

---

最終更新日 : 2012/03/23

# TLVの標準形式変換の外部プロセス化機能 マニュアル

名古屋大学

# 目次

---

## ❖概要

## ❖TLV内部の標準形式変換

## ❖標準形式トレースログの直接入力

### ◆概要

### ◆使い方

### ◆利用の実例

## ❖スクリプト拡張機能を用いた変換

### ◆概要

### ◆動作環境の構築

### ◆使い方

### ◆利用の実例

## ❖変換スクリプトの作成時補足資料

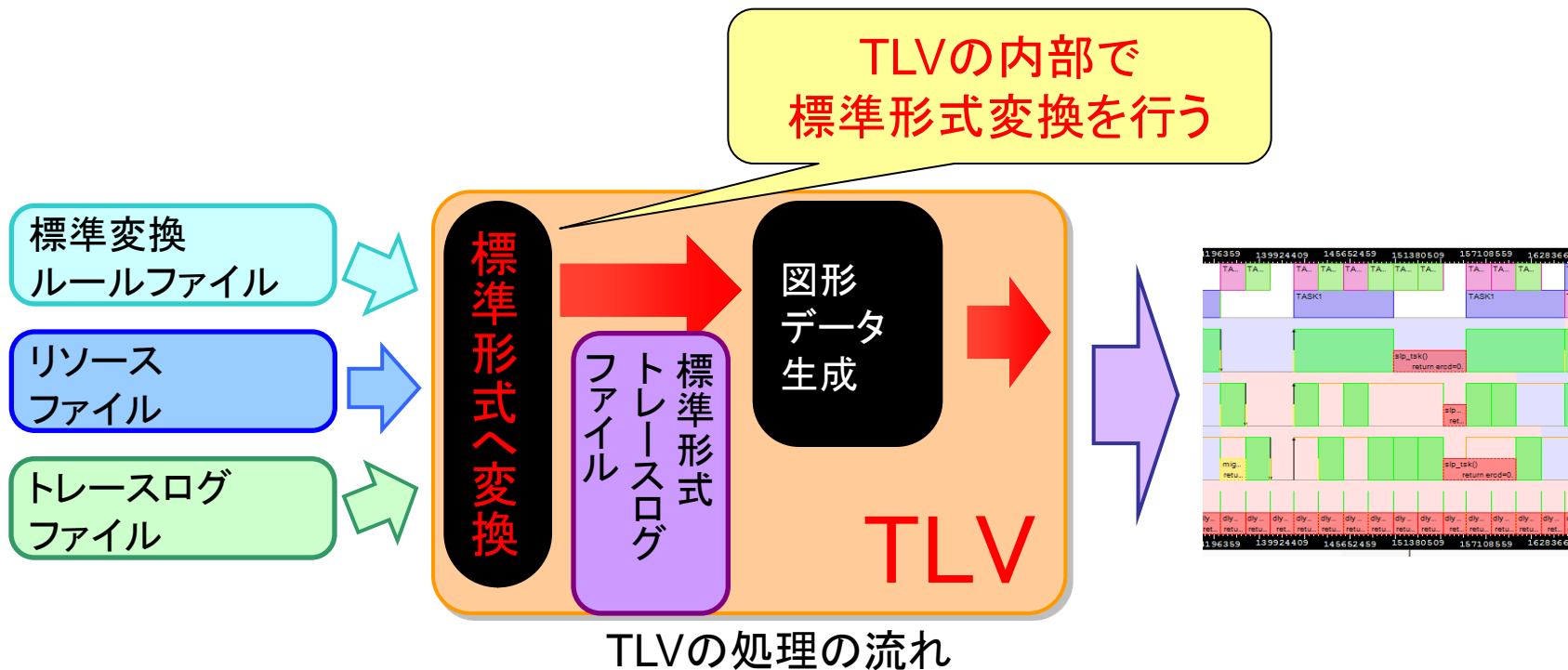
# 概要: TLV標準形式変換の外部プロセス化

---

- ❖ TLV内部で行っているトレースログの標準形式への変換処理を、TLV外部で行うことを可能とする機能です.
- ❖ 以下の2つの機能があります.
  - ◆ 標準形式トレースログファイルの直接入力機能
  - ◆ スクリプト拡張機能を用いた外部変換
- ❖ この機能を用いる利点は以下の通りです.
  - ◆ 変換処理の高速化を実現
    - 標準変換を外部で高速に行うことで、TLV内部の実行時間を短くすることが可能

# TLV内部の標準形式変換

- ❖ 標準形式変換: OSやシミュレータから出力される各種トレースログをある1つの標準形式のトレースログへと変換する機能
  - ◆ 今まではTLVの内部で各種トレースログを標準形式へと変換

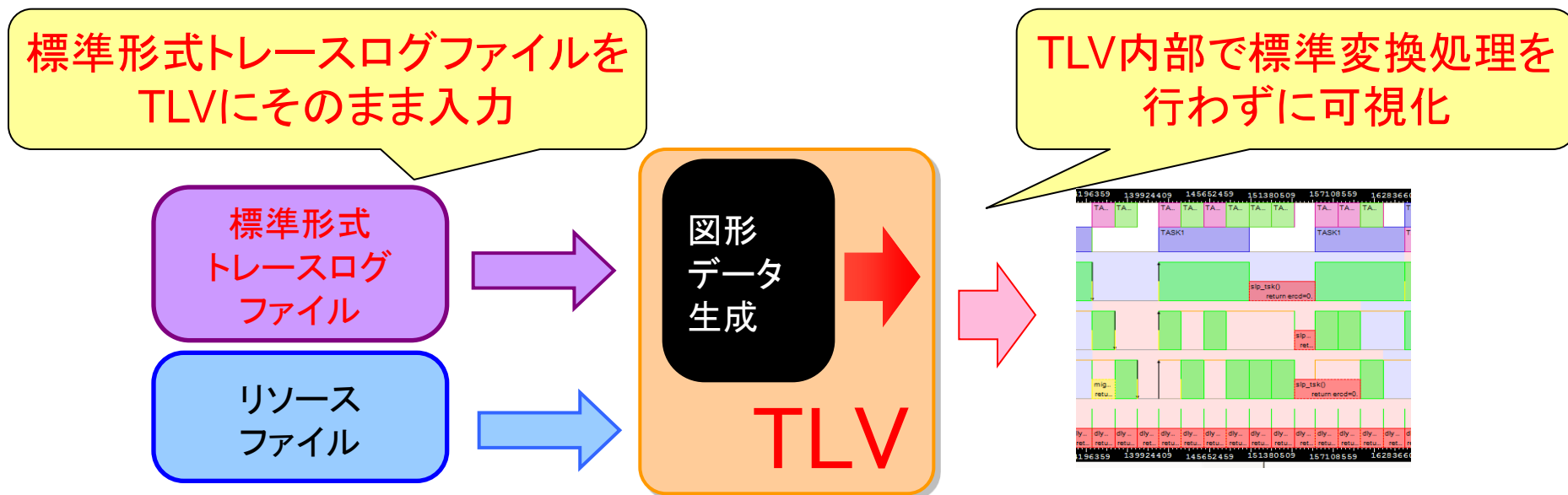


---

# 標準形式トレースログファイルの直接入力機能

# 概要

- ❖ 標準形式トレースログを直接読み込むことができる機能
  - ◆ 標準形式変換が不要(TLVの処理の高速化が可能)
  - ◆ 事前に標準形式トレースログファイルを作成する必要がある



# 使用方法

---

## 1. リソースファイルの設定

- ◆ リソースファイルにおける変換ルールファイルの指定を空白("")とする

## 2. ファイルの読み込み

- ◆ 標準形式トレースログファイルと  
1.で設定したリソースファイルを読み込む

```
{  
    "TimeScale" : "us",  
    "TimeRadix" : 10,  
    "ConvertRules" : [""],  
    "VisualizeRules" : ["toppers", "asp"],  
    "ResourceHeaders" : ["asp"],  
    "Resources":
```

リソースファイルの設定例

# サンプルファイルの可視化

---

❖ サンプルファイル `asp_short_standard_format.res` ,  
`asp_short_standard_format.log` を変換する実例を紹介します

❖ 使用するファイル

◆ リソースファイル

- `sampleFiles/SampleLog/asp/ asp_short_standard_format.res`

◆ トレースログファイル(標準形式)

- `sampleFiles/SampleLog/asp/a asp_short_standard_format.log`



# 利用の実例1. リソースファイルの設定

❖リソースファイルの変換ルールファイルの指定を空白("")とします(下図の赤字部分)

- ◆これにより, TLVが入力ファイルを標準形式トレースログファイルとして処理するようになります. |

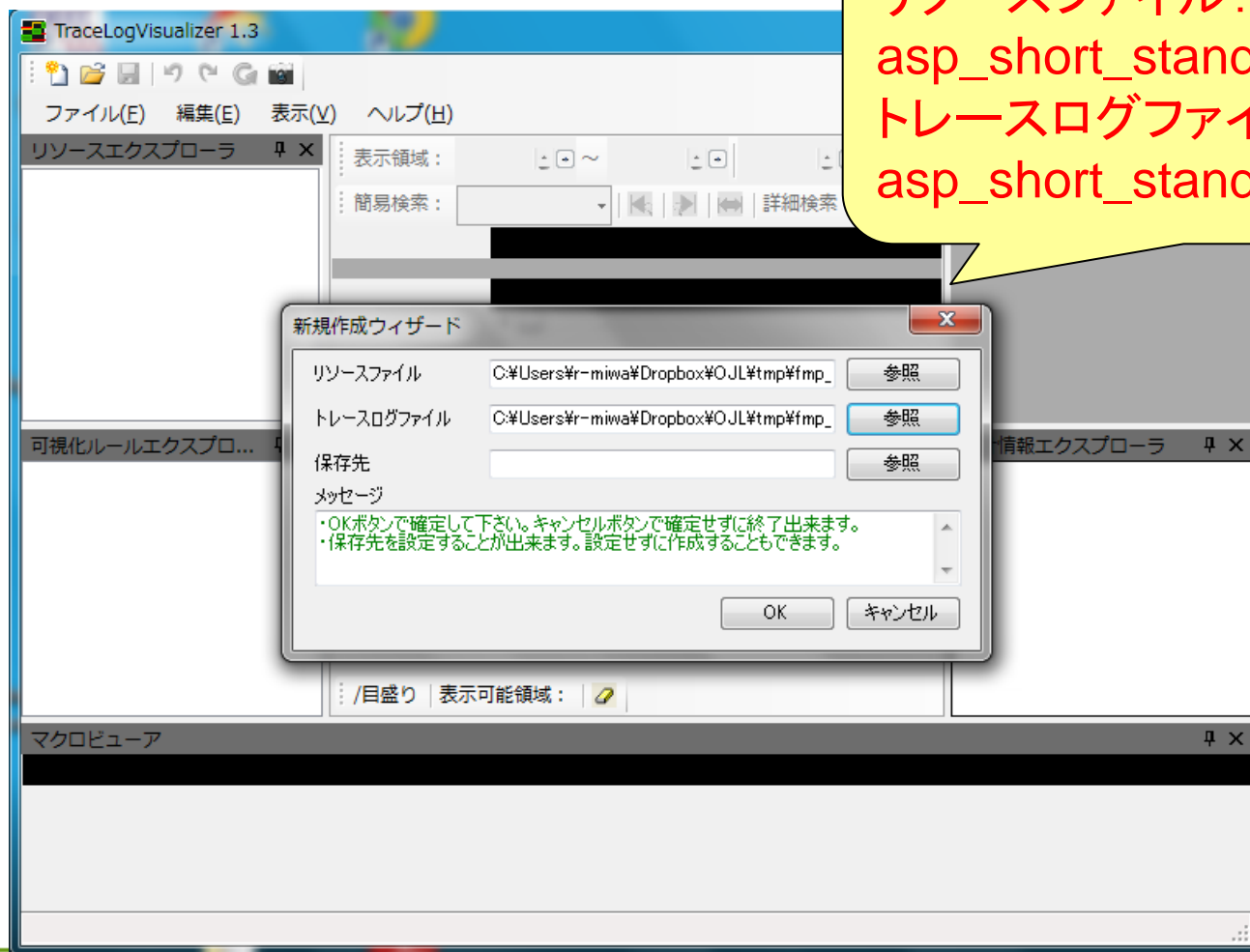
```
{  
    "TimeScale" : "us",  
    "TimeRadix" : 10,  
    "ConvertRules" : [""],  
    "VisualizeRules" : ["toppers", "fmp", "fmp_core2"],  
    "ResourceHeaders" : ["fmp"],  
    "Resources":  
    {
```

空白("")となっている

リソースファイル  
asp\_short\_standard\_format.res

## 利用の実例2. ファイルの読み込み

❖標準形式トレースログと2.で作成したリソースファイルを読み込みます.



リソースファイル:  
asp\_short\_standard\_format.res  
トレースログファイル:  
asp\_short\_standard\_format.log

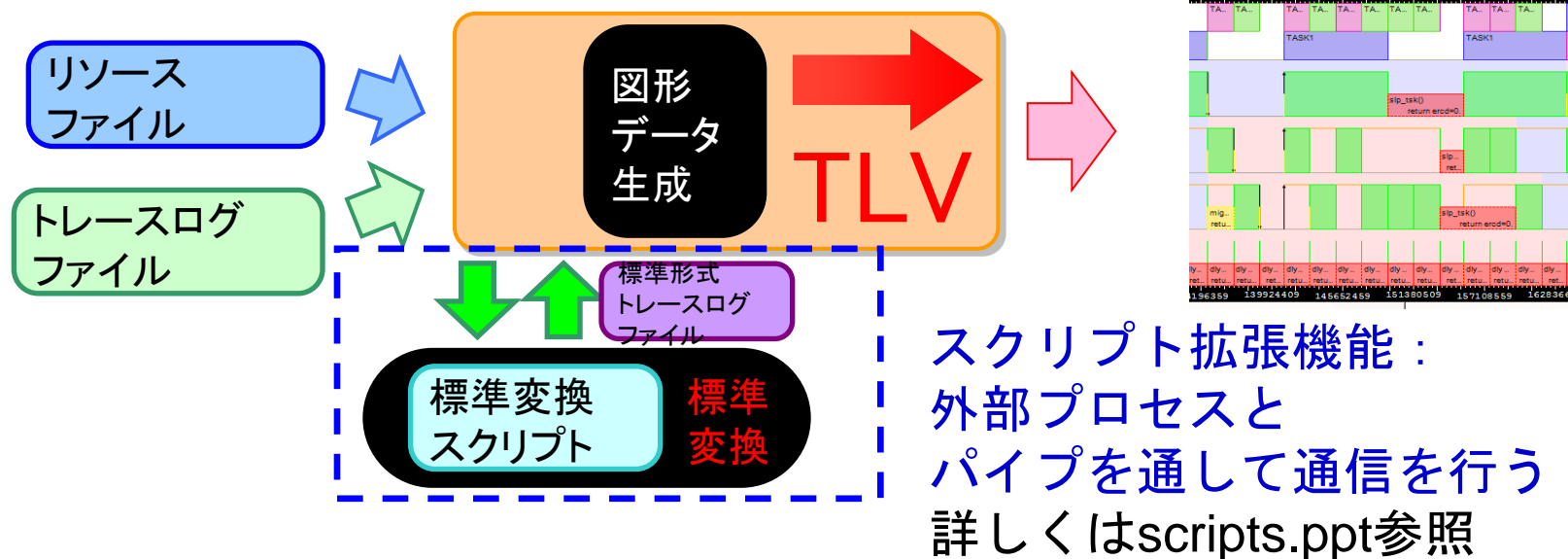
---

# スクリプト拡張機能を用いた外部変換

# 概要: スクリプト拡張機能を用いた外部変換

❖ スクリプト拡張機能を用いて外部変換を行うことで、変換速度の向上を実現

- ◆ TLVから外部のプログラムを呼び出して、外部で変換処理を行うことができる.
- ◆ TLVパッケージ付属のサンプル
  - TLV内部の変換を行わず、外部でRubyスクリプト等を用いて変換
    - 現在, aspとfmp用のRubyスクリプトを用意



# 機能2の利用方法

## ❖手順

- ◆1: 変換ルールとリソースファイルの設定
  - スクリプト拡張機能を用いるようにリソースファイルと変換ルールファイルの設定を行う.
- ◆2: ファイルの読み込み
  - トレースログファイルと1. で作成したリソースファイルを読み込む

```
{  
    "TimeScale" : "us",  
    "TimeRadix" : 10,  
    "ConvertRules" : ["asp_script"],  
    "VisualizeRules" : ["toppers", "asp"],  
    "ResourceHeaders" : ["asp"],  
    "Resources":
```

リソースファイルの設定例

# TLVパッケージ付属のサンプル

## 利用のための動作環境の構築

### 1. Rubyの準備

- ◆ 下記のページから最新版のRubyをインストール
    - [http://rubyforge.org/frs/?group\\_id=167&release\\_id=28426](http://rubyforge.org/frs/?group_id=167&release_id=28426)
- ※Rubygemsを利用します

### 2. Jsonライブラリのインストール

- ◆ 下記URLからWin用のJSONライブラリをgemファイルのあるディレクトリにダウンロード
  - [http://rubyforge.org/frs/?group\\_id=953&release\\_id=38578](http://rubyforge.org/frs/?group_id=953&release_id=38578)
- ◆ gemファイルのあるディレクトリにて以下のコマンドを実行
  - `gem install json -local`

### 3. 環境変数の設定

- ◆ 環境変数に以下を追加

変数名	値
GEM_HOME	gemsのインストール先
PATH	%GEM_HOME%/bin
RUBYLIB	%GEM_HOME%/lib

# 利用の実例

---

❖ サンプルファイルasp\_short.res, asp\_short.logをスクリプト拡張機能を用いて変換する実例を紹介します

❖ 使用するファイル

- ◆ 機能2を用いるように変更したリソースファイル
  - sampleFiles/SampleLog/asp/asp\_short\_script.res
- ◆ トレースログファイル
  - sampleFiles/SampleLog/asp/asp\_short.log
- ◆ 変換スクリプト
  - convertRules/asp\_converter2.rb
  - ※FMPの場合はfmp\_converter2.rbを使用します
- ◆ 変換ルールファイル
  - convertRules/asp\_script.cnv

# 利用の実例1. 変換ルールとリソースファイルの設定

## ❖ 変換ルールの設定

- ◆ 外部の変換プログラム(Rubyプログラム)を使用するよう  
変換ルールを設定

asp\_script.cnv

```
“asp_script”: {  
    ※変換ルールの名前  
    “$STYLE”: “script”, ※外部スクリプトを利用するため”script”と記述  
    “fileName”: “C:/Ruby187/bin/ruby.exe”, ※スクリプトを実行する処理系を記述  
                                     (今回のケースではRubyを使用)  
    “arguments”: “convertRules/asp_converter2.rb” ※変換スクリプトの場所を指定
```

## ❖ リソースファイルの設定

- ◆ ↑で作成した  
変換ルールファイル  
を用いるように設定

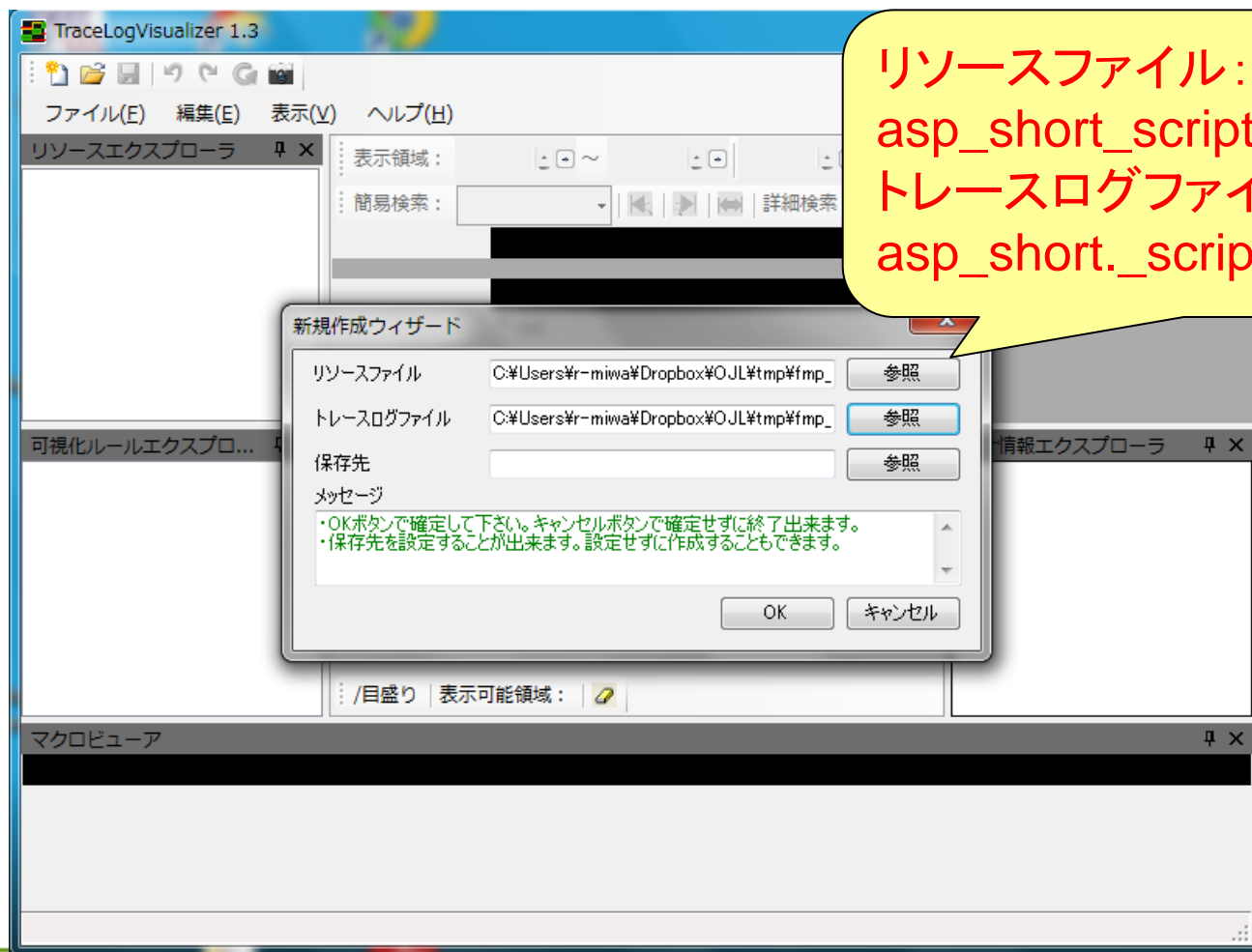
パスの指定は、¥ではなく/で行うこと

```
"TimeScale" : "us",  
"TimeRadix" : 10, ↓※作成した変換ルールを指定  
"ConvertRules" : ["asp_script"],  
"VisualizeRules" : ["toppers", "asp"],  
"ResourceHeaders" : ["asp"],  
"Resources":  
...
```



## 利用の実例2. ファイルの読み込み

❖トレースログと1. で設定したリソースファイルを読み込みます.



リソースファイル:  
asp\_short\_script.res  
トレースログファイル:  
asp\_short.\_script.log

---

# 変換スクリプトの作成時補足資料

# 標準形式トレースログファイルとは

❖ OSが出力した様々のトレースログを, TLVが可視化のために, 認識できるような形式に変換したトレースログファイル

- ◆ 標準変換
- ◆ 詳しくはTLV\_convert\_rules.ppt参照

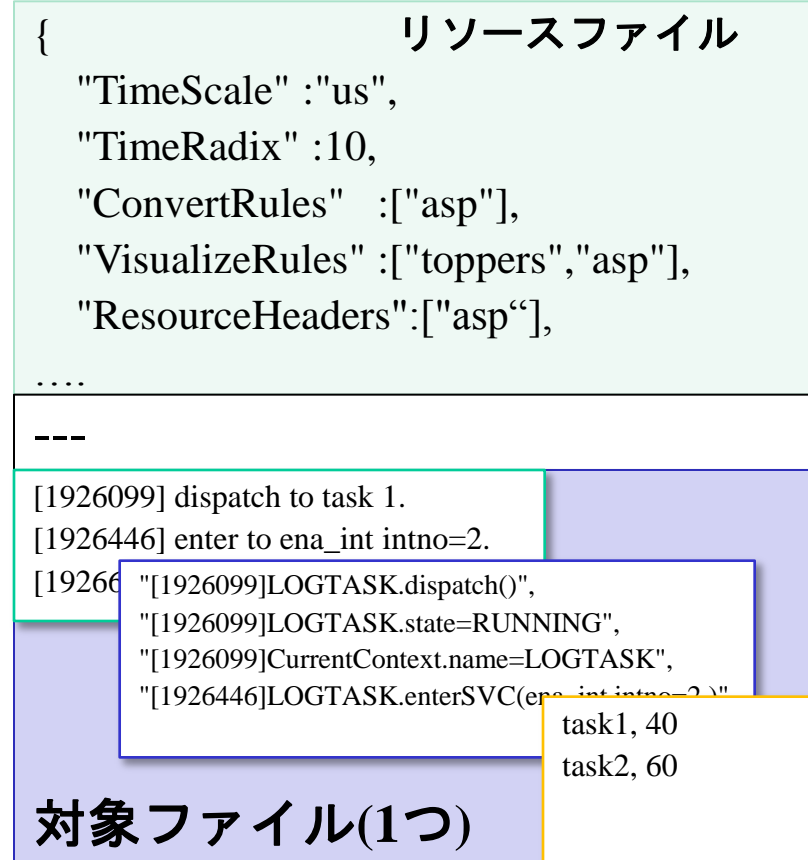
```
1 [2000] dispatch to task 1.
2 [2000] enter to ena_int intno=2.
3 [2000] leave to ena_int ercd=0.
4 [2000] enter to dly_tsk dlytim=10.
5 [2000] task 1 becomes WAITING.
6 [2000] dispatch to task 5.
7 [2000] enter to act_tsk tskid=2.
8 [2000] task 2 becomes RUNNABLE.
9 [2000] leave to act_tsk ercd=0.
10 [2000] enter to act_tsk tskid=3.
11 [2000] task 3 becomes RUNNABLE.
12 [2000] leave to act_tsk ercd=0.
13 [2000] enter to act_tsk tskid=4.
14 [2000] task 4 becomes RUNNABLE.
15 [2000] leave to act_tsk ercd=0.
16 [2000] enter to dly_tsk dlytim=40.
```

## 標準形式変換

```
2 ]LOGTASK.dispatch()",
3 ]LOGTASK.state=RUNNING",
4 ]LOGTASK.enterSVC(ena_int,intno=2)",
5 ]LOGTASK.leaveSVC(ena_int,ercd=0)",
6 ]LOGTASK.enterSVC(dly_tsk,dlytim=10)",
7 ]LOGTASK.wait()",
8 ]LOGTASK.state=WAITING",
9 ]MAIN_TASK.dispatch()",
10 ]MAIN_TASK.state=RUNNING",
11 ]MAIN_TASK.enterSVC(act_tsk,tskid=2)",
12 ]TASK1.activate()",
13 ]TASK1.state=RUNNABLE",
14 ]MAIN_TASK.leaveSVC(act_tsk,ercd=0)",
15 ]MAIN_TASK.enterSVC(act_tsk,tskid=3)",
16 ]TASK2.activate()",
```

# スクリプト拡張機能の仕様

1. TLVが、統計情報ファイル  
**生成用**外部スクリプトを起動する
2. 外部スクリプトの標準入力に  
リソースファイルが書き込まれる
3. 外部スクリプトの標準入力に  
---が書き込まれる
4. 外部**スクリプト**の標準入力に  
対象ファイルが書き込まれる
5. 外部スクリプトが標準出力に  
書きだしたトレースログを  
TLVが読み込む



外部スクリプトに  
入力される情報のイメージ図