

The background of the slide is a photograph of a city street. On the left, there is a large, ornate building with many windows. In the foreground, a red bus is driving away from the camera. The sky is blue.

# 【アセットツール】 DBアクセスログツール **DENEB**ガイド (Java版)

2016年11月15日(火)  
ワールドイツツマイン  
代表取締役副社長  
川上 智弘

# 目次

|                                |              |
|--------------------------------|--------------|
| <b>狙い</b> .....                | <b>P 3</b>   |
| <b>特徴</b> .....                | <b>P 4</b>   |
| <b>事前準備</b> .....              | <b>P 5</b>   |
| <b>対応DBサーバー</b> .....          | <b>P 7</b>   |
| <b>フォルダ構成</b> .....            | <b>P 8</b>   |
| <b>DENEB のダウンロード及び設定</b> ..... | <b>P 1 0</b> |
| <b>DENEB 実行</b> .....          | <b>P 1 1</b> |
| <b>APサーバー早見表</b> .....         | <b>P 1 3</b> |
| <b>アクセスログ機能</b> .....          | <b>P 1 4</b> |
| <b>終りに</b> .....               | <b>P 2 6</b> |

# 狙い

## 課題

画面処理とは違い、DB処理は目に見えません。  
DBを利用した業務アプリケーションが、  
意図しない動作をしている場合、問題部分の分析に工数が掛かります。

## 解決策

- 【DENEb】でSQL処理の実行結果を**見える化**出来ます。
- SQLの[発行]や[結果取得]をログに記録し、AP⇔DBのやり取りの見える化。
  - 業務AP毎のSQL処理の傾向を集計し、問題部分を指摘。

## 利用イメージ

**開発者が実行**をした**プログラム(性能改善対象)**をDBアクセスログツール**【DENEb】**が**常駐・分析**を行い、**テキストログ**及び**ログ集計レポート**を**出力**します。  
テキストログは、SQL文の**実行時刻**、**パラメータの内容**、**応答時間・呼び出し元**、**検索結果**が判る様になっています。  
SQL処理の全課程をテキスト形式のログとして出力します。  
アプリケーションが発行したSQL文とSQL結果が判り、デバッグに有効です。  
DBリソースの開放漏れも自動検出します。  
ログ集計レポートは、SQLの発行回数や累積時間を集計します。  
発行元のクラス名とSQLを対応付けて分析します。

## 効果

- デバッグ時、ステップ実行等をしなくても、問題をSQLログからすぐに特定可能です。
- テスト確認としてもログを利用出来ます。
- DB関連の性能問題の早期発見、問題部分切り分け、障害の再現・解析に活用出来ます。

# 特長

- 「画面の入出力」「DBの入出力」、その両者が見える事が出来るなら、開発者はアプリの動きを手取る様に理解出来ます。
- 発行したSQLだけではなく、DBから応答結果もログに出力します。
- SQLの発行元クラス名もログに出力します。既存プログラムの構造解析にも効果が有ります。

**DB処理のデバッグ・品質向上を進める  
必要があるJava PJで活用下さい！**

# 事前準備01

インストールに必要なソフトウェア。

| ソフトウェア                    | 開発者 | 試験担当者 |
|---------------------------|-----|-------|
| JRE“6”以降                  | ○   | ○     |
| Microsoft Excel “2003” 以上 | ○   | ○     |

JREのバージョンは、コマンドプロンプトを起動して、  
“java -version”を実行したら判別します。

Excelのバージョンは、1番左の[ファイル]タブを選び、  
[アカウント] (下から2番目) ⇒ [Excelのバージョン情報] ボタン (右下) で判  
ります。

ただし後述のClose漏他、ソースファイルの修正が必要になる場合があります  
ので、JREだけではなくJDKも導入する事を強く推奨します。

# 事前準備02

## Javaのバージョン確認。

cmd C:\Windows\System32\cmd.exe

```
C:\WINDOWS\system32>javac -version
javac 1.8.0_102
```

```
C:\WINDOWS\system32>java -version
java version "1.8.0_102"
Java(TM) SE Runtime Environment (build 1.8.0_102-b14)
Java HotSpot(TM) 64-Bit Server VM (build 25.102-b14, mixed mode)
```

```
C:\WINDOWS\system32>cd C:\
```

```
C:\>java Kawakami
```

```
エントリID、 エントリ名、 ねこID、 ねこ名、 ねこ個数、 商品ID、 商品名、 商品数、 植物ID、 植物名、 備品ID、 備品名、 備品数、 食材ID、 食品ID、 食品名、 登録者、 登録日時、 最終更新者、 最終更新日時。
A000000000000001、 最初期エントリ、 0040、 まんぞくさん、 1、 AB003、 商品03、 2、 0009、 ねこじゃらし( 8.0 ~ 8.1 )、 000G、 リュックサック、 1、 0040、 B001、 お菓子の家(¥2000円)、 川上智弘、 2016-05-27 10:30:00.0、 川上智弘、 2016-05-27 10:30:00.0。
A201607071012111、 あいうえお、 0005、 とびみけさん、 2、 AB006、 商品06、 4、 0006、 紫陽花( 1.2 ~ 3.9 )、 000E、 寝袋、 3、 0002、 A001、 トマトスープ(¥150円)、 川上智弘、 2016-07-07 10:12:11.0、 川上智弘、 2016-07-07 10:12:11.0。
A201607071013088、 川上確認エントリ、 0042、 あめしよさん、 3、 AB001、 商品01、 3、 0010、 向日葵( 9.9 ~ 10.0 )、 000A、 カップ、 1、 0040、 F001、 炎の海スペシャル(¥10000円)、 川上智弘、 2016-07-07 10:13:08.0、 川上智弘、 2016-07-07 10:13:08.0。
A201607071015011、 うさまる？、 0019、 おっどさん、 1、 AB012、 商品12、 1、 0002、 デイジー( 3.1 ~ 3.1 )、 000C、 双眼鏡、 9、 0002、 B001、 トマトジュース(¥150円)、 川上智弘、 2016-07-07 10:15:01.0、 川上智弘、 2016-07-07 10:15:01.0。
A201607071018077、 ケータとフミちゃんはラブラヴ、 0002、 くろねこさん、 6、 AB004、 商品04、 9、 0005、 極楽鳥化( 2.5 ~ 7.5 )、 000L、 杖、 4、 0001、 F001、 ぱくだんたまご(¥300円)、 川上智弘、 2016-07-07 10:18:07.0、 川上智弘、 2016-07-07 10:18:07.0。
```

実SQL↓

```
SELECT A.エントリID, A.エントリ名, A.ねこID, B.名前 AS ねこ名, A.ねこ個数, A.商品ID, C.商品名, A.商品数, A.植物ID, IIF(D.植物名 IS NULL, NULL, D.植物名 || '(' || FORMAT(D.保管温度最低値, '0.0') || '~', || FORMAT(D.保管温度最高値, '0.0') || ')') AS 植物名, A.備品ID, E.備品名, A.備品数, A.食材ID, A.食品ID, IIF(F.食品名 IS NULL OR G.食材名 IS NULL, NULL, F.食品名 || '(' || IIF(F.売値 IS NULL, 0, F.売値) || '円') AS 食品名, A.登録者, A.登録日時, A.最終更新者, A.最終更新日時 FROM (((((最初期エントリファイル A LEFT OUTER JOIN 学習用DB B ON (A.ねこID=B.一意ID AND B.非表示フラグ='0')) LEFT OUTER JOIN 商品マスタ C ON (A.商品ID=C.一意ID AND C.非表示フラグ='0' AND C.不良品フラグ='0')) LEFT OUTER JOIN 植物管理マスタ D ON (A.植物ID=D.一意ID AND D.非表示フラグ='0')) LEFT OUTER JOIN 会社備品マスタ E ON (A.備品ID=E.一意ID AND E.非表示フラグ='0')) LEFT OUTER JOIN 食品管理マスタ F ON (A.食材ID=F.食材ID AND A.食品ID=F.食品ID AND F.非表示フラグ='0')) LEFT OUTER JOIN 食材管理マスタ G ON (A.食材ID=G.一意ID AND G.非表示フラグ='0') WHERE A.非表示フラグ='0' ORDER BY A.エントリID
```

```
C:\>
```

# 対応DBサーバー

本ツールに対応しているDBサーバーは以下になります。

- Oracle 11g
- SQL Server 2012
- MySQL 5.5
- PostgreSQL 9.3
- Hyper SQL Database Engine 2.2

ただし、XA対応のJDBCドライバ及びWebLogic ServerにバンドルしているJDBCドライバは**対象外**です。



# DBアクセスログのフォルダ構成

DENEB初回起動時以降に出現します。

## 作業フォルダ

- └ profile\_work
  - └ db
    - └ ICF\_AutoCapsule\_Disabled
      - └ request
  - └ report
    - └ DbAccess
      - └ logs
  - └ svclog
  - └ outputReport.bat
  - └ outputReport.sh





# DBアクセスログのフォルダ説明

| 名称                                      | 用途                                       |
|---|--|
| report¥DbAccess                         | DBアクセスログ集計結果Excelファイル出力フォルダ。             |
| report¥logs                             | DBアクセスログ出力フォルダ。                          |
| svclog                                  | ツール実行時ログ出力フォルダ。                          |
| outputReport.bat                        | DBアクセスログ集計結果Excelコマンド(Windows用)。         |
| outputReport.sh                         | DBアクセスログ集計結果Excelコマンド(Linux用)。           |
| db¥ICF_AutoCapsule_Disabled<br>¥request | DBアクセスログ集計結果Excelコマンドの作業フォルダ(利用者は使用しない)。 |

**集計結果Excelファイルを出力するタイミングは以下の通りです。**

- Javaコマンド終了時。
- outputReport.batまたはoutputReport.shを実行した時。
- deneb.RegularlyReportEnableが**True**の場合。

# DENEbのダウンロード

Javaがインストール済みなら、  
DENEbをダウンロードしましょう。インストールの必要はありません。  
<http://hito.enbbs.jp/essay/file/5/upload/1477062206.jpg>  
上記URLからDL可能です。

DLしたら下記URLからライセンスファイルを取得します。  
<http://img.atwikiimg.com/www9.atwiki.jp/gensouv/attach/346/14/AA.html>

Deneb.jarファイルとNAssetTool.licを**同じフォルダ**に置きます。  
有効期限を超過したら再度DLして下さい。



# CMDでDENEbを実行する場合。

コマンドプロンプトを開き、

cd 【JavaClassファイル】が存在するフォルダ  
を実行後、

```
java -javaagent:【Denebのjarファイルがあるフォルダ】¥Deneb.jar -  
Ddeneb.WorkDir=【ワークファイルフォルダ】¥WorkDir -  
Ddeneb.ClassNameFilter=* -classpath "必要なJDBCドライバの絶対パス";  
【JavaClassファイル】
```

を実行します。

実行後、DBアクセスログ集計結果Excelファイル出力フォルダ  
(report¥DbAccess)、DBアクセスログ出力フォルダ(report¥logs) 及びツール  
実行時ログ出力フォルダ(svclog)の各種ファイルを確認して下さい。

コマンドプロンプトで実施する場合、

[C:¥Program Files¥Java¥[jreフォルダ]¥lib¥ext] フォルダ直下のJDBCドライバ.  
Jarファイルは適当なフォルダにリムーブして下さい。

# APサーバでDENEbを実行する場合。

JDBCドライバをクラスパスで参照する際は、必要なドライバを任意のフォルダに複製して参照設定して下さい。

[C:¥Program Files¥Java¥[jreフォルダ]¥lib¥ext] フォルダ直下のJDBCドライバ参照は非推奨です。

基本的に行う事はオプションに以下を追加。

- 必要なJDBCドライバの絶対パスの追加。
- 前頁にも記述した**DENEbコマンド**の追加。

以上2点です。

今一度書いてあげますが、DENEbコマンドの基本的な記述は以下の通り。

- [-javaagent:【**Denebのjarファイルがあるフォルダ**】¥Deneb.jar]
- [-Ddeneb.WorkDir=【**ワークファイルフォルダ**】¥WorkDir]
- [-Ddeneb.ClassNameFilter=\*]

他のコマンドは必要に応じて追加して下さい。

DENEbコマンドは原則的に [Java Options] の設定になります。

# APサーバ設定早見表

数あるAPサーバーの内、TomCat、Eclipse及びWebLogicの設定表を下に記載します。

| APサーバー   | 設定準備  | JDBCドライバー設定   | DENEBCOMMAND設定                  |
|----------|---|---|---------------------------------|
| TomCat   | [Tomcat8w.exe]を右クリック⇒<br>【管理者として実行】を選択。                                     | [Java]タブの<br>[Java Classpath:]  | [Java]タブの<br>[Java Options:]    |
| Eclipse  | 実行(R)⇒実行構成(N)   | [クラスパス]タブの[ユーザー・エントリー]配下の[デフォルト・クラスパス]にカーソルを当てて[外部JARの追加(X)]ボタンをクリックする。 | [引数]タブの[VM 引数(G)]               |
| WebLogic | [WebLogicインストールフォルダ]¥user_projects¥domains¥base_domain¥bin¥setDomainEnv.cmd | @REM SET THE CLASSPATHの記述部分の直下。   | @REM SET THE CLASSPATHの記述部分の直前。 |

# DBアクセスログ機能01

APサーバーもしくはコマンドプロンプトでJavaClassファイル実行後、  
[作業フォルダ¥profile\_work] フォルダ配下の  
[report¥DbAccess] フォルダにDBアクセスログ集計結果Excelファイル  
が、  
[report¥logs] フォルダにDBアクセスログ出力ファイルが、  
[svclog] フォルダにツール実行時ログ出力ファイルが、  
各々出力されます。



# DBアクセスログ機能02

## 作業フォルダ

- └ profile\_work
  - └ db
    - └ ICF\_AutoCapsule\_Disabled
      - └ request
  - └ report
    - └ DbAccess
      - └ DBLogSummary\_YYYYMMDD\_HH.xls... (DBアクセスログ集計結果)
      - └ logs
        - └ db\_access.log... (アクセスログ)
        - └ db\_close\_detect.log... (クローズ漏ログ)
  - └ svclog
    - └ profile\_svc\_YYYYMMDD\_HHMMSS\_SSS.log... (ツール実行時ログ)
  - └ outputReport.bat
  - └ outputReport.sh

## 青文字のファイル

([DBLogSummary\_YYYYMMDD\_HH.xls] [db\_access.log] [db\_close\_detect.log] [profile\_svc\_YYYYMMDD\_HHMMSS\_SSS.log]) を次の頁から見て行きます。

# DBアクセスログ機能03

DBアクセスログ機能では、データベースアクセス時に下記のイメージのログファイルを出力します。

ログは大きく分けて5種類存在します。

- 生成されたSQL
  - **[SQL生成]**の行に記述。Javaファイルに実際に記述されたSQLを出力します。
- 発行されたSQL
  - **[SQL発行]**の行に記述。パラメータ部分はSQL発行時の値に置き換えて出力されます。
- 結果セット
  - **[SQL正常終了]**の下の方に記述。SQL文の結果が出力されます。

他にも[トランザクション開始][トランザクション終了]の記述がありますが、オートコミットモードの場合は出力されません(通常はオートコミットモードになってます)。



# DBアクセスログ機能04

ログ行の詳細は以下の通りです。

| ログ行の種類              | 出力条件  |
|---------------------|---|
| トランザクション開始          | トランザクションが開始された時点で出力。                            |
| SQL生成               | SQLステートメントの生成に成功した時点で出力。                        |
| SQL生成除外             | SQLステートメントの生成でエラーが発生した時点で出力。                    |
| SQL発行               | SQLを発行した時点で出力。                                  |
| SQL正常終了<br>(+SQL結果) | SQLを発行後、正常終了した時点で出力。<br>(SQL結果はSELECT文などの結果を返す) |
| SQL例外               | SQLを発行後、エラー終了した時点で出力。                           |
| トランザクション終了          | トランザクションがコミット、または<br>ロールバックされた時点で出力。            |

# DBアクセスログ機能05

クローズ漏れ監視ログ出力機能について説明します。

アプリケーションはDBにアクセスした後、一時的に使用したDB上のリソースを開放する必要があります。

アプリケーションがこの処理を怠った場合、処理終了後もDB上のリソースが処理され続け、DBは処理に必要なリソースを継続的に確保出来なくなる可能性があります。

本機能では、DB上のリソースを使用するJDBCオブジェクトがアプリケーション終了後に開放されていない場合に、ログファイルの警告情報を出力し、開発者に改善を促します。

尚、**[クローズ漏れログ 記録開始: YYYY/MM/DD HH:MM:SS. SSS] の記述だけ**の場合、クローズ漏れは**発生していなかった事**になります。

ログ行の詳細は以下の通りです。

| ログ行の種類           | 出力条件  |
|------------------|---|
| Connectionクローズ漏れ | DB接続のclose()を行わず破棄した後、ガベージコレクションが発生した時点で出力。       |
| Statementクローズ漏れ  | SQLステートメントのclose()を行わず破棄した後、ガベージコレクションが発生した時点で出力。 |
| ResultSetクローズ漏れ  | 結果セットのclose()を行わず破棄した後、ガベージコレクションが発生した時点で出力。      |

# DBアクセスログ機能06

DBアクセスログ集計結果ファイルは三つのシートで構成されています。

[SQLパターン一覧] 票は、

生成されたSQL毎の集計情報が出力されます。

生成されたSQL毎のSQL発行回数やSQL実行時間を確認したり、

SQL例外やクローズ漏れが発生しているSQLを特定する事が出来ます。

集計時には、SQL文が同一でも、生成したクラス、メソッド、行番号が同一でない場合、別のレコードとして集計されます。

またSQL文字列は大文字、小文字を区別して集計します。



# DBアクセスログ機能07

SQLパターン一覧票に出力される項目は以下になります。

| 項目名           | 説明                          |
|---------------|-----------------------------|
| 生成クラス名        | SQLを生成したクラスのクラス名。           |
| 生成メソッド名       | SQLを生成したメソッドのメソッド名。         |
| 行番号           | SQLを生成した処理を実行している行番号。       |
| SQLイメージ       | 生成されたSQLの文字列。               |
| SQL生成回数       | SQLが生成された回数。                |
| SQL発行回数       | 生成されたSQLが発行された回数。           |
| SQL累計実行時間(ms) | SQL実行時間の合計(ミリ秒)。            |
| SQL平均実行時間(ms) | SQL実行時間の平均値(ミリ秒)。           |
| SQL最大実行時間(ms) | SQL実行時間の最大値(ミリ秒)。           |
| SQL最小実行時間(ms) | SQL実行時間の最小値(ミリ秒)。           |
| SQL例外発生回数     | SQL発行時に発生したSQL例外の回数。        |
| SQL例外最終発生時刻   | SQL例外が最後に発生した時刻。            |
| クローズ漏れ発生回数    | Statementのクローズ漏れ発生回数。       |
| クローズ漏れ最終発生時刻  | Statementのクローズ漏れが最後に発生した時刻。 |

# DBアクセスログ機能08

- 回数項目は、0回のはものは'0'が表示されます。
- [生成クラス名]は、DBアクセスログにクラス名が生成されていない場合は、生成クラス名【不明】として、一行に纏めて出力されます。
- [SQL発行回数]は、SQLの各生成情報と各発行情報が同一でない場合も、生成したクラス、メソッド、行番号に計上します。
- ソート順は[生成クラス名]の昇順になります。



# DBアクセスログ機能09

**[SQL発行クラス一覧]** 票は、  
クラス毎の集計状況が出力されます。  
クラス毎のSQL発行回数やSQL実行時間を確認したり、SQL例外やクローズ漏が発生しているクラスを特定出来ます。

SQL発行クラス一覧票に出力される項目は以下になります。

| 項目名           | 説明                            |
|---------------|-------------------------------|
| 発行クラス名        | SQLを生成、または発行したクラスのクラス名。       |
| SQL生成回数       | クラスでSQLを生成した回数。               |
| SQL発行回数       | クラスでSQLを発行した回数。               |
| SQL累計実行時間(ms) | クラスで発行されたSQL文の応答時間の合計(ミリ秒)。   |
| SQL例外発生回数     | クラスで発行されたSQLから発生したSQL例外の回数。   |
| SQL例外最終発生時刻   | SQL例外が最後に発生した時刻。              |
| クローズ漏れ発生回数    | クラスで発生したStatementのクローズ漏れ発生回数。 |
| クローズ漏れ最終発生時刻  | Statementのクローズ漏れが最後に発生した時刻。   |

- [発行クラス名] は、DBアクセスログにクラス名が生成されていない場合は、発行クラス名【不明】として、一行に纏めて出力されます。
- ソート順は [発行クラス名] の昇順になります。

# DBアクセスログ機能10

**[アラート一覧]票は、**  
**集計された各項目の値をチェックし、「項目毎に設定された敷居値を超過した」場合にセルを着色して指摘します。**  
**正常値は白（塗らない）、上限注意値は橙色、上限警告値は赤色で着色されます。**  
**色の着色はSQL発行クラス一覧票、SQLパターン一覧票も対象です。**

**アラート一覧票に出力される項目は以下になります。**

| 項目名    | 説明  |
|--------|---|
| 指摘項目   | 指摘された項目。指摘が注意値の場合は橙色に、警告値の場合は赤色で着色して表示される。                |
| メッセージ  | 指摘内容が出力される。   |
| 指摘クラス名 | SQL発行クラス一覧票、SQLパターン一覧票にアラートが出力された場合に、生成クラス名、発行クラス名が出力される。 |
| 指摘メソッド | SQLパターン一覧票にアラートが出力された場合、呼び出し元の生成メソッド名が出力される。              |
| 指摘行番号  | SQLパターン一覧票にアラートが出力された場合、呼び出し元のソースコードの行番号が出力される。           |
| 最終発生時刻 | SQL例外発生回数、クローズ漏れ発生回数にアラートが出力された場合、その最終発生時刻が出力される。         |



# DBアクセスログ機能11

- SQL発行クラス一覧票は[指摘メソッド][指摘行番号]以外が対象、SQLパターン一覧票は全項目が対象です。
- 何も表示が無かった場合はアラートが発生しなかった事になります。  
SQL例外発生回数及びクローズ漏れ発生回数は各々0回を超過すると上限注意値に、5回を超過すると上限警告値になります。  
SQL平均実行時間(ms)及びSQL最大実行時間(ms)は各々3,000msを超過すると上限注意値に、10,000msを超過すると上限警告値になります。  
即ち3秒と10秒が境界線になります。





# DBアクセスログ機能12

ツール実行時ログはDENEb実行時に設定していた引数の内容が表示されます。  
引数名の左側に [-D] を付けて設定して下さい。

| 引数名                            | 用途  | デフォ値     |
|--------------------------------|---|----------|
| deneb. WorkDir                 | DBログ取得対象のプログラム作業フォルダ名。  | カレントフォルダ |
| deneb. DBLogEnable             | DBログ取得機能の有効／無効を設定する。<br>有効はTrue、無効はFalse。                       | True     |
| deneb. RegularlyReportEnable   | DBログ集計結果を定期的に出力するか否かを設定する。<br>出力する場合はTrue、出力しない場合はFalse。        | False    |
| deneb. RegularlyReportInterval | DBログ集計結果を定期的に出力する場合の感覚を[分単位]で指定する。<br>0を指定した場合はデフォ値が適用される。      | 60       |
| deneb. MessageToConsole        | DBアクセスログ引数設定を各コンソールに出力するか否かを設定する。<br>出力する場合はTrue、出力しない場合はFalse。 | False    |
| deneb. ClassNameFilter         | 呼び出し元クラスのクラス名パターン文字列を設定する。<br>ワイルドカード(*)を使用可能。                  | *        |
| deneb. ResultSetLimit          | SELECT文発行時の結果セットの出力最大件数を設定する。<br>0の場合は、結果セットは出力しない。             | ∞        |

スペルミス、大文字小文字が合っていないと [Null] 扱いになるので注意して下さい。例えばClassNameFilterの設定で [classnamefilter] (全て小文字) にするとNullになります。

# 終いに

基本的な説明は以上で終わります。

より発展した内容を知りたいなら

<http://mp.i-revo.jp/user.php/xjwubzee/attach/108/Ainsel2.jpg>

からDLしてガイドのPDFファイルをご覧ください。



# 終了

➤ 終了です。

