形式と異なる場合は、端末でサポートされている画像形式に変換。
- ・ 元画像形式が端末でサポートされている画像形式の場合は画像形式変換はしません。

== 端末でサポートされている場合は ==

- ・ 端末ブラウザの横幅（端末デバイスの横幅ではありません）
- ・ 端末のキャッシュサイズ
- ・ 画像の横幅
- ・ 画像の色深度
- ・ 画像（img タグ）が table タグ内であれば、
テーブル系タグ（table, tr, th, td）の width
（※table タグの変換を無効にした場合のみ考慮されます）

結果以下の調整を行います。

- ・ 画像の縦横幅（調整前と調整後のアスペクト比は同じ）
- ・ 画質（色深度。JPEG の場合は圧縮率も）

15.2 パラメータ指定内容

自動調整に加え、URI にパラメータを指定して調整内容を変更することができます。

- ・ 画像の横幅（パーセント、ピクセル指定）
- ・ 画像のファイルサイズ（変換サイズ上限指定）
- ・ 画像を待ち受け画像（壁紙）に指定



待ち受け画像指定 デフォルト (OFF) と 1 (ON) の違い

- ・ 待ち受け画像の余白背景色



待ち受け画像の余白背景色 デフォルト (黒) と 0000ff (青) の様子

- ・ 画像の無変換指定

15.3 画像の横幅指定

横幅 (width) を指定します。

URI パラメータ名

elx_w

(1) ピクセル指定

横幅をピクセル数で指定します。

記述例 (180px)

```
<img src = "image.jpg?elx_w=180">
```

※ elx は “ELiXir” に由来します。

(2) パーセント指定

画像の横幅を各端末ブラウザの横幅に対してパーセンテージで指定します。

記述例 (80%)

```

```

※ “P” の指定が必要になります

※ アスペクト（縦横）比は固定のため、縦幅（height）の指定は必要ありません。

15.4 画像のファイルサイズ指定

変換ファイルサイズの上限を指定したい場合等に使用します。サイズをバイト数で指定して下さい。端末プロファイルのキャッシュサイズを超えない範囲で調整して出力します。

URI パラメータ名

elx_s

記述例 (2000 バイト)

```

```

15.5 待ち受け画像指定

画像を待ち受け画像（壁紙:Wall Paper）として指定します。

elx_w/elx_s が指定されていた場合でも、端末プロファイルの待ち受け画像の縦横幅が適用されます。

URI パラメータ名

elx_wp (0:OFF, 1:ON)

記述例

```

```

15.6 待ち受け画像の余白背景色指定

画像を待ち受け画像にした時に余白ができる場合があります。その余白部分の色を指定することができます。

URI パラメータ名

elx_wp_bg (RGB 値 16 進数表記。デフォルト黒:000000)

記述例 (青 "0000ff")

```

```

※ elx_wp=1 の指定がない場合は無効になります。

シャープ # は指定しないで下さい。

red 等の色名表記も指定できません。

& は & の文字実体参照です。(URI パラメータのデリミタ)

15.7 画像の無変換指定

あえて変換オリジナルの元コンテンツ を出力させたい場合に使用します。

elx_w/elx_s が指定されていた場合でも、画像の縦横幅やファイルサイズの変換は行われません。

URI パラメータ名

elx_nc (0:OFF, 1:ON)

記述例

```

```

15.8 パラメータの名前指定

URI パラメータ名にはデフォルトで "elx_" 接頭辞が付きますが、バーチャルホスト毎の nginx 設定 (以下「バーチャルホスト設定」とします) で名前を変更することができます。

画像の横幅指定

elx_image_filter_width_param_name

画像のファイルサイズ指定

elx_image_filter_size_param_name

待ち受け画像指定

elx_image_filter_wp_param_name

待ち受け画像の余白背景色指定

elx_image_filter_wp_bg_color_param_name

画像の無変換指定

elx_image_filter_noconv_param_name

※ 設定方法は{[その他の設定](#)}を参照して下さい

15.9 パラメータを付加しない指定

画像にパラメータを付けたくない場合は下記のように指定します。

属性名

elx_asis (0:OFF, 1:ON)

記述例

```

```

16 画像コピー転送防止

エリクサーは、著作権保護に関する機能として、コンテンツ利用者がサイトからダウンロードした画像に対して、メール添付等の利用による画像の端末外への取り出しを禁止することができます。コンテンツ利用者がダウンロードした画像を自由に配布しないように制限したい場合等にご利用下さい。

端末の種類	メール添付	データ転送
docomo	○	○
SoftBank	○	○
au	○	○
Smartphone	×	×
PC	×	×

各端末の種類に対するコピー転送防止機能対応表

※ 端末自体の制限により上記制限が有効とならない場合もあります。

16.1 転送防止の指定方法

バーチャルホスト設定に転送防止する URL のパス（ディレクトリ）を指定します。

設定例（複数のパスを指定）

```
elx_image_filter_copyright_path /nocopy_dir1 /nocopy_dir2;
```

※ 設定の詳細は[{その他の設定}](#)を参照して下さい。

※ “/” を指定すると、全てのパスに対して転送防止が実施されます。

17 端末スペック通知

変換対象コンテンツ（以下「変換元コンテンツ」とします）を持つオリジンサーバ（以下「バックエンド」とします）に変換元コンテンツを取得する際に、アクセスした端末のスペック（仕様）データを、HTTP リクエストヘッダに付加してリクエストを行います。バックエンドのコンテンツサーバ側で受け取ったスペックデータに応じてアプリケーション側で送出するコンテンツをコントロールすることが可能です。

17.1 通知する項目

バーチャルホスト設定ファイル内に指定された項目を通知します。具体的な指定の方法は[\[HTTP ヘッダの編集\]](#)を参照して下さい。指定可能な項目については[\[参考資料\]](#)の弊社ウェブサイト“変数名・端末スペック通知項目一覧”を参照して下さい。

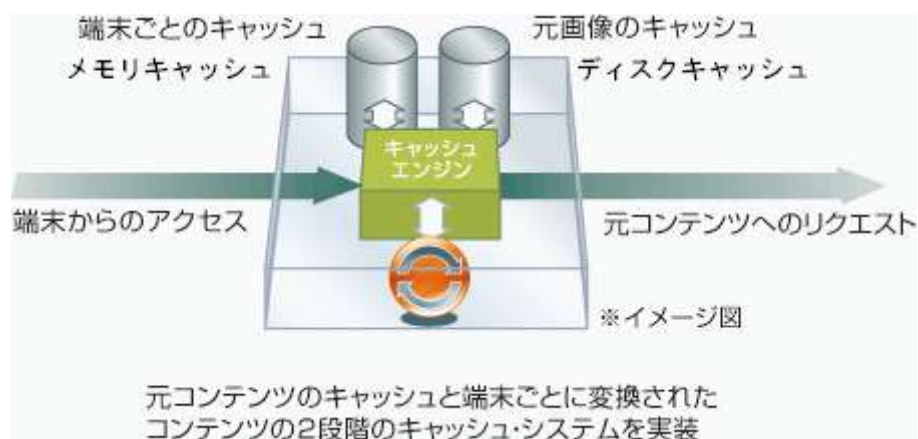
17.2 スペックの名称変更

バーチャルホスト設定ファイル内に指定することにより、通知する項目の名称を変更することが可能です。具体的な指定の方法は[\[HTTP ヘッダの編集\]](#)を参照して下さい。

18 キャッシュ制御

18.1 キャッシュ機構

変換元コンテンツのディスクキャッシュと、端末ごとに変換されたコンテンツのメモリキャッシュの、2つのキャッシュ構造を持ちます。



キャッシュ機構

= 注意 =

キャッシュ機構を正常に動作させるためには、エリクサーが動作しているマシンの時刻を ntp 等を用いてバックエンドと共に正確にする必要があります。

18.2 キャッシュを行いたい場合

デフォルト設定では、キャッシュ（変換元コンテンツのキャッシュ、変換後コンテンツのキャッシュ）を行いますので特に設定を行う必要はありません。

18.3 キャッシュをさせない場合

HTTP レスポンスヘッダの "Cache-Control" に "no-cache" を指定して下さい。
"no-cache" が指定されている場合キャッシュは保存されません。よって、毎回元コンテンツを取得し、毎回変換処理を行いません。

= 注意 =

CGI 等の動的コンテンツの場合は、必ず "Cache-Control: no-cache" の HTTP ヘッダを付加して下さい。

18.4 キャッシュ時間を指定する場合

以下に説明する方法でキャッシュ時間を指定することができます。但し、HTTP ヘッダに "Cache-Control: no-cache" が指定されていた場合はこちらを優先するため、キャッシュされません。

18.4.1 変換後コンテンツ(メモリキャッシュ)のキャッシュ時間指定

キャッシュ時間の指定方法を優先順位の高い順に説明します。

(1) "Expires" ヘッダを指定する

"Expires" で指定された期限までキャッシュを保持します。

(2) "Cache-Control" ヘッダに "max-age" を指定する。

"max-age" で指定された時間内、キャッシュを保持します。

(3) バーチャルホスト設定ファイルで指定する。

指定されたキャッシュ残存時間までキャッシュを保持します。デフォルトは 3600 秒 (1 時間) です。指定項目は以下の通りです。

```
elx_memcached_valid_sec 3600;
```

補足 : 既にキャッシュ時間が指定されている場合でも、より優先順位の高い方法でキャッシュ時間が指定された場合、優先順位の高い方法のキャッシュ時間で上書きされます。

例えば、(3)バーチャルホスト設定ファイルで 3600 秒のキャッシュ時間を指定していた場合でも、(2)の"max-age" で 30 秒と指定された場合。キャッシュ時間は 30 秒となります。

18.4.2 変換元コンテンツ(ディスクキャッシュ)のキャッシュ時間指定

以下にキャッシュ時間の指定方法を説明します。

(1) "Expires" ヘッダを指定する

"Expires" で指定された期限までキャッシュを保持します。

(2) "Cache-Control" ヘッダに "max-age" を指定する。

"max-age" で指定された時間内、キャッシュを保持します。

(3) バーチャルホスト設定ファイルで指定する。

指定されたキャッシュ残存時間までキャッシュを保持します。デフォルトは 3600 秒 (1 時間) です。指定項目は以下の通りです。

```
proxy_cache_valid 200 3600;
```


“200” はバックエンドからの HTTP レスポンスのステータスコードの指定です。この例では、“200” で帰ってきた場合、3600 秒(1 時間) キャッシュを行います。

(4) “nginx.conf” ファイルで指定する。

最後にディスクキャッシュにリクエストが行われてから、指定された時間が経過するまでキャッシュを保持します。デフォルトは 3600 秒(1 時間)です。指定項目は以下の通りです。

```
proxy_cache_path /var/elixir/cache/example.jp .. (省略).. inactive=3600;
```

補足：変換元コンテンツのキャッシュ時間指定の場合、(1) と (2) の優先順位はありません、先に指定された値が優先されます。(3) は (1)、(2) の指定がなかった場合に有効になります。

18.5 キャッシュの削除

エリクサーの各バーチャルホスト毎のキャッシュを削除する場合は、nginx.conf で設定されている各バーチャルホストのキャッシュディレクトリを削除します。ハードディスク上のキャッシュを削除して、エリクサー(nginx と memcached)を再起動してください。

例：example.jp のキャッシュを削除したい場合は下記、example.jp ディレクトリを削除します

```
/var/elixir/cache/example.jp
```

※エリクサー(memcached)のみ再起動した場合、変換後コンテンツのキャッシュが削除されますが、ディスク上の変換元コンテンツのキャッシュは削除されません。

19 未対応端末の動作

リクエストがあった際に HTTP ヘッダの User-Agent で端末の種別を判別し、端末プロファイルをもとに処理内容を決定します。未対応端末での動作は以下の通りです。

19.1 対応している端末

端末プロファイルに登録された端末は変換対象となり、端末種別、機種、キャリア(携帯電話の場合)に応じてコンテンツを変換して送ります。対応端末一覧については[\[参考資料\]](#)の“対応端末一覧”を参照して下さい。

19.2 対応していない端末

3G 以前の旧携帯電話端末等、対応していない端末に対しても変換は行っていますが、動作保証はしておりません。また動作検証も行っておりません。

19.3 未知の端末 (登録されていない端末)

HTTP ヘッダの User-Agent から端末種別やキャリアが判断できる場合は、それらに応じたデフォルト端末 (比較的新しい機種を参考にしたモデル端末) として変換を行います。

19.4 携帯電話向け検索エンジンのクローラ

端末プロファイルに登録されている携帯電話向けの検索エンジンは、携帯端末として処理します。現在登録されているクローラの情報は、[\[参考資料\]](#)の“対応端末一覧”を参照して下さい。

19.5 PC (その他の端末)

HTTP ヘッダの User-Agent が端末プロファイルに定義されていない端末からのアクセスの場合 PC と判断し、無変換で変換元コンテンツを送ります。

20 PC ブラウザでの動作確認方法

実機の端末ではなく、PC ブラウザを使って変換動作確認を行う方法は以下の通りです。通常、最終的な動作確認はあくまでも実機の端末で行いますので、以下参考情報としてご活用下さい。

20.1 確認ツール

Firefox の User-Agent 指定機能等、User-Agent を偽りリクエストを行うことで動作を確認することができます。

※ Firefox のアドオン FireMobileSimulator では問題を確認しております。アドオン UserAgentSwitcher の使用をおすすめします。

※ UID や端末固有番号や HTTP ヘッダ等に関する動作は確認できませんが、端末スペック通知でエリクサーからバックエンドのコンテンツサーバに送出するエリクサー独自の HTTP ヘッダの確認は可能です。

20.2 変換後コンテンツの確認

下記項目でコンテンツの変換の確認ができます。

(1) DOCTYPE 宣言を確認する。

端末の(大枠の)種類によって以下のように DOCTYPE 宣言が変換されます。

端末種類	DOCTYPE 宣言
docomo	i-XHTML
au	OPENWAVE
Softbank	J-PHONE
Smartphone	XHTML1.0 Transitional
PC	無変換

(2) 絵文字変換の確認。

端末の種類によって以下のように絵文字が変換されます。

端末種類	絵文字表現
docomo	Unicode の 16 進数実体参照 (docomo 用)
au	
Softbank	Unicode の 16 進数実体参照 (Softbank 用)
Smartphone	代替絵文字画像
PC	無変換

(3) 画像変換の確認。

端末の種類によって以下のように画像が変換されます。

端末種類	画像変換
docomo	端末のブラウザサイズ及び、URI パラメータで指定された値での変換。
au	docomo と同様
Softbank	docomo と同様
Smartphone	docomo と同様
PC	無変換

(4) 端末スペック通知の確認。

HTTP リクエストヘッダを表示させるスクリプトなどを記述することで確認できます。

端末種類	画像変換
docomo	X-ELX（デフォルト）から始まる HTTP ヘッダとその値が確認できる。
au	docomo と同様
Softbank	docomo と同様
Smartphone	X-ELX（デフォルト）から始まる HTTP ヘッダとその値が確認できる。但し、項目数が少ない場合があります。
PC	Smartphone と同様

21 その他コンテンツ作成時の注意事項

21.1 POST データの文字コード

初期状態ではエリクサーは POST データを透過します。そのため、下記の(1)か(2)の方法で POST データの文字コードを変換する必要があります。

(1) バックエンドで対応する方法

POST されたデータの文字コード変換をバックエンドで行ってください。POST 元のページの送出コンテンツの文字コードは、端末スペック通知項目の“X-ELX-DEVICE-OUTPUT-ENCODING”の値を参考にして下さい。

(2) エリクサーの設定で対応する方法

POST データの Content-Type が“application/x-www-form-urlencoded”の場合にかぎり、POST データを設定項目“elx_post_codeconv_charset”で指定した文字コードに変換します。バックエンドで使用されている文字コードを指定してください。ただし、POST データの Content-Type が“multipart/form-data”の場合は、POST データを透過しますのでご注意ください。

設定項目“elx_post_codeconv_charset”の詳細につきましては{[参考資料](#)}の“設定項目一覧”を参照して下さい。

※エリクサーは GET データも透過しますが、GET データを変換する機能はございません。この場合、method を POST に変更するか、バックエンドで端末スペック通知項目の“X-ELX-DEVICE-OUTPUT-ENCODING”の値を参考に対応してください。

21.2 CSS パースエラー時の対応

CSS level 2.1 に対応していない記述(いわゆる CSS ハック等)をした場合、CSS のパース(構文解析)エラーが発生します。その対応方法を示します。

- ・ エラーの確認方法

error.log 内にエラー内容が記録されます。

```
2011/07/14 15:40:37 [error] .. (省略)..
```

```
5:9:while parsing declaration: this char must be ':'... (省略)
```

内部参照 CSS を記述されていた場合は、コンテンツ内にも下記のようにエラーメッセージを出力します。

```
/*  
++ CSS PARSING ERROR ++  
5:9:while parsing declaration: this char must be ':'  
*/
```

- ・エラーの対処方法

エラーメッセージの最初の数字がエラーが発生した行番号(元コンテンツ内)を示します。よって、上記メッセージの例で言うと、CSSの5行目にパースエラーの原因が存在すると思われます。実際の修正すべきエラー箇所については、どのようなエラーかに依りますので、エラー原因箇所発見の手掛かりとして下さい。

22 独自の変換ルール

ユーザー独自の変換ルール(XSLT スタイルシート)を追加することが可能です。

22.1 変換ルールの作成方法

独自の変換ルールは XSLT と Xpath で記述します。

{[参考資料](#)}の“変換ルールのサンプル”と下記の URL を参考にして変換ルールを作成してください。

XSLT : <http://www.w3.org/TR/xslt/>

XPath : <http://www.w3.org/TR/xpath/>

※ XSLT/Xpath の記述方法、変換ルールのサンプルにつきましてはサポート対象外とさせていただきます。

22.2 変換ルールの設定方法

作成した変換ルールを特定のキャリアのみ、特定のコンテンツのみ、特定フォルダのコンテンツのみなど、様々な条件で設定することが可能です。

具体的な設定方法は、{[参考資料](#)}の“変換ルールの設定例”を参照して下さい。

23 レイアウト変換

エリクサーは、HTML コンテンツに指定されたタグ（XML の要素）とその子要素をひとつのブロック（Block）として捉え、レイアウト変換を行います。レイアウト変換を行う主な目的は、多種多様な PC 用コンテンツを携帯コンテンツらしく表示するためです。エリクサーではこの変換を行う処理単位をメソッド（Method）と呼び、必要なブロックに対して必要なメソッドを呼び出す方式を取ります。コンテンツの再配置を行う場合は出力したいブロックごとにメソッド“block”を指定します。

また、指定して動くメソッドではなく、自動的に実行される基本処理をいくつか用意しており、これらをベーシック変換と呼びます。

※ 説明のためにコンテンツにインデントを付加していますが、パフォーマンス考慮により、実際の変換後コンテンツには付きません。

※ 説明に不必要と思われるタグや属性はHTML コンテンツの妥当性を無視して省略しています。

23.1 ブロック指定方法の種類

レイアウト変換を行うための特定のブロックを指定する方法は2通りあります。

- ・特定の属性にメソッドを指定する方法
- ・メソッドごとに XPath を指定する方法

23.2 特定の属性へのメソッド指定

あるブロックに対してどのメソッドを対応させるのか、エリクサーに知らせるために特定の属性を使用します。その属性を決めるための方法をいくつか用意しています。

- ・デフォルト属性名を使用する
- ・コンテンツの meta タグで指定する
- ・設定ファイルに指定する

(1) デフォルト属性名を使用する

デフォルトの属性名として4種類用意しています。

elxm, elxm1, elxm2, elxm3

※ elxm は“ELiXir Method”に由来します。接頭辞にも使用しています

(2) コンテンツの meta タグで指定する

head タグの子要素として、以下の様な meta タグを記述して下さい。
カンマ区切りにより複数指定可能です。

```
<meta name="elxm_method_attrs" content="attr1, attr2, attr3"/>
<!-- name, content への指定は meta タグとして妥当な記述方法 -->
...
...
<div attr1="method1" attr2="method2, method3" attr3="method4">
...

```

また id、class の様な独自の属性を指定することも可能です。

```
<meta name="elxm_method_attrs" content="id, class"/>
...
...
<div id="method1" class="method2">
...

```

但し、エリクサー内部処理の制限はありませんが、この様な指定方法の場合、HTML コンテンツとしての妥当性に注意する必要があります。

例

- id, class 属性としてカンマの使用は妥当ではない。
- id 属性として同値の属性値を複数指定するのは妥当ではない。

(3) 設定ファイルに指定する

上記 (2) と同様の指定をバーチャルホスト設定ファイル中に設定することも可能です。これによりデフォルト設定が変わり、meta タグへの記述は不要になります。

設定例

```
xslt_stylesheet eug-block.elx elxm_method_attrs='myattr1,myattr2';
```

※ 設定の詳細は [\[その他の設定\]](#) を参照して下さい。

23.3 メソッドごとの XPath 指定

バーチャルホスト設定に XPath を使用して柔軟にレイアウトのブロックを指定することができます。

設定例

id 属性が "main" 直下の div タグにある画像に br タグを付加する

```
xslt_stylesheet eug-block.elx elxm_break_image=' //@id="main"/h%3Adiv' ;
```

※ "break_image" がメソッド名です。XPath で指定する時には、接頭辞 "elxm_" を付けて下さい。

```
<div id="main">
  <div class="content">
    <span>text1</span><b>text2</b>
  </div>
  <table>
    <tr><td></td></tr>
  </table>
  <div class="content">
    <span>text3</span><b>text4</b>
  </div>
</div>
```

※ XPath 内の %3A は XML 名前空間を表す ":" コロンを示します。

コロンは nginx の設定上メタデータであるため、URL エンコードしてエスケープします。また、名前空間 h は "http://www.w3.org/1999/xhtml" のエイリアスで定義しています。

23.4 ベーシック変換機能 OFF

ベーシック変換は基本処理のためデフォルトで実行されますが、バーチャルホスト設定で機能を OFF にすることができます。

23.4.1 テーブル変換、object タグ変換 OFF

(1) object のタグの変換を OFF にする場合。

eug-embed_object.elx' の記述されている行をコメントアウト(行頭に'#')または削除。

設定例

```
xslt_stylesheet eug-embed_object.elx;
```

↓

```
# xslt_stylesheet eug-embed_object.elx;
```

(2) テーブルの変換を OFF にする場合。

eug-table_layout.elx' の記述されている行をコメントアウト(行頭に'#')または削除。

※テーブル変換自体を OFF にする場合。

設定例

```
xslt_stylesheet eug-table_layout.elx;  
↓  
# xslt_stylesheet eug-table_layout.elx;
```

※個別のテーブルに対して変換を OFF にする場合。

属性名'elxm_basic_table_layout'を値'0'で対象テーブル要素内に記述する。

設定例

```
<table elxm_basic_table_layout="0">  
  <tr>  
    <td>table1</td>  
    <td>table2</td>  
  </tr>  
</table>
```

23.4.2 その他のベーシック変換機能 OFF

[ベーシック変換設定名]='0'を'eug-basic.elx'に半角スペースを空けて記述する。複数指定する場合は、':'を間に入れる。

設定例

img タグの border="0" 付加機能と、img タグのサイズ削除機能を OFF にする。

```
xslt_stylesheet eug-basic.elx elxm_basic_hide_image_border='0':  
elxm_basic_strip_image_size='0';
```

23.5 ベーシック変換の内容

(1) ul リストの識別子の非表示。

ul リストに type 属性がない場合または、CSS で list-style-*が指定されていない場合、'list-style:none;'を追加する。

バーチャルホスト設定名

```
elxm_basic_hide_list_style
```

変換前

```
<ul>
  <li>List 1</li>
  <li>List 2</li>
  <li>List 3</li>
</ul>
```

変換後

```
<ul style="list-style:none;">
  <li>List 1</li>
  <li>List 2</li>
  <li>List 3</li>
</ul>
```

(2) 画像境界線を隠す

img タグに border="0" を追加する。すでに border 属性が存在する場合はその値を保持する。

バーチャルホスト設定名

elxm_basic_hide_image_border

変換前

```


```

変換後

```


```

(3) 画像サイズ属性削除

img タグの width、height 属性及び、CSS インライン (style 属性) 記述方式で指定されている width、height の削除。

バーチャルホスト設定名

elxm_basic_strip_image_size

変換前

```
  

```

変換後

```
  

```

(4) テーブル一段組み変換

table のセルの内容を div(初期値) タグで囲んで出力します。

バーチャルホスト設定名

なし：機能を OFF にしたい場合などは下記捕捉を参照して下さい。

例 1：通常のテーブル変換

変換前

```
<table>  
  <tr>  
    <td>Cell A</td>  
    <td>Cell B</td>  
  </tr>  
  <tr>  
    <td>Cell C</td>  
    <td>Cell D</td>  
  </tr>  
</table>
```

変換後

```
<div>  
  <div>Cell A</div>  
  <div>Cell B</div>  
</div>  
<div>  
  <div>Cell C</div>
```

```
<div>Cell D</div>
</div>
```

例 2 : 一行目が表の見出しになっている場合 (th で書かれている場合)

変換前

```
<table>
  <tr>
    <th>Cell A</th>
    <th>Cell B</th>
  </tr>
  <tr>
    <td>Cell C</td>
    <td>Cell D</td>
  </tr>
</table>
```

変換後

```
<div>Cell A</div>
<div>Cell C</div>
<div>Cell B</div>
<div>Cell D</div>
```

補足:

(1) セルの内容を div 以外のタグで囲みたい場合

パーチャルホスト設定の eug-table_layout.elx が記述されている設定行を以下の
ように

変更してください。

タグ名を省略した場合、タグには囲まれずセルの内容のみが出力されます。

別のタグで囲む場合

```
xslt_stylesheet /usr/local/elix/sbin/eug-table_layout.elx
replace_element_name=' タグ名' ;
```

セルの内容のみ出力する場合

```
xslt_stylesheet /usr/local/elix/sbin/eug-table_layout.elx
```

```
replace_element_name='';
```

(5) 埋め込み Object の削除

HTML 内に記述されている Object タグ及び Embed タグを以下の条件で削除します。

条件 1 : Object タグに、端末がサポートしていない classid が指定されていた場合。

条件 2 : Embed タグに、端末がサポートしていない content-type が指定されていた場合。

23.6 レイアウト変換の内容

※ {[メソッドごとの XPath 指定](#)} で使用する時は、接頭辞 “elxm_” を付けて下さい。

例 (メソッド名が “method1” の場合)

```
elxm_method1
```

(1) テキストメニュー

メソッド名

```
text_menu
```

動作

テキスト情報をメニューとして出力する。

以下のタグがあれば削除する。

```
img, object, applet, param, map, area, bgsound, embed, server
```

ul, ol 系のリスト (ul, ol, li) を dl 系 (dl, dt, dd) のリストに変換する。

```
ul, ol => dl
```

```
li => dd
```

dt を dd に変換して全体を dd に揃える。

```
dt => dd
```

変換前

```
<div elxm="text_menu">
```

```
<ul>
```

```
<li>
```

```
<a href="page1.html">
```

```
<span>About page 1<span>
```

```
</a>
```

```
</li>
```

```
<li>
```

```
<a href="page2.html">
  <span>About page 2<span>
</a>
</li>
</ul>
</div>
```

変換後

```
<div elxm="text_menu">
  <dl>
    <dd>
      <a href="page1.html">
        <span>About page 1<span>
      </a>
    </dd>
    <dd>
      <a href="page2.html">
        <span>About page 2<span>
      </a>
    </dd>
  </dl>
</div>
```

※ 変換箇所が分からなくなる様、変換時に参照した属性（上記例なら elxm 属性）は変換後にも残しています。

(2) 画像メニュー

メソッド名

image_menu

動作

画像情報をメニューとして出力する。アンカー (a) タグも残す。

ul, ol 系のリストを dl 系に変換する。

ul, ol => dl

li => dd

dt を dd に変換して全体を dd に揃える。

dt => dd

その他 img, a タグでなければ削除する。

変換前

```
<div elxm="image_menu">
  <ul>
    <li>
      <a href="page1.html">
        <span>About page 1<span>
      </a>
    </li>
    <li>
      <a href="page2.html">
        <span>About page 2<span>
      </a>
    </li>
  </ul>
</div>
```

変換後

```
<div elxm="image_menu">
  <dl>
    <dd>
      <a href="page1.html">
        
      </a>
    </dd>
    <dd>
      <a href="page2.html">
        
      </a>
    </dd>
  </dl>
</div>
```

(3) メニュー

メソッド名

menu

動作

{(1) テキストメニュー} でも {(2) 画像メニュー} でもない一般的なメニューとして出力する。

ul, ol 系のリストを dl 系に変換する。

ul, ol => dl

li => dd

dt を dd に変換して全体を dd に揃える。

dt => dd

変換前

```
<div elxm="menu">
  <ul>
    <li>
      <a href="page1.html">
        <span>About page 1<span>
      </a>
    </li>
    <li>
      <a href="page2.html">
        <span>About page 2<span>
      </a>
    </li>
  </ul>
</div>
```

変換後

```
<div elxm="menu">
  <dl>
    <dd>
      <a href="page1.html">
        <span>About page 1<span>
      </a>
    </dd>
```

```
<dd>
  <a href="page2.html">
    <span>About page 2</span>
  </a>
</dd>
</dl>
</div>
```

(4) タイトルリスト

メソッド名

title_list

動作

タイトル付きのリストを出力する。

div を dl に変換する。

div => dl

h1 から h6 を dt に変換する。

h1-6 => dt

リストの子要素を dd に変換する。

li, dt, dd => dd

リストタグを削除する。

ul, ol, dl

※ 当メソッドの指定は、div タグに対して行って下さい。指定したタグの直下に h1-6 あるいはリスト (ul, ol, dl) 系以外のタグが存在した場合の動作は確定できません。

変換前

```
<div elxm="title_list">
  <h2>Title 1</h2>
  <ul>
    <li>List 1-1</li>
    <li>List 1-2</li>
  </ul>
  <h2>Title 2</h2>
  <ul>
    <li>List 2-1</li>
    <li>List 2-2</li>
```

```
</ul>
</div>
```

変換後

```
<dl elxm="title_list">
  <dt>Title 1</dt>
  <dd>List 1-1</dd>
  <dd>List 1-2</dd>
  <dt>Title 2</dt>
  <dd>List 2-1</dd>
  <dd>List 2-2</dd>
</dl>
```

(5) アンカーリスト

メソッド名

anchor_list

動作

アンカーをリスト化する。

div を dl に変換する。

div => dl

a があれば dd で囲む。

a => dd で囲む

div, a 以外(aの子要素を除く)のタグはすべて削除する。

※ 当メソッドの指定は、div タグに対して行って下さい。また変換後に dl タグが並んでしまう様な div の入れ子(親子)関係の場合の動作は確定できません。

変換前

```
<div elxm="anchor_list">
  <ul>
    <li>
      <a href="page1.html">
        <span>About page 1</span>
      </a>
    </li>
    <li>
```

```
<a href="page2.html">
  <span>About page 2<span>
</a>
</li>
</ul>
<ul>
<li>
  <a href="page3.html">
    <span>About page 3<span>
  </a>
</li>
</ul>
</div>
```

変換後

```
<dl xmlns="anchor_list">
  <dd>
    <a href="page1.html">
      <span>About page 1<span>
    </a>
  </dd>
  <dd>
    <a href="page2.html">
      <span>About page 2<span>
    </a>
  </dd>
  <dd>
    <a href="page3.html">
      <span>About page 3<span>
    </a>
  </dd>
</dl>
```

(6) ALT メニュー

メソッド名

alt_menu

動作

img タグの alt 属性値を抜き出してメニュー出力する。もし属性値に値がなければ、src 属性よりファイル名を抜き出す (“[ファイル名]”)。ファイル名の取得ができなければ、“[画像]”とだけ出力する。

※ ベーシック変換の elxm_basic_list2dl {5-5 (1) リスト・to・DL} が機能 (デフォルト ON) している場合は、ul, ol 系のリストを dl 系に変換します。機能 OFF の場合でも {(3) メニュー} と組み合わせることでリストを変換できます。

変換前

```
<div elxm="alt_menu">
  <ul>
    <li>
      <a href="page1.html">
        
      </a>
    </li>
    <li>
      <a href="page2.html">
        
      </a>
    </li>
  </ul>
</div>
```

変換後

```
<div elxm="alt_menu">
  <ul>
    <li>
      <a href="page1.html">
        Page 1
      </a>
    </li>
    <li>
      <a href="page2.html">
        [page2.gif]
      </a>
    </li>
  </ul>
</div>
```

```
    </li>
  </ul>
</div>
```

(7) 画像のみ出力

メソッド名

only_image

動作

img タグのみ出力する。

変換前

```
<div elxm="only_image">
  <h3>Description 1</h3>
  <a href="page1.html"></a>
  <span>Description 2</span>
</div>
```

変換後

```
<div elxm="only_image">
  
</div>
```

(8) 画像改行

メソッド名

break_image

動作

画像出力後に br タグを挿入する。

変換前

```
<div elxm="only_image">
  <span>Description 1</span>
  
  <span>Description 2</span>
  
</div>
```

```
<span>Description 3</span>
</div>
```

変換後

```
<div elxm="title_list">
  <span>Description 1</span>
  <br style="clear:both"/>
  <span>Description 2</span>
  <br style="clear:both"/>
  <span>Description 3</span>
</div>
```

(9) ID Strip

メソッド名

strip_id

動作

ID 属性を剥ぎ取る。

(10) クラス Strip

メソッド名

strip_class

動作

クラス属性を剥ぎ取る。

(11) ID, クラス Strip

メソッド名

strip_id_class

動作

ID とクラス属性を剥ぎ取る。

24 ページ分割

ページ分割機能とは、変換後コンテンツが各端末のキャッシュサイズを超えると予想される場合、各端末のキャッシュサイズに収まるように自動でページを分割し出力する機能です。

24.1 詳細内容

- (1) form/object/embed/applet/script/select タグの中では分割しません。
- (2) ページ中の画像は変換後に適切なサイズ（ファイルサイズ）あるとして固定値で計算しています。nginx 設定にそれぞれの画像フォーマットの予想ファイルサイズを設定することが可能です。設定方法は[\[各種設定\]](#)の“(2) 予想ファイルサイズ設定”を参照して下さい。
- (3) elx_w の値が 10px 以下の画像は計算から除外しています。
- (4) style 属性中の画像は計算に含まれません。（背景画像など）
- (5) ページ分割時の次ページ前ページのリンクは設定で変更可能です（絵文字設定不可）。挿入するポイントはページの最下部（</body>の直前）。リンクの設定は[\[各種設定\]](#)“(3) リンク HTML の設定”を参照して下さい。

= 注意 =

※ 動的なページでページ分割されて最初にアクセスした時と次ページ前ページでアクセスされた時にページの内容が変わっている場合は分割された1ページ目が出力されます。但し動的なページにも関わらず Cache-Control が no-cache 以外の場合は、変換後のコンテンツがメモリキャッシュに保存されるので、上記の動作にはなりません。

※ エリクサーには、javascript で動的に作成、追加されたエレメントなどのコンテンツは判断できません。

24.2 各種設定

(1) ページ分割機能の ON/OFF

説明 : ページ分割機能の ON/OFF を設定します。

初期値: off (ページ分割機能無効)

設定 : elx_split_page on;

(2) 予想ファイルサイズ設定(単位: バイト)

説明 : 各画像フォーマット毎の変換後の画像サイズの想定値(固定)を設定します。
この値を元にページ全体のサイズを計算します。

初期値:

jpeg 15000(byte)

gif 10000(byte)

png 30000 (byte)

other 20000 (byte)

設定例:

```
elx_split_page_img_jpeg_size 15000;
elx_split_page_img_gif_size 10000;
elx_split_page_img_png_size 30000;
elx_split_page_img_other_size 20000;
```

(3) リンク HTML の設定

説明 : ページ分割時に作成されるリンク (前ページ | 次ページなど) のテンプレートを設定します。

%1\$s : 次ページへのリンク

%2\$s : 前ページへのリンク

%3\$d : 現在のページ番号 (1 から始まる)

%4\$d : 分割したページ数

初期値:

```
<div id="elx_split_page">[<a href="%2$s">%3$d</a>]<a href="%1$s">%4$d</a>]</div>
```

設定例 1 :

```
elx_split_page_link_html ' <div id="elx_split_page">[<a href="%2$s">%3$d</a>]<a href="%1$s">%4$d</a>]</div>';
```

設定例 2 :

```
elx_split_page_link_html ' <div id="elx_split_page">[<a href="%2$s">prev</a>|<a href="%1$s">next</a>]</div>';
```

= 注意 =

※ 文字コードは UTF-8。

※ 次ページと前ページへのリンク (HTTP) に "" (チルダ) が含まれる場合、au 端末では「ページは見つかりませんでした (404)」と表示されます。本現象は、au のプロキシサーバーの問題とされます。HTTPS であれば現象は発生しません。

その他設定についての詳細な説明は、[{参考資料}](#) の "設定項目一覧" を参照して下さい。

25 出力切替

出力切替とは、端末の種類や機能仕様毎に変換出力するコンテンツの出力を切り分ける場合等、記述された条件式によりコンテンツの出力を切り替えることができる機能です。PCの場合、コンテンツを無変換で返します。PCに対して出力切替機能はご利用いただけませんのでご注意ください。

25.1 出力切替の記述方法

出力を切り分けたいコンテンツの外側に div タグを作成し、if 属性と output 属性を記述します。if には条件式を、output には条件式にマッチした場合のコンテンツの動作 (yes:出力する、no:出力しない) を指定します。

```
<div if="条件式" output="yes">
  条件式にマッチした場合に表示したい内容
</div>
<div if="条件式" output="no">
  条件式にマッチした場合に表示したくない内容
</div>
```

※ 出力切替に使用された div タグ自体は削除して出力します。

※ if と output はエリクサー独自の属性です。

25.2 条件式の規則

出力切替機能の条件式は W3C XPath 1.0 の 3 章 “Expressions” 以降の記述規則と同じです。以下、<http://www.w3.org/TR/xpath/#section-Expressions> から要約した EBNF による表現です。

```
boolean-expression ::= Expr | PrimaryExpr
Expr                ::= OrExpr
PrimaryExpr         ::= VariableReference
                    | '(' Expr ')'
                    | Literal
                    | Number
                    | FunctionCall
OrExpr              ::= AndExpr
                    | OrExpr 'or' AndExpr
AndExpr             ::= EqualityExpr
                    | AndExpr 'and' EqualityExpr
```

```

EqualityExpr      ::= RelationalExpr
                   | EqualityExpr '=' RelationalExpr
                   | EqualityExpr '!=' RelationalExpr
RelationalExpr   ::= AdditiveExpr
                   | RelationalExpr '<' AdditiveExpr
                   | RelationalExpr '>' AdditiveExpr
                   | RelationalExpr '<=' AdditiveExpr
                   | RelationalExpr '>=' AdditiveExpr
AdditiveExpr     ::= MultiplicativeExpr
                   | AdditiveExpr '+' MultiplicativeExpr
                   | AdditiveExpr '-' MultiplicativeExpr
MultiplicativeExpr ::= UnaryExpr
                   | MultiplicativeExpr MultiplyOperator UnaryExpr
                   | MultiplicativeExpr 'div' UnaryExpr
                   | MultiplicativeExpr 'mod' UnaryExpr
UnaryExpr        ::= UnionExpr
                   | '-' UnaryExpr
Literal          ::= '"' [^"]* '"'
                   | "'" [^']* "'"
Number           ::= Digits ('.' Digits)?
                   | '.' Digits
Digits           ::= [0-9]+
Operator         ::= OperatorName
                   | MultiplyOperator
                   | '+' | '-' | '=' | '!=' | '<' | '<=' | '>' | '>='
OperatorName     ::= 'and' | 'or' | 'mod' | 'div'
MultiplyOperator ::= '*'
FunctionCall     ::= Name '(' (Argument (',' Argument)*)? ')'
Argument         ::= Expr
VariableReference ::= '$' Name
NameStartChar   ::= [A-Z] | "_" | [a-z]
NameChar        ::= NameStartChar | "-" | "." | [0-9]
Name            ::= NameStartChar (NameChar)*

```

25.3 条件式の例

端末の種類（キャリア）が docomo の場合に出力する

```
<div if="$elx_carrier=' docomo' " output="yes">
```

端末の種類がスマートフォンの場合に出力しない

```
<div if="$elx_carrier=' smartphone' " output="no">
```

端末の種類が Softbank または、au の場合出力する

```
<div if="$elx_carrier=' softbank' or $elx_carrier=' au' " output="yes">
```

端末の種類が docomo で Flash Lite に対応している場合出力する

```
<div if="$elx_carrier=' docomo' and $elx_device_sw_flash_lite_enable=' 1' "
output="yes">
```

端末の種類が docomo または au で Flash Lite に対応している場合出力する

```
<div if="($elx_carrier=' docomo' or $elx_carrier=' au' ) and
$elx_device_sw_flash_lite_enable=' 1' " output="yes">
```

ブラウザの表示幅が 240px 以上の端末の場合出力する

```
<div if="$elx_device_image_browser_width &gt;= ' 240' " output = "yes">
```

※ > と < は比較演算子 > と < の文字実体参照です。

使用できる変数

条件式では、リクエストされた端末のスペック（仕様）情報を変数として使用することができます。具体的な項目は、[\[参考資料\]](#)の“変数名・端末スペック通知項目一覧”を参照して下さい。

付録 A. 独自の用語

エリクサー

[{変換対象コンテンツの仕様}](#) 参照

HTML コンテンツ

[{変換対象コンテンツの仕様}](#) 参照

端末プロファイル

[{画像変換パラメータ}](#) 参照

URI パラメータ、パラメータ

[{画像変換パラメータ}](#) 参照

テーブル系タグ

[{出力自動調整方法}](#) 参照

バーチャルホスト設定

[{パラメータの名前指定}](#) 参照

レイアウト変換、ブロック、メソッド、ベーシック変換

[{レイアウト変換}](#) 参照

elxm 接頭辞

[{特定の属性へのメソッド指定}](#) 参照

ul, ol 系のリスト、dl 系のリスト

[{レイアウト変換の内容}](#) 参照

変換元コンテンツ

[{端末スペック通知}](#) 参照

バックエンド

[{端末スペック通知}](#) 参照

付録 B. 参考資料

nginx (エリクサーがベースとしている HTTP サーバ)

<http://nginx.org/>

nginx の設定情報

<http://nginx.org/en/docs/>

弊社エリクサー製品技術資料

- ・ 対応端末一覧
- ・ マークアップ言語変換ルール
- ・ 絵文字変換ルール
- ・ 変数名・端末スペック通知項目一覧
- ・ 設定項目一覧
- ・ コンテンツ追加機能の設定例
- ・ 変換ルールのサンプル
- ・ 変換ルールの設定例
- ・ よくあるご質問

<http://elixir.neu.co.jp/support/>

Firefox

<http://mozilla.jp/firefox/>

NTT ドコモ 「作ろう i モードコンテンツ」

- ・ i モード対応 XHTML
- ・ i モード対応 HTML
- ・ 絵文字
- ・ i-CSS

HTML Tidy

<http://tidy.sourceforge.net/>

CSS level 2

<http://www.w3.org/TR/CSS2/>

XHTML1.0

<http://www.w3.org/TR/xhtml1/>

HTML4.01

<http://www.w3.org/TR/html401/>

ブロック要素/インライン要素

<http://www.w3.org/TR/html401/struct/global.html#h-7.5.3>

XML

<http://www.w3.org/TR/xml/>

XSLT

<http://www.w3.org/TR/xslt/>

XPath

<http://www.w3.org/TR/xpath/>

XPath “3 Expressions”

<http://www.w3.org/TR/xpath/#section-Expressions>

EBNF

Extended Backus-Naur Form

ISO-14977 等も参考にしてください。

HTTP/1.1

RFC 2616

※ これら URL 参照先のページは変更、削除される場合があります。参照先が見つからない場合は、弊社エリクサー・カスタマサポートまでお問い合わせ下さい。

変更履歴

Changes from 1.05 to 1.06	2014-08-04
*) {リストア手順} を修正	
Changes from 1.04 to 1.05	2013-05-21
*) {XSLT PI 機能} を {独自の交換ルール} に変更	
*) nginx の設定情報の URL を変更	
Changes from 1.03 to 1.04	2013-03-19
*) {SSL 秘密鍵、証明書の設定} に注意点を追記	
Changes from 1.02 to 1.03	2012-11-27
*) 文言を修正	
Changes from 1.01 to 1.02	2012-11-19
*) サイト設定手順の {補足} に追記	
Changes from 1.00 to 1.01	2012-10-31
*) 製品技術資料の URL を変更	
Changes from 0.90 to 1.00	2012-09-26
*) 初期導入編、設定&運用編、コンテンツ作成編を統合	

弊社エリクサー・カスタマサポートへのお問い合わせ

電話からは

082-543-2800（直通）

メールからは

elixir@neu.co.jp

お問い合わせ窓口時間：平日 10:00-12:00 13:00-17:30

免責事項

当マニュアルに記載されている会社名、システム名、製品名は一般に各社の登録商標または商標です。