



Next-L Enju インストールマニュアル (Docker編)

Project Next-L

Version 1.4.x

Table of Contents

1. Enju概要	2
1.1. Enjuとは	2
1.2. Enju開発の経緯	2
2. Enjuを動作させる方法	3
2.1. 概要	3
2.2. Dockerについて	5
2.2.1. Dockerの概要	5
2.2.2. Dockerを利用するメリット	5
2.3. インストールの流れ	6
2.3.1. インストールの全体	6
3. 動作環境の確認とDockerのインストール	7
3.1. 動作環境を確認する	7
3.1.1. Windowsが64bit版であるかの確認，使用可能メモリ容量の確認	7
3.1.2. ハードディスク空き容量の確認	8
3.1.3. IPアドレスの確認	10
3.2. Dockerのダウンロードとインストール	15
3.2.1. Dockerのダウンロード	15
3.2.2. Dockerのインストール	15
4. Enjuのインストール	16
4.1. Enjuの入手	16
4.2. コンテナと初期設定ファイルの作成	16
4.3. Dockerの起動	16
4.4. データベースの作成・初期データの投入	16
4.5. Enjuの起動	17
4.6. Enjuの停止	17
4.6.1. Enjuの再起動	17
4.6.2. メールサーバーの設定	17
5. 検索エンジンの設定	19
5.1. 検索時の異体字を同じ文字として扱う方法	19
5.2. 異体字を同じ文字として扱うようにSolrを設定する	19
5.2.1. マッピングファイルを作成	19
5.2.2. マッピングファイルを配置	20
5.2.3. 設定ファイルにファイル名を書く	20
5.2.4. Enjuの再起動とSolrの再インデックス	21
5.2.5. 参考サイト	21
5.3. 異体字の考え方	21
5.3.1. 時代性	21
5.3.2. 異体字のような別字	21
5.3.3. 代用漢字	22
5.3.4. 中国語の簡体字	22
5.3.5. 検索ノイズとの兼ね合い	22
6. 他のコンピュータからの利用設定	24
7. Enjuのアップデート及び停止・起動	25
7.1. Enjuの停止	25
7.2. 最新ソースコードの取り込み	25
7.3. Enjuを（再）起動	25
8. データのバックアップとリストア	26
9. その他（カスタマイズなど）	27
9.1. Enjuを運用するにあたっての留意点・注意点	27

9.2. Enju を停止する	27
9.3. Enju を再起動する	27
9.4. 「条件を詳しく指定して検索」の画面をカスタマイズする	28
9.5. トップの画面の検索フォームをカスタマイズする	28
9.6. 検索結果一覧画面の検索フォームをカスタマイズする	28
9.7. 検索結果一覧画面に表示項目を追加する	28
9.8. トップ画面やヘルプなどに表示する画像を置く	29
9.9. ヘッダーをカスタマイズする	29
9.10. Sitemapsを設定する	30
10. トラブルシューティング	31
10.1. Q. Dockerでプロキシ環境でも使用できる方法をおしえてください。	31
10.2. Q. VT機能が無効になっているというメッセージが表示されます。	31
10.3. Q. 大きなファイルをTSVインポートしようとするとき"413 Request Entity Too Large nginx/1.4.6 (Ubuntu)"といったエラーメッセージが表示されてできません。	31
10.4. Q. 504 Gateway Timeout というエラーメッセージが表示されて画面が表示されません。	31

本マニュアルは、オープンソース統合図書館システム Next-L Enjuのインストール方法について、説明しています。情報システム担当者がNext-L Enjuをインストール作業をするときに、お読みください。

本マニュアルの作成にあたっては、(株)三菱総合研究所の全面的な支援をいただきました。心より感謝いたします。

Chapter 1. Enju概要

Enjuとはどのようなシステムなのか、概要を説明します。

1.1. Enjuとは

Enjuは、国会図書館、公共図書館、大学図書館、専門図書館、学校図書館などにおける図書館業務に必要な全ての機能を提供する統合図書館管理用ソフトウェアです。図書館で働く人々の総意として必要と思われる機能を実装するほか、図書館利用者が求めるWebサービス機能についても実装していきます。

1.2. Enju開発の経緯

実際の社会に影響を与えるシステムの出現を目指すためには、実験用システムが作られれば良いというものではなく、実際に導入可能なシステムを考える必要があります。そのためには、実際の図書館業務で発生する様々な問題と解決法を、図書館員をはじめとして図書館に関わる様々な方々に提供いただき、細かなノウハウまで取り込んだ理想的な図書館システムを考えていくことが必要となります。

しかし、図書館員の方々をはじめとして、大勢の図書館に関わる方々は、皆さんそれぞれに忙しく、長期間継続して意見やノウハウの提供をいただくことは簡単なことではありません。ましてや、新しいアイデアを考えていただくという負担を過度におかけすることは、とてもできることではありません。

そこで、仕様の策定と並行して小規模図書館であれば導入可能な、統合図書館システムをプロトタイプ的に開発し、これを使っていただくことで、アイデアを出していただくことといたしました。このプロトタイプとして、統合図書館システムNext-L Enjuは開発が行われました。

こうして開発された統合図書館システム Next-L Enjuは、幾度かの改良が行われ、すぐれた開発環境の存在と開発者の技量の高さ、そして何よりも開発者および関係者の熱意によって、現在では、単なるプロトタイプの枠組みを超えて、実際の図書館業務に使用することが可能なシステムへと進化しました。2011年11月11日にリリースされた Next-L Enju Leaf Ver1.0.0 からは、正式名称も「Project Next-Lプロトタイプ Enju」からプロトタイプの文字を取り除き、「Next-L Enju」を正式名称としています。 :toc: macro

Chapter 2. Enjuを動作させる方法

Enjuを動作させる方法について説明します。

2.1. 概要

Enjuを実行させるために、Project Next-LではGitHub上でソースコードを公開しているほか、Docker環境用のコンテナも作成・提供しています。これに対応してEnjuをインストールする方法としては、いくつかの方法がありますが、ここでは2つ紹介します(下図を参照)。

1. Enjuを動作させるのに必要な各種ミドルウェアやモジュール群をインストールし、Enjuのプログラムを実行
2. Dockerがインストールされたコンピュータで、EnjuのDockerコンテナを実行

各図書館の状況や開発形態によって、これらの手法から最適な手法を選択することが必要となります。

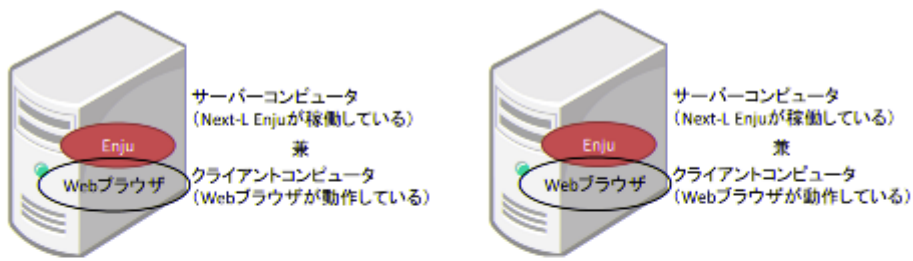
1.の方法はEnjuが動作する環境整備から行っていこうとするもので、実行することができるコンピュータの範囲も広く、取り込むモジュールの選択なども行えるなど最も自由度が高い方法です。コンピュータの能力に応じたパフォーマンスも発揮することができます。その反面、インストールのためにはOSや、Ruby on Railsの開発環境、ネットワークに関わる比較的高度な知識などが必要となります。初心者には、かなりハードルが高いということもできます。各図書館の状況などにあわせ、Enjuの標準的な環境から大幅に手をいれる必要がある場合などには、この方法をとることが必要となることもあると思われます。

2.の方法はWindowsやLinuxなどのような普段使っているOSの上のアプリケーションのひとつとして動くOSの上でEnjuを動かそうとするものです。普段使っているOSの上で異なるOSとアプリケーションを実行するためのソフトウェアを動作させ(これがDockerということになります)、その上でEnjuを動作させることとなります。Docker上で動作するEnju(Enjuコンテナと呼びます)は、Enju本体だけではなく動作環境まで含めた形で提供されるため、Dockerさえ動作するようにすれば、つまり、2.の方法を用いれば、高いITスキルがない人でも、比較的簡単な作業でEnjuを動かすことができます。すなわち、Dockerを利用することにより、基礎的なOS(Windowsなど)の操作と、初歩的なネットワークに関する知識があればインストールが可能です。また、普段は別の用途で使っているPCを必要な時だけEnjuが動作するサーバとして使用することもできます。

さらに、Windows上のWebブラウザ(Microsoft EdgeやGoogle Chrome, Firefoxなど)からDocker上のNext-L Enjuを利用することで、2.の方法を用いれば使い慣れた1台のコンピュータのみで、統合図書館システムNext-L Enjuを運用することもできます。

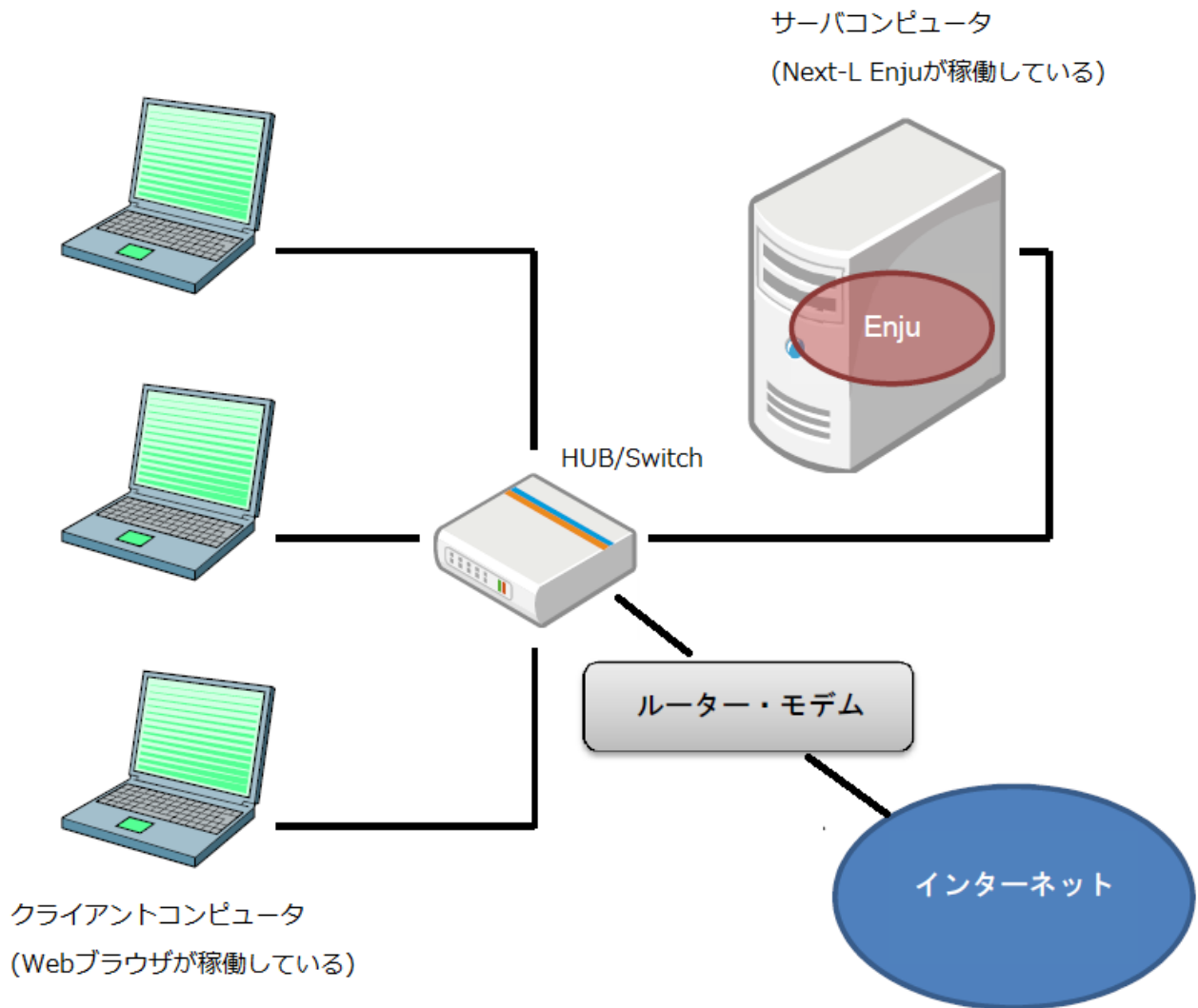
本マニュアルでは、「2. Dockerを用いる方法」について述べます。

まず、「[動作環境の確認とDockerのインストール](#)」と「[Enjuのインストール](#)」の章において、簡便にNext-L Enjuを利用する手法として、1台のコンピュータ上のみでNext-L Enjuを利用する(同じコンピュータ上でサーバとクライアントの両方を動作させる)方法について述べます。もちろん、使用するコンピュータはデスクトップではなくノートブックでも問題ありません。



さらに「他のコンピュータからの利用設定」の章において、設定したNext-L Enjuをネットワークを介して別のコンピュータからも利用する方法について説明していきます。

Next-L Enjuがインストールされたサーバコンピュータにクライアントコンピュータから接続して利用する際、クライアントコンピュータからはブラウザソフトを利用してNext-L Enjuのデータにアクセスします。したがってクライアントコンピュータのOSは問わず、またEnjuの動作はクライアントコンピュータの環境に依存しません。



2.2. Dockerについて

Dockerは、広く利用されているコンテナ実行環境構築・運用ソフトウェアです。

2.2.1. Dockerの概要

Dockerは、Docker社によって提供されているアプリケーション実行環境構築・運用ソフトウェアです。Dockerでは、アプリケーションの入った「コンテナ」を実行することができます。

2.2.2. Dockerを利用するメリット

Dockerを用いた場合、その上で動作するアプリケーションを「コンテナ」、つまりセットアップ済みのファイルとして提供できるメリットがあります。これに対してDockerを使用しない場合は、LinuxなどのOSのインストールからはじめて、Ruby, Ruby on Rails、さらにSolrのようなミドルウェアのインストールが必要になります。またコンテナ環境では、他のパッケージシステムとの競合などに注意を払う必要もありません。比較的簡単に行えるDockerのインストールを行うだけで、Enjuの動作環境の構築を大幅に簡略化することができます。

またDockerは個人用途では無償で提供されており、追加費用なしで動作環境を構築できることも大きなメリットです。

2.3. インストールの流れ

2.3.1. インストールの全体

Dockerを使ったEnjuのインストールは以下の手順で行います。

1. 動作環境の確認
2. Dockerのダウンロード
3. Dockerのインストール
4. Enjuコンテナのダウンロード
5. Enjuコンテナのインストール
6. Enjuコンテナの動作確認
7. Enjuコンテナのネットワーク設定
8. Enjuコンテナの起動 :toc: macro

Chapter 3. 動作環境の確認とDockerのインストール

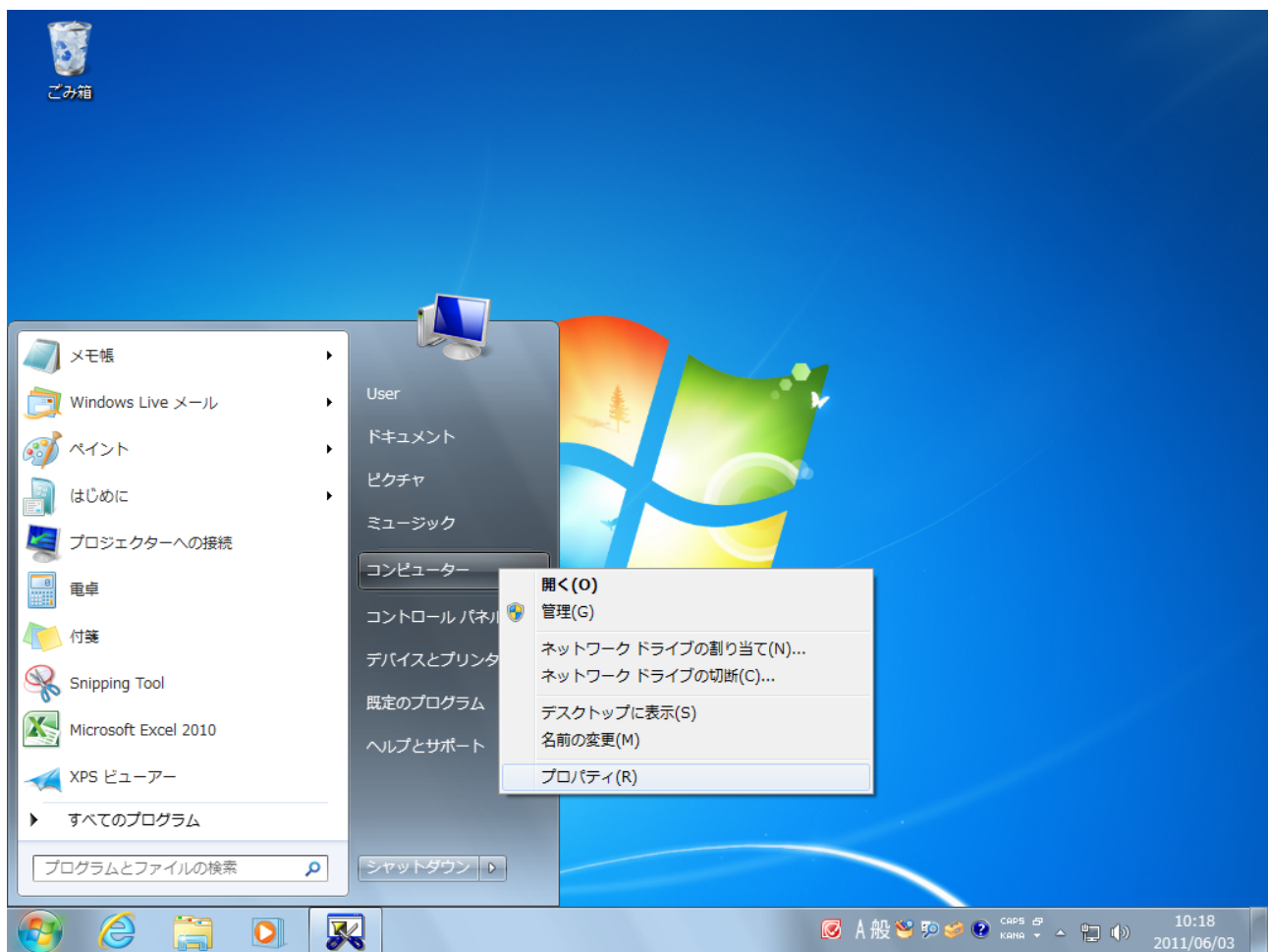
3.1. 動作環境を確認する

作業を開始する前に、Enjuをインストールするコンピュータの環境を確認します。この文書では、Windows 10 でのやり方について説明します。

3.1.1. Windowsが64bit版であるかの確認、使用可能メモリ容量の確認

Windowsが64ビット版であるか、メモリ容量をシステムのプロパティで確認します。※32bit版では動作しません

1. [スタート] ボタンをクリックし、[コンピュータ] を右クリックします。
2. [プロパティ] をクリックします。



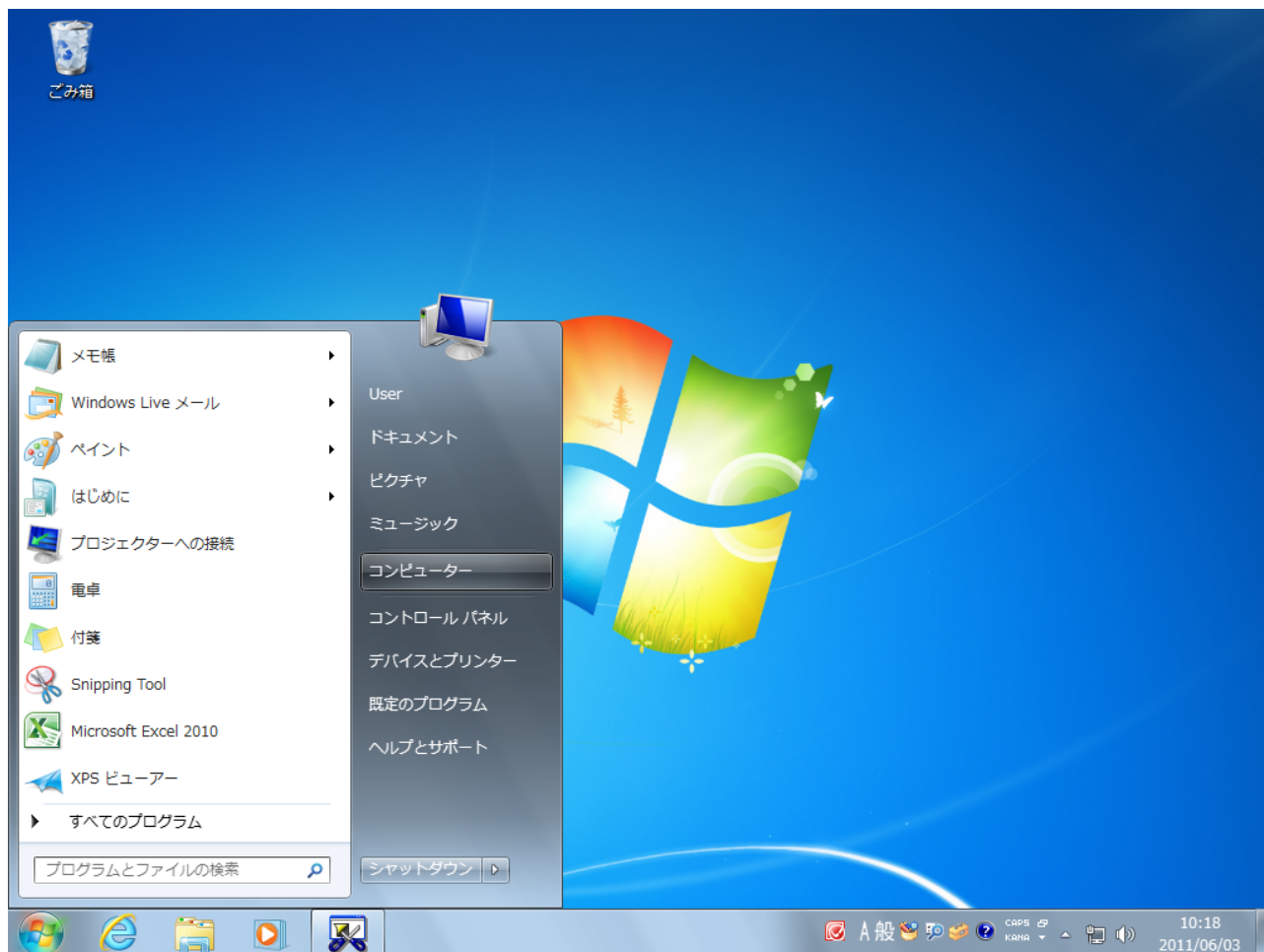
3. 以下の項目を確認します。
 - システムの種類 : 64bitのオペレーティングシステム
 - 実装メモリ : 使用可能容量が 4.00GB 以上



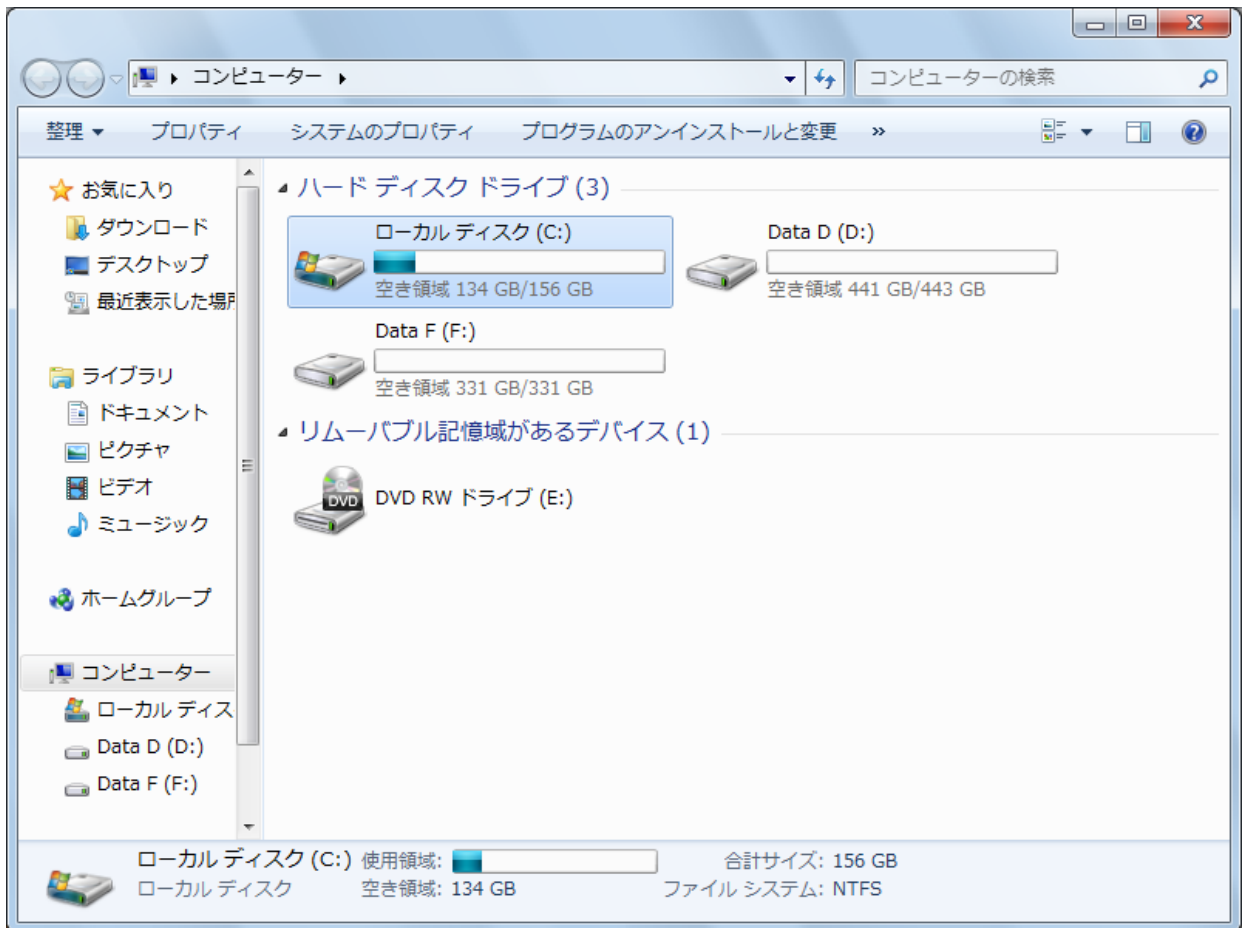
3.1.2. ハードディスク空き容量の確認

Enjuコンテナのインストールと動作に必要なハードディスクの空き容量を確認します。

1. [スタート] ボタンをクリックし， [コンピュータ] を選択します。



2. 以下の項目を確認します。
 - ドライブの空き容量 : 25.0GB以上を推奨



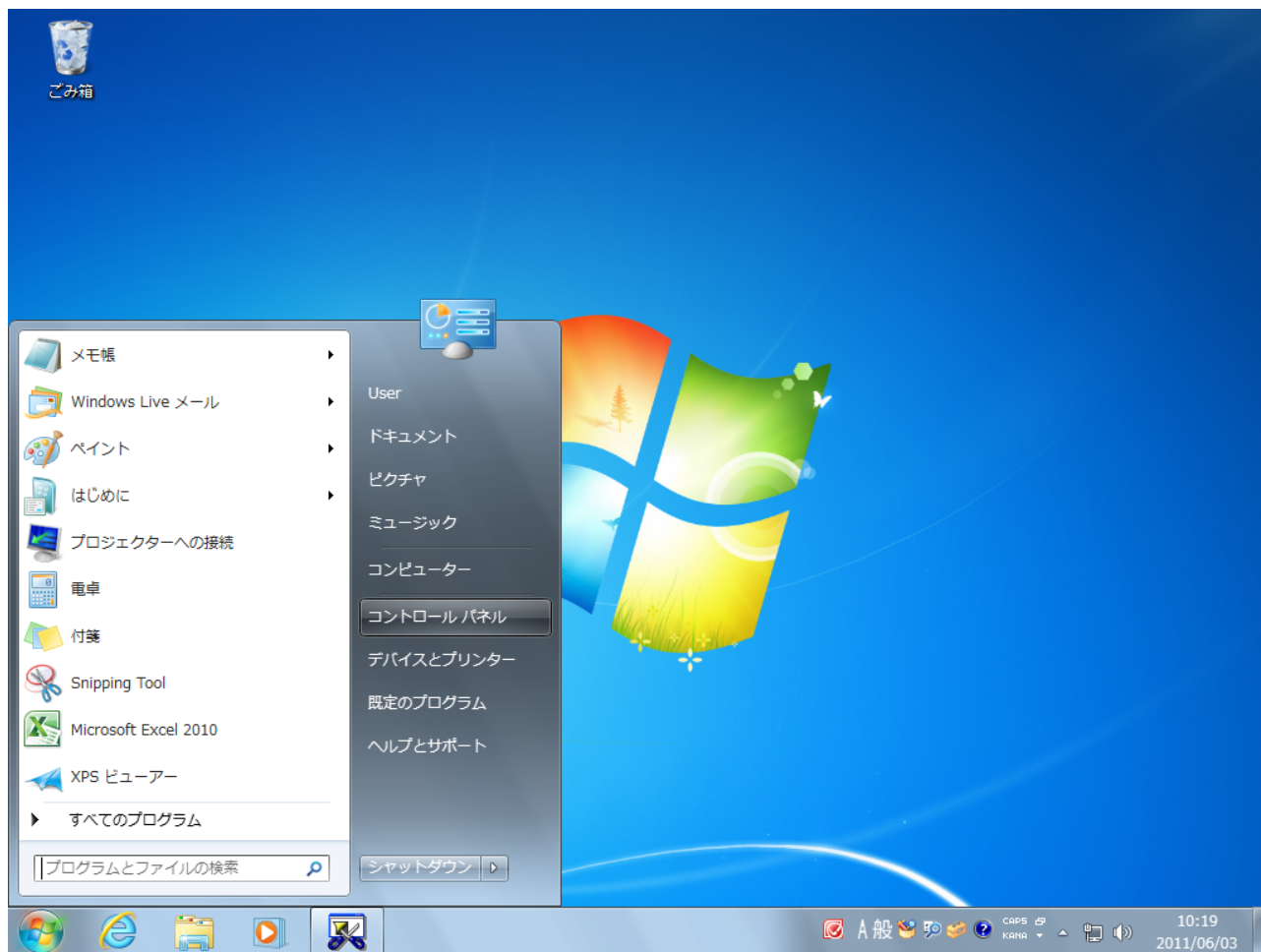
NOTE

ハードディスクに必要な空き容量は、蔵書規模により異なります。25.0GBは蔵書が20万件の規模を想定しています。これよりも蔵書規模が大きい場合は、より多くの容量が必要となります。

3.1.3. IPアドレスの確認

ホストOSのIPアドレスを確認します。

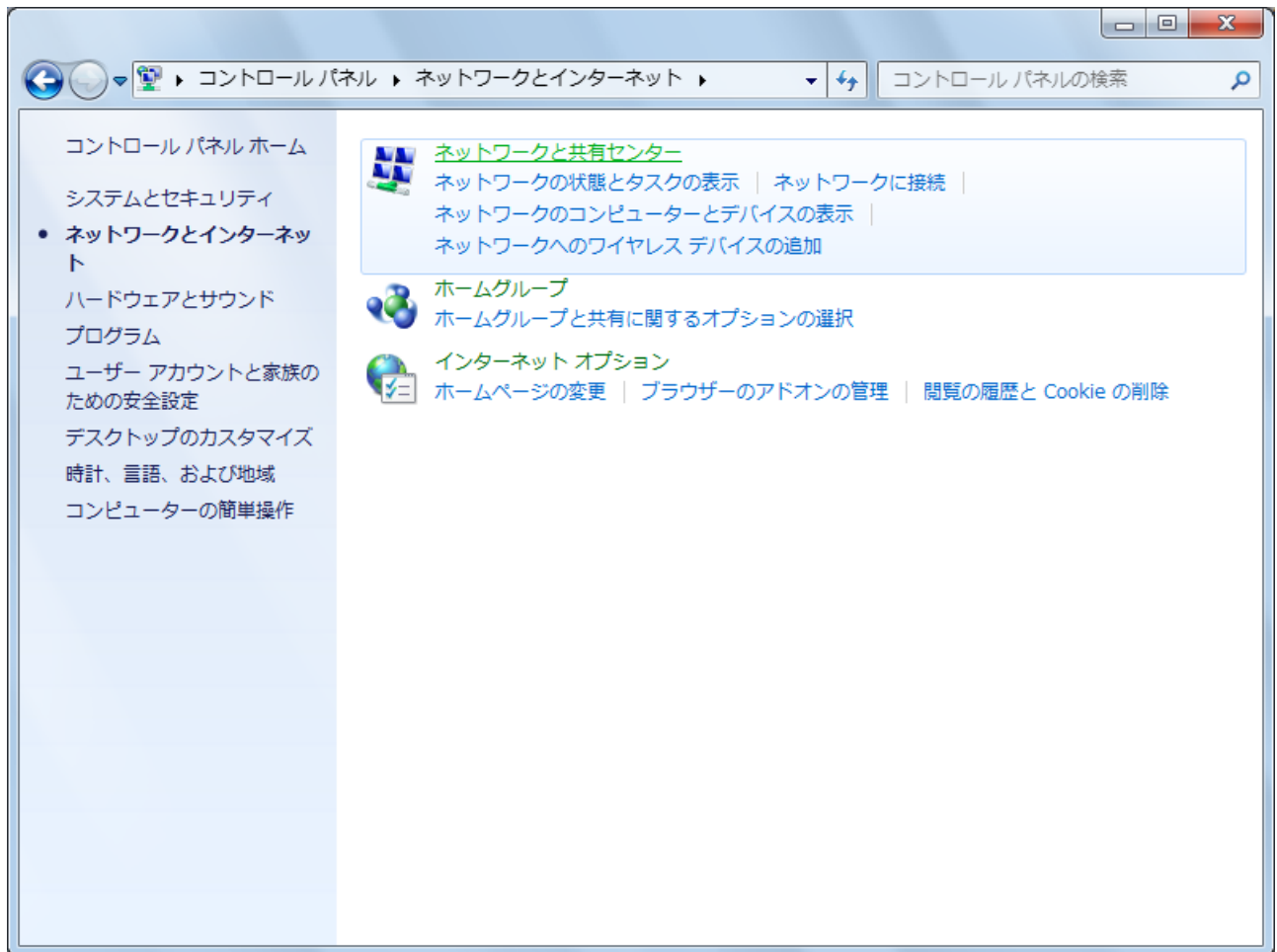
1. [スタート] ボタンをクリックし、[コントロールパネル] を選択します。



2. [ネットワークとインターネット] をクリックします。



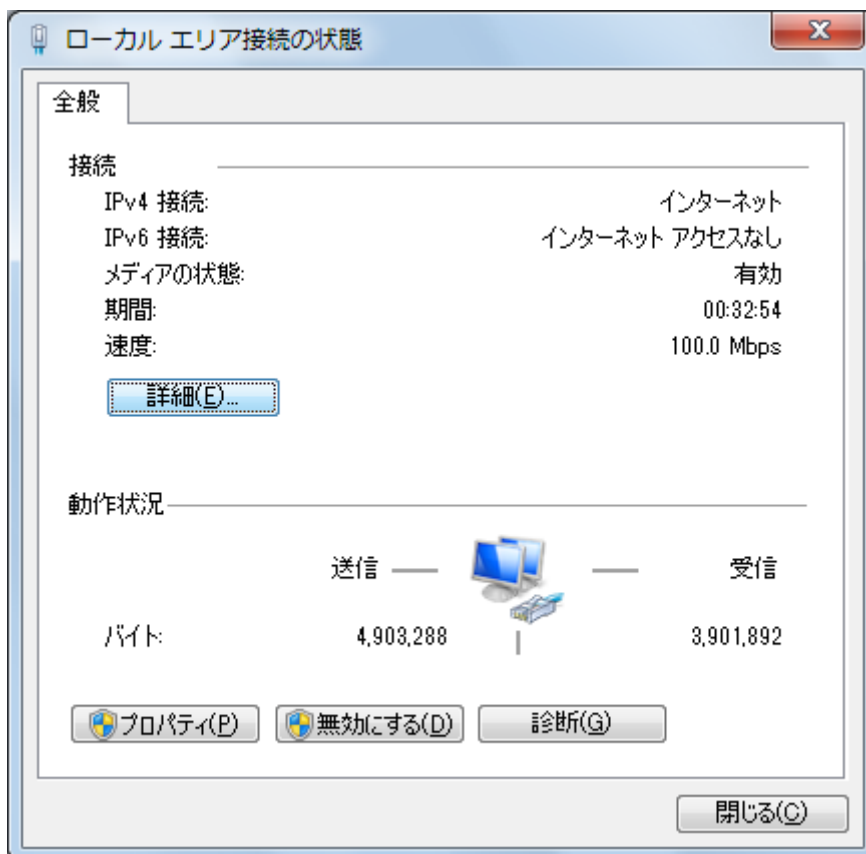
3. [ネットワークと共有センター] をクリックします。



4. [アクティブネットワークの表示] で [ローカルエリア接続] または [ワイヤレスネットワーク接続] をクリックします。

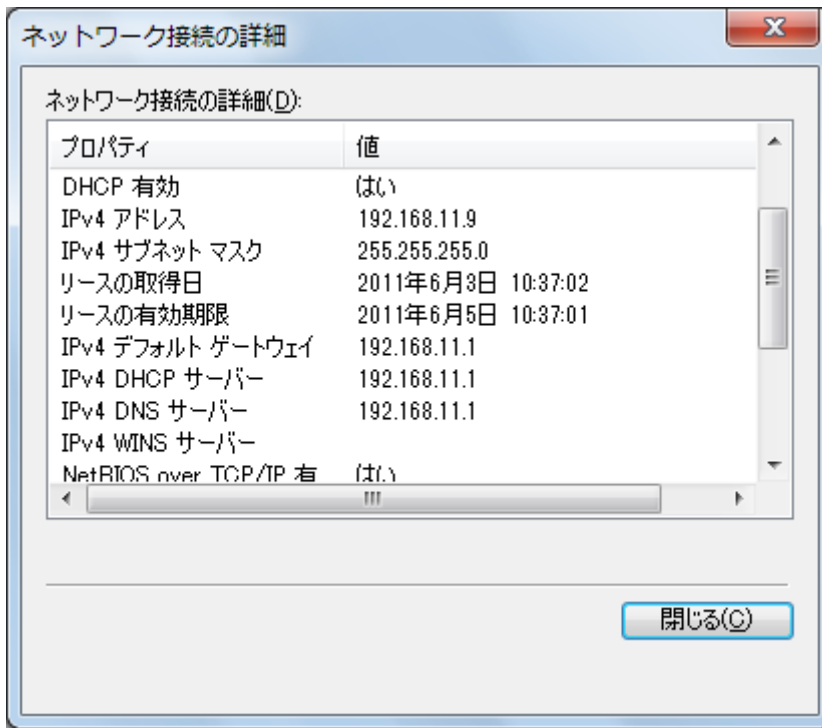


5. [詳細] をクリックします。



6. 表示された画面で、以下の情報をメモなどに記録します。

- DHCPの状態
- IPv4アドレス
- IPv4サブネットマスク
- IPv4デフォルトゲートウェイ
- IPv4 DNSサーバ



3.2. Dockerのダウンロードとインストール

ここではWindowsの場合を説明しますが、Mac、Linuxなど様々なOS上で動きます。

3.2.1. Dockerのダウンロード

Docker DesktopをDocker社のWebサイトからダウンロードします。次のWebページでDownloadsをクリックします。

3.2.2. Dockerのインストール

前項でダウンロードしたファイルを実行し、Docker Desktopをインストールします。 :toc: macro

Chapter 4. Enjuのインストール

4.1. Enjuの入手

Enjuのインストールに必要なパッケージは、すべてネットワーク上で提供されます。

以下のコマンドを入力します。

```
$ git clone https://github.com/next-l/enju_leaf.git -b 1.4
```

4.2. コンテナと初期設定ファイルの作成

以下のコマンドで、コンテナと初期設定ファイルを作成します。

```
$ cd enju_leaf  
$ cp .env.template .env  
$ docker compose build  
$ echo SECRET_KEY_BASE=`docker compose run --rm web bin/rake secret` >> .env
```

4.3. Dockerの起動

Dockerを起動します。

```
$ docker compose up -d
```

4.4. データベースの作成・初期データの投入

以下のコマンドでデータベースを作成します。

```
$ docker compose run --rm web bin/rake db:create  
$ docker compose run --rm web bin/rake db:migrate  
$ docker compose run --rm web bin/rake db:seed
```

CSS・JavaScriptのビルドと、インデックスの更新を行った後、アプリケーションを停止します。

```
$ docker compose run --rm web bin/rake assets:precompile  
$ docker compose run --rm web bin/rake sunspot:reindex  
$ docker compose down
```

4.5. Enjuの起動

Enjuの起動や停止は、enju_leafディレクトリで実行する必要があります。まだenju_leafディレクトリに移動していない場合、以下のコマンドで移動します。

```
$ cd enju_leaf
```

以下のコマンドでEnjuを起動します。

```
$ docker compose up -d
```

Windows上でWebブラウザを起動し、アドレス欄に <http://localhost:8080> と入力してください（ブラウザによってはlocalhost:8080でも可）。

これで、Enjuのサーバに対してアクセスすることが可能になりました。

4.6. Enjuの停止

Enjuを終了(シャットダウン)させる場合には、正しい手順でのシャットダウンを行う必要があります。Dockerをいきなり終了させたり、PCの電源ボタンを押すなどして強制終了させると、使用中のデータベースファイルが壊れて、再起動できなくなるなどの問題が発生する可能性があります。

以下のコマンドを入力します。

```
$ docker compose down
```

Enjuをシャットダウンするタイミングは、図書館の運用方針で決定することができます。起動したら基本的にはずっと動作させ続けていても問題ありません。毎日Enjuコンテナをシャットダウンする、必要な時だけ稼働させるなど、ポリシーに応じた運用をすることができます。

4.6.1. Enjuの再起動

```
$ docker compose down  
$ docker compose up -d
```

これで、再度Enjuが利用できるようになりました。

4.6.2. メールサーバーの設定

パスワードの再発行や資料予約通知などのメール送信機能を使うために必要です。

config/environments/production.rb を以下のように変更します。

```
Rails.application.configure do
```

```
# 中略

# hostオプションはEnjuを動作させているホストのホスト名を指定
config.action_mailer.default_url_options = {host: 'enju.example.jp'}

# SMTPサーバの設定
# address, port, domain, user_name, passwordなどを変更してください
# 不要な設定はコメントアウトできます。設定例は以下も参照してください
# https://railsguides.jp/action\_mailer\_basics.html#mail%E7%94%A8%E3%81%AEaction-mailer%E8%A8%AD%E5%AE%9A
config.action_mailer.delivery_method = :smtp
config.action_mailer.smtp_settings = {
  address: 'yoursmtpserver.example.jp',
  port: 25,
  domain: 'example.jp',
  user_name: 'smtpusername',
  password: 'smtppassword',
  authentication: 'plain',
  enable_starttls_auto: true,
  open_timeout: 5,
  read_timeout: 5
}
end
```

Chapter 5. 検索エンジンの設定

5.1. 検索時の異体字を同じ文字として扱う方法

書誌情報には、「森鷗外／森鷗外」や「内田百閒／内田百閒」、「芥川龍之介／芥川竜之介」といった異体字が混在しています。これらの文字を別のものとして扱うと、検索結果に漏れが生じる可能性があります。たとえば、「森鷗外」で検索すると「森鷗外」が含まれない、といった問題が起きるのです。

こうした問題を解決するために、検索時には「鷗」と「鷗」などの異体字を同じ文字として扱う設定が必要です。

この設定を行うには、検索エンジンで適切な設定を施す必要があります。本章では、Enjuと連携して動作する全文検索エンジン「Apache Solr」での設定方法を解説します。

5.2. 異体字を同じ文字として扱うようにSolrを設定する

SolrのCharFilterという機能を使えば異体字を同じ文字として検索させることができます。

例えば「島」「嶋」「鳶」を検索上同じ文字として扱いたいとします。その場合、代表の文字を一つ、例えば「島」と決めます。そして、「嶋」「鳶」を「島」に変換するような設定を行います。

するとSolrは、書誌情報から検索用データを作成する際に、「嶋」「鳶」を「島」に変換します。

また、検索を行う際も、検索語の中の「嶋」「鳶」を「島」に変換します。

これにより、「大嶋」で「大島」も「大鳶」もヒットするようになります。

5.2.1. マッピングファイルを作成

以下のような形式のマッピングファイルを作ります。

```
"嶋" => "島"  
"鳶" => "島"
```

文字コードはUTF-8、改行はLFとします。ファイル名は例えばja-variants.txtなど好きなように付けます。

ナンバー記号 (#) で始まる行はコメントとみなされます。しかし、以下のような形式のコメントは許されません。

```
"嶋" => "島" # コメント (これはダメ)
```

「⇒」の前後のスペースは略すこともできます。

Unicodeのコード番号で文字を指定することもできます。4桁の16進数で `\uXXXX` のように書きます。

4桁で表せるのは、Unicodeの基本多言語面 (BMP: Basic Multilingual Plane) 内の文字だけです。BMP外の

文字は5桁ないし6桁になりますので、この形式は使えません。文字そのものをじかに書く必要があります。BMP外には、「𠀎」（U+2123D）、「𠀏」（U+2000B）のようなありふれた異体字もあります。

変換元、変換先は2文字以上書くこともできます。よって、

```
"麿" => "麻呂"
"\u5715" => "図書館"
"\u309F" => "より"
```

と書いておけば、

- 「人麿」で「柿本人麻呂」が
- 「図書館用品」で「圖用品研究」が
- 「善藏より聞覚」で「御大工棟梁善藏方聞覚控」が

検索できるようになります。（「圖」〔U+5715〕は図書館学者の杜定友による1926年頃の造字で「圖書館」を一字にしたもの；「方」〔U+309F〕は「より」の合字）

変換元は空文字列にできませんが、変換先は空文字列を指定することができます。

5.2.2. マッピングファイルを配置

Enju インストールディレクトリーの下に solr/conf があります。

前節で作成したマッピングファイルをここにに入れておきます。

5.2.3. 設定ファイルにファイル名を書く

solr/conf ディレクトリーの中に schema.xml というXMLファイルがあります。

このXMLに `fieldType[name="text"]` という要素があり、その直下に `analyzer` という要素があります。その内容に、以下のように `charFilter` 要素を追加します。

```
<fieldType name="text" class="solr.TextField" omitNorms="false">
  <analyzer>
    <charFilter class="solr.MappingCharFilterFactory" mapping="ja-variants.txt" />
```

`mapping` 属性にファイル名を指定します。

ファイル名は

```
<charFilter class="solr.MappingCharFilterFactory" mapping="vars1.txt,vars2.txt"/>
```

のようにカンマ区切りで複数指定することもできます。

複数ファイルに異体字マッピングファイルを分けておくと、たとえば、a) 確かな典拠に基づくメインファ

イルと、b) 館の都合に合わせて追加／修正するカスタマイズファイル、といったようにファイルを分割管理することもできます。

また、漢字の異体字とラテン文字の置き換え（「æ」→「ae」など）を別ファイルにする、といった分け方もできます。

5.2.4. Enjuの再起動とSolrの再インデックス

設定が終わったら、設定を反映するために、Enju を再起動（リスタート）して、書誌情報を再度インデックスします。

```
docker compose down
docker compose up -d
docker compose run --rm web rake sunspot:reindex RAILS_ENV=production
```

これで異体字でも検索ができるようになるはずです。

5.2.5. 参考サイト

[solr.MappingCharFilterFactory](#)

上記ページにCharFilterの仕様などが詳しく書かれています。公式サイトです。

5.3. 異体字の考え方

異体字のマッピングファイルを作るにあたって注意すべきことなどを述べます。

5.3.1. 時代性

現在、「著」と「着」は別字であり、「著しい」「著す」「著作」／「着る」「着く」「着信」などを使い分け、交換できません。

しかし「着」はもともと「著」の異体字でした。幸田文に「著物」という作品があり、もっと時代の新しい三島由紀夫も作品中で「著る」を使っています。

また、「丼」はもともと「井」の異体字ですが、現在ではドンブリの意味でしかふつつ使いません。これを異体字として扱うと、天井（てんじょう）を調べていて天井（てんどん）が出てくることになります。

このように、ある二つの漢字が異体字であるかどうかはそれが使われた時代によって変わります。

5.3.2. 異体字のような別字

「斎」と「齊」は異体字の関係にあります。「齋」と「齎」もそうです。しかし、「斎・齊」と「齋・齎」とは別字です。「書齋」は「書齊」とは書けず、「一斎」は「一齋」とは書けません。

とはいえ、人名（サイトウ）に限って言えばこの四字は異体字のように使われており、異体字だと思っているサイトウさんも少なくないようです。

検索上は異体字かどうか拘泥せず、この四字を同じ文字として扱うのがよいかもしれません。

5.3.3. 代用漢字

「国際連盟」「連合艦隊」はそれが存在した時代には「國際聯盟」「聯合艦隊」と書かれていました。しかし、「聯」は「連」の異体字ではありません。

この場合の「連」は、当用漢字に含まれなかった「聯」を、同音で意味的にもあまり無理のない当用漢字の別字に置き換えたものです。

このような置き換え字は「代用漢字」などと呼ばれ、「午后」→「午後」、「遵守」→「順守」、「附録」→「付録」、「蒐集」→「収集」、「編輯」→「編集」、「沙漠」→「砂漠」、「日蝕」→「日食」、「訊問」→「尋問」、「脈搏」→「脈拍」、「叛乱」→「反乱」、「扮飾」→「粉飾」、「長篇」→「長編」、「哺育」→「保育」、「繙帯」→「包帯」、「拋物線」→「放物線」、「彎曲」→「湾曲」など多数あります。

こういったものも異体字のように扱うかべきどうかは場合によるでしょう。

単語単位で「午后」を「午後」に変換するのでなく文字単位で「后」を「後」に変換する場合、「后」（きさき）による検索で大量の検索ノイズが発生するかもしれません。

5.3.4. 中国語の簡体字

中国の「広東省」は簡体字で「广东省」と書きます。「广」／「広」、「东」／「東」は異体字の関係にあるので、このペアもマッピングファイルに入れておけば、中国語の書誌情報も手軽に検索できてすこぶる便利ではないでしょうか。

ところが、そう簡単ではありません。

「機」の簡体字は「机」です。「葉」は「叶」、「幹」は「干」です。このように、中国語では異体字であるものが、日本語では全くの別字ということがあるので。

このように、どの字とどの字が異体字の関係になるかは、言語によっても異なります。

5.3.5. 検索ノイズとの兼ね合い

上で見たように、異体字マッピングを拡充すれば検索漏れが減らせる一方で検索ノイズが増えることがあります。再現率（検索漏れの少なさ）と適合率（検索ノイズの少なさ）は一般にはトレードオフの関係にあります。

そのため、資料群の特性を考えて異体字の範囲を定める必要があるでしょう。

もう一つ検討すべきことは、検索ノイズにも利用者を悩ませるものとそれほどでもないものがあることです。

異体字の例ではありませんが、「国語辞典」で検索して「中国語辞典」までヒットした場合、利用者はなぜそのような検索ノイズが生じたのか、たいてい理解できます。

しかし、アガサ・クリスティーに「機上のナントカ」（『機上の死』）があったと思って「機上」で検索したら『ナントカ辞典机上版』の類がどっと出てきた、という場合、中国語を学んだことのない利用者には理解不能です。

叶（かのう）さんの著作を検索したら樋口一葉の作品まで出てきた。これも同様です。

自分の操作が悪いのか、システムの不具合なのか。利用者は無用のストレスを与えることになります。

蔵書構成だけでなく、利用者層の特性も考慮して決めるとよいでしょう。

なお、異体字対策については、検索機能でなくデータ側で頑張る方針もあります。異体字の範囲はほどほどにとどめ、書誌情報に異表記を積極的に追加して再現率を上げるやり方です。 :toc: macro

Chapter 6. 他のコンピュータからの利用設定

以下のWebサイトをごらんください。 https://github.com/next-l/enju_leaf/wiki/Production

Chapter 7. Enjuのアップデート及び停止・起動

Enjuは、以下の手順で、Enjuのソースコードが公開されているGitHubから変更されたファイルを取り込み最新版にすることができます。

7.1. Enjuの停止

以下のコマンドを入力します。

```
$ docker compose down
```

7.2. 最新ソースコードの取り込み

1. 以下のコマンドを入力します。

```
$ cd enju_leaf
```

2. 以下のコマンドを入力します。

```
$ git pull  
$ docker compose pull  
$ docker compose build  
$ docker compose run --rm web rake db:migrate  
$ docker compose run --rm web rake assets:precompile
```

7.3. Enjuを（再）起動

以下のコマンドを入力します。

```
$ docker compose up -d
```

Chapter 8. データのバックアップとリストア

https://github.com/next-l/enju_leaf/wiki/Backup を参照してください。

Chapter 9. その他 (カスタマイズなど)

9.1. Enjuを運用するにあたっての留意点・注意点

Enjuサーバは、外部からネットワーク経由でアクセスすることができるサービスとして運用されます。したがって、そのセキュリティ管理などには十分に気を配る必要があります。Enjuサーバ自体は、配信時点での最新の状況に対応できるようにセキュリティ対策を講じていますが、日進月歩のネットワーク環境では、新しいネットワーク上の脅威が発生する可能性もあります。このような状況に対応するため、運用に際しては、特に以下の2つにご注意ください。

1. 最新の配信パッケージを使用すること：Enjuコンテナについても、今後、バージョンアップを重ねるたびに配信を予定しています。機能面での充実というだけでなく、セキュリティ面からも最新のパッケージを使用ください。
2. 不必要な外部からのアクセスを禁止すること：場合によっては、ファイアウォールの導入やリモートルータでのフィルタリングについても検討した方が良いでしょう。

9.2. Enju を停止する

この章では頻繁にEnjuを停止する方法の説明がありますので、停止する方法についてまず説明しておきます。

1. Enjuをインストールしたディレクトリに移動します。

```
cd enju_leaf
```

2. Enju を停止します。

```
docker compose down
```

9.3. Enju を再起動する

この章では頻繁にEnjuを再起動する方法の説明がありますので、再起動する方法についてまず説明しておきます。

1. Enjuをインストールしたディレクトリに移動します。

```
cd enju_leaf
```

2. Enjuを再起動します。

```
docker compose up -d
```

9.4. 「条件を詳しく指定して検索」の画面をカスタマイズする

Enju 「条件を詳しく指定して検索」の画面をカスタマイズする方法を説明します。設定する画面がないため、これはサーバー上の作業をする必要があります。

1. [Enjuを停止する](#) を実行して、Enjuを停止します。
2. 以下のファイルの中身を編集して保存します。

```
app/views/page/advanced_search.html.erb
```

3. [Enjuを再起動する](#) を実行してEnjuを再起動し、「条件を詳しく指定して検索」の画面をリロードして確認します。修正が必要なら 1. と 2. の手順を行います。

9.5. トップの画面の検索フォームをカスタマイズする

Enju トップの画面の検索フォームをカスタマイズする方法を説明します。設定する画面がないため、これはサーバー上の作業をする必要があります。

1. [Enjuを停止する](#) を実行して、Enjuを停止します。
2. 以下のファイルの中身を編集して保存します。

```
app/views/page/_search_form.html.erb
```

3. [Enjuを再起動する](#) を実行してEnjuを再起動し、トップの画面をリロードして確認します。修正が必要なら 1. と 2. の手順を行います。

9.6. 検索結果一覧画面の検索フォームをカスタマイズする

Enju 検索結果一覧画面の検索フォームをカスタマイズする方法を説明します。設定する画面がないため、これはサーバー上の作業をする必要があります。

1. [Enjuを停止する](#) を実行して、Enjuを停止します。
2. `app/views/manifestations/_index_form.html.erb` を編集します。
3. [Enjuを再起動する](#) を実行してEnjuを再起動し、検索して検索結果一覧画面を確認します。修正が必要なら 1. と 2. の手順を行います。

9.7. 検索結果一覧画面に表示項目を追加する

Enju 検索結果一覧画面に表示される書誌情報や所蔵情報の表示内容をカスタマイズする方法を説明します。設定する画面がないため、これはサーバー上の作業をする必要があります。

1. [Enjuを停止する](#) を実行して、Enjuを停止します。
2. `app/views/manifestations/_manifestation.html.erb` を編集します。例えば、件名などを追加表示し

たい場合は以下のようなコード片を挿入します:

```
<%- manifestation.subjects.each do |subject| -%>
  <%= link_to "#{subject.subject_heading_type.display_name.localize}:
  #{subject.term}", manifestations_path(query: "subject_sm:\#{subject.term}\") -%>
<%- end -%>
```

3. **Enjuを再起動する** を実行してEnjuを再起動し、検索して検索結果一覧画面を確認します。修正が必要なら 1. と 2. の手順を行います。

9.8. トップ画面やヘルプなどに表示する画像を置く

1. 置きたい画像を用意します。（ここでは例として logo.png とします）
2. インストールしてあるEnju の app/assets/images/ 以下に画像ファイルを置きます。※ custom フォルダを作成し、その下にファイルを置くことを推奨します。

NOTE

- フォルダを作成しその下にファイルを置くことも可能です。
- ファイル名やフォルダ名は任意に作成できます（ただし、Enjuが使用するものと衝突する場合は動作保証しかねます）。
- customフォルダ以下のファイルはEnjuが用意した画像と衝突しないことが保証されます。

3. 以下のコマンドを実行します。

```
$ bundle exec rake assets:precompile
```

4. 参照するURLについて

以下のようなURLになりますのでこのURLを使って参照することができます。

- app/assets/images/ 以下に、 custom フォルダを作成して、その下に logo.png を置いた場合
- URL例（デモサーバー） : <https://enju.next-l.jp/assets/custom/logo.png>
- URL例（コンテナ） : <http://localhost:8080/assets/custom/logo.png>

※ 画像ファイルをブラウザから置けるようにする機能を開発予定です（[詳細 #1133](#)）。

9.9. ヘッダーをカスタマイズする

ヘッダーをカスタマイズする方法を説明します。現在は、ヘッダーに表示されるタイトルしか変更できないため、たとえば、バナー画像を使いたい場合などは、サーバー上の作業をする必要があります。

1. **Enjuを停止する** を実行して、Enjuを停止します。
2. バナー画像があれば、画像ファイルをEnju に置きます。
 - 画像ファイルを置く方法は、[トップ画面やヘルプなどに表示する画像を置く](#)を参照してください。

- 説明で使う例として：置いたファイル：custom/logo.png
 - バナー画像の大きさは横：640ピクセル 縦：65ピクセル にするとちょうどよいです。大きすぎるとはみ出て表示されます。
3. `app/views/page/_header.html.erb`を編集します。例えば、バナー画像をタイトルの代わりにつけた場合は、以下の記述を変更します。

```
<h1><%= link_to image_tag('custom/logo.png'), root_path %></h1>
```

4. [Enjuを再起動する](#) を実行してEnjuを再起動し、リロードして画面（どの画面でもよい）を確認します。修正が必要なら 1. と 2. 3.の手順を行います。

9.10. Sitemapsを設定する

以下のコマンドでサイトマップを作成・更新します。ファイルは`public/sitemap.xml.gz`に作成されます。定期的にサイトマップを更新する場合、以下のコマンドをcronなどに登録してください。

```
docker compose run --rm web rake sitemap:create
```

サイトマップは以下のようなURLでアクセスできます。

<https://enju.example.jp/sitemap.xml.gz>:toc: macro

Chapter 10. トラブルシューティング

10.1. Q. Dockerでプロキシ環境でも使用できる方法をおしえてください。

(TBD)

10.2. Q. VT機能が無効になっているというメッセージが表示されます。

(TBD)

10.3. Q. 大きなファイルをTSVインポートしようとするとき"413 Request Entity Too Large nginx/1.4.6 (Ubuntu)"といったエラーメッセージが表示されてできません。

(TBD)

10.4. Q. 504 Gateway Timeout というエラーメッセージが表示されて画面が表示されません。

設定の誤り等により起動時にエラーが発生している場合、ログファイルを確認する必要があります。以下のコマンドで、エラーメッセージが出ていないかを確認してください。

```
docker compose logs -f web
```