

トランザクション追跡機能JProT
(JBoss Profiler with Method Flow Tracer)
使用手順書

日立製作所

1	はじめに	3
2	動作環境構築手順.....	4
2.1	JProTインストール手順.....	4
2.1.1	JBoss Profilerのダウンロード	4
2.1.2	JProTパッチの適用/プログラム生成.....	4
2.2	ログ蓄積環境における動作環境構築手順	4
2.2.1	ログ蓄積部の配置.....	5
2.2.2	開始・終了指示部のデプロイ	5
2.2.3	JVMオプションの設定.....	5
2.3	ログ解析環境における動作環境構築手順	6
2.3.1	処理部・表示部のデプロイ	6
2.3.2	Javaアプレットをブラウザで使用するための設定.....	6
3	使用手順.....	7
3.1	ログ蓄積環境におけるログ蓄積手順	7
3.1.1	JBossの起動.....	7
3.1.2	ログ蓄積の開始	7
3.1.3	Javaプログラムの実行.....	9
3.1.4	ログ蓄積の終了	10
3.2	ログ解析環境における処理・表示手順.....	10
3.2.1	JBoss Profiler画面の起動	10
3.2.2	ログ蓄積先ディレクトリの指定	10
3.2.3	プロセスID(PID)の指定.....	11
3.2.4	JProT機能の選択	13
3.2.5	入口メソッドの指定	14
3.2.6	トランザクション検索・絞込み/一覧表示画面.....	15
3.2.7	メソッド実行経路の表示.....	19

1 はじめに

JProT(JBoss Profiler with Tracer)は、Java プログラム内部で実行されるトランザクション(メソッドの実行経路・呼び出し関係)を追跡し可視化するツールである。JProT は、JBoss コミュニティにて開発中のプロファイラ・ツール JBoss Profiler をベースとして開発した拡張機能であり、JBoss Profiler の仕組みに大きく依存する。

JProT は、Java プログラムを実行し、そのログを蓄積する環境(以下、ログ蓄積環境とする)と、ログを処理・表示する環境(以降、ログ解析環境とする)に大別できる。これらは異なる JBoss 上で動作させることも、同一の JBoss 上で動作させることも可能である。

2 および 3 において、動作環境構築手順ならびに使用手順について説明する。なお、使用手順は、既存 JBoss Profiler と共通となる部分も多いため、別途 JBoss Profiler 付随の手順書も参照されたい。JBoss Profiler の各種ドキュメントは、JBoss Profiler のインストール・ディレクトリ配下の docs ディレクトリに格納されている。

以降、本報告では、説明の容易化のため、表 2.1-1 に示す凡例を使用する。

表 2.1-1 本報告における凡例

略号	説明
PROFILER_DIR	JBoss Profiler ダウンロード先ディレクトリ
JBOSS_DIR	JBoss インストール先ディレクトリ

なお、説明に際して、コマンドやパラメタ値と明示的に区別するために、上記凡例および、環境に応じて変わる変数等は**太字**により表記するものとする。

2 動作環境構築手順

2.1 に、JProT のインストール手順を示す。2.2 ならびに 2.3 に、ログ蓄積環境およびログ解析環境における、動作環境構築手順を示す。

2.1 JProT インストール手順

JProT のインストール手順を示す。本手順は、任意のマシン上で実施すれば良い。ただし、あらかじめ JDK/JBoss/Ant/gcc がインストールされていることを前提とする。

2.1.1 JBoss Profiler のダウンロード

以下のサイトで紹介されている手順に従い、JBoss Profiler をダウンロードする。

```
http://wiki.jboss.org/wiki/Wiki.jsp?page=JBossProfiler
```

なお、現在の JProT は、2005/07/04 時点の JBoss Profiler をベースとして開発しており、それ以外のバージョン上での動作を保証しない。JBoss Profiler のダウンロードにあたっては、"-D"オプションにより該当バージョンを指定する。

2.1.2 JProT パッチの適用/プログラム生成

JProT 付属の setup.sh ファイルを編集し、冒頭の"JBOSS_PROFILER"変数に、JBoss Profiler のダウンロード先ディレクトリ PROFILER_DIR を指定する。

```
JBOSS_PROFILER=PROFILER_DIR
```

指定後、setup.sh ファイルの編集内容を保存し、実行する。

```
$/setup.sh
```

setup.sh の実行が正常に完了すると、JBoss Profiler に JProT パッチが適用され、JProT の動作に必要なファイルが生成される。以下 3 つのファイルが生成されていることを確認する。

```
PROFILER_DIR/build/jboss-profiler-noAOP.sar  
PROFILER_DIR/build/jboss-profiler.war  
PROFILER_DIR/native/linux/libjbossInspector.so
```

2.2 ログ蓄積環境における動作環境構築手順

ログ蓄積環境における動作環境構築手順を示す。ログ蓄積環境を構築するマシンには、

あらかじめ、JDK/JBoss がインストールされていることを前提とする。

2.2.1 ログ蓄積部の配置

以下のライブラリを、適当なディレクトリに配置する。

```
PROFILER_DIR/native/linux/libjbossInspector.so
```

上記ライブラリを配置したディレクトリを、環境変数”LD_LIBRARY_PATH”で指定する。

```
export LD_LIBRARY_PATH=<ライブラリ配置先ディレクトリ>
```

2.2.2 開始・終了指示部のデプロイ

PROFILER_DIR の build ディレクトリ配下にある jboss-profiler-noAOP.sar ファイルを、JBoss にデプロイする。

```
$cd PROFILER_DIR/build/  
$scp jboss-profiler-noAOP.sar JBOSS_DIR/server/<設定セット名>/deploy/
```

2.2.3 JVM オプションの設定

JVM がログ蓄積部 (libJbossInspector.so ライブラリ) を使用できるように、変数 JAVA_OPTS を設定する。また、オプション変数により、ログ蓄積先ディレクトリや解析対象とするパッケージを指定する。

```
JAVA_OPTS= " -XrunjbossInspector:<蓄積先ディレクトリ>,  
             include=<対象パッケージ>,  
             ignore=*,  
             memory=false,  
             tracer=true "
```

表 2.2-1 オプション変数

変数	説明	備考
include	追跡対象とするメソッドをパッケージ単位で指定する。	複数パッケージを指定する場合は、以下のように列挙。 <pre>include=<対象パッケージ>, include=<対象パッケージ>,...,</pre>

変数	説明	備考
ignore	追跡対象外とするメソッドをパッケージ単位で指定する。“*”を指定すると、include で指定したパッケージ以外は追跡しない。	基本的には“*”を指定。
memory	メモリ・プロファイラ用のログを蓄積する。(デフォルト: true)	JProT 使用時には“false”を指定。
tracer	JProT 固有のオプション。例外発生時に、発生した例外を含む実行経路を取得する。(デフォルト: false)	JProT 使用時には“true”を指定。

2.3 ログ解析環境における動作環境構築手順

ログ解析環境における動作環境構築手順を示す。ログ解析環境を構築するマシンには、あらかじめ、JDK/JBoss がインストールされていることを前提とする。

なお、既存 JBoss Profiler の備えるアプレット表示機能を使用しない場合、2.3.2 に示す手順は必須ではない。

2.3.1 処理部・表示部のデプロイ

PROFILER_DIR の build ディレクトリ配下にある jboss-profiler-noAOP.sar および jboss-profiler.war を、JBoss にデプロイする。

```
$cd PROFILER_DIR/build/
$cp jboss-profiler-noAOP.sar JBOSS_DIR/server/<設定セット名>/deploy/
$cp jboss-profiler.war JBOSS_DIR/server/<設定セット名>/deploy/
```

2.3.2 Java アプレットをブラウザで使用するための設定

JBoss Profiler は、Java アプレットによる表示機能を備えている。Java アプレットをブラウザ上で動作させるには、ブラウザに Java プラグインを設定する必要がある。設定方法は、以下のサイトで紹介されている手順に従う。

```
http://www.knowd.co.jp/staff/rotake/java-plugin.html
http://java.sun.com/j2se/1.4.2/ja/manual_install_linux.html
```

例として、RHEL3(Mozilla-1.4.3)における設定方法を以下に示す。

```
#cd /usr/lib/mozilla-1.4.3/plugins/
```

```
#ln s $JAVA_HOME/jre/plugin/i386/ns610-gcc32/libjavaplugin_oji.so
```

なお、ln コマンド行の i386 ディレクトリ配下には、同名のライブラリを格納する ns610 ディレクトリも存在する。誤ってこちらのライブラリにシンボリックリンクを張った場合、Mozilla ブラウザ自体が起動しなくなるため注意されたい。

3 使用手順

ログ蓄積環境およびログ解析環境における、使用手順を示す。

3.1 ログ蓄積環境におけるログ蓄積手順

以下は、ログ蓄積環境における一般的な使用手順である。ただし、JBoss Profiler は、ログ蓄積を任意時点で開始・終了できる特徴機能を備えており、Java プログラム実行最中の開始・終了も可能である。

3.1.1 JBoss の起動

JBoss を起動する。

```
$cd JBOSS_DIR/bin/  
$./run.sh -c <設定セット名>
```

JBoss が起動すると、JProT も起動する。JBoss の起動コンソールに、以下のように表示されれば、JProT が正常に起動したことを示す。ただし、 で囲んだ数値は、JVM を起動したプロセス ID(PID)であり、一般的には JVM 起動毎に異なる。

```
Running process 4684  
JBoss Profiler: Dissabling memory events  
JBoss Tracer: Enabling Exceptions/Errors
```

3.1.2 ログ蓄積の開始

以下の URL にアクセスすると、JBoss の JMX 管理コンソール画面が表示される。

```
http://localhost:8080/jmx-console/
```

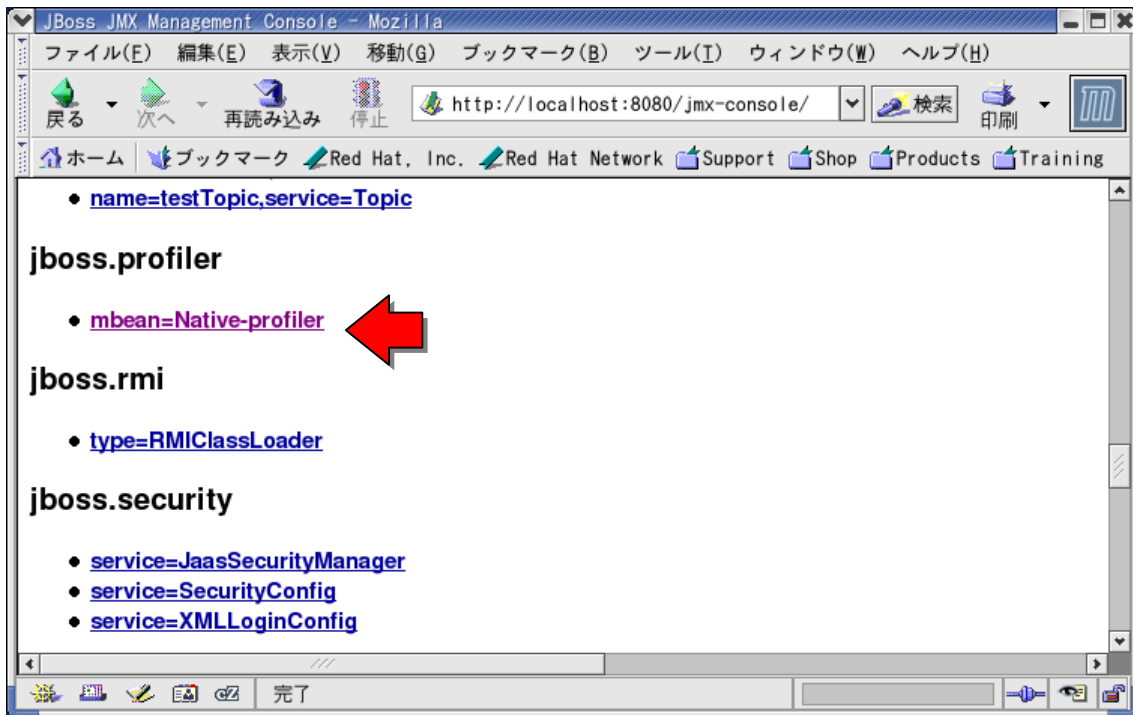


図 3.1-1JMX 管理コンソール画面

“jboss.profiler”ドメイン以下の”mbean=Native-profiler”を選択すると、図 3.1-2 の画面が表示される。

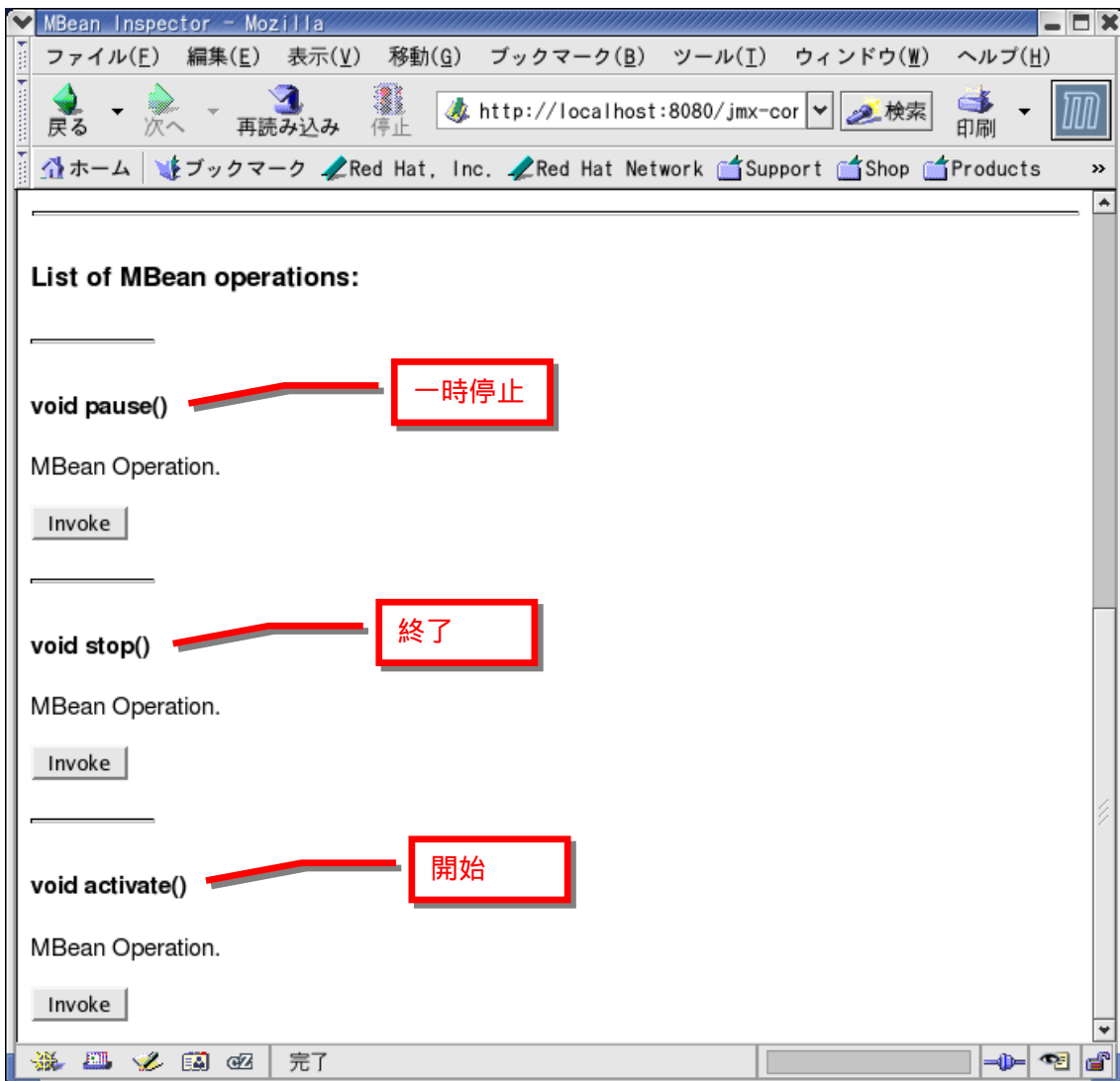


図 3.1-2JBoss Profiler/JProT のオペレーション

表示される画面の最下部にある”active”オペレーションを実行する。画面に、以下のメッセージが表示される。

```
Operation completed successfully without a return value.
```

同時に、3.1.1 で JBoss を起動したコンソールに、以下のメッセージが表示されれば、ログ蓄積部が正常に開始したことを示す。

```
JBoss Profiler: Initializing DataGathering
```

3.1.3 Java プログラムの実行

2.2.3 で指定した、解析対象とするパッケージを含む Java プログラムを実行する。

3.1.4 ログ蓄積の終了

3.1.2 の”active”オペレーションと同じ画面にある”stop”オペレーションを実行する。画面に、以下のメッセージが表示される。

```
Operation completed successfully without a return value.
```

同時に、3.1.1 で JBoss を起動したコンソールに、以下のメッセージが表示されれば、ログ蓄積部が正常に終了したことを示す。

```
Stopping DataGathering
```

3.2 ログ解析環境における処理・表示手順

ログ解析環境において、3.1 で蓄積したログを処理・表示する手順を示す。なお、ログ蓄積環境とログ解析環境が同一の場合、3.1.1 で起動した JBoss プロセス上で処理・表示手順を実施することも可能であるが、解析中もログ蓄積部が動作するため、パフォーマンスが劣化することに注意する必要がある。

なお、以下 3.2.1 ~ 3.2.3 は、既存 JBoss Profiler と共通となる手順である。JProT の解析機能は、手順 3.2.1 ~ 3.2.3 を実施することではじめて使用可能となる。手順 3.2.1 ~ 3.2.3 記載以外の既存 JBoss Profiler の使用手順については、本報告では割愛する。

3.2.1 JBoss Profiler 画面の起動

以下の URL にアクセスする。

```
http://localhost:8080/jboss-profiler/
```

3.2.2 ログ蓄積先ディレクトリの指定

解析対象とするログが蓄積されているディレクトリを指定する。ディレクトリは、ローカルに存在する必要がある。ログ蓄積環境と同一マシン上で動作させる場合、一般的に、2.2.3 の JAVA_OPTS で指定したディレクトリを指定する。ログ蓄積環境と異なるマシン上で動作させる場合は、あらかじめログ・ファイルをローカルに移動させておく必要がある。

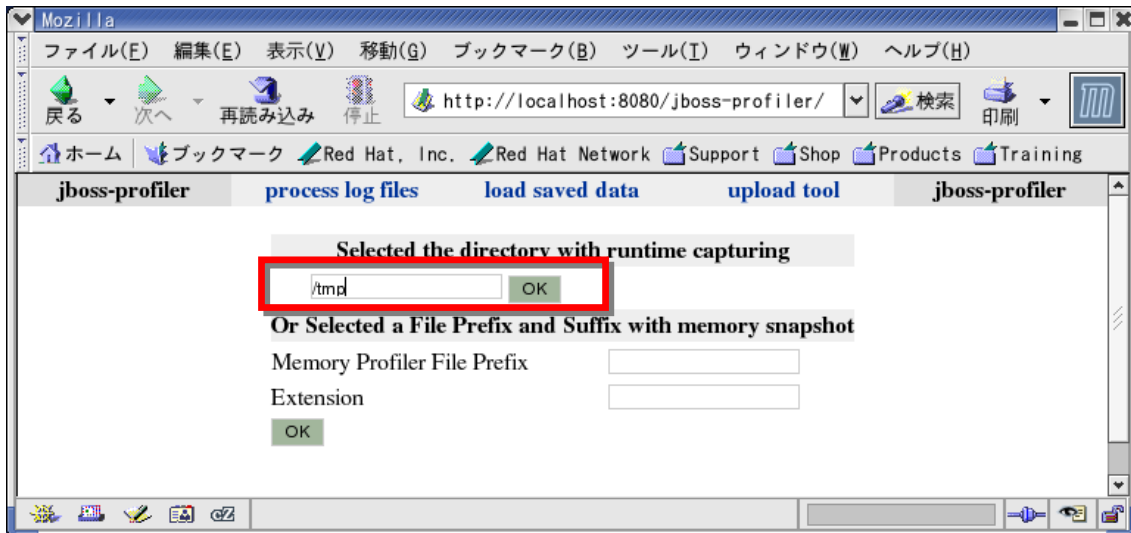


図 3.2-1 ディレクトリ指定画面

ディレクトリの指定後、“ok”をクリックすると、3.2.3 に示 PID 指定画面に進む。

3.2.3 プロセス ID(PID)の指定

解析対象とする PID を指定する。コンボボックスに、3.2.1 で指定したディレクトリに格納される全てのログの PID が表示されるので、その中から選択する。

その下に表示される 3 つのチェックボックスは、デフォルトでチェックされている。2.2.3 で“memory=true”を指定した場合はそのままでも良いが、そうでない場合はチェックを外す。(チェックの有無により、既存 JBoss Profiler の処理に要する時間が大きく異なる場合がある。)

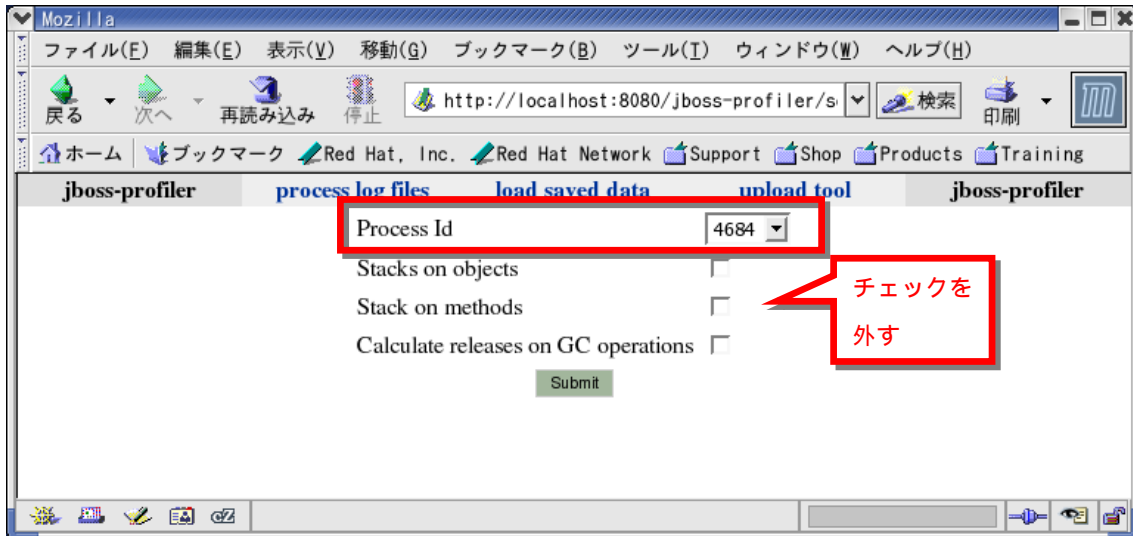


図 3.2-2PID 指定画面

指定以後、“Submit”をクリックすると、サーブレットが処理を開始する。画面には、処理するログ・ファイル名と、処理の進捗状況(%)が表示される。処理が終了すると、画面最下部に“[click here](#)”というリンクが表示される。このリンクをクリックすると、3.2.4 に示す JBoss Profiler画面に進む。

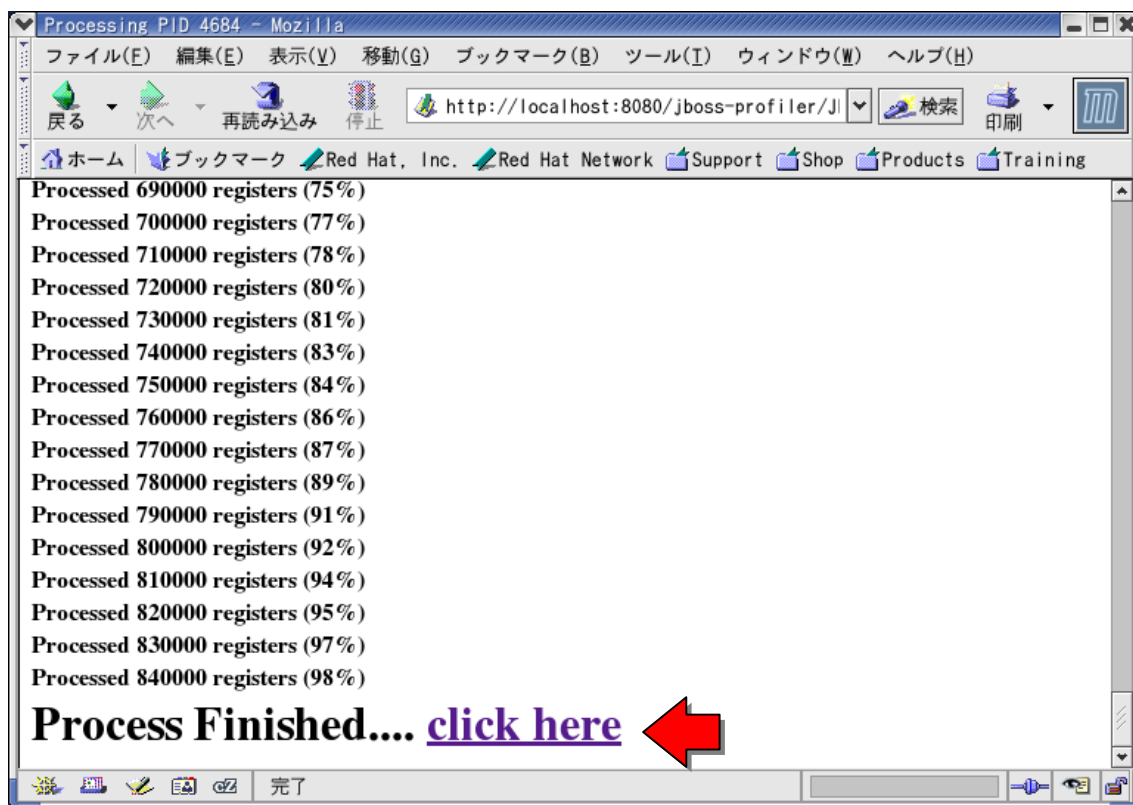


図 3.2-3 サブレット終了画面

3.2.4 JProT 機能の選択

前記 3.2.3 までの手順により表示されるのは、既存 JBoss Profiler による処理結果を示す最初の画面である。既存 JBoss Profiler の様々なビューアーの詳細については割愛する。

ここでは、JProT 機能を選択する。“jboss-tracer”メニューの transaction tracer”リンクをクリックすると、3.2.5 に示す JProT 画面に進む。

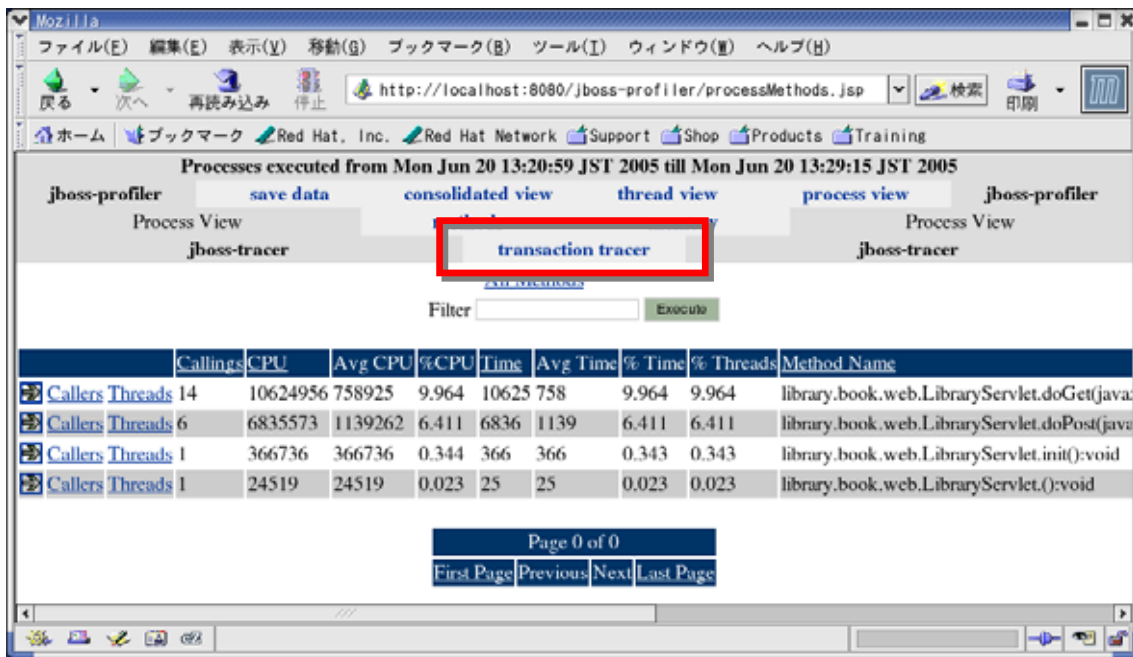


図 3.2-4 JBoss Profiler 画面と JProT へのリンク

3.2.5 入口メソッドの指定

トランザクションの入口となるメソッドを指定する。メソッドは、コンボボックスに候補が一覧表示されるので、その中から選択するか、テキストボックスにメソッド名を入力することで指定する。

テキストボックスに入力する場合は、メソッドの完全名称またはその一部を文字列で指定する。メソッドの完全名称は以下の形式である。図 3.2-4 に示す画面から、“Method Name”の文字列をコピーして貼り付けるのが容易である。

パッケージ名.メソッド名.(引数,引数,...):メソッドの型

なお、コンボボックスおよびテキストボックスの両方に入力した場合は、これ以降の処理において、コンボボックスの入力内容が優先される。

また、画面最下部のチェックボックスはデフォルトでチェックされているが、チェックを外すと、後述のサーブレット処理画面に表示されるログ・ファイル名を非表示にし、処理の進捗状況(%)のみを表示することができる。

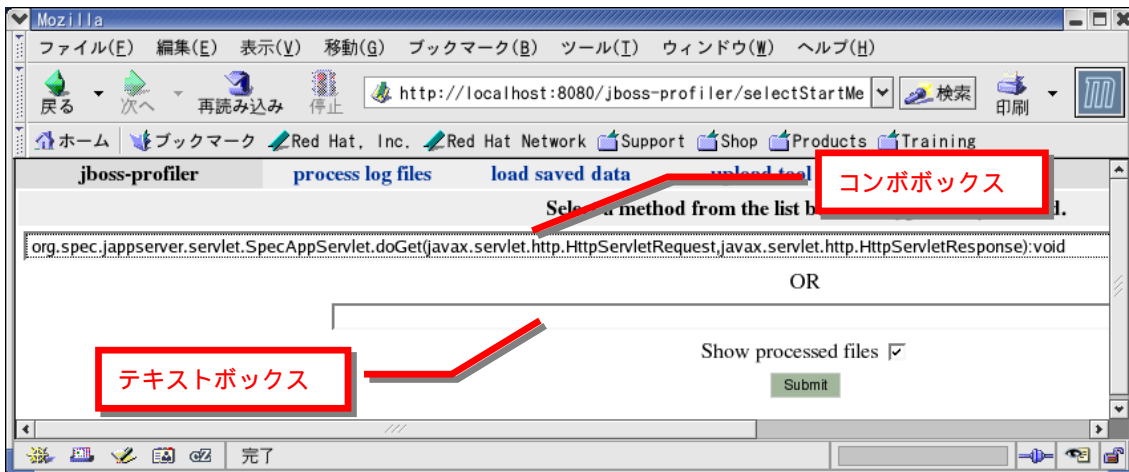


図 3.2-5 入口メソッド指定画面

メソッド指定後、“Submit”をクリックすると、サーブレットが処理を開始する。

テキストボックスで指定した文字列を含むメソッドが存在しない場合、その旨が画面に表示されるので、ブラウザの「戻る」ボタンによりメソッド指定画面に戻り再指定を行う。複数存在する場合は、その文字列を含む全てのメソッドがコンボボックスに一覧表示されるので、その中から選択し“Submit”をクリックする。

指定したメソッドが 1 つ存在する場合に、サーブレットは追跡処理を行う。前述のチェックボックスのチェックを外している、ログ・ファイル名を非表示にし、進捗状況(%)のみを表示することができる。処理が終了すると、3.2.3 と同様、画面最下部に“[click here](#)”というリンクが表示される。このリンクをクリックすると、3.2.6 のトランザクション検索・絞込み/トランザクション一覧表示画面に進む。

3.2.6 トランザクション検索・絞込み/一覧表示画面

図 3.2-6 画面の上半分には、3.2.5 において指定したメソッドを入口メソッドとする全てのトランザクションが表示される（トランザクション一覧表示画面）。下半分には、検索・絞込みを行う画面が表示される（トランザクション検索・絞込み画面）。

トランザクション検索・絞込み画面を 3.2.6.1 で、トランザクション一覧表示画面を 3.2.6.2 で説明する。

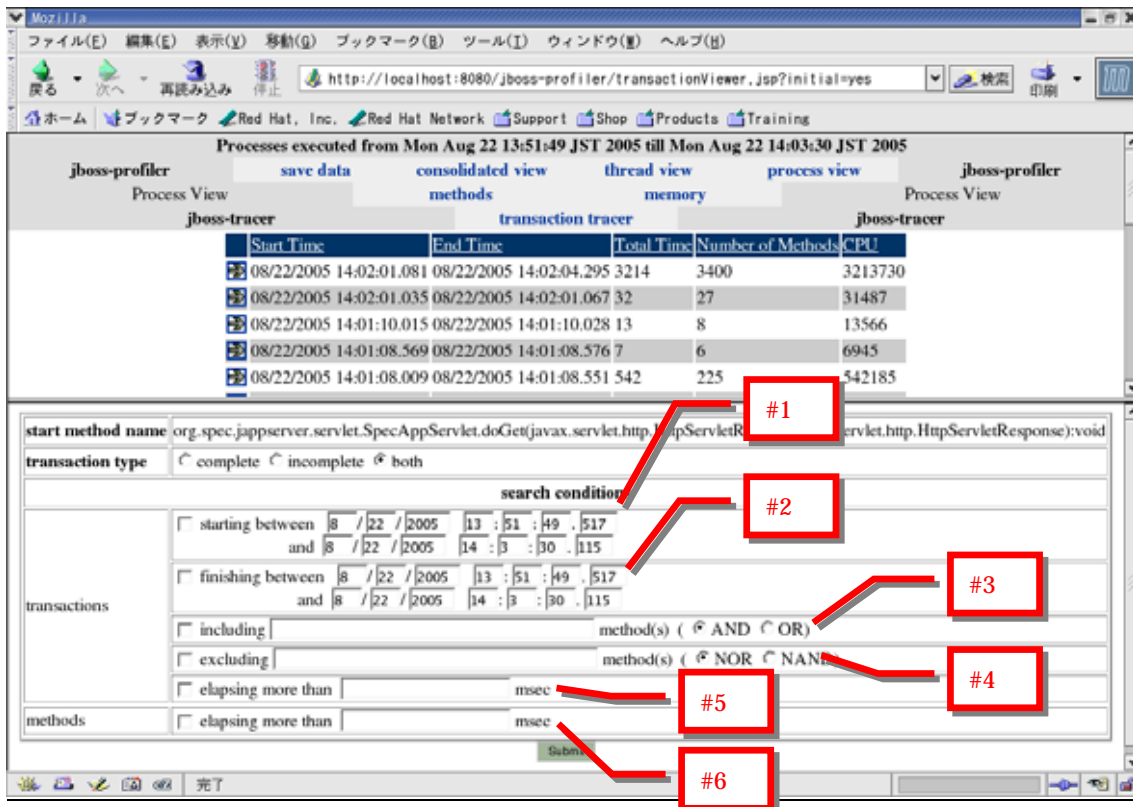


図 3.2-6 トランザクション検索・絞り込み/一覧表示画面

3.2.6.1 トランザクション検索・絞り込み画面

トランザクション検索・絞り込み画面（図 3.2-6 下図）に表示される項目について、以下に説明する。

表 3.2-1 検索・絞り込み画面の大項目

大項目	説明	備考
start method name	3.2.5 で指定したメソッド名が表示される。ここでメソッド名を変更することはできない。	
transaction type	検索対象とするトランザクションのタイプを指定する。	詳細は表 3.2-2 を参照。
Search conditions	検索条件を指定する。6 種類の条件が指定可能である。複数の条件を選択した場合、それらの AND 条件とみなす。	詳細は表 3.2-3 を参照。

表 3.2-1 に示した”transaction type”のオプションについての説明を、表 3.2-2 に示す。

表 3.2-2 transaction type のオプション

オプション	説明
complete	入口メソッドの開始から終了まで正常に完了したトランザクションを表示する。
incomplete	入口メソッドが終了することなく途中で途切れてしまったトランザクションを表示する。
both	complete、incomplete 両方を表示する。(デフォルト)

表 3.2-1 に示した”search conditions”で指定可能な条件を表 3.2-3 に、その指定方法を表 3.2-4 に示す。なお、表 3.2-3 ならびに表 3.2-4 の項番(#)は、図 3.2-6 に示す番号に対応する。

表 3.2-3 search conditions で指定可能な条件

#	説明
1	トランザクションの開始時間帯により絞り込む。デフォルトで、ログ蓄積開始時刻から終了時刻までが表示されるため、適宜変更のこと。
2	トランザクションの終了時間帯により絞り込む。デフォルトで、ログ蓄積開始時刻から終了時刻までが表示されるため、適宜変更のこと。
3	特定メソッドを含むトランザクションを検索する。指定したメソッドが複数個の場合、それら全てを含む AND 検索、または、いずれか 1 つ以上を含む OR 検索を選択できる。
4	特定メソッドを含まないトランザクションを検索する (NOT検索)。指定したメソッドが複数個の場合、それらをいずれも含まない NOR 検索と、いずれか 1 つ以上を含まない NAND 検索を選択できる。
5	トランザクション全体 (入口メソッドの開始から終了まで) の経過時間が一定時間を超えるようなトランザクションを検索する。
6	トランザクション内部の任意のメソッドの経過時間が一定時間を超えるようなトランザクションを検索する。

表 3.2-4 条件の指定方法

#	説明
1	検索時間帯の開始時刻および終了時刻を、以下の形式で指定する。 「月(MM)/日(dd)/年(yyyy) 時(hh):分(mm):秒(ss):ミリ秒(SSS)」
2	検索時間帯の開始時刻および終了時刻を、以下の形式で指定する。 「月(MM)/日(dd)/年(yyyy) 時(hh):分(mm):秒(ss):ミリ秒(SSS)」
3	メソッド名に含まれる文字列を指定する。指定された文字列を含む全てのメソッドが検索対象となる。複数個のメソッドを指定するには、スペース“ ”で区切って列挙する。
4	メソッド名に含まれる文字列を指定する。指定された文字列を含む全てのメソッドが検索対象となる。複数個のメソッドを指定するには、スペース“ ”で区切って列挙する。
5	閾値となる時間をミリ秒(SSS)で指定する。
6	閾値となる時間をミリ秒(SSS)で指定する。

検索条件の指定後、“Submit”をクリックすると、検索条件に従い絞り込まれたトランザクションの一覧が、トランザクション一覧表示画面に表示される。

3.2.6.2 トランザクション一覧表示画面

トランザクション一覧表示画面（図 3.2-6 上図）には、3.2.6.1 で指定した検索条件に従い絞り込まれたトランザクションの一覧が、テーブル形式により表示される。

テーブルの各項目について、以下に説明する。

表 3.2-5 トランザクション一覧表示画面の各項目

項目	説明
最左列の矢印アイコン	メソッド実行経路画面へのリンク
Start Time	トランザクション（入口メソッド）の開始時刻
End Time	トランザクション（入口メソッド）の終了時刻
Total Time	トランザクション全体（入口メソッドの開始から終了まで）の経過時間（単位：ミリ秒）
Number of Methods	トランザクションに含まれるメソッドの数（入口メソッド自身を含む）
CPU	トランザクション実行に要した CPU 時間

なお、この画面では、以下の操作が可能である。

- 1 画面（ページ）に表示されるトランザクションは最大 20 トランザクションである。それ以上のトランザクションが存在する場合は、複数ページに渡って表示される。他の画面へは、画面最下部にある "First Page"（最初のページ）、"Previous"（前のページ）、"Next"（次のページ）、"Last Page"（最後のページ）をクリックすることにより移動可能である。
- "Start Time"、"End Time"、"Total Time"、"Number of Methods"、"CPU"をキーにしたソートが可能である。いずれかをクリックする度に、昇順/降順を切り替えた表示を行う。
- 入口メソッドを再指定する場合は、"jboss-tracer"メニューの"transaction tracer"リンクをクリックすることで、3.2.5 の入口メソッド指定画面に戻ることができる。

トランザクションを選択し、最左列の矢印アイコンをクリックすると、3.2.7 に示すメソッド実行経路表示画面が別画面で表示される。

3.2.7 メソッド実行経路の表示

画面には、3.2.6.2 で選択したトランザクション内部で実行されたメソッドの経路が表示される。トランザクション内部で例外が発生した場合は、発生経路も表示される。

Methods Order	Start Time	End Time	Total Time	CPU Time
org.spec.jappserver.servlet.SpecAppServlet.doGet	08/22/2005 14:00:57.268	08/22/2005 14:00:57.303	35	34847
org.spec.jappserver.servlet.SpecAppServlet.performTask	08/22/2005 14:00:57.268	08/22/2005 14:00:57.303	35	34813
org.spec.jappserver.servlet.helper.SpecServletAction.doVehicleQuotes	08/22/2005 14:00:57.270	08/22/2005 14:00:57.303	33	32933
org.spec.jappserver.servlet.helper.SpecAction.getVehicleQuotes	08/22/2005 14:00:57.270	08/22/2005 14:00:57.288	18	18123
org.spec.jappserver.servlet.helper.VehicleSearch.	08/22/2005 14:00:57.273	08/22/2005 14:00:57.273	0	31
org.spec.jappserver.orders.itemBrowserSes.ejb.ItemBrowserSesEJB.getItems	08/22/2005 14:00:57.275	08/22/2005 14:00:57.279	4	3310
org.spec.jappserver.orders.itement.ejb.ItemCmp20EJBSProxy.toString	08/22/2005 14:00:57.277	08/22/2005 14:00:57.277	0	93
org.spec.jappserver.orders.itement.ejb.ItemCmp20EJB.getDataBean	08/22/2005 14:00:57.277	08/22/2005 14:00:57.278	1	679
org.spec.jappserver.orders.helper.ItemsDataBean.	08/22/2005 14:00:57.277	08/22/2005 14:00:57.277	0	14
org.spec.jappserver.orders.itement.ejb.ItemCmp20EJBSProxy.getDescription	08/22/2005 14:00:57.277	08/22/2005 14:00:57.277	0	93
org.spec.jappserver.orders.helper.ItemsDataBean.setDescription	08/22/2005 14:00:57.277	08/22/2005 14:00:57.277	0	10
org.spec.jappserver.orders.itement.ejb.ItemCmp20EJBSProxy.getDiscount	08/22/2005 14:00:57.277	08/22/2005 14:00:57.277	0	70
org.spec.jappserver.orders.helper.ItemsDataBean.setDiscount	08/22/2005 14:00:57.277	08/22/2005 14:00:57.277	0	10
org.spec.jappserver.orders.itement.ejb.ItemCmp20EJBSProxy.getName	08/22/2005 14:00:57.277	08/22/2005 14:00:57.277	0	65
org.spec.jappserver.orders.helper.ItemsDataBean.setName	08/22/2005 14:00:57.277	08/22/2005 14:00:57.277	0	8
org.spec.jappserver.orders.itement.ejb.ItemCmp20EJBSProxy.getPrice	08/22/2005 14:00:57.277	08/22/2005 14:00:57.278	1	81
org.spec.jappserver.orders.helper.ItemsDataBean.setPrice	08/22/2005 14:00:57.278	08/22/2005 14:00:57.278	0	7
org.spec.jappserver.orders.itement.ejb.ItemCmp20EJBSProxy.getId	08/22/2005 14:00:57.278	08/22/2005 14:00:57.278	0	67
org.spec.jappserver.orders.helper.ItemsDataBean.setId	08/22/2005 14:00:57.278	08/22/2005 14:00:57.278	0	7
org.spec.jappserver.orders.itement.ejb.ItemCmp20EJBSProxy.getCategory	08/22/2005 14:00:57.278	08/22/2005 14:00:57.278	0	79
org.spec.jappserver.orders.helper.ItemsDataBean.setCategory	08/22/2005 14:00:57.278	08/22/2005 14:00:57.278	0	7
org.spec.jappserver.orders.itemBrowserSes.ejb.ItemBrowserSesEJB.getTotalItems	08/22/2005 14:00:57.287	08/22/2005 14:00:57.287	0	41
org.spec.jappserver.servlet.helper.SpecConfig.getPage	08/22/2005 14:00:57.288	08/22/2005 14:00:57.288	0	11
org.spec.jappserver.servlet.helper.SpecServletAction.requestDispatch	08/22/2005 14:00:57.288	08/22/2005 14:00:57.303	15	14478
org.spec.jappserver.servlet.helper.SpecUtils.formatCurrency	08/22/2005 14:00:57.296	08/22/2005 14:00:57.297	1	709
org.spec.jappserver.servlet.helper.SpecUtils.formatCurrency	08/22/2005 14:00:57.298	08/22/2005 14:00:57.298	0	623
org.spec.jappserver.servlet.helper.SpecUtils.formatNumber	08/22/2005 14:00:57.298	08/22/2005 14:00:57.299	1	610

図 3.2-7 メソッド実行経路表示画面

表示される各項目について、以下に説明する。

表 3.2-6 メソッド実行経路表示画面の各項目

項目	説明
Methods Order	メソッドを時系列表示する。インデントの異なるメソッドが前後する場合、それらのメソッドが入れ子関係にあることを示す。 3.2.6.1 で特定メソッドを含むトランザクションの検索を指定した場合、該当メソッドは、それ以外のメソッドとは異なる色で表示される。
Start Time	各メソッドの開始時刻。

項目	説明
End Time	<p>各メソッドの終了時刻。</p> <p>3.2.6.1 の”transaction type”で”both”または”incomplete”を指定した場合、終了しなかったメソッドについては、”NOT EXIT”と表示される。</p>
Total Time	<p>各メソッドの開始から終了までの経過時間。(単位:ミリ秒)</p> <p>3.2.6.1 の”transaction type”で”both”または”incomplete”を指定した場合、終了しなかったメソッドについては、”UNKNOWN”と表示される。</p> <p>3.2.6.1 で任意メソッドの経過時間による検索を指定した場合、閾値を超えるメソッドの経過時間は、異なる色で表示される。</p>
CPU Time	<p>各メソッドの実行に要した CPU 時間。</p>