

# TOPPERS 活用アイデア・アプリケーション開発 コンテスト

部門 : アプリケーション開発部門

作品のタイトル : TOPPERS Configuration Generator (TCG)

作成者 : 富士ソフト株式会社 (代表: 嶋原一人)

共同作業 : -

対象者 : TOPPERS 第3世代カーネル利用者

使用する開発成果物 : TOPPERS 第3世代カーネル

## 目的・狙い

静的 API によるコンフィギュレーションを行うには、静的 API のシンタックスやパラメータ・属性の意味を覚えるか、仕様書を見ながら記述する必要がある。そこで、GUI ツールを用いて、視覚的にコンフィギュレーションを行えるツールがあれば、静的 API を覚えたり、仕様書を見る必要がなくなり、有用であると考えた。

## アプリケーションの概要

TOPPERS 第3世代カーネルに対するコンフィギュレーションを行う GUI ツール: TCG を開発した。Windows/MacOS/Linux で動作すること、および保守性を考慮し、Ruby/GTK3 を使用した。TCG でコンフィギュレーションを行った後、cfg ファイルの出力に加え、静的 API を表示(プレビュー)する機能を設けた。

## TCG 実行環境

Ruby および GTK3 実行環境で動作する.

Windows の場合, 以下を参考にする.

<https://qiita.com/doublev80/items/6b0f623b2e4aa63c4bb6>

MacOS, Linux の場合, Ruby 実行環境を用意した上で, gem により GTK3 をインストールする.

```
$ gem install gtk3
```

## TCG を用いるメリット

<コンフィギュレーションの簡易化>

ASP3 カーネルだけでも, 静的 API は 17 個存在し, 各 API のシンタックス, パラメータ, 属性の意味を覚えるのは困難である. TCG では, 各パラメータ, 属性のツールチップに, 統合仕様書の説明を表示しているため, 統合仕様書を見なくても, TCG だけでコンフィギュレーションを行うことが可能である.

<プロファイル変更>

ASP3/FMP3/HRP3/HRMP3 のコンフィギュレーションをラジオボラン 1 つで切り替えられるため, 例えば, ASP3 向けのコンフィギュレーション情報を, 簡単に別のプロファイル用のコンフィギュレーションへ変更することができる.

<マルチプラットフォーム>

TCG は, Ruby/GTK3 で開発しているため, 無償の実行環境を用意すれば, Windows, Linux, MacOS のいずれでも使用可能である.

<保守性>

有償の開発環境やコンパイラなどが不要であり, Ruby による開発が可能であるため, 機能拡張や, 不具合修正, デバッグを容易に行うことができる.

<拡張性>

TOPPERS 第 3 世代カーネルでは, コンフィギュレータを Ruby で開発しているため, 将来的に, コンフィギュレータによるエラーチェックやコード生成処理を, TCG に取り込んだり, TCG から実行するといったことも可能である.

## TCG の主な使用方法

Ruby で `tcg.rb` を実行すると、以下の画面が表示される。

The screenshot shows the TOPPERS Configuration Generator interface. The 'Profile' section is set to ASP3. The 'タスク' (Task) sidebar is active, and the configuration fields for a task are displayed. The 'タスクID' field is empty. The 'クラス' (Class) is set to CLS\_ALL\_PRC1. The '属性' (Attributes) section has checkboxes for TA\_ACT and TA\_NOACTQUE. The '拡張情報' (Extension Information) field is empty. The '先頭番地' (Start Address) field is empty. The '起動時優先度' (Startup Priority) field is empty. The 'スタックサイズ' (Stack Size) field is empty. The 'スタック先頭番地' (Stack Start Address) field is empty. The 'システムスタックサイズ' (System Stack Size) field is empty. The 'システムスタック先頭番地' (System Stack Start Address) field is empty. The '保護ドメイン' (Protection Domain) field is empty. The 'アクセス許可パターン' (Access Permission Patterns) section has four fields for '通常操作1', '通常操作2', '管理操作', and '参照操作', all of which are empty. The '表示' (Show) and 'ファイル出力' (File Output) buttons are visible at the bottom right.

初期状態では、何も設定がない状態である。左側のカーネルオブジェクトのリストから、設定を行うものを選択することで、右側の設定項目が選択したカーネルオブジェクトに応じて切り替わる。

右上の **Profile** にて、対象のカーネルを選択でき、選択したカーネルで使用可能なオブジェクト、パラメータのみ編集が可能となる。

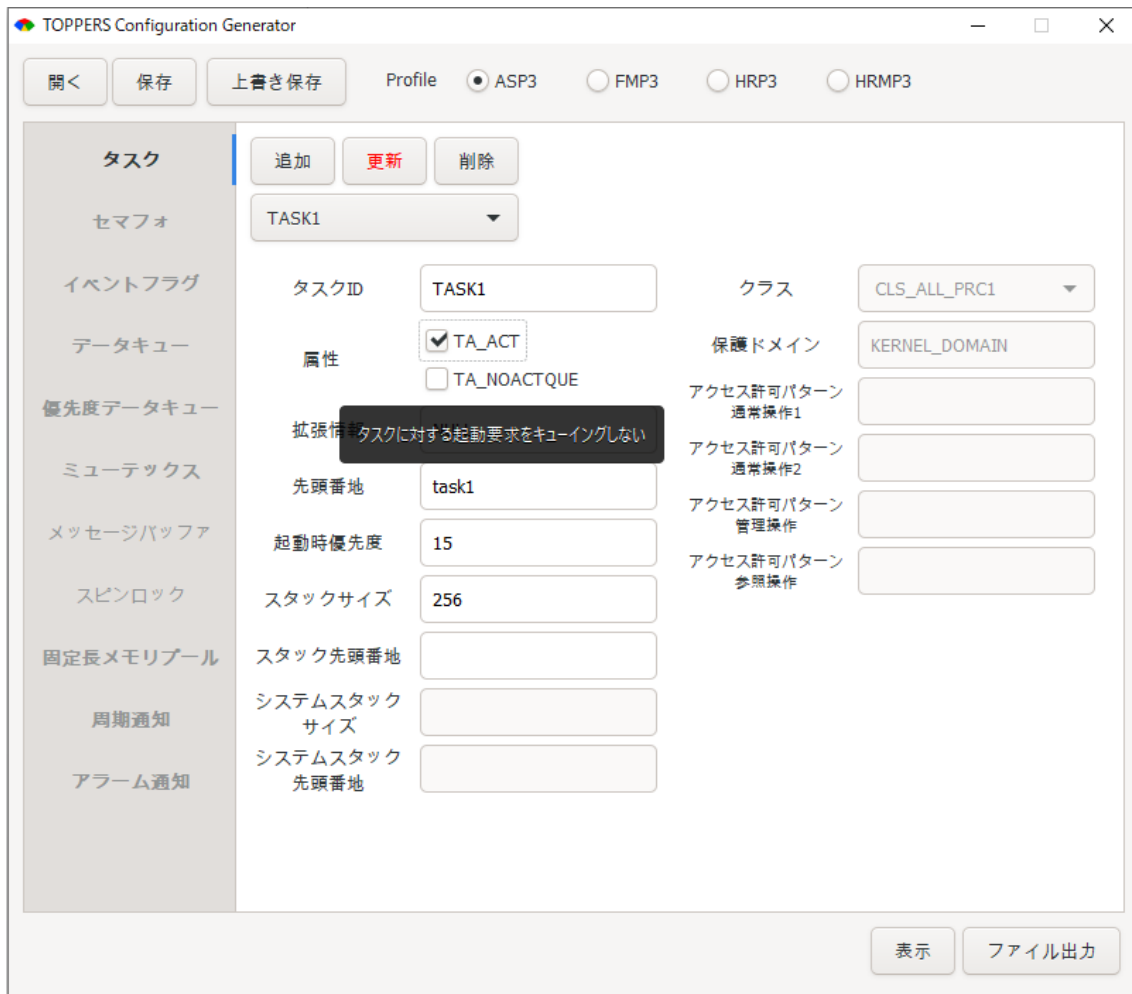
「開く」ボタン：保存したファイルを読み込む。

「保存」ボタン：設定内容をファイル(yaml形式)に名前を指定して保存する。

「上書き保存」ボタン：編集中のファイルに上書き保存する。

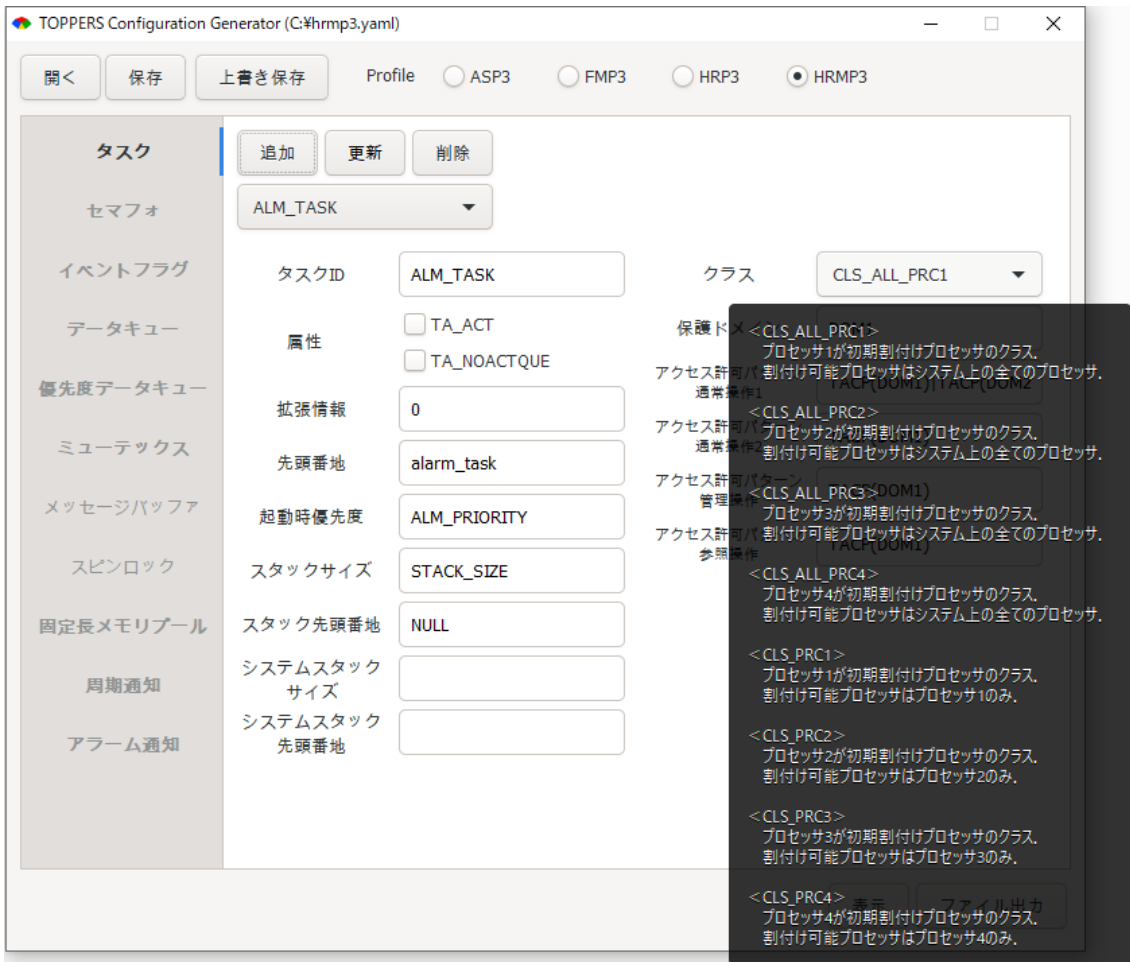


カーネルオブジェクトにマウスを重ねると、対象オブジェクトの説明が表示される。

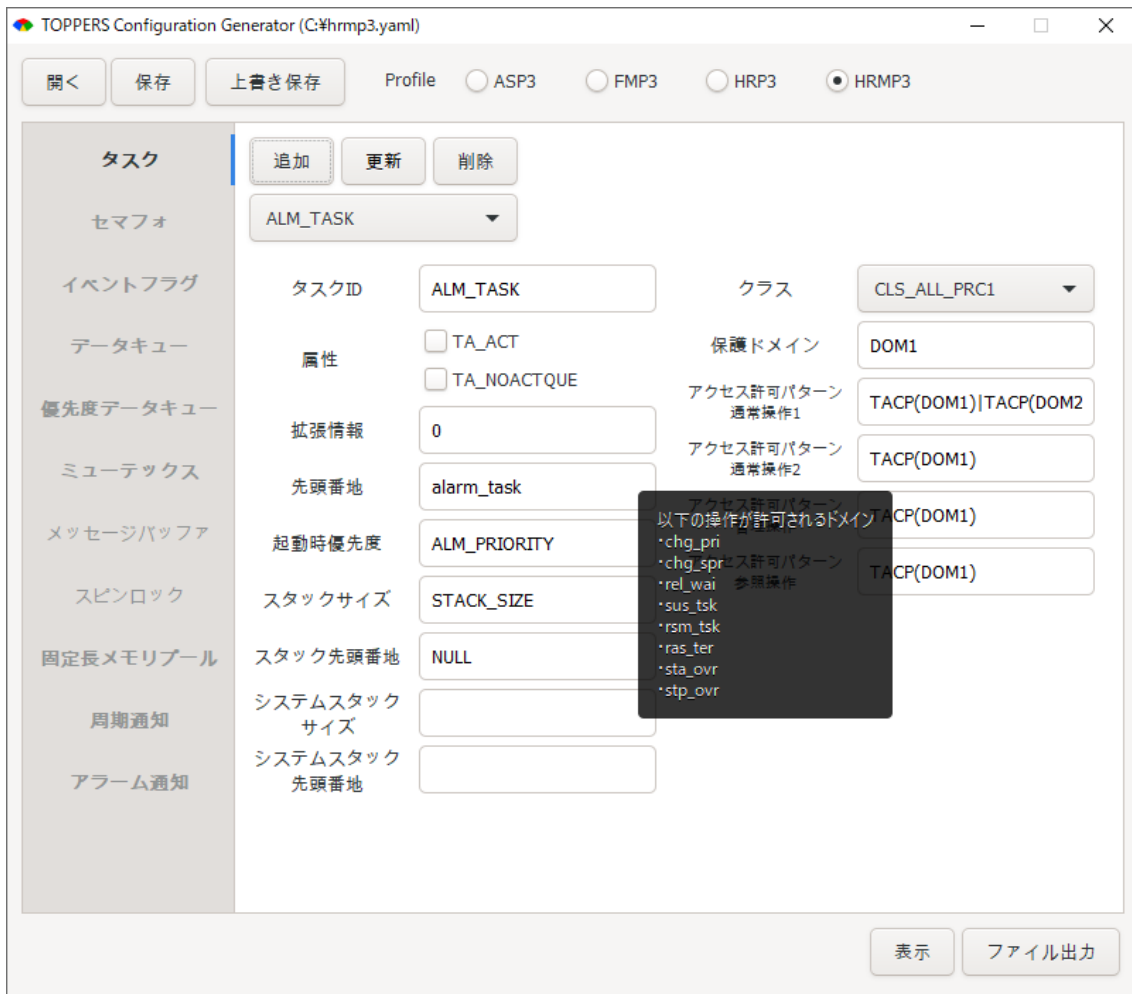


「追加」ボタン押下により、オブジェクトが追加され、デフォルト値が設定される。  
各パラメータを変更後、「更新」ボタンを押下することで、設定内容が反映される。  
※パラメータや属性の選択肢にマウスを重ねること、説明が表示される。  
※反映されていない状態の場合「更新」ボタンの文字が赤くなる。

「追加」「更新」「削除」ボタンの下のコンボボックスで、設定対象のオブジェクトを変更する。「削除」ボタン押下により、設定中のオブジェクトを削除する。



マルチプロセッサ対応カーネルにおいて、所属プロセッサを指定するクラスはコンボボックスから選択できる。コンボボックスにマウスを重ねると、各クラスの説明が表示される。



保護機能対応カーネルにおいて、アクセス許可ベクタを設定する場合、各アクセス許可パターンにマウスを重ねると、許可される操作が表示される。

TOPPERS Configuration Generator (C:\hmp3.yam)

開く 保存 上書き保存 Profile  ASP3  FMP3  HRP3  HRMP3

タスク 追加 更新 削除

セマフォ ALMHDR1

イベントフラグ

データキュー

優先度データキュー

ミューテックス

メッセージバッファ

スピンロック

固定長メモリアル

周期通知

アラーム通知

アラーム通知ID ALMHDR1 クラス CLS\_ALL\_PRC1

保護ドメイン DOM1

アクセス許可パターン 通常操作1

アクセス許可パターン 通常操作2

アクセス許可パターン 管理操作

アクセス許可パターン 参照操作

タイムイベントの通知の種類 TNFY\_ACTTSK タスクID ALM\_TASK

エラーの通知の種類 TNFY\_SNDTQ データキューID DATAQUEUE1

送信する値 1

表示 ファイル出力

周期通知，アラーム通知では，タイムイベントの通知の種類をコンボボックスから選択することで，対応する付随情報の設定に切り替わる．エラーの通知の種類も同様に，設定した種類に応じた付随情報の設定に切り替わる．

タスクやセマフォを選択する通知の場合，設定済みのカーネルオブジェクトを，コンボボックスから選択することができる．



```
View Static API
CLASS(CLS_ALL_PRC1) [
    DOMAIN(DOM1) [
        CRE_TSK(TASK1, [TA_NULL, 1, task, MID_PRIORITY, STACK_SIZE, NULL])
        SAC_TSK(TASK1, [TACP(DOM1)|TACP(DOM2), TACP(DOM1), TACP(DOM1), TACP(DOM1)])
        CRE_TSK(TASK2, [TA_NULL, 2, task, MID_PRIORITY, STACK_SIZE, NULL])
        CRE_TSK(ALM_TASK, [TA_NULL, 0, alarm_task, ALM_PRIORITY, STACK_SIZE, NULL])
        SAC_TSK(ALM_TASK, [TACP(DOM1)|TACP(DOM2), TACP(DOM1), TACP(DOM1), TACP(DOM1)])
        CRE_ALM(ALMHDR1, [TA_NULL, [TNFY_ACTTSK, ALM_TASK]])
    ]
    DOMAIN(DOM2) [
        CRE_TSK(TASK3, [TA_NULL, 3, task, MID_PRIORITY, STACK_SIZE, NULL])
    ]
    KERNEL_DOMAIN [
        CRE_TSK(MAIN_TASK, [TA_ACT, 0, main_task, MAIN_PRIORITY, STACK_SIZE, NULL])
        CRE_TSK(EXC_TASK, [TA_NULL, 0, exc_task, EXC_PRIORITY, STACK_SIZE, NULL])
        CRE_CYC(CYCHDR1, [TA_NULL, [TNFY_HANDLER, 0, cyclic_handler], 2000000, 0])
    ]
}
]
```

「表示」ボタンを押下することで、設定中のコンフィギュレーションに応じた静的 API を表示するプレビューウィンドウが起動する。「ファイル出力」ボタン押下により、この内容を名前を指定して `cfg` ファイルに保存する。

※プレビューウィンドウの表示からコピー&ペーストして使用してもよい。

※プレビューウィンドウを表示したままにしおけば、編集した内容が即時で反映される。

## 今後の展望

現時点では、コンテスト用の開発であるため、すべての静的 API に対応しておらず、厳格なエラーチェックなども行っていない。今後、有用なコンフィギュレーションツールとして、TOPPERS 第 3 世代カーネルに同梱して頂くなど、一般に使用して頂ける目処があれば、TOPPERS の trac/svn にて管理して、機能拡張・メンテナンスしていきたい。

主に、以下のような機能拡張を検討している。

- ・未対応の静的 API への対応。
- ・メモリ保護、時間保護に対するコンフィギュレーションの可視化。
- ・不正操作等への厳格なエラーチェック。
- ・`cfg.rb` の取り込み(`cfg.rb` 側の構成やターゲット依存部との切り分け等の課題あり)。
- ・既存の `cfg` ファイルをパースして取り込む機能。

以上。