

ネット公開を目的としたマルチウィンドウアプリ用フレームワーク 簡易マニュアル

1. 動作環境

1.1 サポート OS

公式サポートは Unix 系 OS とする（開発時の主たるテスト OS は Linux）。
ただし、公式サポート外でも次項の必要物が揃えられる環境であれば、動作する可能性が高い。

1.2 必要物

- a) Xvnc : inetd 対応 (標準入出力を RFB プロトコル通信用に用いることができる) のものであること。通信量の点から Tight VNC を推奨する。
- b) Tcl/Tk : (a) の Xvnc に表示できるものであること。現在のサポート状況から Tcl/Tk 8.4 系を推奨する。
- c) Ruby : 1.8.5 以降または 1.8.5 以降の Ruby に付属の Ruby/Tk をバックポートしたもの (1.8.5-*preview3* 以降でも可)。Ruby/Tk が (b) の Tcl/Tk ライブラリを用いることができるように適切にコンパイルされている必要がある。
- d) その他 : 公開するアプリケーションで必要とされるライブラリ類が適切にインストールされていること。

2. 利用方法

次のステップによって、Ruby/Tk で書かれた GUI アプリケーションをネットワークに公開する。

- i. ローカルのウィンドウシステムで動く Ruby/Tk アプリケーションを用意する。
- ii. `tkorca.rb` を用いてローカルのウィンドウシステム上で実行テストを行う。
- iii. `tkorca_server.rb` や `tkorca_daemon.rb` を用いてネットワークサービスを行う。

3. `tkorca.rb`

Ruby/TkORCA の中心。単独で動かした場合はローカルのウィンドウシステムへの出力となるため、実行テストにも用いることができる。コマンドラインは次の形式である。

```
tkorca.rb [ ...options... ] [--] [ ...applications... ]
```

“--” はアプリケーションが “-” で始まらない場合は省略できる。

オプションの一部を次に示す (他のオプションについては `tkorca/opt_parse.rb` を参照)。

- D, --trust_mode : safe Tk インタープリタを用いない
- A, --auth_check : login ダイアログを表示する (添付の `sysconf/authority_check.rb` は試験用のダミーであり、実働には同ファイル中の `AUTH_PROC` を適切な認証処理に置き換える必要がある)
- B, --broadband : 通信帯域が広い場合の動作モードを用いる
- I, --interactive : IRB 相当のインタラクティブ操作を有効にする
- w size, --width size : 画面幅を size に設定する (デフォルトは 800)
- h size, --height size : 画面高さを size に設定する (デフォルトは 600)
- W size, --vroot_width size : 仮想画面幅を size に設定する (デフォルトは 1600)
- H size, --vroot_height size : 仮想画面高さを size に設定する (デフォルトは 1200)
- T sec, --timeout sec : 連続利用可能時間を sec 秒に制限する
- m file, --mother_rcfile file : 起動時に mother で実行するファイルを指定する

アプリケーション引数は次の形式を取り、指定ごとに一つの daughter が起動する。

```
app_path[, [trusted_ip][, [safe_level][, [use_console][, [init_script]]]]]
```

- app_path : 起動するアプリケーションのパス。もしフルパスでなければ、インストールされた Ruby/TkORCA の `scripts` ディレクトリからの相対パスと見做される。
- trusted_ip : “true” または “false”。true の場合は safe Tk インタープリタを使わない。デフォルトは false。
- safe_level : アプリケーションを実行する sandbox 内での Ruby のセーフレベル値。デフォルトは 4 (最も制約が厳しい)。
- use_console : “true” または “false”。標準入出力を繋ぐ疑似コンソールを利用するかどうかを指定する。ただし現在のバージョンでは、出力は可能だが入力としてはまだ不完全である。デフォルトは false。
- init_script : アプリケーション起動時に実行されるスクリプトの指定。“{ ... }” というように波括弧で囲まれている場合はそれがスクリプトを直接記述したものと見做し、そうでない場合はスクリプトファイルを指定したものと見做す。

もしアプリケーションが一つも指定されなかった場合には、いくつかのアプリケーション候補を表示し、利用者にその中から一つを選択してもらうように起動する。こうしたアプリケーション選択動作は、`scripts/application_selector.rb` で規定しているので、これを置き換えることで動作仕様を変更することができる。

4. tkorca_server.rb

接続要求に対して tkorca.rb を起動してサービスを提供するサーバスクリプト。コマンドラインは次の形式である。

```
tkorca_server.rb [ ...options... ] [--] [ ...tkorca_options... ]
```

“*tkorca options*” には、起動する tkorca.rb に渡すコマンドラインオプションをそのまま書く。

もし “*tkorca options*” がアプリケーション引数のみで tkorca.rb のオプション(“-” で始まるもの)を全く含まないのであれば “-” は省略することができる。

オプションの一部を次に示す(他のオプションについては server/opt_parse.rb を参照)。

```
-L, --reverse_lookup : 接続相手の IP アドレスからの逆引きを行なう
-p port, --port port : サーバの受け付けポートを port に設定する(デフォルトは 5940)
-c port, --control_port port : 制御用接続ポートを port に設定する(デフォルトは 5941)
-C path, --control_passwd path : 制御用接続ポートへの接続認証用パスワードファイル
-b path, --ruby_bin path : Ruby バイナリのフルパス
-a path, --xauth_path path : xauth コマンドのフルパス
-x path, --xvnc_path path : Xvnc コマンドのフルパス
```

5. tkorca_daemon.rb

デーモンとして動作するために必要な設定をした後、tkorca_server.rb を起動するためのスクリプト。コマンドラインで与えられた引数は、そのまま tkorca_server.rb に渡される。

6. 管理ツール

6.1 tkorca_logrotate.rb

ログファイルの名前を変更し、新しいログファイルに切替える。

6.2 tkorca_shutdown_server.rb

稼働中の Ruby/TkORCA デーモンを停止する。

6.3 server/controller_passwd.rb

Ruby/TkORCA サーバのコントロールポートに接続する際の認証用パスワードの生成・管理を行う。

6.4 telnet を使った接続

Ruby/TkORCA サーバのコントロールポートに接続し、制御を行うことができる。利用可能なコマンドは “?” を入力することで表示される。

“connect” コマンドを使うことで、稼働中のアプリケーションサービスに接続して直接にコントロールすることができる。この状態は、tkorca.rb でのインタラクティブ操作そのものである。

7. 補足

Xvnc を X サーバとして見た場合、近年の X サーバがディスプレイカードの機能を活かしてサポートしているような拡張モジュールはほぼサポートしていない。そのため、OS にデフォルトでインストールされている X 関係のライブラリを使用したのではうまく動かず、互換ライブラリのインストール等を必要とする場合がある。例えば、OpenGL を使っているアプリケーションでは、X の拡張モジュールを使わないように make された Mesa (OpenGL 互換ライブラリ)が必要となるだろう。

Ruby/TkORCA 自体には secure な通信を行う仕組みは組み込んでいないが、通信経路を secure にしたい場合は、stunel などを用いて RFB プロトコルによる通信を SSL または SSH で wrap することで容易に実現できる (SSL 受け付けポートから Ruby/TkORCA サーバの受け付けポートにフォワードするように設定して稼働させておけばよい)。VNC クライアントには VNC over SSL に対応したものが存在するので、そうしたクライアントを使えば楽であろう。