最終更新日 ： 2012/02/22

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
- ❖従来のTLV標準形式変換
- ❖機能１：標準形式トレースログの直接入力機能
  - ◆概要
  - ◆利用の流れ
  - ◆利用の実例
- ❖機能２：スクリプト拡張機能を用いた変換
  - ◆概要
  - ◆動作環境
  - ◆利用の流れ
  - ◆利用の実例

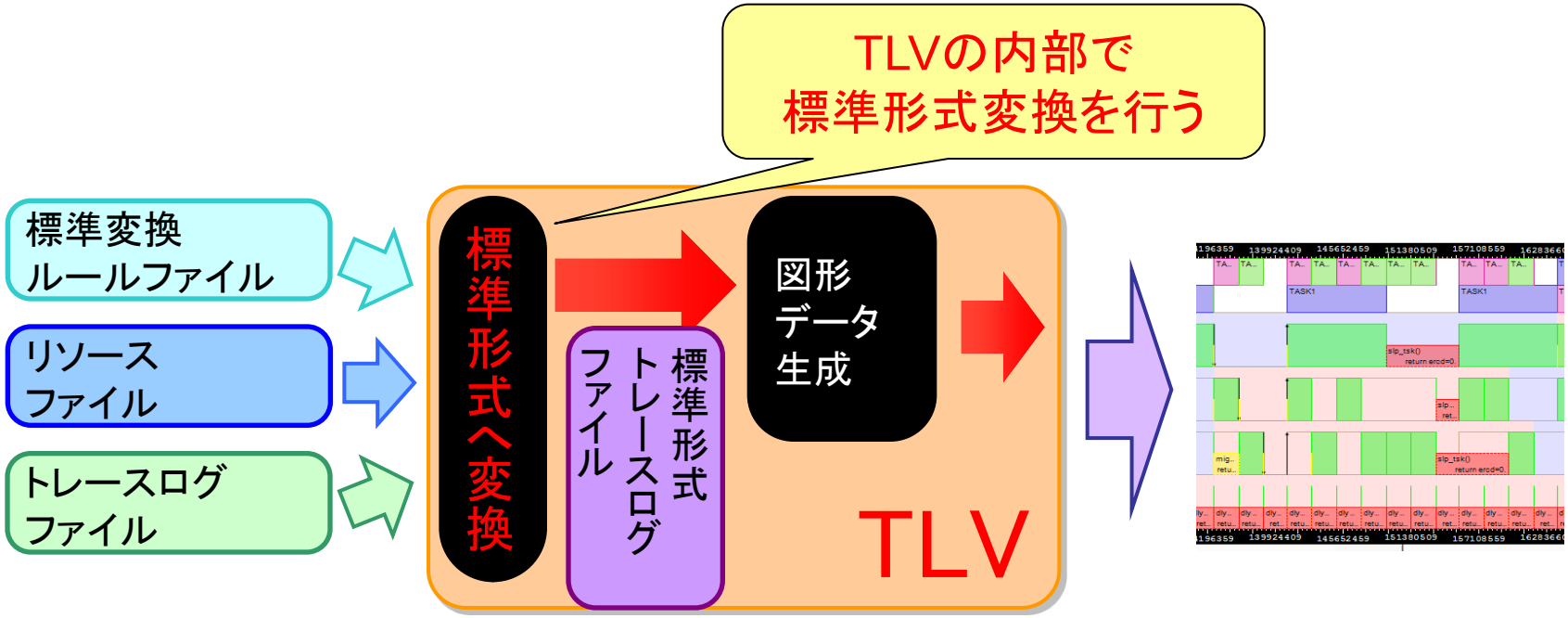
概要：TLV標準形式変換の外部プロセス化

- 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．
  - 機能1：標準形式トレースログファイルの直接入力機能
  - 機能2：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．
  - 直接標準形式のトレースログを入力することが可能
  - 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

従来のTLV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

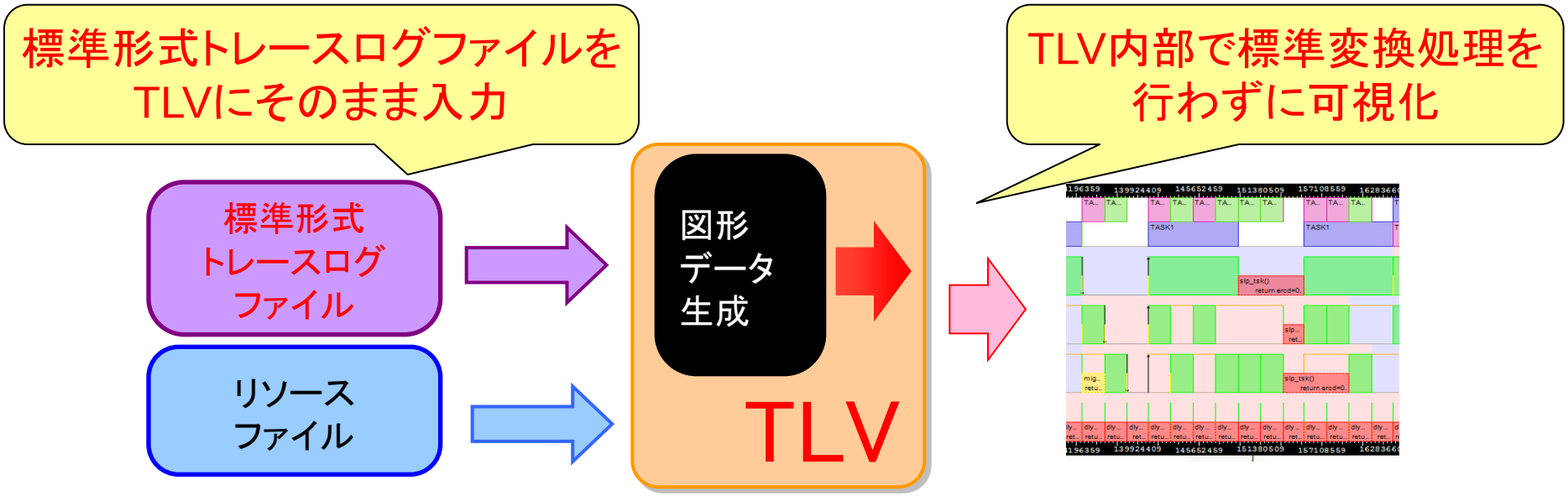


TLVの処理の流れ

機能1：
標準形式トレースログファイルの直接入力機能

機能1の概要

❖TLVに直接標準形式トレースログを読み込むことができる機能
- ◆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



標準形式トレースログファイル

- 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

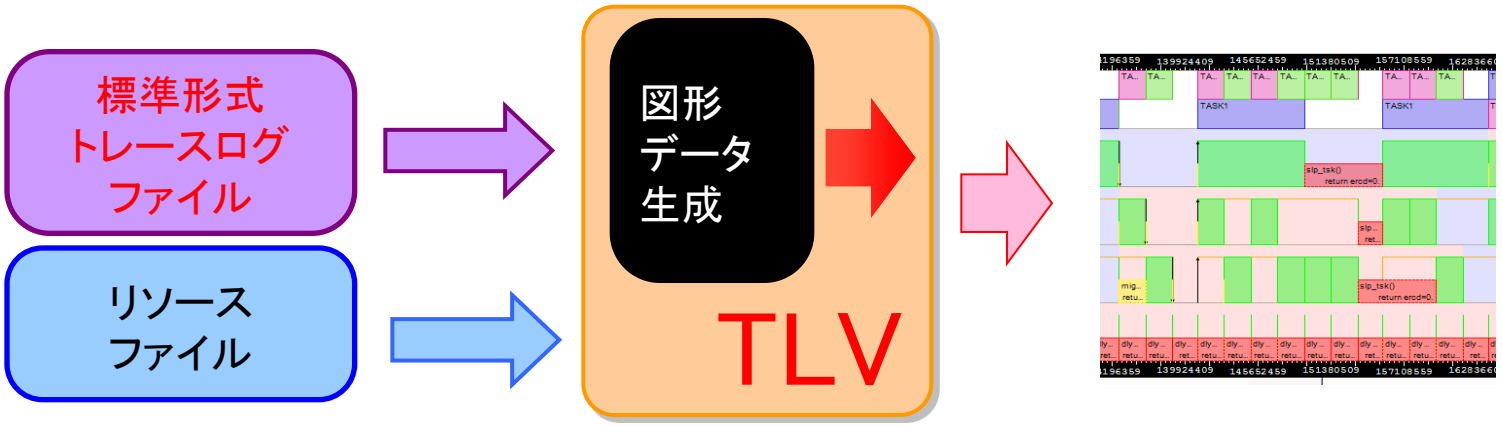
```
 1 [2000] dispatch to task 1.
 2 [2000] enter to ena_int intno=2.
 3 [2000] leave to ena_int ercd=0.
 4 [2000] enter to dly_tsk dlytim=10.
 5 [2000] task 1 becomes WAITING.
 6 [2000] dispatch to task 5.
 7 [2000] enter to act_tsk tskid=2.
 8 [2000] task 2 becomes RUNNABLE.
 9 [2000] leave to act_tsk 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()",
```

利用の流れ

1. 標準形式トレースログを作成
   - シミュレータから出力
   - トレースログを標準形式に変換
2. リソースファイルの設定
   - 読み込むファイルが標準形式トレースログであることを設定
3. ファイルの読み込み
   1. 1で作成した標準形式トレースログファイルと2で設定したリソースファイルを読み込む



利用の実例

❖サンプルファイルasp_short_standard_format.res, asp_short_standard_format.logを標準形式で読み込ませる方法を例に説明します.

❖登場するファイル

- sampleFiles/SampleLog/asp/asp_short.log
- sampleFiles/SampleLog/asp/asp_short_standard_format.res
- sampleFiles/SampleLog/asp/asp_short_standard_format.log

利用の実例1．標準形式トレースログを作成

❖標準形式のトレースログを作成します．

- ◆今回のケースでは，asp_short.logを標準形式に変換した asp_short_standard_format.loを作成しました．
- ◆標準形式については，doc/TLV_convert_rulus.pptを参照してください．

```
[954133]LOGTASK1.dispatch()
[954133]LOGTASK1.state=RUNNING
[954133]CurrentContext_PRC1.name=LOGTASK1
[954639]LOGTASK1.enterSVC(ena_int,intno=65538.)
[954863]LOGTASK1.leaveSVC(ena_int,ercd=0.)
[954123]LOGTASK2.dispatch()
[954123]LOGTASK2.state=RUNNING
…
```

標準形式トレースログファイル asp_short_standard_format.log

利用の実例2. 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. I

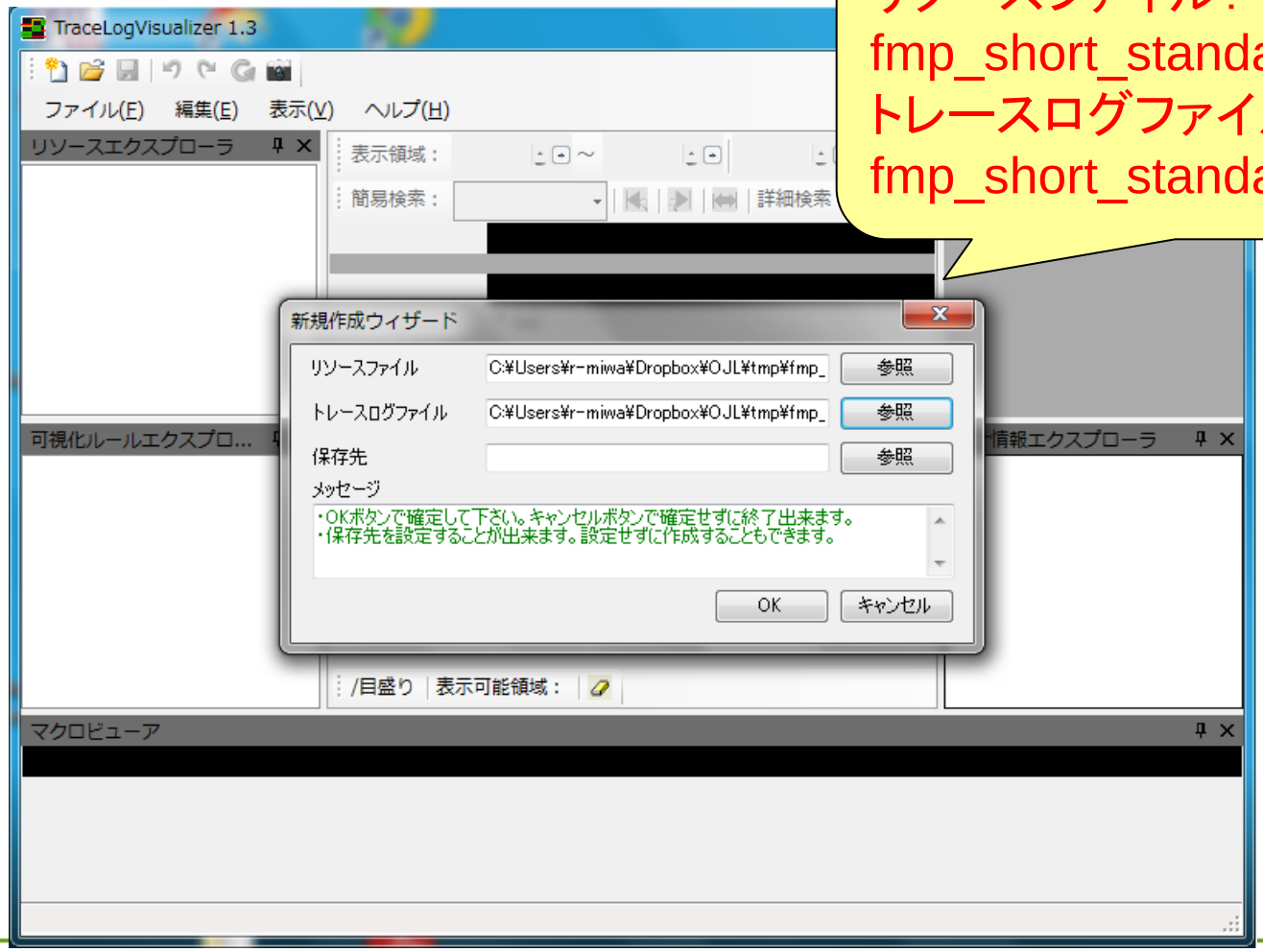
```
{
        "TimeScale" :"us",
        "TimeRadix" :10,
        "ConvertRules"   :[""],
        "VisualizeRules" :["toppers","fmp","fmp_core2"],
        "ResourceHeaders":["fmp"],
        "Resources":
        {
```

空白(“”)とする

リソースファイル
asp_short_standard_format.res

利用の実例3．ファイルの読み込み

❖1．，2．で作成した標準形式トレースログ，リソースファイルを読み込みます．

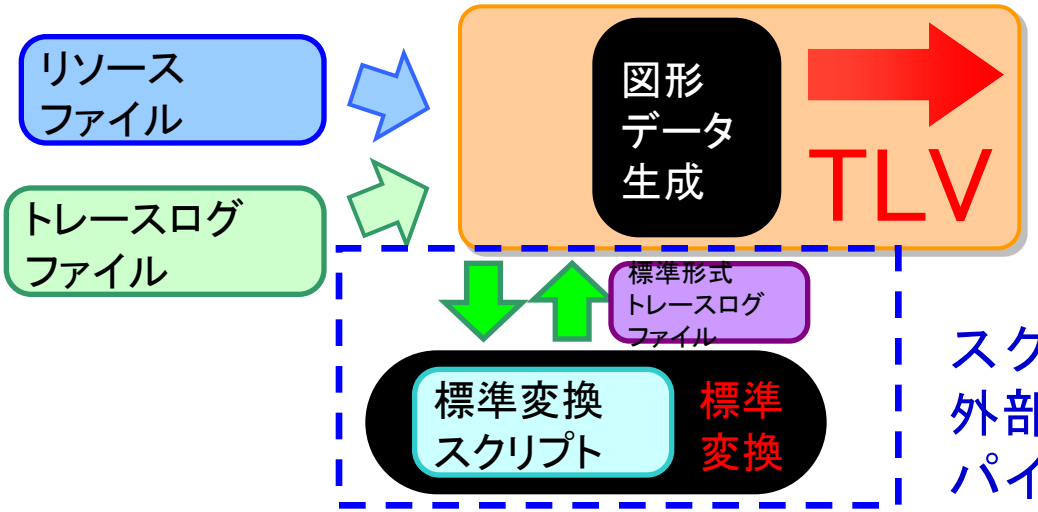


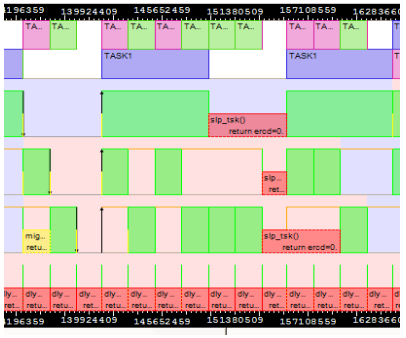
リソースファイル：
fmp_short_standard_format.res
トレースログファイル：
fmp_short_standard_format.log

機能２：
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の実行途中でパイプラインを用いて外部で変換処理を行うことができる.
  - 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





スクリプト拡張機能：
外部プロセスと
パイプを通して通信を行う
詳しくはscripts.ppt参照

機能2の利用方法

❖動作環境

- Rubygemsをインストールされている必要があります.

❖手順

- 1：変換スクリプトの作成
  - スクリプト拡張機能の仕様に合わせて，変換スクリプトを作成する
- 2：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.
- 3：ファイルの読み込み
  - トレースログファイルと2．で作成したリソースファイルを読み込む

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

❖登場するファイル

- sampleFiles/SampleLog/asp/asp_short.res：リソースファイル
- sampleFiles/SampleLog/asp/asp_short_script.res：
  今回の機能を用いるように変更したリソースファイル
- sampleFiles/SampleLog/asp/asp_short.log：トレースログファイル
- convertRules/asp_converter.rb：変換スクリプト
- convertRules/asp_script.cnv：変換ルールファイル

利用の実例1．変換スクリプトの作成

- スクリプト拡張機能の仕様に合わせて変換スクリプトを作成します．
  - スクリプト拡張機能の仕様（次項）に合わせてトレースログを標準形式へと変換するスクリプトを作成します．
    - 今回のケースではRubyを用いて作成

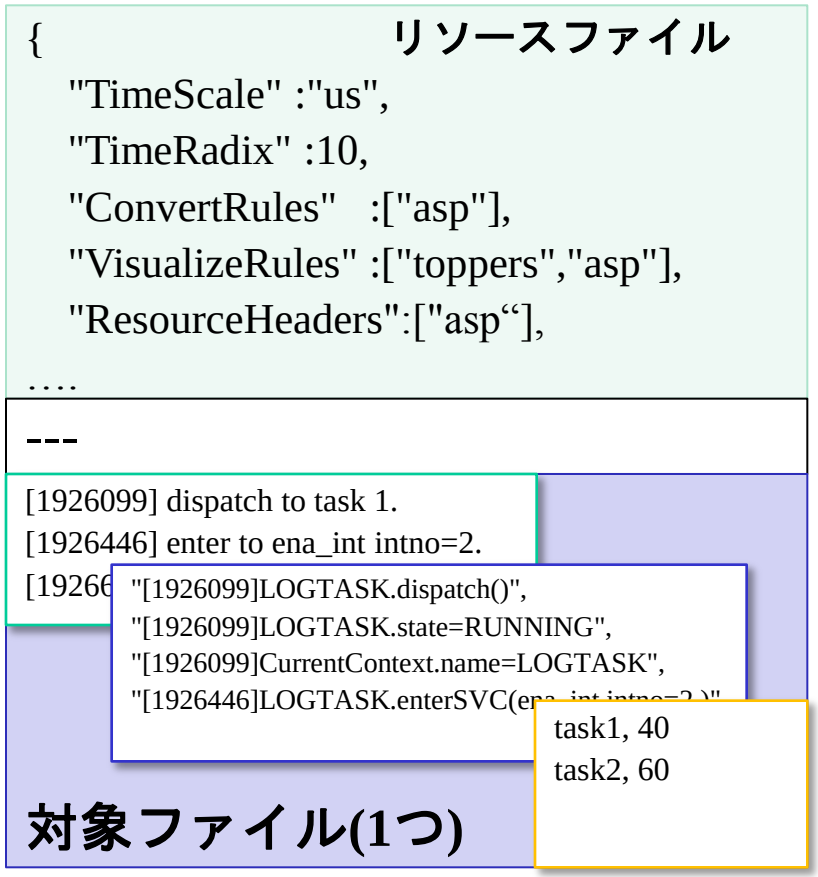
```
if FmpResource.ExistRunning?(prcid,"Task")
    print "[",time,"]",$Context[prcid][-1],".preempt()¥r¥n"
    print "[",time,"]",$Context[prcid][-1],".state=RUNNABLE¥r¥n"
    FmpResource.ChangeAtr(prcid,$Context[prcid][-1],"Task","RUNNABLE")
  end
  print "[",time,"]",FmpResource.ResName(prcid,$1,"Task"),".dispatch()¥r¥n"
  print "[",time,"]",FmpResource.ResName(prcid,$1,"Task"),".state=RUNNING¥r¥n"
  print "[",time,"]CurrentContext_PRC",prcid,".name=",FmpResource.ResName(prcid,$1,"Task"),"¥r¥n"
  $Context[prcid].push FmpResource.ResName(prcid,$1,"Task")

  FmpResource.ChangeAtr(prcid,FmpResource.ResName(prcid,$1,"Task"),"Task","RUNNING")
```

Rubyによる変換スクリプトの例:fmp_converter.rb
※asp用のファイル(asp_converter.rb)も存在します．

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

利用の実例2．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

- 変換スクリプトを使用するよう
  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:/cygwin/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_converter.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

```
"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3．ファイルの読み込み

❖1．，2．で作成した標準形式トレースログ，リソースファイルを読み込みます．