



Copyright © 2014 NTT DATA INTRAMART CORPORATION

目次

- 1. 改訂情報
- 2. はじめに
 - 2.1. 本書の内容
 - 2.2. 対象読者
 - 2.3. 本書に記載されている外部サイトのURL
 - 2.4. 用語解説
 - 2.5. 本書の構成
- 3. Solrの概要
 - 3.1. Apache Solr とは
 - 3.2. RDBとの比較
- 4. Solrのセットアップ
 - 4.1. 前提条件
 - 4.2. セットアップ手順
 - 4.3. 利用するJavaのバージョンについて
- 5. Solrの設定
 - 5.1. コアの追加・削除 (solr.xml)
 - 5.2. コアごとの設定
- 6. Solrの操作
 - 6.1. インデックスの削除
 - 6.2. Solrの移行
 - 6.3. Solrのバックアップ

改訂情報

変更年月日	変更内容
2014-05-30	初版
2014-12-01	第2版 下記を追加・変更しました <ul style="list-style-type: none">「利用するJavaのバージョンについて」に「Java SE Development Kit 7u55」以降ではJVM起動オプションを指定する必要がない旨を追加「Solr管理画面の確認」に説明を追加
2017-04-01	第3版 下記を追加・変更しました <ul style="list-style-type: none">「Solrのセットアップ」のセットアップツールのダウンロード方法の説明を修正

はじめに

本書の内容

本書では IM-ContentsSearch for Accel Platform で利用している全文検索エンジンサーバの Apache Solr について説明します。

IM-ContentsSearch モジュールの利用方法は「[intra-mart Accel Platform セットアップガイド](#)」 - 「[IM-ContentsSearch](#)」を参照してください。

テナント環境セットアップ時の設定については「[intra-mart Accel Platform セットアップガイド](#)」 - 「[Apache Solr接続情報](#)」を参照してください。

対象読者

以下の利用者を対象としています。

- IM-ContentsSearch をセットアップする方
- IM-ContentsSearch を運用・管理する方

本書に記載されている外部サイトのURL

本書内で記載されている外部URLは 2014年5月 現在のものとなります。

用語解説

本書中の表記について以下に記載します。

- IM-ContentsSearch for Accel Platform を省略して IM-ContentsSearch と表記します。
- Apache Solr を省略して Solr と表記します。
- リレーショナルデータベース を省略して RDB と表記します。
- Apache Solr がデプロイされたWeb Application Serverを Solrサーバ と表記します。
- セットアップツールのファイルを展開したディレクトリを %SOLR_SETUP_HOME% と表記します。
- Resin がインストールされているディレクトリを %RESIN_HOME% と表記します。
- Tomcat がインストールされているディレクトリを %TOMCAT_HOME% と表記します。
- Web Application Server内に配置したApache Solrのディレクトリを %SOLR_HOME% と表記します。

本書の構成

本書は以下のような構成となっています。

- [Solrの概要](#)

Solr の概要・基本的な知識を紹介します。

- [Solrのセットアップ](#)

IM-ContentsSearch を利用するための Solr のセットアップ方法について説明します。

- [Solrの設定](#)

Solr の設定項目について説明します。

- [Solrの操作](#)

IM-ContentsSearch を利用するにあたって必要となる Solr の運用上の手順などについて説明します。

Solrの概要

ここでは Solr の概要や、基本的な知識について紹介します。

項目

- [Apache Solr とは](#)
- [RDBとの比較](#)

Apache Solr とは

Apache Solr は、高性能の全文検索エンジンライブラリ Lucene を使って構築されたJavaベースの全文検索エンジンサーバです。

Solr は全文検索を行う対象となる文書・データから索引語を抽出して、文書群への位置情報を単語ごとに保持した転置インデックスを保持します。

この転置インデックスをあらかじめ作成しておくことで、RDBでは実現が難しい高速な全文検索が実現可能となります。

Solr は Tomcat などのサーブレットコンテナ上にデプロイして利用します。



コラム

Solr についての詳しい情報は、Apache Software Foundation のWeb サイトでご確認ください。

<http://lucene.apache.org/solr/> (English)

また「Solr Wiki」では、Solr に関する様々な資料が掲載されていますのでご活用ください。

<http://wiki.apache.org/solr/> (English)

RDBとの比較

RDB と Solr における用語の対比表は以下の通りです。

用語比較

RDB	Solr	備考
スキーマ	→ コア	詳細については次項にて説明します。
テーブル	→ スキーマ	詳細については次項にて説明します。
レコード	→ ドキュメント	IM-ContentsSearch では コンテンツ という形でラップしています。
カラム	→ フィールド	

RDB	Solr	備考
プライマリキー	→ IDフィールド	
ユーザ	→ なし	
コミット	→ コミット	詳細については次項にて説明します。
ロールバック	→ ロールバック	詳細については次項にて説明します。
なし	→ 最適化	詳細については次項にて説明します。

詳細説明

■ コア

RDBのスキーマに相当する機能で、1つの Solr 上に複数のコアを保持することが可能です。Solr に対して複数コアの設定を行った場合、コアごとにアクセスするURLが異なります。尚、IM-ContentsSearch 用に配布している Solr では、標準で **default** という名前のコアを定義しています。

■ スキーマ

RDBのテーブルに相当する機能です。Solr では1コアに1スキーマのみ定義可能です。スキーマ上のプライマリキーに相当するIDフィールドは、システムを通して一意となる値を設定する必要があります。

また、Solr のスキーマには以下の特徴があります。

1. 配列型の値をサポートしている
2. 動的フィールド（データ型のみ指定し後から自由に追加できるフィールド）がある

以上の特徴により、RDBよりも柔軟なデータ構造を持たせることができます。

■ 更新処理

IM-ContentsSearch で利用している Apache Solr 3.6.x にはRDBのupdate文のようにレコード内の特定カラムを更新する機能がありません。

そのため、いずれかの値に変更があった場合には該当のドキュメントを再度作成する必要があります。

■ トランザクション

Solr には分散トランザクションの概念がありません。

そのため複数のプロセスから同時に登録処理が行われた場合、先行するプロセスのコミット処理によって、後続するプロセスが登録したデータが途中でコミットされる可能性があります。

この問題を回避するために IM-ContentsSearch では、クローラジョブは1つのジョブネットにまとめ、更にそのジョブネットは並列実行を許可しない設定としています。

■ コミット

Solr におけるコミットは、登録済みデータを検索結果に反映させる処理です。

RDBのようにトランザクション処理を確定するためのコミット処理ではありません。

- ロールバック

Solr におけるロールバックは、登録済みでコミット前のデータを破棄する処理です。先述の通り Solr には分散トランザクションの概念がないため、複数のプロセスから同時に登録処理が行われていた場合、別プロセスが登録したデータも削除されます。

- 最適化

Solr ではコミット処理を行うたびに索引情報を保持したバイナリファイルが作成されるため、コミットを繰り返すとファイル数が増え続けます。ファイル数が増大することで、以下の問題が発生します。

1. 検索時にロードするバイナリファイルが増加しI/O負荷が高くなることで、検索速度が低下する。
2. ファイルディスクリプタ、またはファイルハンドルの消費量が多くなり、場合によっては不足する可能性がある。

この問題を回避するために、Solr には最適化 (optimize) 処理が用意されています。最適化処理を実行することで、バイナリファイルを最小限にまとめ直し、上記の問題を回避することが可能です。

IM-ContentsSearch では、以下の方法で Solr に対して最適化処理を実行することが可能です。

1. クローラジョブネットの最後に最適化ジョブを実行する。 **(推奨)**
2. APIから最適化処理を実行する。 **(非推奨)**
3. ジョブの終了処理にて上記の最適化APIを実行する。 **(非推奨)**



注意

最適化処理はバイナリファイルのサイズ、すなわち登録されたデータの量によって処理時間が増加するため、実行を最小限にする必要があります。すべてのコミット処理が終了するタイミングで最適化処理を実行するために、ジョブネットの実行ジョブの最後に最適化ジョブを設定することを推奨します。

Solrのセットアップ

IM-ContentsSearch を利用するための Solr のセットアップ方法について説明します。

項目

- 前提条件
- セットアップ手順
 - セットアップツールの取得・展開
 - セットアップツールのコマンド一覧
 - 環境構築手順
 - Jetty（開発向け）
 - Resin
 - Tomcat
 - Solr管理画面の確認
- 利用するJavaのバージョンについて
 - Jettyを利用する場合

前提条件

- リリースノートに記載されているシステム要件を満たしている必要があります。
 - 詳細は「リリースノート」 - 「[Apache Solr システム要件](#)」を参照してください。



注意

Solr はディスクアクセスが大量に発生する場合があります。

そのため、intra-mart Accel Platform がセットアップされたサーバと同一筐体に構築した場合、パフォーマンスに影響が出る可能性があります。

運用環境では intra-mart Accel Platform と Solr を別サーバに構築することを強く推奨します。

セットアップ手順

IM-ContentsSearch では、Solr のセットアップツールを提供しています。

セットアップツールで構築される Solr は、IM-ContentsSearch 専用のカスタマイズが設定済みの状態です。

セットアップツールを利用することで以下の処理を行うことが可能です。

- アプリケーションサーバ「Jetty」を利用した構築および起動
- アプリケーションサーバ「Resin」上へ構築するためのファイル群の出力
- アプリケーションサーバ「Tomcat」上へ構築するためのファイル群の出力

i コラム

弊社から提供しているセットアップツールはアプリケーションサーバの Jetty を同梱しています。

Jetty は主に開発・検証用の利用を想定しています。

Solr を構築するアプリケーションサーバ製品は任意に選択することが可能ですが、運用環境では Resin または Tomcat の利用を推奨します。

セットアップツールの取得・展開

1. セットアップツールの取得

Solr のセットアップツールを取得します。

セットアップツールは以下のURLよりダウンロードできます。ダウンロードには製品のライセンスキーが必要です。

[プロダクトファイルダウンロード](#)

2. セットアップツールの展開（解凍）

セットアップツールのファイル <solr_setup.zip> を取得後、任意のパスに展開します。

本書では、例として以下のディレクトリを指定します。

- Windowsの場合

```
C:\solr\solr_setup
```

- Linuxの場合

```
/usr/local/solr
```

セットアップツールのファイルを展開したディレクトリを、以下 %SOLR_SETUP_HOME% と略します。

セットアップツールのコマンド一覧

以下のコマンドを入力してセットアップツールを実行するとセットアップツールのコマンド一覧とその説明が表示されます。

- Windowの場合

```
C:\solr\solr_setup> setup.bat
```

- Linuxの場合

```
/usr/local/solr/setup.sh
```

以下はWindowsでの出力例です。

```

C:\solr\solr_setup>
C:\solr\solr_setup> setup.bat
Buildfile: C:\solr\solr_setup\build\build.xml

usage:
  [echo] Welcome to the Apache Solr Installation!
  [echo] Use " to display usage
  [echo] Use 'usage' to display usage
  [echo] Use 'jetty' to download the jetty zip file, solr zip file and the solr.war and unzip the zip
files and config the jetty port
  [echo] Use 'jetty.start' to download the jetty zip file, solr zip file and the solr.war and unzip
the zip files and config the jetty port and start the jetty
  [echo] Use 'resin' to download the solr zip file and the solr.war for resin and unzip the zip file
  [echo] Use 'tomcat' to download the solr zip file and the solr.war for tomcat and unzip the
zip file

BUILD SUCCESSFUL
Total time: 1 second

```

セットアップツールの各コマンド一覧と説明は以下の通りです。

コマンドは、`setup.bat`（Windowsの場合）または `setup.sh`（Linuxの場合）のパラメータとして指定します。

- パラメータを指定するコマンドの例

Windowの場合 `C:\solr\solr_setup> setup.bat resin`

Linuxの場合 `/usr/local/solr/setup.sh resin`

* - (なし) - セットアップツールの説明を表示します。

- ■ usage
 - セットアップツールの説明を表示します。
- ■ jetty
 - Solr が利用可能な状態の Jetty を構築します。
起動ポートは <%SOLR_SETUP_HOME%/build_user.properties>に指定されたポートに変更されています。デフォルトの起動ポートは `8983` です。
- ■ jetty.start
 - jettyコマンドを実行後、アプリケーションサーバ Jetty を起動します。
- ■ resin
 - Resin へのセットアップに必要なファイル群を `dist` ディレクトリへ出力します。
- ■ tomcat
 - Tomcat へのセットアップに必要なファイル群を `dist` ディレクトリへ出力します。

環境構築手順

Jetty（開発向け）

Jetty を利用してSolrサーバを構築する場合は以下の手順を行ってください。

1. コマンドの実行

- Windowsの場合、コマンドプロンプトを起動して以下のコマンドを実行します。

```
C:\solr\solr_setup> setup.bat jetty
```

- Linuxの場合、以下のコマンドを実行します。

```
/usr/local/solr/setup.sh jetty
```

処理が完了するとセットアップに必要なファイル群が <%SOLR_SETUP_HOME%/dist/jetty> ディレクトリに出力されます。

2. サーバの起動

- Windowsの場合

```
C:\solr\solr_setup> setup.bat jetty.start
```

- Linuxの場合

```
/usr/local/solr/setup.sh jetty.start
```

コラム

Jettyの起動時、以下のようなエラーが発生する場合、ポート番号の設定を変更してください。

<%SOLR_SETUP_HOME%/dist/jetty/jetty/etc/jetty.xml> にてポート番号の設定ができます。

エラーログ出力例

```
2012-09-24 13:34:33.043:WARN::failed SocketConnector@0.0.0.0:8983:
java.net.BindException: Address already in use: JVM_Bind
2012-09-24 13:34:33.044:WARN::failed Server@9aba32:
java.net.BindException: Address already in use: JVM_Bind
2012-09-24 13:34:33.045:WARN::EXCEPTION
java.net.BindException: Address already in use: JVM_Bind
```

Resin

Resin を利用してSolrサーバを構築する場合は以下の手順を行ってください。

1. コマンドの実行

- Windowsの場合、コマンドプロンプトを起動して以下のコマンドを実行します。

```
C:\solr\solr_setup> setup.bat resin
```

- Linuxの場合、以下のコマンドを実行します。

```
/usr/local/solr/setup.sh resin
```

処理が完了するとセットアップに必要なファイル群が `<%SOLR_SETUP_HOME%/dist/resin>` ディレクトリに出力されます。

2. 出力ファイルのコピー

`<%SOLR_SETUP_HOME%/dist/resin/resin>` ディレクトリ配下を `<%RESIN_HOME%>` へコピーします。

3. ホームロケーションの設定

`<%RESIN_HOME%/conf/resin.xml>` の `</resin>` の前に `<%SOLR_SETUP_HOME%/dist/resin/env-entry.txt>` の内容を追加します。

`env-entry.txt`

```
<env-entry>
  <env-entry-name>solr/home</env-entry-name>
  <env-entry-type>java.lang.String</env-entry-type>
  <env-entry-value>./solr</env-entry-value>
</env-entry>
```

4. サーバの起動

Resinの起動に関しては「[intra-mart Accel Platform セットアップガイド](#)」 - 「[Web Application Server の起動・停止](#)」を参照してください。

Tomcat

Tomcat を利用してSolrサーバを構築する場合は以下の手順を行ってください。

1. コマンドの実行

- Windowsの場合、コマンドプロンプトを起動して以下のコマンドを実行します。

```
C:\solr\solr_setup> setup.bat tomcat
```

- Linuxの場合、以下のコマンドを実行します。

```
/usr/local/solr/setup.sh tomcat
```

処理が完了するとセットアップに必要なファイル群が `<%SOLR_SETUP_HOME%/dist/tomcat>` ディレクトリに出力されます。

2. 出力ファイルのコピー

`<%SOLR_SETUP_HOME%/dist/tomcat/tomcat>` ディレクトリ配下を `<%TOMCAT_HOME%>` へコピーします。

3. URLEncodingの設定

<%TOMCAT_HOME%/conf/server.xml> の `Connector` タグに、属性 `URLEncoding="UTF-8"` を追加します。

設定例 (Tomcat7.0.30)

```
<Connector port="8080" protocol="HTTP/1.1"
  connectionTimeout="20000"
  redirectPort="8443"
  URLEncoding="UTF-8"/>
```

4. サーバの起動

Tomcatの起動手順については構築した環境に応じた方法で起動してSolrが動作していることを確認してください。

Solr管理画面の確認

Solr管理画面へ接続し、「search」ボタンが正常に動作することを確認します。

1. 以下のURLをWebブラウザのアドレスバーに入力して「Solr管理」画面が表示できることを確認します。

```
http://<HOST>:<PORT>/solr/default/admin/
```

図 Solr管理画面

2. QueryStringのテキストエリアに「*:*」と入力されていることを確認し、「Search」ボタンをクリックします。
3. 検索に成功した場合、ブラウザに以下のようなXMLが表示されます。

```

<response>
  <lst name="responseHeader">
    <int name="status">0</int>
    <int name="QTime">58</int>
    <lst name="params">
      <str name="hl.snippets">5</str>
      <str name="hl.useFastVectorHighlighter">>true</str>
      <str name="echoParams">all</str>
      <str name="sort">record_date desc</str>
      <str name="start">0</str>
      <str name="q">*:*</str>
      <str name="rows">10</str>
      <str name="indent">on</str>
      <str name="version">2.2</str>
    </lst>
  </lst>
  <result name="response" numFound="0" start="0"/>
</response>

```

利用するJavaのバージョンについて

「Java SE Development Kit 7u40」、 「Java SE Development Kit 7u45」、 「Java SE Development Kit 7u51」 のバージョンで Solr を利用する場合、Solr のインデックスが破損して正常に動作しない不具合が確認されています。

そのため、不具合を回避するために下記のJVM起動オプションを必ず指定してください。

```
-XX:-UseSuperWord
```

「Java SE Development Kit 7u25」以前のバージョンを利用する場合は指定する必要はありません。また、「Java SE Development Kit 7u55」以降のバージョンをご利用になる場合についても、上記問題は発生しません。

JVM起動オプションの指定方法については、利用する各アプリケーションサーバごとの設定方法を確認してください。

コラム

不具合の詳細については、Java SE Development Kit 7u40 のリリースノート内に記述されている、

「Irregular crash or corrupt term vectors in the Lucene libraries.」
の項を参照してください。

Update Release Notes - Java SE Development Kit 7, Update 40

<http://www.oracle.com/technetwork/java/javase/7u40-relnotes-2004172.html>
(English)

Jettyを利用する場合

セットアップツールに同梱されている Jetty を利用する場合には以下の手順でJVM起動パラメータを変更してください。

1. OSに応じた起動スクリプトをエディタで開きます。

- Windowsの場合

```
C:\solr\solr_setup\dist\jetty\jetty\startup.bat
```

- Linuxの場合

```
/usr/local/solr/dist/jetty/jetty/startup.sh
```

2. いずれのファイルにも共通して `java -jar start.jar` の記述があることを確認します。

3. 確認した記述内容を `java -XX:-UseSuperWord -jar start.jar` と変更してください。

Solrの設定

Solr の設定項目について説明します。

項目

- コアの追加・削除 (solr.xml)
 - コアとは
 - intra-mart Accel Platform の複数テナント環境におけるSolrサーバの構成
 - コアの追加
 - コアの削除
- コアごとの設定
 - 基本設定 (solrconfig.xml)
 - スキーマの設定 (schema.xml)

コアの追加・削除 (solr.xml)

コアとは

コアはRDBのスキーマに相当し、コアごとにスキーマ定義やクエリの設定を持つことができます。

1つのSolrインスタンスの中に複数のコア（マルチコア）を設定することで、コアごとにインデックスが保持されます。

弊社から提供しているセットアップツールで作成するSolr環境は初期状態で **default** というコアが設定されています。

セットアップツールが出力するコアの設定ファイル (<schema.xml>, <solrconfig.xml> など) には IM-ContentsSearch用の設定が最初から記載されています。

intra-mart Accel Platform の複数テナント環境におけるSolrサーバの構成

intra-mart Accel Platformでは、WARファイルによる複数テナントおよび、バーチャルテナントによる複数テナントで運用する場合には、

テナントごとにコアを設定し、テナント単位でインデックスの管理を行います。

WARファイルによる複数テナントおよび、バーチャルテナントによる複数テナント環境でのSolrサーバの構成は以下の2つのパターンが考えられます。

1. テナントごとにSolrサーバを構築する
2. 1つのSolrサーバで、テナントごとにインデックスを管理する（コアを複数作成する）。

2の構成にする場合には、Solrコア構成ファイル <%SOLR_HOME%/solr.xml> の編集が必要となります。

データの増加に応じてインデックスが増加することにより検索や最適化の処理速度が低下する場合がありますため、

可能であれば、テナントごとにSolrのインスタンスを分けて（1の構成で）運用することを推奨します。

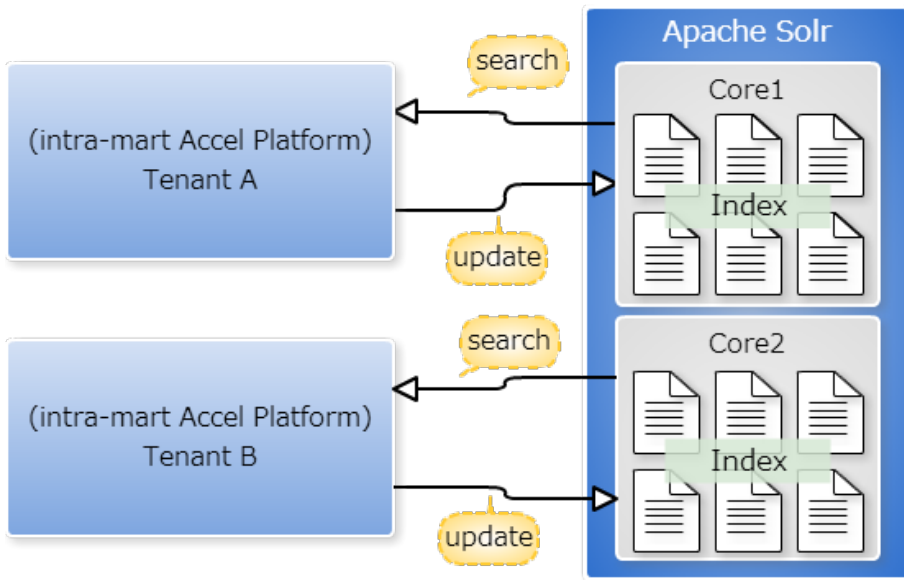


図 1つのSolrインスタンスに複数のコアを設定した場合のWARファイルによる複数テナントおよび、バーチャルテナントによる複数テナント環境の概念



注意

IM-ContentsSearchではコンテンツを識別するユニークキーに **ID** フィールドを指定しており、別々のテナントで生成されたコンテンツのIDが入ってしまうため、複数のテナントに同じSolrコアを設定することはできません。

コアの追加

sample というコアを作成する例の手順を紹介します。
default のコアがあることを前提とします。

1. <solr.xml>の編集

<%SOLR_HOME%/solr.xml>の `core` タグを以下のように追加します。

```
<cores adminPath="/admin/cores" >
  <core name="default" instanceDir="default" />
  <core name="sample" instanceDir="sample" />
</cores>
```

2. インスタンスディレクトリの作成

%SOLR_HOME%/default をコピーして %SOLR_HOME% ディレクトリ内に貼り付け、コピーしたディレクトリ名を **sample** に変更します。

3. Solrの起動

アプリケーションサーバを起動して、正常に起動することを確認します。

起動時に以下のようなエラーが表示される場合は設定が正しく行われていない可能性がありますので、設定内容を確認してください。

```
java.lang.RuntimeException: Can't find resource 'solrconfig.xml' in classpath or
'solr\sample\conf\'
```

4. Solr管理画面の確認

ブラウザでSolr管理画面が表示できることを確認してください。

```
http://<HOST>:<PORT>/solr/default/admin/
```

コアの削除

sample というコアを削除する例の手順を紹介します。

1. <solr.xml>の編集

<%SOLR_HOME%/solr.xml>のコア名 **sample** が記述されている `core` タグを削除します。

2. データ保存ディレクトリの削除

<%SOLR_HOME%/<コア名>/conf/solrconfig.xml> の `dataDir` タグに指定されているデータ保存ディレクトリを削除します。

データ保存ディレクトリについては [基本設定 \(solrconfig.xml\)](#) を参照してください。

3. インスタンスディレクトリの削除

%SOLR_HOME%/sample のディレクトリを削除します。

4. Solrの起動

アプリケーションサーバを起動して、正常に起動することを確認します。

起動時に以下のようなエラーが表示される場合は設定が正しく行われていない可能性がありますので、設定内容を確認してください。

```
java.lang.RuntimeException: Can't find resource 'solrconfig.xml' in classpath or
'solr\sample\conf\'
```

5. Solr管理画面の確認

ブラウザでSolr管理画面が表示できることを確認してください。

```
http://<HOST>:<PORT>/solr/default/admin/
```

コアごとの設定

コアごとに下記の設定を行うことが可能です。

基本設定 (solrconfig.xml)

弊社から提供しているセットアップツールで出力する<solrconfig.xml>はIM-ContentsSearch用の設定が記述されています。

<solrconfig.xml>は %SOLR_HOME%/<コア名>/conf ディレクトリ内に保存されています。

以下に記載する項目以外の内容は変更しないでください。

- インデックス保存ディレクトリ

インデックスが保存される場所です。初期状態では %SOLR_HOME%/コア名%/data ディレクトリに保存されます。

相対パスで指定した場合はアプリケーションサーバを起動したディレクトリによってカレントディレクトリが変わるため、絶対パスでの指定を推奨します。

```
<dataDir>${solr.data.dir:}</dataDir>
```

スキーマの設定 (schema.xml)

弊社から提供しているセットアップツールで出力する<schema.xml>はIM-ContentsSearch用のスキーマ定義が記述されています。

IM-ContentsSearchのためのフィールドやアナライザの定義が記載されているため、必要がない限り<schema.xml>の内容を変更しないでください。

<schema.xml>は %SOLR_HOME%/コア名%/conf ディレクトリ内に保存されています。

コラム

IM-ContentsSearch が利用するSolrサーバのフィールド (<schema.xml>で定義されているフィールド) は、IM-ContentsSearchの検索フィールド設定 (solr-field-config) によって指定されています。

詳細は以下のドキュメントを参照してください。

- 「IM-ContentsSearch プログラミングガイド」 - 「フィールド」
- 「設定ファイルリファレンス」 - 「検索フィールド設定」

Solrの操作

IM-ContentsSearch を利用するにあたって必要となる Solr の運用上の手順などについて説明します。

項目

- インデックスの削除
- Solrの移行
 - Tomcatを利用している場合
 - Resinを利用している場合
 - Jettyを利用している場合
- Solrのバックアップ

インデックスの削除

インデックスを削除する方法について説明します。

開発時や環境再構築時に一旦インデックスを削除したい場合などに実行してください。

curlコマンドを実行してSolrサーバに対してXMLデータを送信、またはWebブラウザのアドレスバーにURLを指定して削除を実行することができます。いずれも、Solrサーバが起動されている状態で実行する必要があります。

- コア名「default」内のインデックスをすべて削除する操作

Linuxの場合：curlコマンドを実行

```
curl "http://localhost:8983/solr/default/update?commit=true" --data-binary '<delete>
<query>*:*</query></delete>' -H 'Content-type: text/xml; charset=utf-8';
```

Windowsの場合：WebブラウザのアドレスバーにURLを指定

```
http://localhost:8983/solr/default/update?stream.body=<delete><query>*:*</query>
</delete>&commit=true
```

コラム

IM-ContentsSearch では、インデックスを削除するためのジョブを用意しています。
「[ジョブ・ジョブネットリファレンス](#)」 - 「[削除クローリング](#)」 を参照してください。

Solrの移行

Solrサーバに保存されている設定ファイルやインデックスの移行手順についてアプリケーションサーバごとに説明します。

移行先のサーバでインデックスを再度作成する場合にはインデックスのコピーは必要ありません。

本書では同じアプリケーションサーバ製品への移行を想定した手順を記載します。

**注意**

Solrサーバのデータの移行作業を行う前には、バックアップを行うことを推奨します。

Tomcatを利用している場合

Tomcat でSolrサーバを構築している場合のデータ移行手順について説明します。

- 作業は必ず Tomcat を停止させている状態で開始してください。
- 移行元の Tomcat のインストールディレクトリを %TOMCAT_OLD_HOME% と表記します。
- 移行先の Tomcat のインストールディレクトリを %TOMCAT_NEW_HOME% と表記します。

%TOMCAT_OLD_HOME%:

移行元の Tomcat のインストールディレクトリ

%TOMCAT_NEW_HOME%:

移行先の Tomcat のインストールディレクトリ

1. WARファイルのコピー

<%TOMCAT_OLD_HOME%/webapps/solr.war> を
<%TOMCAT_NEW_HOME%/webapps> ディレクトリ配下にコピーします。

2. solrディレクトリのコピー

<%TOMCAT_OLD_HOME%/solr> ディレクトリを <%TOMCAT_NEW_HOME%> ディレクトリ配下にコピーします。

3. インデックス保存ディレクトリのコピー

インデックスが保存されているディレクトリをコピーしてください。

- <%TOMCAT_OLD_HOME%/solr/%コア名%/solrconfig.xml>の `dataDir` に指定しているディレクトリを変更していない場合

<%TOMCAT_OLD_HOME%/solr/%コア名%> ディレクトリ配下に `data` ディレクトリがあるため、手順2でコピー済みです。

- <%TOMCAT_OLD_HOME%/solr/%コア名%/solrconfig.xml>の `dataDir` に指定しているディレクトリを変更していない場合

複数のコアを設定している場合はそれぞれのディレクトリと <solrconfig.xml> の `dataDir` の設定を確認してください。

4. URLEncodingの設定

<%TOMCAT_NEW_HOME%/conf/server.xml> の `Connector` 要素に、属性 `URIEncoding="UTF-8"` を追加します。

5. その他環境設定

Tomcat の設定をデフォルト設定から変更している場合は、移行先の環境に合わせて変更してく

ださい。

6. Tomcatの起動

Tomcat を起動し、正常に起動されることを確認してください。

7. Solr管理画面の確認

ブラウザでSolr管理画面が表示できることを確認してください。

```
http://<HOST>:<PORT>/solr/default/admin/
```

Resinを利用している場合

Resin でSolrサーバを構築している場合の移行手順について説明します。

- 作業は必ず Resin を停止させている状態で開始してください。
- 移行元の Resin のインストールディレクトリを `%RESIN_OLD_HOME%` と表記します。
- 移行先の Resin のインストールディレクトリを `%RESIN_NEW_HOME%` と表記します。

`%RESIN_OLD_HOME%`:

移行元の Resin のインストールディレクトリ

`%RESIN_NEW_HOME%`:

移行先の Resin のインストールディレクトリ

1. WARファイルのコピー

`<%RESIN_OLD_HOME%/webapps/solr.war>` を `<%RESIN_NEW_HOME%/webapps>` ディレクトリ配下にコピーします。

2. solrディレクトリのコピー

`<%RESIN_OLD_HOME%/solr>` ディレクトリを `<%RESIN_NEW_HOME%>` ディレクトリ配下にコピーします。

3. インデックス保存ディレクトリのコピー

インデックスが保存されているディレクトリをコピーしてください。

- `<%RESIN_OLD_HOME%/solr/%コア名%/solrconfig.xml>` の `dataDir` に指定しているディレクトリを変更していない場合

`<%xxx_HOME%/solr/%コア名%>` ディレクトリ配下に `data` ディレクトリがあるため、手順2でコピー済みです。

- `<%RESIN_OLD_HOME%/solr/%コア名%/solrconfig.xml>` の `dataDir` に指定しているディレクトリを変更していない場合

複数のコアを設定している場合はそれぞれのディレクトリと `<solrconfig.xml>` の `dataDir` の設定を確認してください。

4. ホームロケーションの設定

<%RESIN_NEW_HOME%/conf/resin.xml> の </resin> の前に以下の内容を追加します。

```
<env-entry>
  <env-entry-name>solr/home</env-entry-name>
  <env-entry-type>java.lang.String</env-entry-type>
  <env-entry-value>./solr</env-entry-value>
</env-entry>
```

5. その他環境設定

Resin の設定をデフォルト設定から変更している場合は、移行先の環境に合わせて変更してください。

6. Resinの起動

Resin を起動し、正常に起動されることを確認してください。

7. Solr管理画面の確認

ブラウザでSolr管理画面が表示できることを確認してください。

```
http://<HOST>:<PORT>/solr/default/admin/
```

Jettyを利用している場合

Jetty でSolrサーバを構築している場合のデータ移行手順について説明します。

- 作業は必ず Jetty を停止させている状態で開始してください。
- 移行元の Jetty のインストールディレクトリを %JETTY_OLD_HOME% と表記します。
- 移行先の Jetty のインストールディレクトリを %JETTY_NEW_HOME% と表記します。
- 移行先の Jetty は弊社提供のセットアップツールで構築されている環境とします。

%JETTY_OLD_HOME%:

移行元の Jetty のインストールディレクトリ

%JETTY_NEW_HOME%:

移行先の Jetty のインストールディレクトリ

1. solrディレクトリのコピー

<%JETTY_OLD_HOME%/solr> ディレクトリを <%JETTY_NEW_HOME%> ディレクトリ配下にコピーします。

2. インデックス保存ディレクトリのコピー

インデックスが保存されているディレクトリをコピーしてください。

- <%JETTY_OLD_HOME%/solr/%コア名%/solrconfig.xml>の dataDir に指定しているディレクトリを変更していない場合

<%xxx_HOME%/solr/%コア名%> ディレクトリ配下に data ディレクトリがあるた

- <%JETTY_OLD_HOME%/solr/%コア名%/solrconfig.xml>の `dataDir` に指定しているディレクトリを変更していない場合

複数のコアを設定している場合はそれぞれのディレクトリと <solrconfig.xml> の `dataDir` の設定を確認してください。

3. その他環境設定

Jetty の設定をデフォルト設定から変更している場合は、移行先の環境に合わせて変更してください。

4. Jettyの起動

Jetty を起動し、正常に起動されることを確認してください。

5. Solr管理画面の確認

ブラウザでSolr管理画面が表示できることを確認してください。

```
http://<HOST>:<PORT>/solr/default/admin/
```

Solrのバックアップ

1. solrディレクトリのバックアップ

%SOLR_HOME% ディレクトリをバックアップしてください。

<solrconfig.xml> の `dataDir` の指定を変更していない場合は %SOLR_HOME% ディレクトリに設定ファイルとインデックスが保存されているため、下記の手順を行う必要はありません。

2. インデックスのバックアップ

インデックスが保存されているディレクトリをバックアップしてください。

複数のコアを設定している場合はそれぞれの <solrconfig.xml> の `dataDir` に指定されているディレクトリをそれぞれバックアップしてください。