

---

最終更新日 : 2012/02/22

# TLV 統計情報表示機能マニュアル

名古屋大学

# 目次 ([下線付き青文字](#)はクリックすると関連ページに飛びます)

## ❖ [概要](#)

## ❖ [ファイル構成](#)

- ◆ [統計情報生成ルールとは？](#)
- ◆ [統計情報ファイルとは？](#)

## ❖ [関連ウィンドウ](#)

- ◆ [統計情報エクスプローラ](#)
- ◆ [統計情報エクスプローラの表示/非表示](#)
- ◆ [統計情報ビューア](#)
- ◆ [統計情報ビューアの表示/非表示](#)

## ❖ [利用方法](#)

- ◆ [利用の流れ](#)
- ◆ [利用の流れ 実例](#)

## ❖ [生成モード説明](#)

- ◆ [データ読み取りモード](#)
- ◆ [基本解析モード](#)
- ◆ [スクリプト拡張モード](#)
- ◆ [統計情報ファイル入力モード](#)

## ❖ [ファイルフォーマット](#)

- ◆ [本機能が扱う全ファイルフォーマット共通注意事項](#)
- ◆ [統計情報生成ルールファイルフォーマット](#)
- ◆ [Settingのフォーマット \(グラフ設定\)](#)
- ◆ [グラフ設定と統計情報ビューアの対応](#)
- ◆ [円グラフについて](#)
- ◆ [ヒストグラムについて](#)
- ◆ [RegexpRuleのフォーマット](#)
- ◆ [BasicRuleのフォーマット](#)
- ◆ [BasicRuleのフォーマット When, From, To](#)
- ◆ [ScriptExtensionのフォーマット](#)
- ◆ [InputRuleのフォーマット](#)
- ◆ [グラフ設定の優先順位](#)
- ◆ [統計情報ファイルのフォーマット](#)

## ❖ [サンプルファイルの説明](#)

# 概要

- ❖ トレースログを解析して統計情報を取得・可視化する機能です
- ❖ 統計情報は、円グラフなどのグラフで可視化されます
- ❖ TLVファイルを出力すれば、ファイルとして統計情報を得られます



例: タスクのCPU利用率の可視化

# ファイル構成

---

## ❖ TLVの関連ファイル

### ◆ 統計情報生成ルールファイル

statisticsGenerationRules/task\_dispatch\_count.sgr

### ◆ 統計生成スクリプト

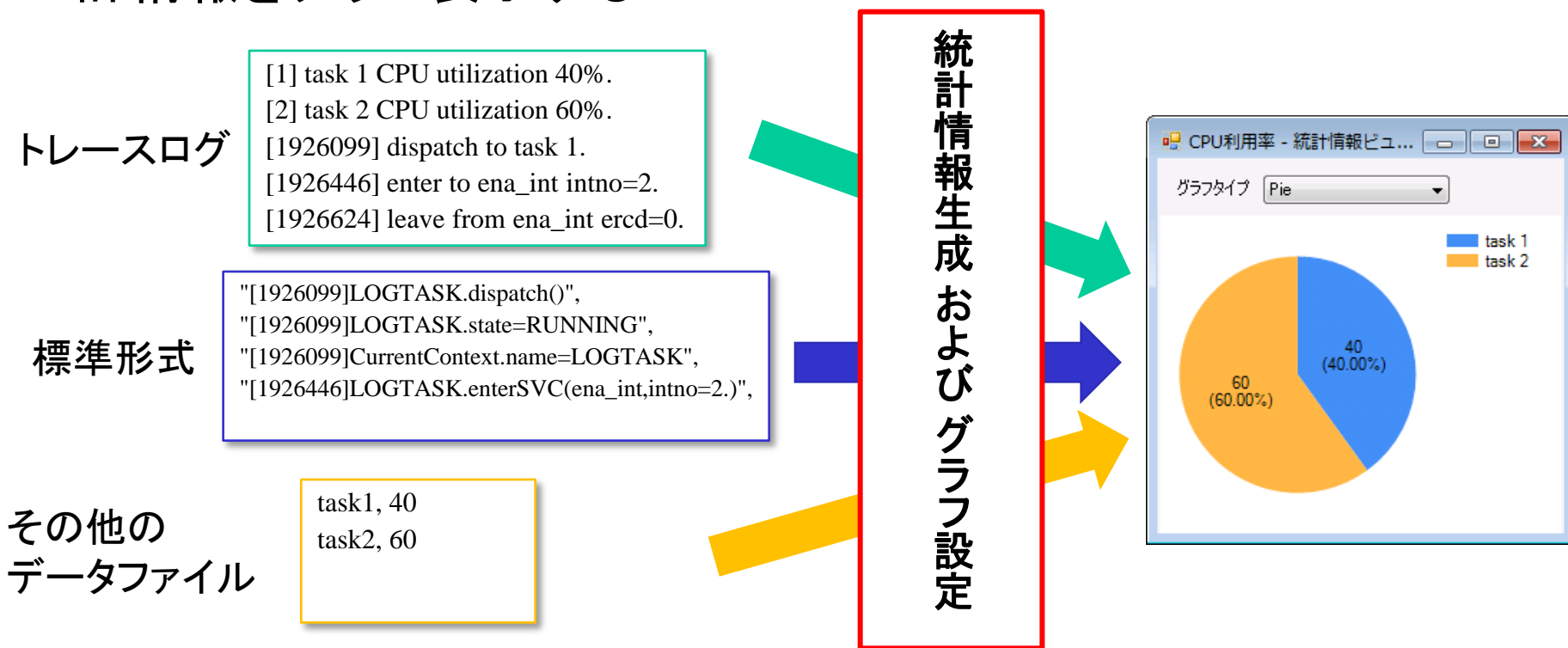
statisticsGenerationScript /task\_dispatch\_count.rb

### ◆ 統計情報ファイル サンプル

sampleFiles/statistics/cpu\_utilization.sta

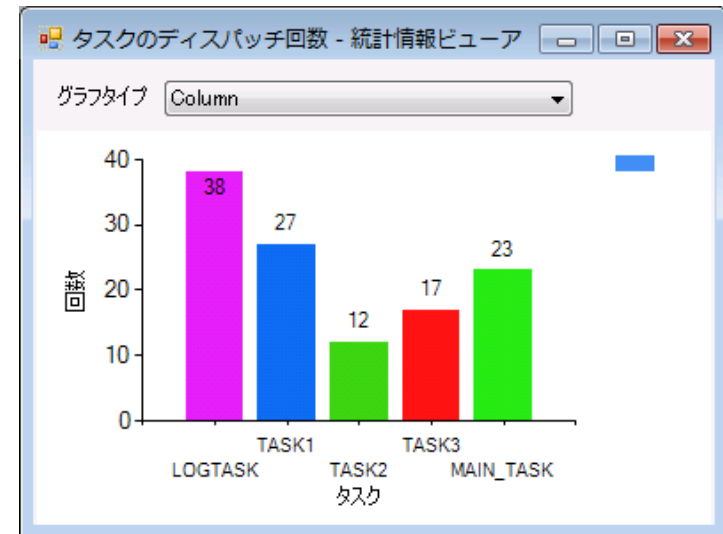
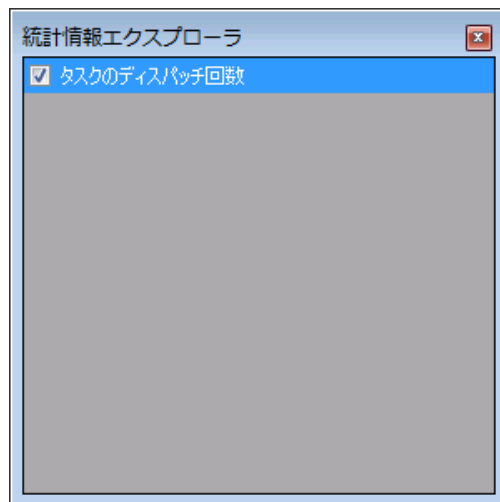
# 統計情報生成ルールとは？

- ❖ 統計情報ファイルを生成する規則
- ❖ OSが出力した様々なトレースログや、標準形式トレースログ、そのほかのデータファイルから統計情報を生成・取得し、得た統計情報をグラフ表示する



# 統計情報ファイルとは？

- ❖ 統計情報生成ルールによって生成された統計情報とグラフ設定を記録したファイル
- ❖ このファイル情報をもとに統計情報エクスプローラに表示する一覧や統計情報ビューアに表示するグラフを生成する



---

統計情報表示機能に関連するウィンドウの概要、利用方法を紹介します

# 関連ウィンドウ

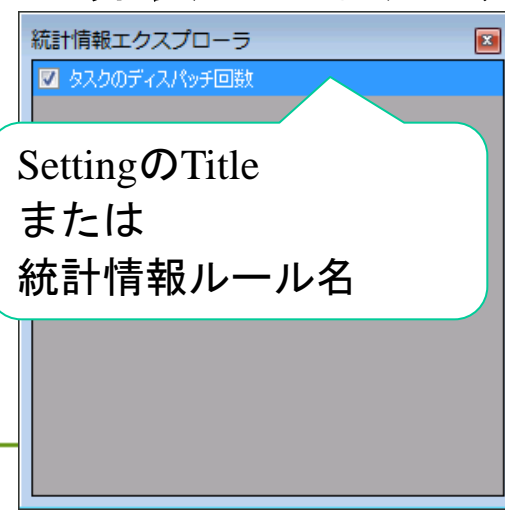
# 統計情報エクスプローラ

## ❖ 生成された統計情報の一覧が表示されるウィンドウ

- ◆ 表示される名前は、統計情報生成ルールのSettingにあるTitleで設定したものです。Titleの設定を省略した場合、統計情報生成ルール名が表示されます。
- ◆ エラーなどで生成できなかった統計情報は表示されません

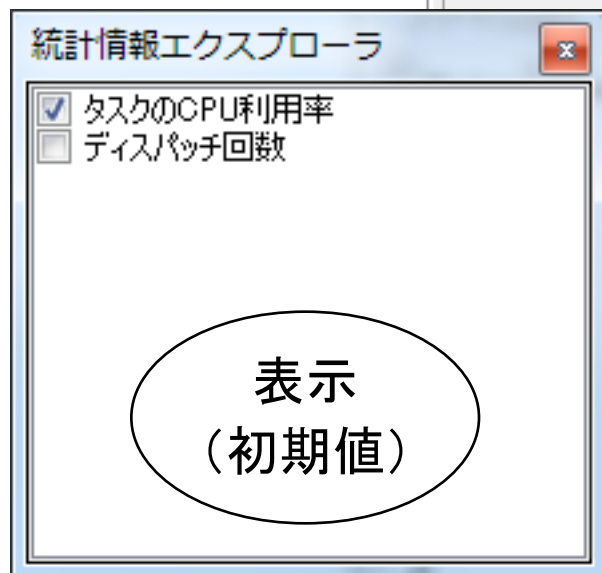
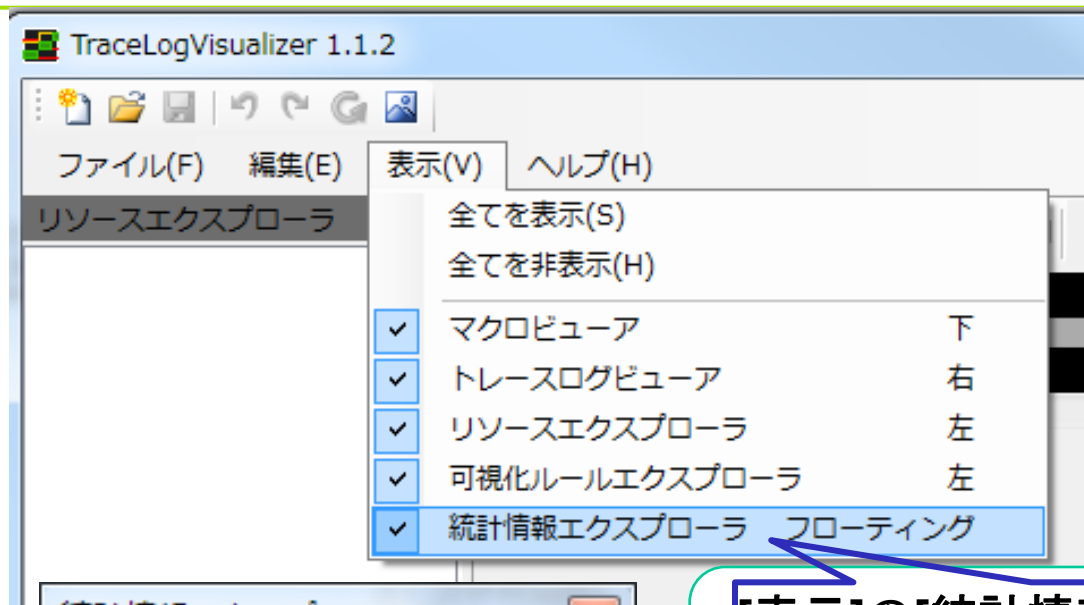
## ❖ 利用方法

- ◆ 名前の横にあるチェックボックスをチェックすると、その統計情報をグラフ表示した統計情報ビューアが出現します
- ◆ 逆に、チェックを外すとその統計情報ビューアは非表示になります
- ◆ TLV本体にドッキングできます
  - リソースエクスプローラ等と同様
  - このウィンドウの表示/非表示に関しても他のドッキング可能なウィンドウと同じです

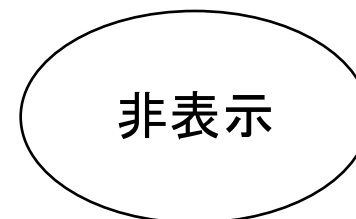
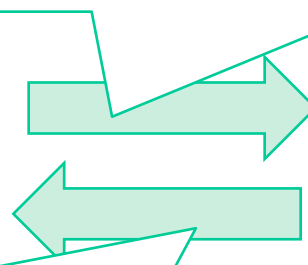




# 統計情報エクスプローラの表示/非表示



- ・[表示]の[統計情報エクスプローラ]のチェックを外す
- ・統計情報エクスプローラの[閉じる]ボタンを押す
- ・[表示]の[全てを非表示]を押す



- ・[表示]の[統計情報エクスプローラ]にチェックを入れる
- ・[表示]の[全てを表示]を押す

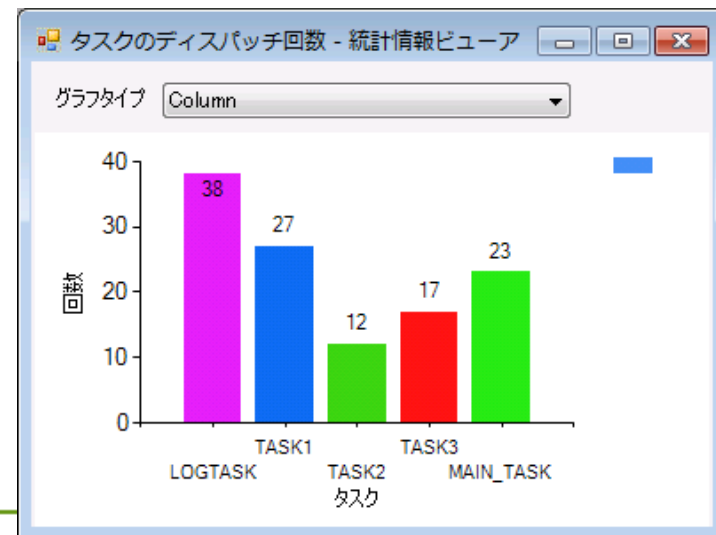
# 統計情報ビューア

## ❖ 統計情報をグラフで表示するビューア

- ◆ 統計情報生成ルールを用いて生成した統計情報ファイルを読み込んで表示しています

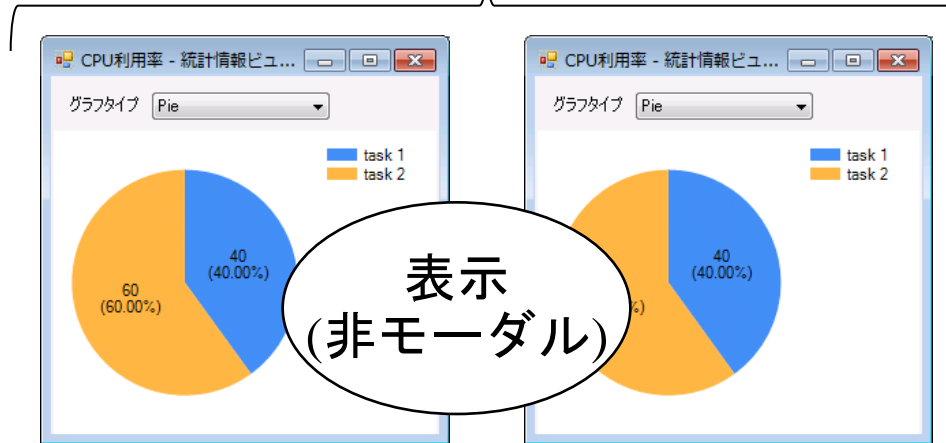
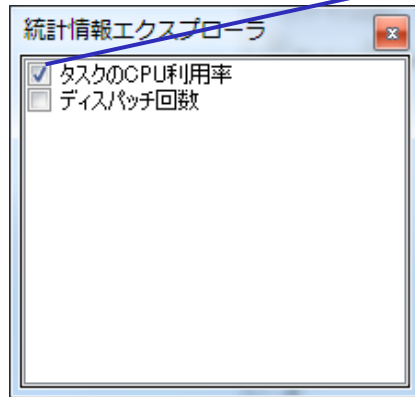
## ❖ 利用方法

- ◆ 表示するには、[統計情報エクスプローラ](#)をご利用ください
- ◆ 統計情報ビューアは各統計情報につき1つ表示されますので、画面上に複数の統計情報ビューアを表示できます
- ◆ このウィンドウは、TLV本体にドッキングできません
- ◆ グラフ設定がどのように反映されているかは「[グラフ設定と統計情報ビューアの対応](#)」を参照して下さい



# 統計情報ビューアの表示/非表示

チェックした数 = 統計情報ビューアの数



・ 統計情報エクスプローラにある項目をチェックする

・ 統計情報エクスプローラにある項目のチェックを外す  
・ 統計情報ビューアの[閉じる]ボタンを押す

非表示  
(初期値)

統計情報エクスプローラを閉じてても、  
統計情報ビューアは閉じない

---

# 利用方法

# 利用の流れ

---

前準備: 統計情報生成ルールファイルを作成しておきます  
([統計情報生成ルールフォーマット](#)、[生成モード](#)参照)

1. リソースファイル (.res) に適用する統計情報生成ルールを指定します
2. トレースログ (.log) とリソースファイルをTLVに入力します  
(トレースログの可視化を開始する方法と同じです)
3. 可視化後、統計情報エクスプローラに表示された一覧から見たい統計情報にチェックを入れますと、その情報をグラフ表示した統計情報ビューアが表示されます
4. 統計情報を保存したい場合、tlvファイルを保存してください

# 利用の流れ 実例

---

❖ 標準形式トレースログからTASKの属性がRUNNINGに変化したイベント回数を取得してグラフ表示する方法を例にして説明します

## ❖ 登場するファイル

❖ 統計情報生成ルールファイル

statisticsGenerationRules/task\_dispatch\_count.sgr

❖ トレースログファイル

sampleFiles/asp/asp\_stats.log

❖ リソースファイル

sampleFiles/asp/asp\_stats.res

# 1. リソースファイル (.res) に適用する統計情報生成ルールを指定します

統計情報生成ルールファイル  
task\_dispatch\_count.sgr

```
{  
  "task_dispatch_count" : {  
    "Setting" : { ... },  
    ...  
  }
```

リソースファイル  
asp\_stats.res

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

❖ リソースファイルに赤字の行を記述することで、統計情報生成ルールを指定します。

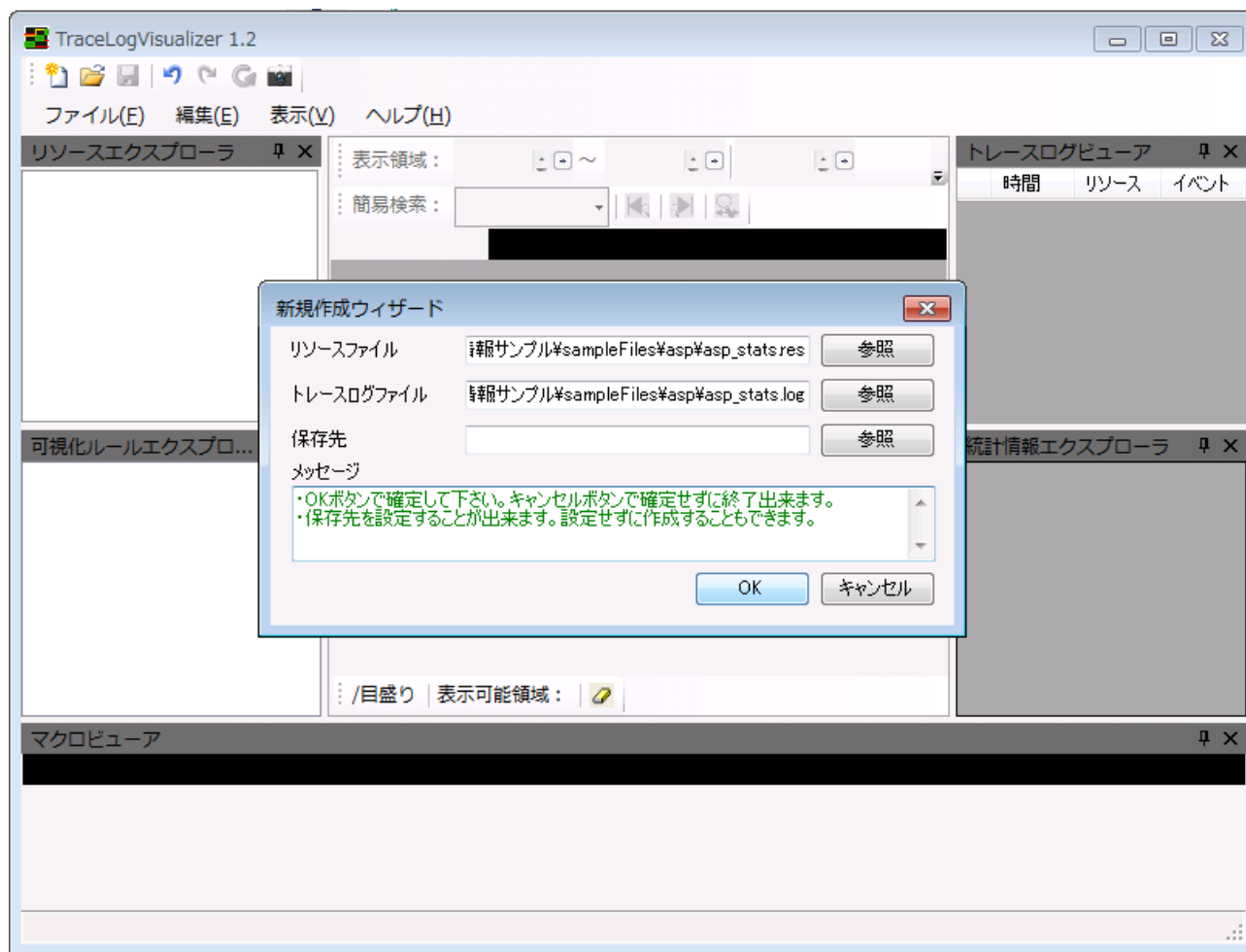
◆ VisualizeRules等と同様に、複数のルールを指定可能

❖ 統計情報生成ルール名は、統計情報生成ルールファイル内の青字で示したものです

◆ 統計情報生成ルールのフォーマットの詳細は、「[統計情報生成ルールフォーマット](#)」参照

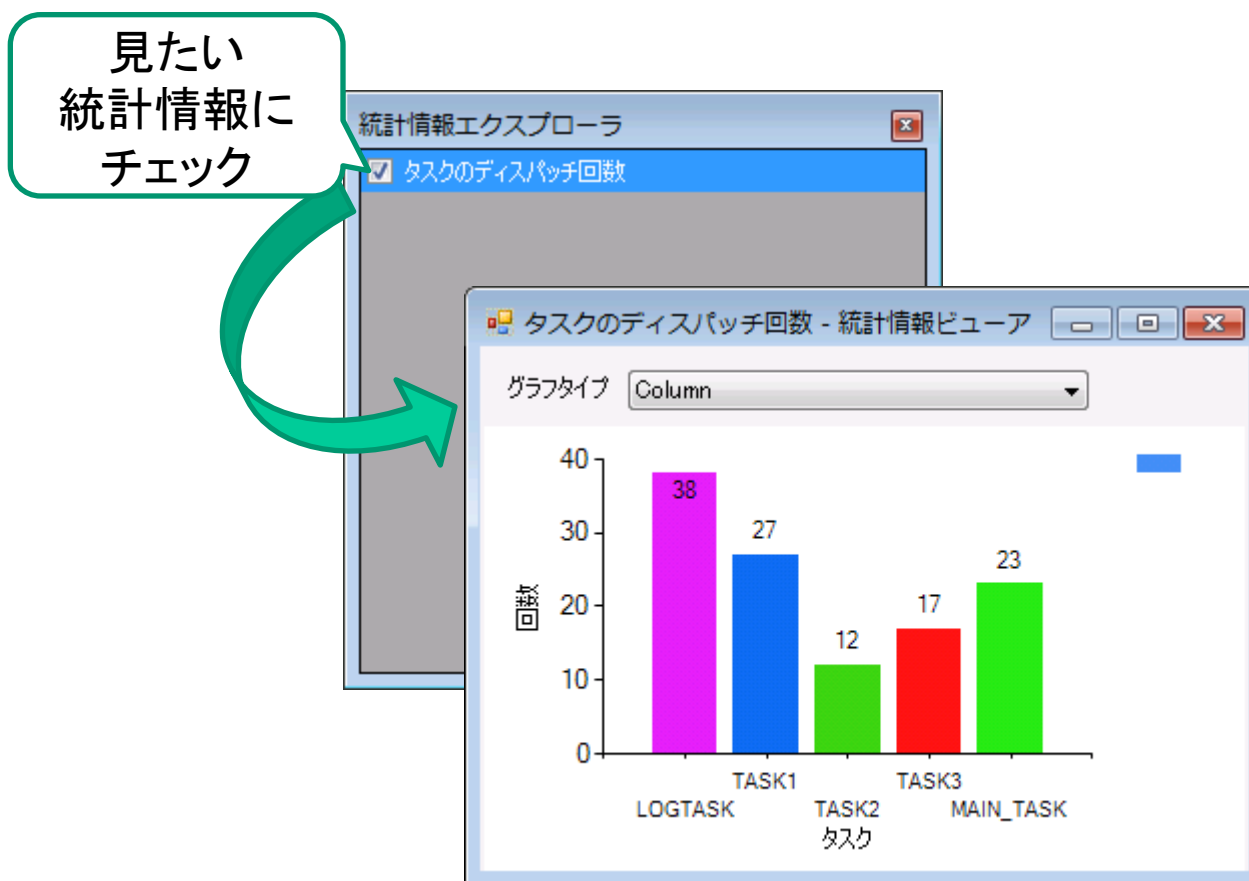
## 2. トレースログ (.log) とリソースファイルをTLVに入力します

❖従来通り、ドラッグ & ドロップも可能です

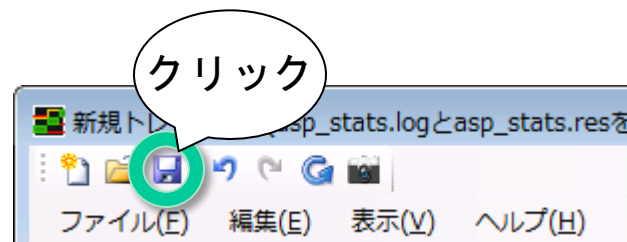




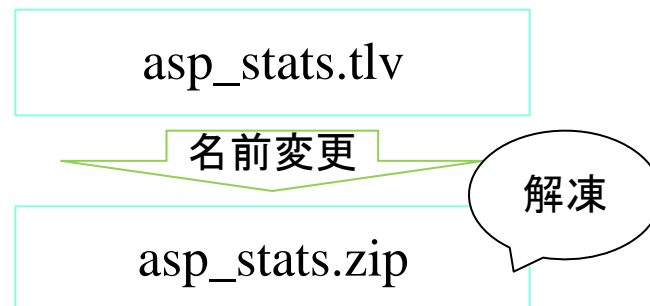
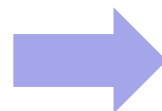
### 3. 統計情報エクスプローラに表示された一覧から 見たい統計情報にチェックを入れます



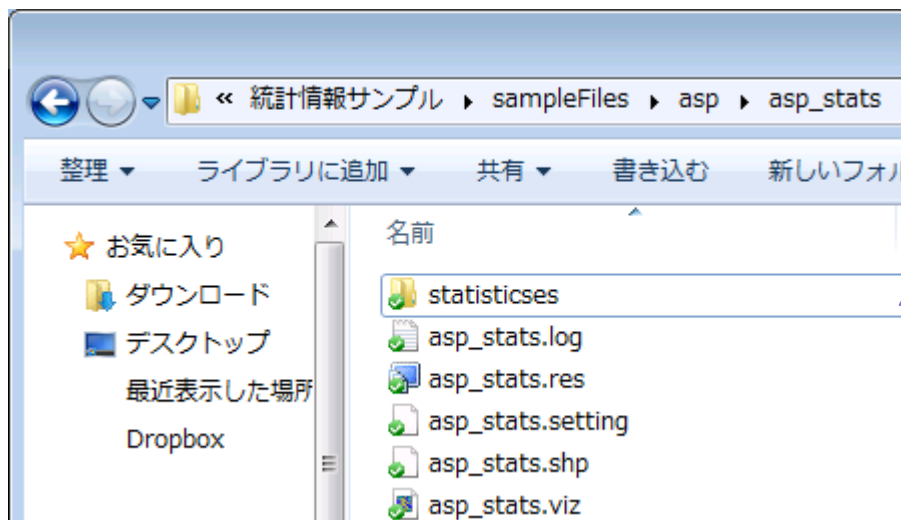
## 4. 統計情報を保存したい場合、tlvファイルを保存してください



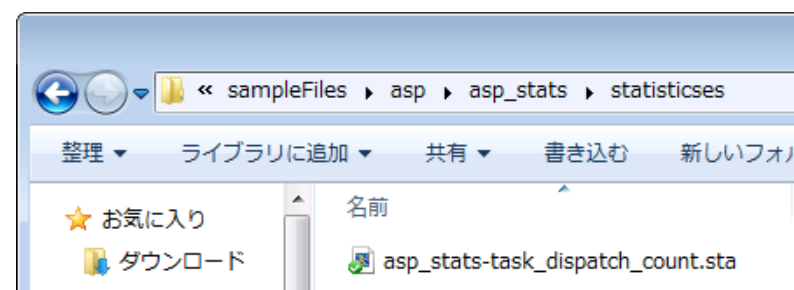
asp\_stats.tlvと名前を付けて保存



拡張子をzipに変更して解凍



解凍すると上図のファイル群が生成されます



statisticsesフォルダ内に生成した統計情報ファイルが保存されています

---

統計情報、統計情報ファイルを得る方法の総称を「生成モード」と名付け、いくつかの生成モードを用意しました

## 生成モード説明

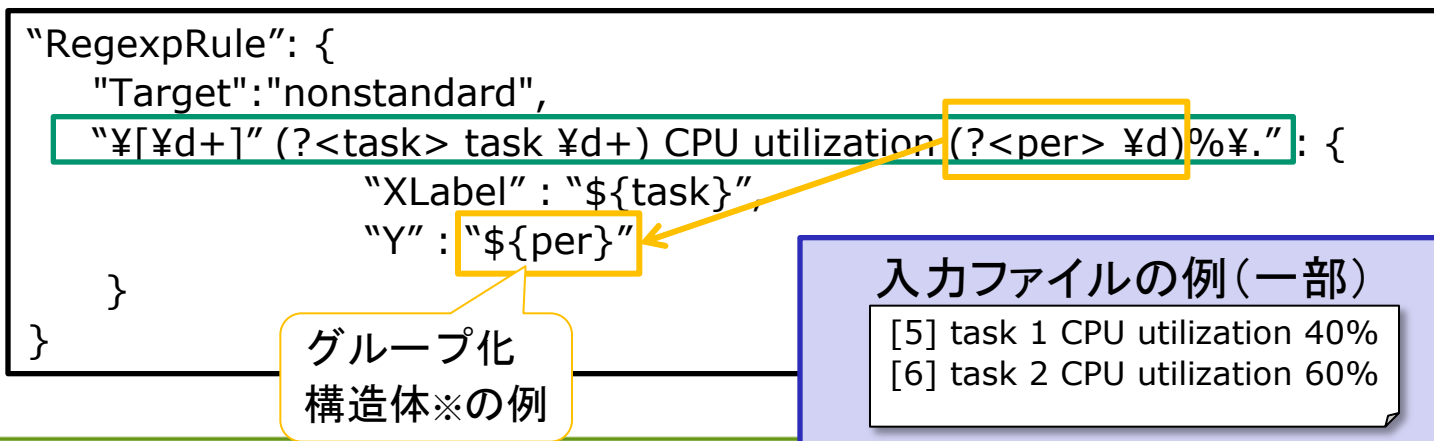
- [データ読み取りモード](#)
- [基本解析モード](#)
- [スクリプト拡張モード](#)
- [統計情報ファイル入力モード](#)

# データ読み取りモード

## 概要

- ❖ ログファイルやログファイル以外のファイルに記録された統計情報を読み込むモード
- ❖ 統計情報を取り出すには **正規表現** を使用
  - ◆ ファイル形式に依存しない読み込みが可能

### データ読み取りモードのルール例



# データ読み取りモード

---

## 統計情報生成ルール上の扱い

- ◆Mode指定: Regexp
- ◆ルール記述: [RegexpRule](#)

## 主な利用目的

- ◆ファイルに記録された統計情報、Color値を用いて統計情報のグラフ表示をしたい

## 対象ファイル

- ◆入力するトレースログ
- ◆標準形式トレースログ
- ◆上記以外のテキスト形式のファイル

# 基本解析モード

---

## 概要

- ❖ 簡単なルール記述で、標準形式トレースログを解析して統計情報を取得できるモード
- ❖ 高度な計算や取得方法には対応していないため、簡単に取得できるような統計情報を得るのに適しています

# 基本解析モード

---

## 利用可能な解析メソッド※

### ❖ Count: 指定したイベントの出現回数のカウント

- ◆Intervalを設定すると、ログをInterval値ずつに区切り、その各区間  
で出現したイベント数をカウントします([図解](#)参照)
  - この場合、X軸が各区間、系列がリソースとなるので、対象リソースは  
1つのみとなります。系列は1つのみの対応となっているためです。

### ❖ Measure: 指定したイベント間の間隔測定

- ◆現在、平均のみをグラフに描画します。統計情報ファイルのSeries  
のPointsには、平均値が記録されます。
- ◆最大・最小は[YLabel](#)で平均値とともに記録されます。

# 基本解析モード

---

## 統計情報生成ルール上の扱い

- ◆Mode指定: Basic
- ◆ルール記述: [BasicRule](#)

## 対象ファイル

- ◆標準形式トレースログ

## 主な利用目的

- ◆簡単な方法で得られる統計情報を、スクリプト作成より手軽に取得したい

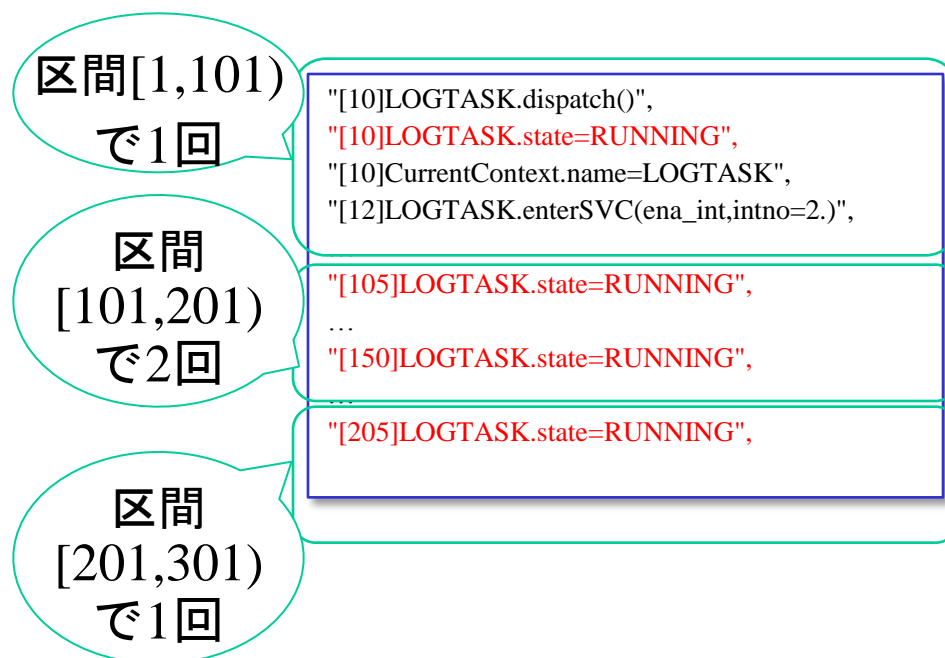


# 基本解析モード

## 解析メソッドの図解

### ❖ Count (Interval設定時)

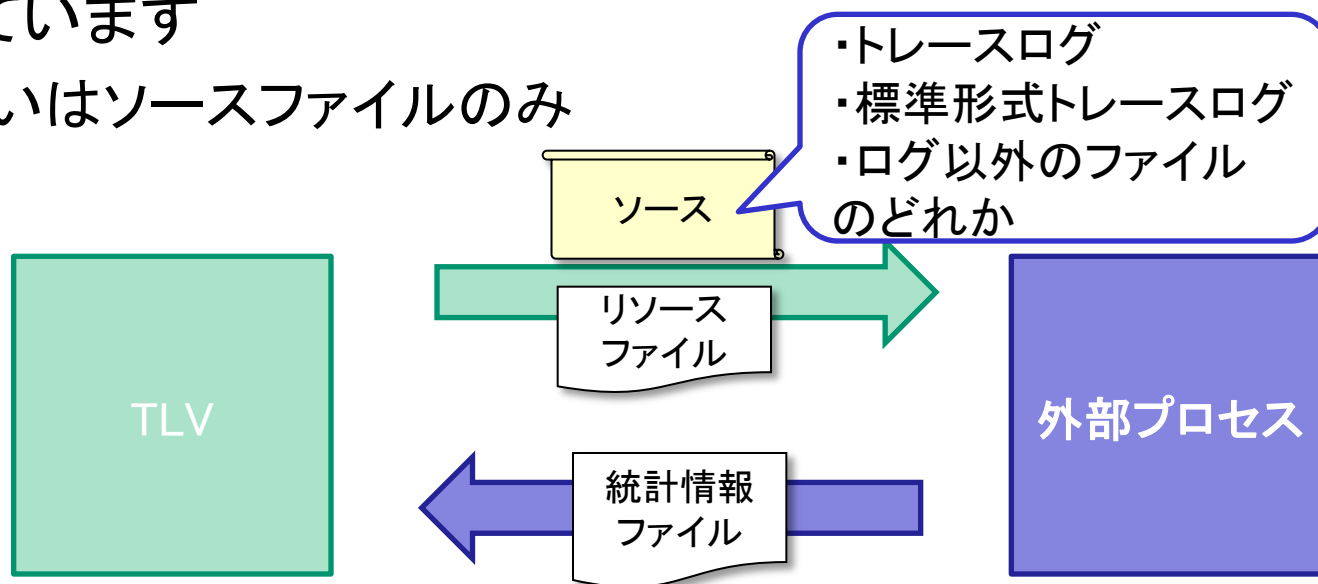
- 時刻1から始まるログで、Interval=100とした場合のLOGTASK.stateがRUNNINGになるイベント回数をカウント



# スクリプト拡張モード

## 概要

- ❖ 外部プロセスで統計情報ファイルを作成するスクリプトやアプリケーションを実行して統計情報ファイルを取得するモード
- ❖ 複雑な手段を用いて統計情報やグラフ設定を得るのに適しています
- ❖ 変換・可視化ルールにおけるスクリプト拡張とほぼ同じ仕様になっています
  - ◆ 違いはソースファイルのみ



# スクリプト拡張モード

---

## 統計情報生成ルール上の扱い

- ◆Mode指定 : Script
- ◆ルール記述 :  
[ScriptExtension](#)

## 対象ファイル

- ◆入力するトレースログ
- ◆標準形式トレースログ
- ◆上記以外のテキスト形式のファイル

## 主な利用目的

- ◆基本解析モードでは取得不可能な統計情報を取得したい
- ◆すでにスクリプトやアプリがあるのでそれを利用したい

# スクリプト拡張モード

---

## 外部スクリプト、アプリケーションの仕様

- ❖ 以下の処理の流れに対応している必要があります
  - ❖ ここでは「スクリプト」として記述していますが、アプリも同様です
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統計情報ファイルを、TLVが読み込む

# スクリプト拡張モード

外部スクリプト、アプリケーションの標準入力に書き込まれる  
ファイルイメージ

```
{                                リソースファイル
  "TimeScale" : "us",
  "TimeRadix" : 10,
  "ConvertRules" : ["asp"],
  "VisualizeRules" : ["toppers", "asp"],
  "ResourceHeaders" : ["asp"],
  ....
}
```

[1] task 1 CPU utilization 40%.

[2] task 2 CPU utilization 60%.

[192600] "[1926099]LOGTASK.dispatch()",

[192640] "[1926099]LOGTASK.state=RUNNING",

[192660] "[1926099]CurrentContext.name=LOGTASK",

"[1926446]LOGTASK.enterSVC(esp, int, int, 2)"

task1, 40

task2, 60

対象ファイル(1つ)

# 統計情報ファイル入力モード

---

## 概要

❖手元にある統計情報ファイルを入力するモード

## 利用可能な入力方法

❖統計情報ファイルの絶対パス指定

❖統計情報ファイルの内容をルールファイル内に記述

## 注意事項

❖このモードで作成される統計情報ファイルの統計情報名は、指定した統計情報ファイルにある統計情報名ではなく、このモードを指定した統計情報生成ルール名が適用されます。そのため、このモードで入力した統計情報ファイルと、それをもとに作成した統計情報ファイルでは、統計情報名が異なる場合があります

# 統計情報ファイル入力モード

---

## 統計情報生成ルール上の扱い

- ◆Mode指定: Input
- ◆ルール記述: [InputRule](#)

## 対象ファイル

- ◆統計情報ファイル

## 主な利用目的

- ◆以前の結果の再利用をしたい
- ◆他のログで得た統計情報を一緒に表示したい

---

# ファイルフォーマット



# 本機能が扱う全ファイルフォーマット共通注意事項

---

- ❖ 日本語を扱う際には、文字コードをUTF-8で保存したファイルを使用してください。
- ❖ 次ページ以降で紹介するフォーマットには、便宜上、コメント行として「**#**～」という表現を使用しています。しかし、JSON形式ではこのようなコメントは非対応なので、実際に記述するとエラーになります。

# 統計情報生成ルールファイルフォーマット

```
{  
  “統計情報生成ルール名” : {  
    “Setting” : { 「Settingのフォーマット」 参照 }, #省略時 : Setting内全てが省略時設定  
    “Mode” : ” 生成モード” #” Regex” :データ読み取りモード、” Basic” :基本解析モード、  
      #” Script” :スクリプト拡張モード、” Input” :統計情報ファイル入力モード  
      #省略するとエラーになります  
    “UseResourceColor” : Boolean #リソースファイルで設定したColorを使うか (省略はfalse)  
  
    “RegexRule” : { 「 RegexRuleのフォーマット」 参照 },  
    “BasicRule” : { 「 BasicRuleのフォーマット」 参照 },  
    “ScriptExtension” : { 「 ScriptExtensionのフォーマット」 参照 },  
    “InputRule” : { 「 InputRuleのフォーマット」 参照 }  
  },  
  “統計情報生成ルール名2” : { … } #ひとつのファイルに複数のルールを記述可能です  
}
```

→ これらは、1つのルール内に同時に定義できます。ただし、適用されるのは“Mode”で**指定した生成モードに関するもののみ**です。生成モードに関するものを省略するとエラーになります。

統計情報生成ルールファイルは、1つのルール定義を複数のファイルに分割することも可能です。その際は、統計情報生成ルール名を一致させること、定義の重複をなくすことに気を付けてください。

# Settingのフォーマット(グラフ設定)

“Title”: “グラフのタイトル (統計情報エクスプローラに表示)”, #省略時: 統計情報生成ルール名

“AxisXTitle”: “X軸のタイトル”, #省略時: 空文字列

“AxisYTitle”: “Y軸のタイトル”, #省略時: 空文字列

“SeriesTitle”: “系のタイトル”, #省略時: 空文字列

“DefaultType”: “最初のグラフの種類(右表)”, #省略時: ” Pie”

“MajorTickMarkInterval”: 主目盛幅(数値), #省略時: 自動設定

“MinorTickMarkInterval”: 補助目盛幅(数値), #省略時: 自動設定

“MajorGridVisible”: 主グリッドの有無(boolean), #省略時: false (なし)

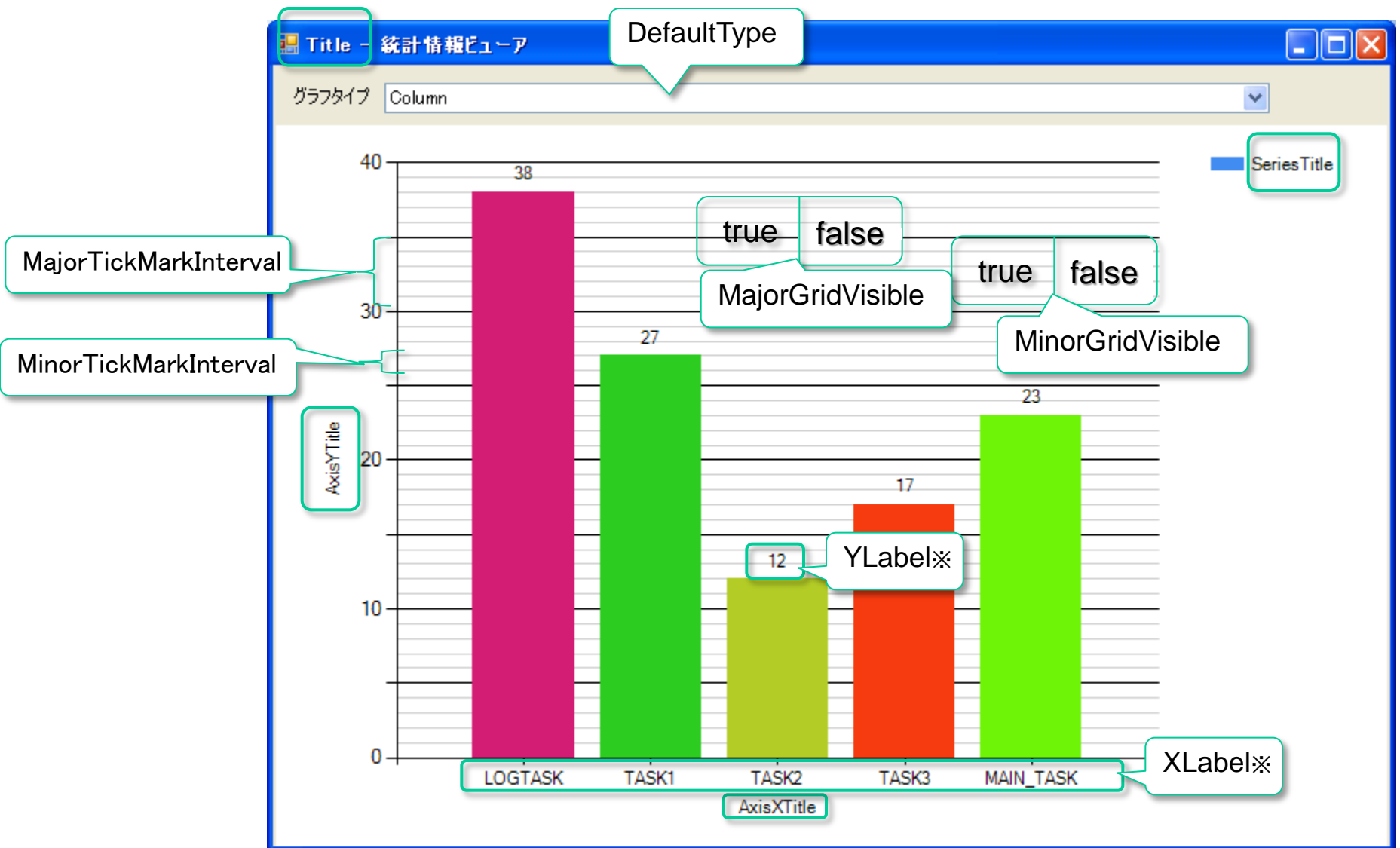
“MinorGridVisible”: 補助グリッドの有無(boolean) #省略時: false (なし)

DefaultTypeへの設定値と  
グラフの種類との対応表

設定値	グラフ種類
” Pie”	円
” Bar”	横棒
” Column”	縦棒
” Line”	折れ線
“Histogram”	ヒストグラム

全部省略可能

# グラフ設定と統計情報ビューアの対応



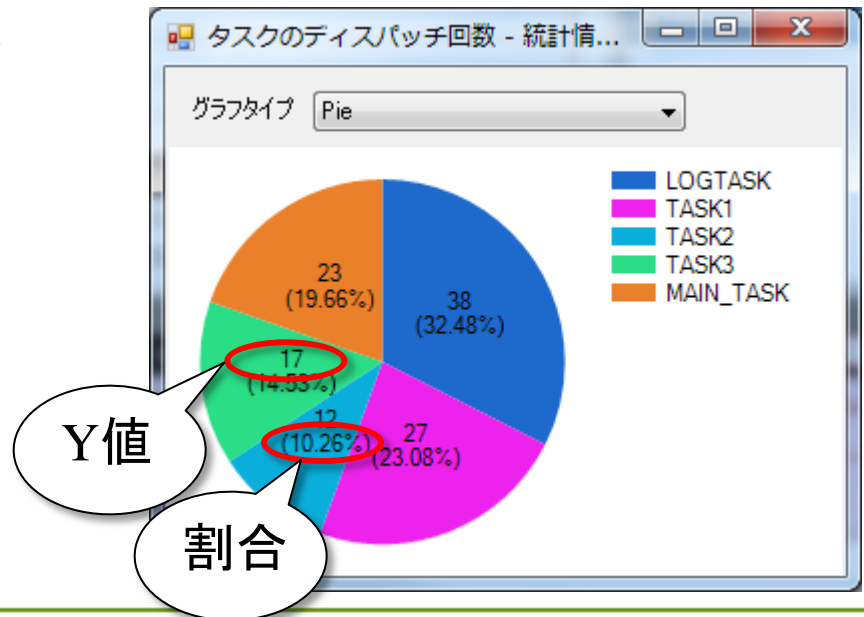
※Setting内の設定ではありませんが、グラフに関する事項なので記載しました。

具体的には、[RegexpRule](#)、[統計情報ファイル](#)で関係します。

# 円グラフについて

## ❖表示方法

- ◆円グラフは、全データポイントのY値合計を100%として、各データポイントのY値の割合で円を分割します。
- ◆各データポイントの全体に対するY値の割合は、統計情報ファイルに記録されません。
- ◆円グラフでは、YLabel が反映されません。TLV内部で円グラフ用に最適化されています。
  - Y値の下に円グラフ上の割合を括弧で囲って表示



# ヒストグラムについて

---

## ❖ 利用方法

- ◆ 生成する統計情報ファイルのPointsの各XValueは、本来の区間の中間点を設定してください。内部でのX軸のオフセット操作により、棒の両端に目盛が配置されます。
  - 例: XValueが100～200の点の場合、XValueは150と設定してください。

## ❖ 注意事項

- ◆ XLabelが設定されているとX軸のラベル調整がされません。ご利用の際は、XLabelを設定しないようにしてください。
- ◆ 現在、ヒストグラムを正しく表現するには、スクリプト拡張モードを利用する以外ありません。

# RegexRuleのフォーマット

“Target” : ” 生成元” , #standard:標準形式、nonstandard:入カログ、ファイルパス:ログ以外のファイル  
#省略するとエラーになります

“Regexps” : { #省略するとエラーになります

“正規表現” : #.NETの正規表現に準じます。

{ #次の4項目の値には、” グループ化構造体※1” が指定可能です。

“XLabel” : ” X値の文字列ラベル” , #省略時 : Xvalueの数字

“XValue” : “X値” , #省略時 : ” 0”

“Yvalue” : “Y値” , #省略時 : ” 0”

“Color” : “色 (#AARRGGBB : 16進数)” #省略時 : .NETライブラリの既定値※2

},

“正規表現” : {…} #複数のルールが記述可能です※3。

#ただ、定義した正規表現の回数、データ全体を読み直すので、

#定義数、データ数・行数によっては生成に時間がかかる場合があります。

}

※1 : 詳細はrules-manual.pdfの「2.2.4 変換ルールファイル(JSON形式)」4段落目など参照

※2 : ” UseResourceColor” が true ならば、省略するしないに関わらず、リソースに設定した色を使用します。詳細は「[グラフ設定の優先順位](#)」参照。

※3 : 正規表現を表す文字列の重複は許しませんが、文字列が異なれば同じ意味となる正規表現の重複は許します。たとえば、各タスクのColorをルール内で静的に定義したい場合などにご利用頂けます。

# BasicRuleのフォーマット

“BasicRule” : {

“Method” : “統計収集方法名”, # Count: カウント、Measure : 計測

“Interval” : 時間幅(単位はリソースファイルで指定したもの),

“When” : {

“ResourceType” : [ “リソースタイプ名” ],

“ResourceNames” : [ ” リソース名” ],

“AttributeName” : ” 属性名” ,

“AttributeValue” : 属性値,

“BehaviorName” : ” 振る舞い名” ,

“BehaviorArg” : ” 振る舞いの引数” #省略可

},

“From” : {Whenと同じ}

“To” : {Whenと同じ}

}

指定したリソースタイプに属するリソースを対象とします

両方記述可能※1。

両方省略はエラーです。

属性変化か振る舞いか

どちらか一方のみ記述可能。

両方記述または省略はエラーです

When、From、Toに  
ついては次ページ参照

※1 : この場合、対象となるリソースの集合をA、指定したリソースタイプに属するリソースの集合をT、指定したリソースの集合をRとしたとき、

$$A = (T \cup R) - (T \cap R)$$

となります。

順序はResourceNamesの配列、ResourceTypeの配列となります。



# BasicRuleのフォーマット When, From, To

---

- ❖ある1つのイベントのみを対象としたい
  - ◆Whenを利用する
- ❖あるイベントを始点として、あるイベントを終点としたい
  - ◆From: 始点となるイベント
  - ◆To: 終点となるイベント
- ❖From-Toに設定するResourceType,ResourceNamesについて
  - ◆リソースは設定順に適応していきます。総当たりではありません。
    - 例: FromのResourceNames の1番目とToのResourceNames の1番目
- ❖注意事項
  - ◆WhenとFrom-Toは1つのルール内で同時に利用できません。

# ScriptExtensionのフォーマット

```
“ScriptExtension” : {  
    “Target” :” 生成元” , #standard:標準形式、nonstandard:入力ログ、  
        #ファイルパス:ログ以外のファイル  
        #省略するとエラーになります。  
  
    “FileName” : “スクリプト処理系(またはアプリケーション)の絶対パス” , #省略するとエラー  
    “Arguments” :” 実行時に渡される引数。外部スクリプトなどを指定。”  
        #” {0}” とした場合、” Script” に記述されたスクリプト文を含んだ  
        #一時ファイル名が指定されます。  
        #省略するとエラーになりますが、どうしても省略したい場合は” ” のみを記述  
    “Script” :” スクリプト文”  
}
```

基本的に変換・可視化時のスクリプト拡張機能と同じです(Targetと大文字小文字の違い以外同じ)。  
ここの内容だけでは理解できなかった場合、TLV外部スクリプトマニュアル(doc/scripts.pdf)を参照  
するとよいかもしれません。

# InputRuleのフォーマット

```
“InputRule” : {  
  “FileName” : “統計情報ファイルの絶対パス”  
  “Data” : { “統計情報名” : { 以降は「統計情報ファイルのフォーマット」参照 } }  
  #統計情報ファイルをルール内に直接記述します。記述できるのは1つの統計情報ファイルのみです。  
}
```

どちらか一つのみ記述。  
両方記述した場合、FileNameが優先されます。

# グラフ設定の優先順位

## Settingの優先順位

### ❖最高

- ◆モードのルールを用いて生成した際に得られた設定
  - スクリプト拡張  
"ScriptExtension"※1
  - 統計情報ファイル入力モード  
"Input"
- ◆統計情報生成ルールのSetting

### ❖最低

## Colorの優先順位

### ❖最高

- ◆UseResourceColor(trueの場合)
- ◆モードのルールを用いて生成した際に得られた設定
  - データ読み取りモード"Regexp"
  - スクリプト拡張"ScriptExtension"
  - 統計情報ファイル入力モード  
"Input"

### ❖最低

※1: Settingの各項目ごとに優先順位が適用されます。つまり、統計情報生成ルールで定義されたSettingの項目aを外部プロセスで設定された項目aで上書きします。項目a以外の項目で重複がない場合は、それらの項目は上書きしません。

# 統計情報ファイルのフォーマット

```
{
  “統計情報名” : {
    “Setting” : { 「Settingのフォーマット」 参照 }, #省略時 : Setting内全てが省略時設定
    “Series” : {
      “Points” : [ #Xvalueが同じ値の場合は記載順、異なれば昇順でグラフ表示されます
        {
          “XLabel” : “X値の文字列ラベル”, #省略時 : Xvalueの数字
          “XValue” : “X値”, #省略時 : ” 0”
          “YLabel” : “Y値の文字列ラベル※1, ※2”, #省略時 : 空字(グラフにはY値が表示されます)
          “Yvalue” : “Y値”, #省略時 : ” 0”
          “Color” : “色(#AARRGGBB : 16進数)” #省略時 : .NETライブラリの既定値
        },
        { ... }
      ]
    }
  }
}
```

※1 YLabelには、マクロが利用できます

マクロ	置き換えられるデータ
#VALX	X値
#VALY	Y値
#PERCENT	系列全体に対するY値の割合
#TOTAL	系列のすべてのY値の合計

一つの統計情報ファイルに統計情報を一つとしています。  
複数記述すると、**エラー**が発生します。主に、TLVファイル読み込み、統計情報ファイル入力モード、スクリプト拡張モードに関係します。

※2 円グラフ(Pie)、ヒストグラム(Histogram)には適用されません。

# サンプルファイルの説明

## sampleFiles/SampleLog/asp/asp\_stats.log

- ❖ 内容は asp\_short とほぼ同様です。
- ❖ asp\_short との違いは、最初の 2 行が追加されている点です。

### 【追加された2行のログ】

```
[1] task 1 CPU utilization 40%.  
[2] task 2 CPU utilization 60%.
```

- ❖ 追加されたログは、架空のトレースログで、ターゲットシステム内で得られた統計情報をトレースログとして出力した場合を想定したものです。現在(TLV2.0)の logtrace フォルダ内のものは対応していません。

### ※ 【追加された記述】

```
"StatisticsGenerationRules":["asp_cpu_utilization", "task_dispatch_count"],
```

## sampleFiles/SampleLog/asp/asp\_stats.res

- ❖ 内容は asp\_short とほぼ同様です。
- ❖ asp\_short との違いは、統計情報生成ルール指定をする記述が追加されている点です※。
- ❖ 現在(TLV2.0)、この指定はリソースファイル(.res)の生成時に記述されていないため、各自で追記する必要があります。生成時に記述したい場合、logtraceフォルダの tl.v.tf ファイルを修正してください。