

# dbrtcbuilder version 1.0.0 ユーザーマニュアル

株式会社 会津ラボ



1. 内容物について	.....	03
2. 概要、動作要件	.....	04
3. スタートガイド	.....	05
4. DBRTC 生成、反映	.....	06
5. オプションについて	.....	07
6. プログラムについて	.....	08
7. 補足説明	.....	09
8. 更新履歴	.....	10

## dbrtcbuilder version 1.0.0

package	dbrtc	dbrtcauxiliary.py	dbrtc実行時のDB用補助モジュール
		dbrtcenv.py	dbrtc環境設定ファイル
		dbrtckv.py	mongodb 用補助モジュール
	dbrtcbuilder	_default.xml	生成時のデフォルト設定ファイル例
		dbrtcbuilder.py	DBRTC 生成本体プログラム
		template_rtc.py	rtc ファイル生成のテンプレート
		template_rtc_conf.py	rtc.conf ファイル生成用テンプレート
		template_user_conf.py	user.conf ファイル生成用テンプレート
	man	dbrtc_man.pdf	このファイル

## 概要

dbrtcbuilder とは、データベース (DB) への書き込み、読み込みを行うRTCを生成するツール群です。

ここでは、ツールより生成された RTC を DBRTCと呼ぶことにします。

(生成ファイル名は、ユーザー様が自由に設定できます。)

ユーザー様は、自分の RTC と dbrtcbuilder で生成された DBRTC とを接続し、DB へのアクセスを実現できます。

## 動作要件

OpenRTM-aist version 1.1.2 の動作環境同様

対象DB : MongoDB (3.2 系)

Python 2.7 系

pymongo, json, time, datetime を利用

## 手順

- ①データ送信用RTCを作成します。（通常、RTC Builder利用）
- ②dbrtcbuilder.py を用いてDBアクセス用RTC（DBRTC）を作成します。

①で作成された RTC.xml を入力ファイルとして渡します。

出力フォルダは予め作成しておきます。

例) -i : 入力ファイル / -o : 出力ファイル

```
> python dbrtcbuilder.py -i c:¥input¥RTC.xml -o c:¥output¥dbrtc.py
```

- ③dbrtcフォルダ内の3ファイル(dbrtcauxiliary.py, dbrtcenv.py, dbrtckv.py )  
を出力フォルダにコピーする等して同じフォルダ内に配置します。

- ④dbrtcenv.pyを環境にあわせ編集します。(L15 - L18)

```
DB_HOST = 'localhost'
```

```
DB_PORT = 27017
```

```
DB_DBNAME = 'DBNAME'
```

```
DB_COLLECTIONNAME = 'COLLECTIONNAME'
```

- ⑤RTSystemEditorを起動します。

送信用RTC、DBRTCを起動し、接続、Activate、DBアクセスを実現します。

## DBRTC生成について

DBRTC は `dbrtcbuilder.py` に `RTC.xml` 指定することで生成されます。

`RTC.xml` は通常 `rtcbuilder` プロジェクト内に作成されています。

`dbrtcbuilder.py` は Python ファイルです。

Python を用いて、**ターミナル**、または**コマンドプロンプト**にて実行します。

作成例 ( `RTC.xml` を入力ファイルとし、`DBRTC.py` ファイルを出力する例)

```
> python dbrtcbuilder.py -i c:¥test¥RTC.xml -o c:¥test¥DBRTC.py
```

出力フォルダは予め作成してください。

※ オプションについては、次ページにて紹介します。

## dbrtcbuilder オプションについて

オプション	引数	備考
-i	RTC.xml	入力ファイル、RTC.xmlを指定します。
-o	(DBRTC).py	出力ファイル
-e	エンドポイント	出力RTCのエンドポイントを指定します。 デフォルトは 127.0.0.1 です。
-c	RTC サイクル	RTC の サイクルを指定します。 デフォルトは、2.0 です。
-d	(default).xml	アクセス要素を共通化させる場合、共通化 RTC.xmlとして利用できます。 dbrtcbuilder.py と同階層に配置してください。 デフォルトは、default.xml です。(※補足説明参照) _default.xml を同梱。有効化にはファイル名を変更してください。
-m	w / r	DB へのモード。w=write / r=read 用RTCを生成します。 デフォルトは、 w : 書込用RTCです。
-t	0 / 1	DB書込時、Inport.read() を発生する条件を指定できます。 0 : isNew() 、 1 : isEmpty() を判定しDBへ書き込みます。 csv 等より、DB へ書き込む列を揃えたい場合、1 を指定してください。 ただし、送信側で接続全ポートにデータがあることを保証してください。 デフォルトは、0 : isNew() を用いて判断します。

## DBRTCの動作について

### 出力ファイル群

-o オプションにより、DBRTC として 3ファイル生成させます。

- ・ (DBRTC).py : DBRTC 本体ファイル
- ・ rtc.conf : rtc コンフィグファイル
- ・ (DBRTC).conf : DBRTC コンフィグファイル

デフォルトは、サーバー配置前提のコンフィグファイルとなっています。

ローカルでご利用の場合は適宜修正してください。

### DBRTC 実行には

dbrtc フォルダ以下に、dbrtcauxiliary.py、dbrtcenv.py、dbrtckv.py ファイルがあります。

DB へアクセスするために必要なファイル群になります。

出力ファイル群と上記 3 ファイルを同じフォルダ内へ配置してください。

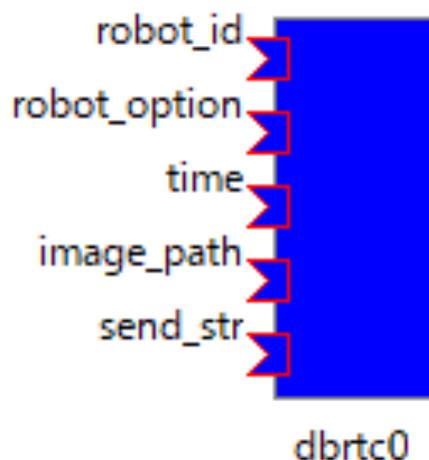
### dbrtcenv.py について

DBRTC 動作定義があります。

ローカルでご利用する場合は、適宜修正してください。

このファイルに検索条件を指定すればDBから検索ができます。

## default.xml について



default.xml は以下のポートを追加する設定が記載されています。

robot\_id : ロボットID、RTC::TimedString 型

robot\_option : ロボットオプション、RTC::TimedString 型

time : クライアント実行時時間、RTC::TimedString 型

image\_path : 関連イメージのパス指定、RTC::TimedString 型

出力RTC(DBRTC) に上記ポートが不要な場合、default.xml をリネームまたは削除してください。  
ユーザー様で別途 default.xml を用意されましたら、そのポート情報が反映されます。

## server\_time について

DB へ書込時、サーバー側の時間のデータが付加されます。(version 0.5.0 ~)

server\_time : DB への書込時間、RTC::TimedString 型

DB検索時に利用する等の用途に利用します。

## バージョンを確認する方法

dbrtcbuilder.py を引数無しで実行すると、バージョンが確認できます。

### version 1.0.0

- ・公開用にコード修正、マニュアル修正

### version 0.7.0

- ・著作権表記の修正
- ・DBからのリードRTCは件数の処理が終了したら、Deactivate へ移行するように変更

### version 0.6.0

- ・下記オプション追加、不具合修正、CSV 等からの書込時用モード追加(-t 同義)等
- ・DB からの読み込みモード追加、読み込み用RTC生成 (-m)
- ・DB 書き込み時に、Inport における判定関数を選択できるように修正 (-t)
- ・default.xml が存在する時、共通内容のポートを作成するように変更 (-d)

### version 0.5.X

- ・conf ファイルを出力するように変更。ユーザー様の設定にて反映できるよう修正

### version 0.1.0

- ・DBRTCをサーバー側配置し、DB 接続できるよう更新
- ・DB 環境を設定できるように更新

### version 0.0.1

- ・ファーストバージョン
- ・ローカルから直接DBへ接続する構造