# OpenRTM-aist (Java) - 機能 #3385

## ①　コンポーネント操作関数セットの実装

2015/12/11 01:47 - n-ando

| | | | |
|---|---|---|---|
| **ステータス:** | 担当 | **開始日:** | 2015/12/11 |
| **優先度:** | 通常 | **期日:** | |
| **担当者:** | win-ei | **進捗率:** | 100% |
| **カテゴリ:** | | **予定工数:** | 0.00時間 |
| **対象バージョン:** | RELEASE_1_2_0 | | |

**説明**

Rtshell[3]ライクに、コンポーネントの各種操作を簡単に行うことができる関数セットを実装すること。
[1]　rtshell: http://openrtm.org/openrtm/ja/node/869

### 関係しているリビジョン

**リビジョン 782 - 2016/03/01 17:13 - win-ei**

Added CORBA_RTCUtil. refs #3385

**リビジョン 783 - 2016/03/01 17:24 - win-ei**

Added an unit test file for CORBA_RTCUtil class. refs #3385

**リビジョン 784 - 2016/03/01 17:55 - win-ei**

Fixed build.xml for unit tests.. refs #3385

**リビジョン 785 - 2016/03/01 20:08 - win-ei**

Adjusting function names and interfaces. refs #3385

**リビジョン 786 - 2016/03/02 13:10 - win-ei**

implemented methods of CORBA_RTCUtil class. refs #3385

**リビジョン 787 - 2016/03/02 19:31 - win-ei**

Fix bugs of CORBA_RTCUtil class. refs #3385

**リビジョン 1031 - 2017/08/30 15:00 - t-katami**

[compat,bugfix,->releng_1_2] fixed the bug. refs #3385

### 履歴

**#1 - 2016/02/18 12:04 - n-ando**

以下の仕様でお願いします。

```
 *  @brief   RTC操作ユーティリティー関数群
 *
 *  RTC操作ユーティリティー関数群
 *  ============================
 *
 *  CORBA_RTCUtilではRTC操作のための種々のインターフェースを提供する
 *
 *  RTC基本操作系
 *  -------------
 *  -   get_component_profile(rtc):
 *          RTCのProfileを取得
 *  -   is_existing(rtc):
 *          RTCが存在しているかを確認
 *  -   is_alive_in_default_ec(rtc):
 *          RTCがデフォルトの実行コンテキストでalive状態かを判定する
 *
 *  ECおよび状態操作系
 *  ------------------
 *  -   get_actual_ec(rtc,   ec_id   =   0):
```

```
 *          RTコンポーネントに関連付けした実行コンテキストから指定したIDの
 *          実行コンテキストを取得
 *  -  get_ec_id(rtc,  ec):
 *          対象のRTコンポーネントから指定した実行コンテキストのIDを取得す
 *          る
 *  -  activate(rtc,  ec_id  =  0):
 *          RTCを指定した実行コンテキストでアクティブ化する
 *  -  deactivate(rtc,  ec_id  =  0):
 *          RTCを指定した実行コンテキストで非アクティブ化する
 *  -  reset(rtc,  ec_id  =  0):
 *          RTCを指定した実行コンテキストでリセットする
 *  -  get_state(rtc,  ec_id=0,  rtc  =  []):
 *          対象のRTコンポーネントの指定した実行コンテキストでの状態を取得
 *  -  is_in_inactive(rtc,  ec_id  =  0)
 *          対象のRTコンポーネントの指定した実行コンテキストでINACTIVE状態
 *          かどうか判定
 *  -  is_in_active(rtc,  ec_id  =  0):
 *          対象のRTコンポーネントの指定した実行コンテキストでACTIVE状態か
 *          どうか判定
 *  -  is_in_error(rtc,  ec_id  =  0)
 *          対象のRTコンポーネントの指定した実行コンテキストでERROR状態か
 *          どうか判定
 *  -  get_default_rate(rtc):
 *          RTCのデフォルトの実行コンテキストの実行周期を取得する
 *  -  set_default_rate(rtm,  rate):
 *          RTCのデフォルトの実行コンテキストの実行周期を設定する
 *  -  get_current_rate(rtc,  ec_id):
 *          RTCの指定IDの実行コンテキストの周期を取得
 *  -  set_current_rate(rtc,  ec_id,  rate):
 *          RTCの指定IDの実行コンテキストの周期を設定
 *  -  add_rtc_to_default_ec(localcomp,  othercomp):
 *        対象のRTCのデフォルトの実行コンテキストに指定のRTCを関連付ける
 *  -  remove_rtc_to_default_ec(localcomp,  othercomp):
 *        対象のRTCのデフォルトの実行コンテキストの指定のRTCへの関連付け
 *        を解除する
 *  -  get_participants_rtc(rtc):
 *          RTCのデフォルトの実行コンテキストに参加しているRTCのリストを取
 *          得する
 *
 *  ポート操作系
 *  ------------
 *  -  get_port_names(rtc):
 *          指定したRTCの保持するポートの名前を取得
 *  -  get_inport_names(rtc):
 *        指定したRTCの保持するインポートの名前を取得
 *  -  get_outport_names(rtc):
 *          指定したRTCの保持するアウトポートの名前を取得
 *  -  get_svcport_names(rtc):
 *          指定したRTCの保持するサービスポートの名前を取得
 *  -  get_port_by_name(rtc):
 *          対象のRTCから指定した名前のポートを取得
 *  -  get_connector_names(rtc,  port_name):
 *          指定したポートの保持しているコネクタの名前のリストを取得
 *  -  get_connector_ids(rtc,  name):
 *  -  get_connector_ids(port):
 *          指定したポートの保持しているコネクタのIDのリストを取得
 *  -  create_connector(name,  prop_arg,  port0,  port1):
 *          指定したポートを接続するためのコネクタプロファイルを取得
 *  -  already_connected(port0,  port1):
 *          指定したポート同士が接続されているかを判定
 *  -  connect(name,  prop,  port0,  port1):
 *          指定したポートを接続する
 *  -  connect_multi(name,  prop,  port,  target_ports):
 *          指定したポートと指定したリスト内のポート全てと接続する
 *  -  connect_by_name(name,  prop,  rtc0,  portName0,  rtc1,  portName1):
 *          対象のRTCの指定した名前のポートを接続する
 *  -  disconnect(connector_prof):
 *
 *      disconnect_by_portref_connector_id(port_ref,  conn_id)
 *      disconnect_by_portref_connector_name(port_ref,  conn_name)
 *      disconnect_by_portname_connector_id(port_ref,  conn_id)
 *      disconnect_by_portname_connector_name(port_ref,  conn_name)
 *      disconnect_by_portrefs(port0_ref,  port1_ref)
 *      disconnect_all_by_ref(port_ref)
 *      disconnect_all_by_name("rtcloc://hostname/ConsoleIn.out")
```

```
    *
    *           指定のコネクタを切断する
    *  -  disconnect_by_connector_name(port,  name):
    *           対象のポートで指定した名前のコネクタを切断
    *  -  disconnect_by_connector_id(port,  id):
    *           対象のポートで指定したIDのコネクタを切断
    *  -  disconnect_by_port_name(port0,  port1):
    *           対象ポートと接続しているポートで指定したポート名と一致した場合
    *           に切断
    *
    *  コンフィギュレーション系
    *  ------------------------
    *  -  get_configuration(rtc,  conf_name):
    *           指定したRTコンポーネントのコンフィギュレーション取得
    *  -  set_configuration(rtc,  confset_name,  key,  value):
    *           コンフィギュレーションパラメータを設定
    *  -  get_parameter_by_key(rtc,  confset_name,  value_name):
    *           指定したコンフィギュレーションセット名、パラメータ名のコンフィ
    *           ギュレーションパラメータを取得
    *  -  get_active_configuration_name(rtc):
    *           対象のRTCのアクティブなコンフィギュレーションセット名を取得する
    *  -  get_active_configuration(rtc):
    *           アクティブなコンフィギュレーションセットを取得
    *  -  set_active_configuration(rtc,  value_name,  value):
    *           コンフィギュレーションパラメータを設定
```

**#2 - 2016/02/29 15:43 - win-ei**

*- 進捗率 を 0 から 30 に変更*

チェックインしました。
r781
python版をベースに実装しましたので、後で、仕様と整合しているか確認します。

**#3 - 2017/08/30 15:02 - katami**

*- ステータス を 新規 から 担当 に変更*

*- 担当者 を win-ei にセット*

*- 進捗率 を 30 から 100 に変更*

**#4 - 2017/08/31 10:46 - katami**

*- 対象バージョン を RELEASE_1_2_0 にセット*