TLV外部スクリプトマニュアル

TLV開発チーム

平成21年9月26日

目 次

第 1 章　概要 **1**
1.1　外部スクリプトの概要 1
1.2　外部スクリプトの利点 1
1.3　その他に参照するべきマニュアル 1
1.4　用語の定義/略語の説明 1

第 2 章　変換ルール **3**
2.1　変換ルール用外部スクリプトの仕様 3
2.2　*.cnv ファイルの記述方法 3

第 3 章　可視化ルール **5**
3.1　可視化ルール用外部スクリプトの書き方 5
3.2　*.viz ファイルの記述方法 6
3.3　例: CPU 利用率可視化表示 7

第1章　概要

1.1　外部スクリプトの概要

Ruby や Perl などの任意の言語で変換ルール・可視化ルールを記述できる機能です。

任意の言語で記述した変換ルールや可視化ルールを**外部プロセス**と呼びます。

外部プロセスとの通信はパイプを通して行なわれる。

1.2　外部スクリプトの利点

以下の利点があります。

- 自由度が非常に高い
- チューニングにより高速化可能

1.3　その他に参照するべきマニュアル

『TLV 変換ルール・可視化ルールマニュアル』も参照してください。

1.4　用語の定義/略語の説明

表 1.1: 用語定義

用語・略語	定義・説明
TLV	Trace Log Visualizer
標準形式トレースログファイル	本ソフトウェアが扱うことの出来る形式をもつトレースログファイル。各種トレースログファイルは、この共通形式トレースログファイルに変換することにより本ソフトウェアで扱うことが出来るようになる。
*.cnv ファイル 変換ルール	変換ルールを記述するファイル トレースログファイルを標準形式トレースログファイルに変換する際に用いられるルール。
可視化ルール	標準形式トレースログファイルを可視化する際に用いられるルール。
*.viz ファイル	変換ルールを記述するファイル
外部スクリプト	任意の言語で記述された、変換や可視化を行なうためのスクリプト。
TLV ファイル	本ソフトウェアが中間形式として用いるファイル。前述の標準形式トレースログファイルは、この TLV ファイルの一部である。

第2章　変換ルール

2.1　変換ルール用外部スクリプトの仕様

変換ルール用外部スクリプトは、リソースファイルとトレースログを受け取り、標準形式トレースログを出力します。

変換時の処理の流れは以下のようになります。

1. TLV が、変換ルール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トレースログ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出した標準形式トレースログファイルを、TLV が読み込む。

リソースファイルの形式は、『TLV 変換ルール・可視化ルールマニュアル』を参照してください。

2.2　*.cnv ファイルの記述方法

変換に用いる外部スクリプトを指定するために、cnv ファイルは表 2.1 の要素が追加されています。

リスト 2.1 のように、`arguments` を用いて外部スクリプトを指定します。`arguments` は TLV.exe との相対パスも利用できます。

リスト 2.1: 外部スクリプトを指定する変換ルールの例

```
1 {
2   "asp2": {
3     "$STYLE": "script",
4     "fileName": "c:/cygwin/bin/ruby",
```

```
5       "arguments": "conv.rb",
6     }
7   }
```

あるいは、`script` を用いて、リスト 2.2 のように*.cnv ファイル内にスクリプトを直接記述することもできます。

リスト 2.2: 直接記述する変換ルールの例

```
1   {
2     "asp2": {
3       "$STYLE": "script",
4       "fileName": "c:/cygwin/bin/ruby",
5       "arguments": "{0}",
6       "script" : "puts '[1]TASK1.state=RUNNING'"
7     }
8   }
```

表 2.1: 追加された要素

要素	内容
$STYLE	旧ルールと区別するための要素。常に script と記述する
fileName	スクリプトを実行する処理系
arguments	実行時に渡される引数。`{0}`は一時ファイル名に置き換えられる。
script	一時ファイルの内容

第3章　可視化ルール

3.1　可視化ルール用外部スクリプトの書き方

可視化ルール用外部スクリプトは、リソースファイルと標準形式トレースログを受け取り、図形を出力します。

変換時の処理の流れは以下のようになります。

1. TLV が、可視化ルール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標準形式トレースログ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出した図形データを、TLV が読み込む。

図形データの形式は、リスト 3.1 のように基本図形を JSON の配列形式で並べたものになります。基本図形については、『TLV 変換ルール・可視化ルールマニュアル』の『図形の定義』を参照してください。

リスト 3.1: 図形データの例

```
1  [
2   {
3     "Type":"Rectangle",
4     "Size":"100,80",
5     "Pen":{"Color":"ff00ff00","Width":1},
6     "Fill":"6600ff00"
7   },
8   {
9     "Type":"Rectangle",
10    "Size":"100,80",
11    "Pen":{"Color":"ff00ff00","Width":1},
```

```
12      "Fill":"6600ff00"
13    }
14  ]
```

3.2 *.viz ファイルの記述方法

可視化に用いる外部スクリプトを指定するために、*.viz ファイルに表 3.1 の表 3.1 の要素が追加されている。

リスト 3.2 のように、`arguments` を用いて外部スクリプトを指定します。`arguments` は TLV.exe からの相対パスも利用できます。

`script` を用いると、リスト 3.3 のように*.viz ファイル内にスクリプトを直接記述することができる。

リスト 3.2: 外部ファイルを指定する可視化ルールの例

```
1  {
2    "asp2":{
3      "VisualizeRules":{
4        "taskStateChange":{
5          "Style": "script",
6          "DisplayName 状態遷移":"",
7          "Target":"Task",
8          "FileName": "c:/cygwin/bin/ruby",
9          "Arguments": "viz.rb",
10       }
11     }
12   }
13 }
```

あるいは、`script` を用いて、リスト 3.3 のようにルール内にスクリプトを直接記述することもできます。

リスト 3.3: 直接記述する可視化ルールの例

```
1  {
2    "asp2":{
3      "VisualizeRules":{
4        "taskStateChange":{
5          "Style": "script",
6          "DisplayName 状態遷移":"",
7          "Target":"Task",
8          "FileName": "c:/cygwin/bin/ruby",
9          "Arguments": "{0}",
10         "Script" : "puts '{ \"Type\":\"Rectangle\", ... }'"
```

```
11          }
12        }
13      }
14    }
```

表 3.1: 追加された要素

要素	内容
Style	旧ルールと区別するための要素。常に script と記述する
FileName	スクリプトを実行する処理系
Arguments	実行時に渡される引数。{0}は一時ファイル名に置き換えられる。
Script	一時ファイルの内容

3.3 例: CPU 利用率可視化表示

外部スクリプトを用いた可視化ルールは、visualizeRules/ディレトリに asp_cpu.viz、fmp_cpu.viz として同梱されています。

初期状態では無効化されています。ファイル中の"asp_"を"asp"に、"fmp_"から"fmp"に直すと、有効化されます。

動作には，Ruby と JSON ライブラリ (http://flori.github.com/json/) が必要です．開発には Cygwin 版の Ruby 1.8.7 を用いています．

可視化項目は，以下の通りです．

ASP CPU 使用率 (その時点で起動しているタスクの数)，平均使用率 (使用率の平均)，割り込み回数の積算

FMP CPU 使用率，平均使用率