

JEUS JPAガイド

JEUS v8.0



Copyright © 2016 TmaxSoft Co., Ltd. All Rights Reserved.

Copyright Notice

Copyright © 2016 TmaxSoft Co., Ltd. All Rights Reserved.

45, Jeongjail-ro, Bundang-gu, Seongnam-si, Gyeonggi-do, 13613, South Korea

Restricted Rights Legend

All TmaxSoft Software (JEUS®) and documents are protected by copyright laws and international convention. TmaxSoft software and documents are made available under the terms of the TmaxSoft License Agreement and this document may only be distributed or copied in accordance with the terms of this agreement. No part of this document may be transmitted, copied, deployed, or reproduced in any form or by any means, electronic, mechanical, or optical, without the prior written consent of TmaxSoft Co., Ltd.

Nothing in this software document and agreement constitutes a transfer of intellectual property rights regardless of whether or not such rights are registered) or any rights to TmaxSoft trademarks, logos, or any other brand features. This document is for information purposes only. The company assumes no direct or indirect responsibilities for the contents of this document, and does not guarantee that the information contained in this document satisfies certain legal or commercial conditions. The information contained in this document is subject to change without prior notice due to product upgrades or updates. The company assumes no liability for any errors in this document.

このソフトウェア(JEUS®)マニュアルの内容とプログラムは、日本国の著作権法および国際条約によって保護されています。マニュアルの内容とプログラムは、TmaxSoft Co., Ltd.との使用許諾契約書の下でのみ使用することができ、マニュアルは使用許諾契約で許可されている範囲を除いては、配布または複製することができません。TmaxSoftの書面による事前の承諾を得ることなく、このマニュアルの全部または一部を電子的または機械的な方法を問わず、転送、複製、配布したり、または二次的著作物を作成する等の行為を一切禁じます。

このソフトウェアのマニュアルとプログラムの使用許諾契約は、いかなる場合においても、マニュアル及びプログラムと関連する知的財産権(登録の有無を問わず)を譲渡するものと解釈されず、TmaxSoftのブランド、ロゴ、商標等の使用権限を与えるものではありません。マニュアルは、情報を提供する目的でのみ提供しており、これに伴う契約上の直接的ないしは間接的な責任を負わず、マニュアルの内容は法律上もしくは商業的な特定の条件が満たされることを保証しません。マニュアルの内容は、製品のアップグレード及び修正により、その内容が予告なく変更されることがあり、内容上の誤りがないことを保証しません。

Trademarks

JEUS® is registered trademark of TmaxSoft Co., Ltd.

JEUS®は、TmaxSoft Co., Ltd.の登録商標です。

Java and Solaris are registered trademarks of Oracle Corporation and its subsidiaries and affiliates.

Java、Solarisは、Oracle Corporation及びその子会社、関連会社の登録商標です。

Microsoft, Windows, and Windows NT are registered trademarks or trademarks of Microsoft Corporation.

Microsoft、Windows、Windows NTは、Microsoft Corporationの登録商標または商標です。

HP-UX is a registered trademark of Hewlett Packard Enterprise Company.

HP-UXは、Hewlett Packard Enterprise Companyの登録商標です。

AIX is a registered trademark of International Business Machines Corporation.

AIXは、International Business Machines Corporationの登録商標です。

UNIX is a registered trademark of X/Open Company, Ltd.

UNIXは、X/Open Company, Ltd.の登録商標です。

Linux is a registered trademark of Linus Torvalds.

Linuxは、Linus Torvaldsの登録商標です。

Other products and company names are trademarks or registered trademarks of their respective owners.

その他、記載されている会社名、製品名などは、各社の商号、商標または登録商標です。

The names of companies, systems, and products mentioned in this manual may not necessarily be indicated with a trademark symbol (TM, ®).

本マニュアルに記載されている会社名、システム名、製品名などには必ずしも商標表示(TM、®)を付記しておりません。

Open Source Software Notice

Some modules or files of this product are subject to the terms of the following licenses. : APACHE2.0, CDDL1.0, EDL1.0, OPEN SYMPHONY SOFTWARE1.1, TRILEAD-SSH2, Bouncy Castle, BSD, MIT, SIL OPEN FONT1.1

Detailed Information related to the license can be found in the following directory : \${INSTALL_PATH}/lib/licenses

この製品の一部ファイルまたはモジュールは、APACHE2.0、CDDL1.0、EDL1.0、OPEN SYMPHONY SOFTWARE1.1、TRILEAD-SSH2、Bouncy Castle、BSD、MIT、SIL OPEN FONT1.1のライセンスに準拠します。

文書情報

文書名: JEUS JPAガイド

発行日: 2016年10月14日

ソフトウェアバージョン: JEUS v8.0

ガイドバージョン: v2.1.1

目次

このガイドについて	xi
第1章 はじめに	1
第2章 プロバイダーの設定	3
2.1. データベースの設定	3
2.1.1. 使用環境別の設定	3
2.1.2. データベース・タイプの設定	5
2.1.3. スキーマ自動生成の設定	6
2.2. キャッシュ	8
2.3. クエリー・ヒント	10
2.4. ロギング設定	11
第3章 プロバイダーの変更	13
3.1. パーシスタンス・プロバイダーの変更	13
3.2. 使用可能なパーシスタンス・プロバイダー	14
索引	15

表目次

[表 2.1]	Java SEモードのデータベース・プロパティ	5
[表 2.2]	eclipselink.target-databaseプロパティ	5
[表 2.3]	JavaタイプとSQLタイプのマッピング	7

例目次

[例 2.1]	Java EEモードでのデータベースの設定例	4
[例 2.2]	Java SEモードでのデータベースの設定例	4
[例 2.3]	データベース・タイプの設定例	5
[例 2.4]	スキーマの自動生成の設定例	6
[例 2.5]	キャッシュ方式の設定例	8
[例 2.6]	クエリー・ヒントの使用例	10
[例 2.7]	ロギングの設定例	11
[例 3.1]	パーシスタンス・プロバイダーの変更例	13

このガイドについて

対象読者

Java Persistence API(JPA)は、リレーショナル・データベースにアクセスするための標準ORM技術を提供しており、既存のEJBで提供していたエンティティーBeanを代替する技術です。

本書は、JEUS[®](以下、JEUS)でJPAを使用して開発を行う開発者を対象としています。

前提知識

JEUSの基本的な使用方法と製品を理解するには、以下のガイドについてあらかじめ熟知することをお勧めします。

- 『JEUS 紹介ガイド』
- 『JEUS インストール & スタートガイド』

本書のすべてのサンプルと環境構成は、UNIXスタイルに準拠します。Microsoft Windows[™](以下、Windows)など他の環境で作業を行う場合は、次のような事項を考慮してください。

たとえば、Windowsプラットフォームでは、ディレクトリー区切り子をUNIXスタイルのスラッシュ(/)からWindowsスタイルのバックスラッシュ(\)に変えて使用してください。また、環境変数もWindowsスタイル(%%)に変更して使用してください。

本書で触れているJEUS_HOMEは、JEUSがインストールされているディレクトリーです。

制限事項

本書の内容は、Java標準に準拠して作成されていますが、本書で触れているJava EE仕様については詳しく取り上げていません。関連内容についてはJava EE関連ドキュメントを参照してください。

本書の構成

本書は、計3章で構成されています。

- 「[第1章 はじめに](#)」

JPAとJEUSについて説明します。

- 「[第2章 プロバイダーの設定](#)」

JEUSの基本プロバイダーの設定について説明します。

- 「[第3章 プロバイダーの変更](#)」

JEUSの基本プロバイダーを変更する方法について説明します。

表記上の規則

表記	意味
<<AaBbCc123>>	プログラム・ソースコードのファイル名
<Ctrl>+C	CtrlキーとCキーを同時に押す
[Button]	GUIのボタン、メニュー名
太字	強調
「」、『』（鍵カッコ）	関連文書、あるいはガイド内の他の章および節の表示
「入力項目」	画面UI上の入力項目
ハイパーリンク	メール・アカウント、Webサイト
>	メニューの実行順
+----	下位ディレクトリー/ファイル有り
----	下位ディレクトリー/ファイル無し
<div>参考</div>	参照/注意事項
<div>注</div>	注意事項
[図 1.1]	図の名称
[表 1.1]	表の名称
AaBbCc123	Javaコード、XMLドキュメント
[<i>command argument</i>]	オプション・パラメータ
< xyz >	「<」と「>」の間の内容は実際に使用される特定の名称または値で置き換えられる
	構文の中の相互に排他的な選択項目の選択肢を示す 例) A B: AとBのいずれかを選択
...	パラメータ、値、または他の情報が繰り返される
\${ }	環境変数

システム要件

	要求事項
プラットフォーム	Solaris 9, 10, 11
	HP-UX 11.x, 11i, 11iV2
	IBM AIX 5L, 6L, AIX 7L
	MS Windows 2008, 2012, Vista, 7, 8
ハードウェア	最小2GB以上、推奨20GBのハードディスク容量
	推奨1GB以上のメモリー容量
JDK	JDK 7, JDK 8

関連文書

ガイド	説明
JEUS 紹介ガイド	JEUSサーバーについて全般的に紹介し、JEUSのアーキテクチャーを含む各構成要素について記述しています
JEUS インストール&スタートガイド	JEUSについて紹介し、JEUSのインストールおよび開始方法について記述しています
JEUS サーバガイド	JEUSシステムおよびサーバーの概要とシステムの管理方法について記述しています
JEUS WebAdminガイド	JEUSのWeb管理ツールであるWebAdminを利用したJEUSの設定および制御、モニタリング、クラスタリング、リソースの設定および管理について記述しています

参考文献

- Pro EJB 3 Java Persistence API, Mike Keith and Merrick Schincariol, Apress
- Enterprise JavaBeans 3.0 5th ed., Bill Burke and Richard Monson-Haefel, O'Reilly
- Pro JPA 2 Mastering the Java Persistence API, Mike Keith and Merrick Schincariol, Apress

お問合せ先

Korea

TmaxSoft Co., Ltd.
45, Jeongjail-ro, Bundang-gu,
Seongnam-si, Gyeonggi-do, 13613
South Korea
Tel: +82-31-8018-1000
Fax: +82-31-8018-1115
Email: info@tmax.co.kr
Web (Korean): <http://www.tmaxsoft.com>
TechNet: <http://technet.tmaxsoft.com>

USA

TmaxSoft Inc.
101 North Wacker Drive, Suite 2014,
Chicago, IL 60606
U.S.A
Tel: +1-312-525-8330
Email: info@tmaxsoft.com
Web (English): http://www.tmaxsoft.com/us_en/home

Japan

TmaxSoft Japan Co., Ltd.
5F Sanko Bldg, 3-12-16 Mita, Minato-Ku, Tokyo, 108-0073
Japan
Tel: +81-3-5765-2550
Fax: +81-3-5765-2567
Email: info@tmaxsoft.co.jp
Web (Japanese): <http://www.tmaxsoft.co.jp>

China

Beijing TmaxSoft System Software Co., Ltd.
Room103, No.2 Huizhong Building, Seven Street Shangdi,
Haidian District, Beijing, 100085
P.R.China
Tel: +86-10-6298-8827
Email: info@tmaxsoft.com.cn
Web (Chinese): http://www.tmaxsoft.com/cn_en/home_cn_en

Brazil

Tmax Brasil Sistemas e Serviços Ltda.
Av. Copacabana, 177, sala 32~35 Empresarial 18 do Fortel
Alphaville Barueri, Sao Paulo, 06472-001
Brazil
Tel: +55-11-4191-3100
Fax: +55(11) 4191-3705 (extension#112)
Email: info.bra@tmaxsoft.com
Web (Portuguese): http://www.tmaxsoft.com/br_en/home_br_en

Russia

Tmax Rus L.L.C.
Leninsky prospekt, 113/1 (Park Place Moscow),
Office 318e, Moscow, 117198
Russia
Tel: +7(495)970-01-35
Email: info.rus@tmaxsoft.com
Web (Russian): http://www.tmaxsoft.com/ru_ru/home_ru_ru

Singapore

Tmax Singapore Pte. Ltd.
430 Lorong 6, Toa Payoh #10-02,
OrangeTee Building, 319402
Singapore
Tel: +65-6259-7223
Fax: +65-6258-7112
Email: info.sg@tmaxsoft.com
Web (English): http://www.tmaxsoft.com/sg_en/home_sg_en

United Kingdom

TmaxSoft UK Ltd.
215 Knyvett House, Watermans Business Park,
The Causeway, Staines TW18 3BAB
United Kingdom
Tel: +44-1784-895005
Email: info.uk@tmaxsoft.com
Web (English): http://www.tmaxsoft.com/gb_en/home_gb_en

Canada

TmaxSoft Canada, Inc.
2425 Matheson Blvd East, 8th floor,
Unit 824 Mississauga, ON, L4W 5K4
Canada
Tel: +1-905-361-2888
Email: info.canada@tmaxsoft.com
Web (English): http://www.tmaxsoft.com/ca_en/home_ca_en

Australia

TmaxSoft Proprietary Limited
L32, 101 Miller Street, North Sydney 2060
Australia
Tel: +91-9845-330-704
Email: info.aus@tmaxsoft.com
Web (English): http://www.tmaxsoft.com/au_en/home_au_en

India

TmaxSoft Technologies Private Limited
Sobha Alexander Plaza, 3rd Floor,
16/2 Commissariat Road, Bangalore-560025
India
Tel: +91-9845-330-704
Email: info.india@tmaxsoft.com
Web (English): http://www.tmaxsoft.com/in_en/home_in_en

Turkey

TmaxSoft Co., Ltd. Turkey Liaison Office
Windowist Tower. Eski Buyukdere Cad. No:26,
Maslak 34467 Istanbul
Turkey
Tel: +90-544-553-6045
Email: cslee@tmaxsoft.com
Web (English): http://www.tmaxsoft.com/tr_en/home_tr_en

第1章 はじめに

Java Persistence API(以下、JPA)はリレーショナル・データベースにアクセスするための標準ORM技術を提供しており、既存のEJBで提供していたCMPエンティティBeanを代替する技術です。

JEUSはJPA仕様の全機能をサポートしています。JPAは、JSR 338で公開されたJPA 2.1仕様に定義されており、EJBコンテナに依存せずに、EJBとWebモジュール、Java SEスタンドアロン・クライアントですべて使用することができます。

JPAパーシスタンス・プロバイダーは、ユーザーが選択して使用することができます。JEUSでは基本的にEclipse Persistence Services Projectの**EclipseLink**実装を提供しています。

必要に応じて他の実装を使用することができます。関連する内容については、「[3.1. パーシスタンス・プロバイダーの変更](#)」を参照してください。

実際にJPAを使用するときは、JPA仕様で提供する基本的なAPIや設定、プロバイダーの特性を合わせて考慮する必要があります。特に、キャッシングなどは重要な特性であり、これを考慮せずにアプリケーションを開発すると、望むとおり結果が得られないこともあり得ます。そのため、環境に合わせてJPAを適切に構成し開発する必要があります。その例については、今後、多様なパターンで紹介されると考えられるので、EclipseLink JPAのWebサイトを随時参照することを推奨します。

- EclipseLink

<http://www.eclipse.org/eclipselink/>

本書では、JEUSの基本プロバイダーであるEclipseLinkをJEUSで使用するために必要な設定についてのみ説明しています。JPA技術の全般やプログラミング方法については、本書に記載されている[参考文献](#)を参照してください。

第2章 プロバイダーの設定

本章では、JEUSの基本プロバイダーであるEclipseLinkの設定方法について説明します。この設定は、JPA仕様に定義されていない機能を実装するために必要であり、それぞれのアプリケーションに合わせて正しく設定する必要があります。

2.1. データベースの設定

使用環境とデータベースのタイプに合わせて設定する方法と、データベースのスキーマを自動生成する設定について説明します。

2.1.1. 使用環境別の設定

データベースの設定方法は使用環境によって異なります。

Java EE環境

Java EE環境(モード)は、JEUSのManaged Server(以下、MS)上のWebコンテナ、EJBコンテナ、アプリケーション・クライアント・コンテナを意味します。

参考

より正確には、各コンテナでコントロールするスレッドであり、例えばWebエンジンに設定するWebスレッドプールのスレッドなどです。アプリケーションが直接作成したスレッドプールのスレッドのように、コンテナが管理しないスレッドではJava SE環境と同じです。

使用するデータベースは、persistence.xml記述子に設定します。使用するトランザクションのタイプによって、<jta-data-source>と<non-jta-data-source>値を設定します。

- グローバル・トランザクションを使用する場合
 - <transaction-type>の値を「JTA」に設定します。
 - <jta-data-source>に使用するデータソースのJNDI名を設定します。
- ローカル・トランザクションを使用する場合
 - <transaction-type>の値を「RESOURCE_LOCAL」に設定します。

- **<non-jta-data-source>**に使用するデータソースのJNDI名を設定します。

[例 2.1] Java EEモードでのデータベースの設定例

```
<?xml version="1.0" encoding="UTF-8"?>
<persistence xmlns="http://java.sun.com/xml/ns/persistence">
  <persistence-unit name="em" transaction-type="JTA">
    <jta-data-source>jdbc/MyDB</jta-data-source>
  </persistence-unit>
</persistence>
```

参考

1. Java EE環境では、<transaction-type>の値を設定しない場合、JTAトランザクションが基本設定となります。
 2. JEUSでDBデータソースを設定する方法については、『JEUS サーバガイド』の「第6章 DBコネクション・プールとJDBC」を参照してください。
-

Java SE環境

Java SE環境(モード)は、Java EEコンテナではなく、Javaスタンドアロン・クライアントなどの環境を意味します。この環境ではローカル・トランザクションのみ使用することができ、使用するデータベースのJDBC設定をプロパティで設定する必要があります。

以下の例に示したように、<transaction-type>の値を**RESOURCE_LOCAL**に設定します。

[例 2.2] Java SEモードでのデータベースの設定例

```
<?xml version="1.0" encoding="UTF-8"?>
<persistence xmlns="http://java.sun.com/xml/ns/persistence">
  <persistence-unit name="em" transaction-type="RESOURCE_LOCAL">
    <properties>
      <property name="eclipselink.jdbc.driver"
        value="org.apache.derby.jdbc.ClientDriver"/>
      <property name="eclipselink.jdbc.url"
        value="jdbc:derby://localhost:1527/testdb"/>
      <property name="eclipselink.jdbc.user" value="APP"/>
      <property name="eclipselink.jdbc.password" value="APP"/>
    </properties>
  </persistence-unit>
</persistence>
```

以下は、プロパティ値についての説明です。

[表 2.1] Java SEモードのデータベース・プロパティ

プロパティ値	説明
eclipselink.jdbc.driver	対象データベースのJDBCドライバー・クラスの名前です
eclipselink.jdbc.url	対象データベースのJDBCのURLです
eclipselink.jdbc.user	対象データベースのユーザー名です
eclipselink.jdbc.password	対象データベースのパスワードです

2.1.2. データベース・タイプの設定

データベースのタイプは基本的にJDBCコネクション情報を通じて自動で検知されます。自動検知機能がうまく動作しない場合や、別のデータベースを使う場合は、「eclipselink.target-database」プロパティを設定することができます。

参考

データベースのタイプは、JDBCドライバーのDatabaseMetaData.getDatabaseProductName()を使って、データベースのベンダー名を正規表現(regular expression)で比較する方式を使用します。

[例 2.3] データベース・タイプの設定例

```
<?xml version="1.0" encoding="UTF-8"?>
<persistence xmlns="http://java.sun.com/xml/ns/persistence">
  <persistence-unit name="em">
    <jta-data-source>jdbc/MyDB</jta-data-source>
    <properties>
      <property name="eclipselink.target-database" value="DB2"/>
    </properties>
  </persistence-unit>
</persistence>
```

以下は、サポートするデータベース・タイプの値です。

[表 2.2] eclipselink.target-databaseプロパティ

値	説明
Auto	自動検知(デフォルト値)
Cloudscape	Cloudscape DBMS
DB2	IBM DB2 DBMS
DB2Mainframe	IBM DB2 Mainframe DBMS
Derby	Apache Derby DBMS

値	説明
HSQL	HSQL DBMS
JavaDB	JavaDB DBMS
MySQL4	MySQL DBMS
Oracle	Oracle DBMS
PostgreSQL	PostgreSQL DBMS
SQLServer	Microsoft SQLServer DBMS
Sybase	Sybase DBMS
Customized class name	デフォルトでサポートしないDBMSを追加するときに使用
その他	その他のDBMSについては、eclipselinkのPersistence Property Extensions Referenceページでtarget-database部分を参照してください

基本的にサポートされないDBMSについては、DBMS対応機能を別途実装することで使用することができます。そのような場合には、該当するクラス名を指定します。より詳しい内容については、本書に記載されている[参考文献](#)を参照してください。

2.1.3. スキーマ自動生成の設定

自動でDBスキーマを作成する機能を使用する場合に設定します。この機能を利用すれば、アプリケーションをデプロイするときに、データベースのテーブルと制約(Constraints)が自動で作成されます。

[例 2.4] スキーマの自動生成の設定例

```
<?xml version="1.0" encoding="UTF-8"?>
<persistence xmlns="http://java.sun.com/xml/ns/persistence">
  <persistence-unit name="em">
    <jta-data-source>jdbc/MyDB</jta-data-source>
    <properties>
      ...
      <property name="eclipselink.ddl-generation" value="create-tables" />
      ...
    </properties>
  </persistence-unit>
</persistence>
```

関連するプロパティの設定は以下のとおりです。

項目	説明
eclipselink.ddl-generation	スキーマDDL(Data Descriptor Language)を作成する方法を設定します – none : 何も実行しません(デフォルト値)

項目	説明
	<ul style="list-style-type: none"> – create-tables : 新規テーブルを作成し、既存のテーブルをそのまま残します – drop-and-create-tables : 既存のテーブルを削除し、新規テーブルを作成します – create-or-extend-tables : テーブルを作成しますが、既に存在するテーブルがある場合は、存在しない列のみを追加します

DDLを作成するとき、以下のようにデータベースのSQLタイプに合わせてJavaタイプを作成します。

[表 2.3] JavaタイプとSQLタイプのマッピング

Java Type	Derby, JavaDB, Cloudscape	Oracle	DB2	Sybase	SQLServer	MySQL
boolean, Boolean	SMALLINT	NUMBER(1)	SMALLINT	BIT	BIT	TINYINT(1)
int, Integer	INTEGER	NUMBER(10)	INTEGER	INTEGER	INTEGER	INTEGER
long, Long	BIGINT	NUMBER(19)	INTEGER	NUMERIC(19)	NUMERIC(19)	BIGINT
float, Float	FLOAT	NUMERIC(19,4)	FLOAT	FLOAT(16)	FLOAT(16)	FLOAT
double, Double	FLOAT	NUMERIC(19,4)	FLOAT	FLOAT(32)	FLOAT(32)	DOUBLE
short, Short	SMALLINT	NUMBER(5)	SMALLINT	SMALLINT	SMALLINT	SMALLINT
byte, Byte	SMALLINT	NUMBER(3)	SMALLINT	SMALLINT	SMALLINT	SMALLINT
java.lang. Number	DECIMAL	NUMBER(38)	DECIMAL(15)	NUMERIC(38)	NUMERIC(28)	DECIMAL(38)
java.math. BigInteger	BIGINT	NUMBER(38)	BIGINT	NUMERIC(38)	NUMERIC(28)	BIGINT
java.math. BigDecimal	DECIMAL	NUMBER(38)	DECIMAL(15)	NUMERIC(38)	NUMERIC(28)	DECIMAL(38)
java.lang. String	VARCHAR(255)	VARCHAR(255)	VARCHAR(255)	VARCHAR(255)	VARCHAR(255)	VARCHAR(255)
char, Character	CHAR(1)	CHAR(1)	CHAR(1)	CHAR(1)	CHAR(1)	CHAR(1)
byte[], Byte[], java.sql.Blob	BLOB(64000)	LONG RAW	BLOB(64000)	TEXT	TEXT	TEXT(64000)

Java Type	Derby, JavaDB, Cloudscape	Oracle	DB2	Sybase	SQLServer	MySQL
char[], Character[], java.sql.Clob	CLOB(64000)	LONG	CLOB(64000)	TEXT	TEXT	TEXT(64000)
java.sql.Date	DATE	DATE	DATE	DATETIME	DATETIME	DATE
java.sql.Time	TIME	DATE	TIME	DATETIME	DATETIME	TIME
java.sql.Timestamp	TIMESTAMP	DATE	TIMESTAMP	DATETIME	DATETIME	DATETIME

2.2. キャッシュ

JPAでは基本的にパーススタンス・コンテキストというファースト・レベル・キャッシュをサポートしています。一般にパーススタンス・コンテキストはトランザクション別に生成されるため(拡張パーススタンス・コンテキストは除く)、トランザクション間のキャッシュはサポートしません。それを補うために、Eclipse Linkではセカンド・レベル・キャッシュ機能を提供しています。

セカンド・レベル・キャッシュは、エンティティ・マネージャー・ファクトリー(EntityManagerFactory)レベルでサポートしているため、同じEntityManagerFactoryで生成されたすべてのエンティティ・マネージャーはこの共有キャッシュを使うことになります。つまり、パーススタンス・コンテキストに存在しないエンティティを取得するときは、セカンド・レベル・キャッシュを確認して、そこに存在すれば、そのエンティティを取得します。そのため、読み取り(Read)作業を繰り返す場合に性能向上を図ることができます。

[例 2.5] キャッシュ方式の設定例

```
<?xml version="1.0" encoding="UTF-8"?>
<persistence xmlns="http://java.sun.com/xml/ns/persistence">
  <persistence-unit name="em">
    <jta-data-source>jdbc/MyDB</jta-data-source>
    <properties>
      ...
      <property name="eclipselink.ddl-generation" value="create-tables" />
      <property name="eclipselink.cache.type.default" value="NONE" />
      <property name="eclipselink.cache.size.default" value="999" />
      <property name="eclipselink.cache.shared.default" value="false" />
      ...
    </properties>
  </persistence-unit>
</persistence>
```

以下は、キャッシュ作業のオプションについての説明です。

項目	説明
eclipselink.cache.type.default	<p>キャッシュ方式を設定します。</p> <p>以下のオプション以外のオプションやオプションについての詳細設定については、EclipseLink User Guideの「Cache Type and Size」項目を参照してください</p> <ul style="list-style-type: none"> – Full : オブジェクトをハード・リファレンスでキャッシュします。エンティティーは削除されるまで常にキャッシュに存在することになります – Weak : オブジェクトをウィーク・リファレンスでキャッシュします。オブジェクトはガベージ・コレクション(GC)が発生すると削除されます – Soft : Weakとほぼ同じですが、メモリーが足りない場合にのみオブジェクトを削除します(デフォルト値) – NONE : オブジェクトをキャッシュに保存しません。この方法は使用を抑えることを勧めます。キャッシュ機能を使いたくない場合は、eclipselink.cache.shared.xxxプロパティを設定することを推奨します
eclipselink.cache.size.default	<p>キャッシュに保存される最大のオブジェクト数を指定します(デフォルト値 : 1000)</p>
eclipselink.cache.shared.default	<p>共有キャッシュを使用するかどうかを設定します</p> <ul style="list-style-type: none"> – true : オブジェクトを共有キャッシュに保存します。すべてのエンティティー・マネージャーはキャッシュを使うことになります(デフォルト値) – false : オブジェクトを共有キャッシュに保存しません。したがって、エンティティー・マネージャー別にキャッシュを共有しません。この設定は、セカンド・レベル・キャッシュを使用しない効果があります
eclipselink.cache.type.<ENTITY>	<p>エンティティー別にキャッシュする方式を設定します。</p> <p><ENTITY>にエンティティー名、または完全修飾クラス名を指定します。指定するエンティティーと関連のあるすべてのエンティティーは同じ設定を持つ必要があります。</p> <p>指定できる値は、eclipselink.cache.type.defaultと同じです</p>
eclipselink.cache.size.<ENTITY>	<p>エンティティー別にキャッシュ・サイズを指定します。<ENTITY>にエンティティー名、または完全修飾クラス名を指定します。</p>

項目	説明
	指定できる値は、eclipselink.cache.type.defaultと同じです
eclipselink.cache.shared.<ENTITY>	<p>エンティティー別に共有キャッシュを使用するかどうかを設定します。</p> <p><ENTITY>にエンティティー名、または完全修飾クラス名を指定します。指定するエンティティーと関連のあるすべてのエンティティーは同じ設定を持つ必要があります。</p> <p>指定できる値は、eclipselink.cache.type.defaultと同じです</p>

注

セカンド・レベル・キャッシュを使うときに注意することは、外部アプリケーションによって、あるいは直接データベースのデータを変更する場合に、変更内容がキャッシュに反映されないということです。そのような場合に、データベースの最新値ではなく、キャッシュされた値が返されるため、アプリケーション側でこのことを考慮する必要があります。

アプリケーションの環境を考慮してキャッシュ・オプションを適切に設定するか、EntityManager.refresh()、eclipselink.refreshクエリー・ヒント、またはロッキング(pessimistic/optimistic)などを使って、こうした問題を防ぐことができます。

2.3. クエリー・ヒント

クエリー・ヒントはクエリー・オブジェクトを使用するときに、プロバイダーが提供する機能を使用できるようにします。

クエリー・ヒントは、下の例のように、クエリーを実行するときに設定できます。名前付きクエリー(Named query)を使用する場合は、@QueryHintアノテーションを使って設定することもできます。

[例 2.6] クエリー・ヒントの使用例

```
List employees
= em.createQuery("SELECT e FROM Employee e WHERE e.name = :name")
    .setParameter("name", name)
    .setHint("eclipselink.refresh", true)
    .getResultList();
```

提供するクエリー・ヒントは以下のとおりです。

項目	説明
eclipselink.pessimistic-lock	SELECTの際に、ペシミスティック・ロッキング(Pessimistic Locking)を使うかどうかを設定します

項目	説明
	<ul style="list-style-type: none"> – NoLock : 使用しない(デフォルト値) – Lock : SELECT ... FOR UPDATE文を使ってロッキングします – NoLockWait : SELECT ... FOR UPDATE NO WAIT文を使ってロッキングします
eclipselink.refresh	<p>データベースから最新値を取得してキャッシュをアップデートするかどうかを設定します</p> <ul style="list-style-type: none"> – true : 最新値を取得し、キャッシュをアップデートします – false : キャッシュの値を使用します(デフォルト値)

2.4. ロギング設定

ロギングのレベルを設定することで詳細ログを見ることができます。

基本的にロギングのレベルは、JEUSサーバー全体に適用されるレベル(基本的にINFO)に従います。これをパーシスタンス・ユニットごとに変更する場合は、eclipselink.logging.levelプロパティで設定することができます。

ロガーは、Java EEモードでは基本的にJEUSで提供されるロガー(JEUS Logger)を使います。Java SEモードでは、標準出力に出力するデフォルト・ロガー(DefaultLogger)を使います。これを変更したい場合は、eclipselink.logging.loggerプロパティで設定することができます。

[例 2.7] ロギングの設定例

```
<?xml version="1.0" encoding="UTF-8"?>
<persistence xmlns="http://java.sun.com/xml/ns/persistence">
  <persistence-unit name="em">
    <jta-data-source>jdbc/MyDB</jta-data-source>
    <properties>
      ...
      <property name="eclipselink.logging.level" value="FINE"/>
      <property name="eclipselink.logging.logger" value="DefaultLogger"/>
      ...
    </properties>
  </persistence-unit>
</persistence>
```

以下は、ロギング設定についての説明です。

項目	説明
eclipselink.logging.level	ロギングのレベルを指定します <ul style="list-style-type: none">– OFF: ロギングしない– SEVERE– WARNING– INFO(デフォルト値)– CONFIG– FINE: SQL関連情報を照会する場合に設定します– FINER– FINEST
eclipselink.logging.logger	使用するロガーを指定します <ul style="list-style-type: none">– JEUSLogger : JEUSで提供するロガー (Java EEモードのデフォルト値)– DefaultLogger : 標準出力ロガー (Java SEモードのデフォルト値)– JavaLogger : java.util.loggingロガー– Custom class name : 別途のロガーを実装した場合に指定します

参考

EclipseLinkの詳細設定については、<http://wiki.eclipse.org/EclipseLink/UserGuide/JPA>を参照してください。

第3章 プロバイダーの変更

本章では、JEUSの基本プロバイダーを変更する方法について説明します。

3.1. パーシスタンス・プロバイダーの変更

JEUSで基本的に提供しているプロバイダー以外のプロバイダーを使用する場合は、JPA仕様で提供する設定を使用します。

まず、必要なライブラリーをJEUS_HOME/lib/applicationにコピーするか、またはアプリケーション別にパッケージングします。そして、プロバイダーのクラス名をpersistence.xmlの<provider>値に設定すると、パーシスタンス・ユニット別に設定したプロバイダーを使用します。

プロバイダーのクラス名と関連プロパティ値はプロバイダーごとに異なるので、関連文書を参考して設定する必要があります。たとえば、ハイバネート(Hibernate)を使用する場合は以下のように設定します。

[例 3.1] パーシスタンス・プロバイダーの変更例

```
<?xml version="1.0" encoding="UTF-8"?>
<persistence xmlns="http://java.sun.com/xml/ns/persistence">
  <persistence-unit name="em">
    <provider>org.hibernate.ejb.HibernatePersistence</provider>
    <jta-data-source>jdbc/MyDB</jta-data-source>
    <properties>
      <!-- add Hibernate properties here -->
    </properties>
  </persistence-unit>
</persistence>
```

前述のように、パーシスタンス・ユニットごとにプロバイダーを変更することも可能ですが、JEUSのデフォルト・プロバイダーを変更することもできます。JEUSを起動させるとき、システム・プロパティのjeus.persistence.defaultProviderを変更するプロバイダーのクラス名に設定すると、プロバイダーを設定していないすべてのパーシスタンス・ユニットは指定したデフォルト・プロバイダーを使用することになります。

参考

JEUSのシステム・プロパティの設定については、『JEUSサーバガイド』および『JEUSリファレンスガイド』を参照してください。

3.2. 使用可能なパーシスタンス・プロバイダー

他のパーシスタンス・プロバイダーについては、以下のWebサイトを参照してください。

- Hibernate EntityManager

<http://www.hibernate.org>

- OpenJPA

<http://openjpa.apache.org>

- BEA Kodo

http://docs.oracle.com/cd/E13189_01/kodo/docs40/index.html

索引

ら

ロガー(logger), 11

E

EclipseLink, 1
eclipselink.cache.shared.<ENTITY>, 10
eclipselink.cache.shared.default, 9
eclipselink.cache.size.<ENTITY>, 9
eclipselink.cache.size.default, 9
eclipselink.cache.type.<ENTITY>, 9
eclipselink.cache.type.default, 9
eclipselink.ddl-generation, 6
eclipselink.jdbc.driver, 5
eclipselink.jdbc.password, 5
eclipselink.jdbc.url, 5
eclipselink.jdbc.user, 5
eclipselink.logging.level, 12
eclipselink.logging.logger, 12
eclipselink.pessimistic-lock, 10
eclipselink.refresh, 11
eclipselink.target-databaseプロパティ, 5

J

Java EE環境プロバイダーの設定, 3
Java SEモードのデータベース・プロパティ, 5
Java SE環境プロバイダーの設定, 4
JavaタイプとSQLタイプのマッピング, 7
JPA, 1

か

キャッシュ, 8
クエリー・ヒント, 10

さ

セカンド・レベル・キャッシュ, 8

は

パーシスタンス・プロバイダー, 1, 13

