

次世代ロボット知能化技術開発プロジェクト
作業知能（社会・生活分野）の開発

操作手順書
タウンマネジメントシステム RTC

V e r . 0.40

2011年12月1日

九州大学

改版履歷

[illegible]

目次

改版履歴	i
目次	ii
1. はじめに	1
1. 1. 本書の適用範囲	1
1. 2. 関連文書	1
1. 3. 本書を読むにあたって	1
1. 4. 動作環境	2
2. ディレクトリ構成	3
3. ソフトウェアインストール	4
3. 1. 基本環境	4
4. 実行	6
4. 1. ネームサーバの起動	6
4. 2. RTSystemEditor の起動	6
4. 3. TmsRtProvider の起動	7
4. 4. TmsRtConsumer の起動	8
4. 5. ログイン ID の設定	9
4. 6. 接続	10
4. 7. 活性化	11
4. 8. 実行	11
4. 9. 終了手順	14
5. トラブルシューティング	15
6. 特記事項	16

1. はじめに

1. 1. 本書の適用範囲

本書はタウンマネジメントシステム RTC を、産業技術総合研究所が開発した RT ミドルウェア OpenRTM-aist を用いて動作させるための手順について記述している。

1. 2. 関連文書

本書の関連文書は下表の通り。

No.	文書名	備考
1	タウンマネジメントシステム RTC 機能仕様書	タウンマネジメントシステム RTC の 仕様について記載。

1. 3. 本書を読むにあたって

本書は RT ミドルウェア、RT コンポーネント(以下、RTC)、およびタウンマネジメントシステム(以下、TMS)に関する基本知識を備えた利用者を対象としている。RT ミドルウェア、RTC、TMS については下記を参照のこと。

OpenRTM-aist Official Website:

<http://www.openrtm.org/>

タウンマネジメントシステム Website:

http://openrtm.org/openrtm/ja/project/NEDO_Intelligent_PRJ_ID121

1. 4. 動作環境

検証に用いた動作環境は以下のとおりである。

OS	Windows XP Professional SP3
RT ミドルウェア	OpenRTM-aist-1.0.0 (JAVA)
開発言語	JAVA
依存ライブラリ (OpenRTM)	JDK 6
依存ライブラリ (その他)	TMS Server (タウンマネジメントシステム Website より取得)
	TMS API (タウンマネジメントシステム Website より取得)

2. ディレクトリ構成

本書は下表のディレクトリ構成のもとに記述されている。ダウンロードしたタウンマネジメントシステム RTC を解凍し、同じ構成となるよう配置しておく。ディレクトリ構成が下表と異なる場合は、適宜その環境に合わせて読み替える必要がある。

ディレクトリ	言語	内容	備考
C:\¥ TMS2011_bin	-	-	-
└─TmsRtConsumer	-		-
└─TmsRtProvider			

3. ソフトウェアインストール

動作に必要なソフトウェアを以下に記す。

3. 1. 基本環境

3. 1. 1. OpenRTM-aist-1.0.0 (JAVA)

JAVA 言語で実装された RTC を実行するための RT ミドルウェアをインストールする。インストールの詳細は、以下のサイトを参照のこと


- ・ダウンロード先 URL : OpenRTM-aist-1.0.0 (JAVA)

<http://www.openrtm.org/pub/Windows/OpenRTM-aist/java/OpenRTM-aist-Java-1.0.0.msi>

- ・ダウンロード先 URL : JDK6

<http://www.oracle.com/technetwork/java/javase/downloads/index.html>

JDK6 を先にインストールしてください。

インストーラ	インストールされるプログラム
OpenRTM-aist-Java-1.0.0.msi 	OpenRTM-aist-1.0.0 (JAVA)
JDK6: Java SE Development Kit Web ページ	JDK6: Java SE Development Kit

3. 1. 2. TMS Server

タウンマネジメントシステムのサーバ機能を提供するためのソフトウェアをインストールする。同梱されているインストーラを用いる。「TMS サーバ導入マニュアル.pdf」を参照すること。

インストーラ	インストールされるプログラム
TmsServerLibSetup.exe	TMS Server

3. 1. 3. TMS API

タウンマネジメントシステムが提供する API を利用するためのソフトウェアをインストールする。同梱されているインストーラを用いる。「TMS API 導入マニュアル.pdf」を参照すること。

インストーラ	インストールされるプログラム
TmsApiSetup.exe	TMS API

3. 1. 4. タウンマネジメントシステム RTC

タウンマネジメントシステム RTC を、2 章で示したディレクトリ構成と同じ構成となるよう配置する。

4. 実行

本章では、タウンマネジメントシステムの実行手順を示す。なお、TMS Server は起動しているとする。

4. 1. ネームサーバの起動

OpenRTM-aist の “start-orbd.vbs ” を実行してネームサーバを起動する。

4. 2. RTSystemEditor の起動

OpenRTM-aist の、RTSystemEditor を起動する。RTSystemEditor の NameServiceView でネームサーバが起動していることを確認する（図 5. 1）。RTSystemEditor の使用法は以下のサイトを参照のこと。

RTSystemEditor マニュアル URL :

<http://www.openrtm.org/openrtm/ja/node/1175>

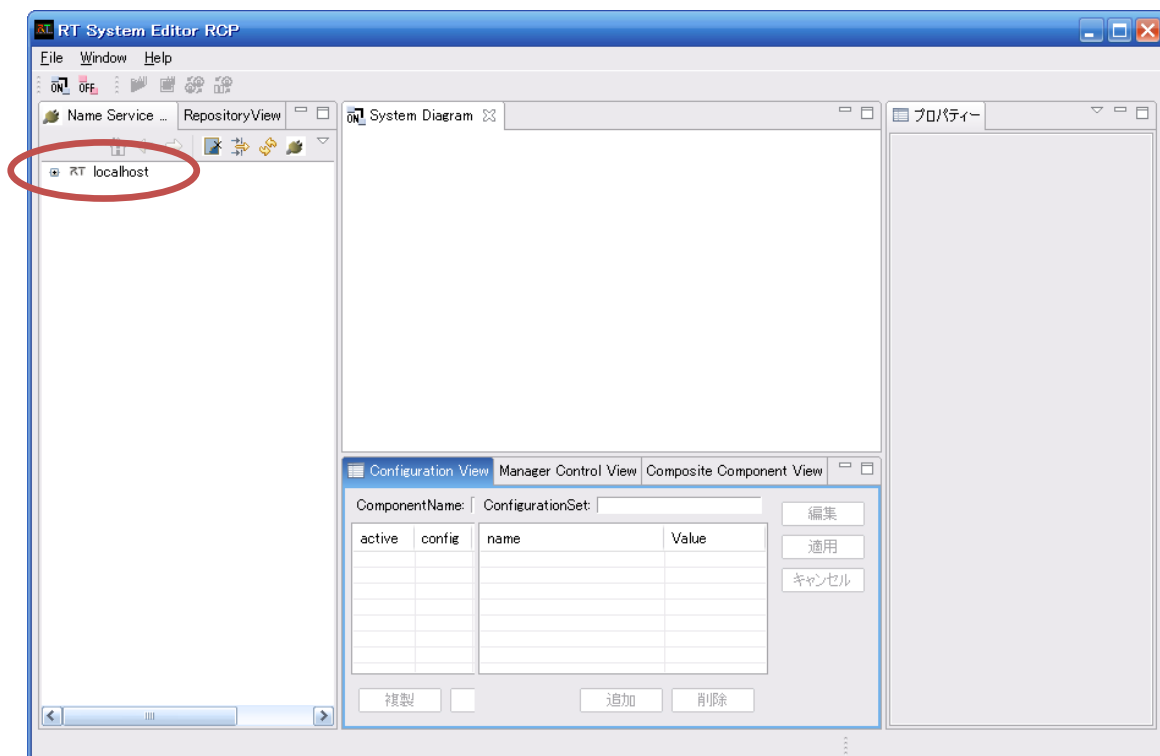


図 5. 1 RTSystemEditor の起動

4. 3. TmsRtProvider の起動

以下のバッチファイルを実行して TmsRtProvider を起動する。

```
C:\¥TMS2011_bin¥TmsRtProvider¥ TmsRtProviderComp.bat
```

起動後に図 5. 2 のように Eclipse の NameServiceView に、TmsRtProvider が登録されていることを確認する。

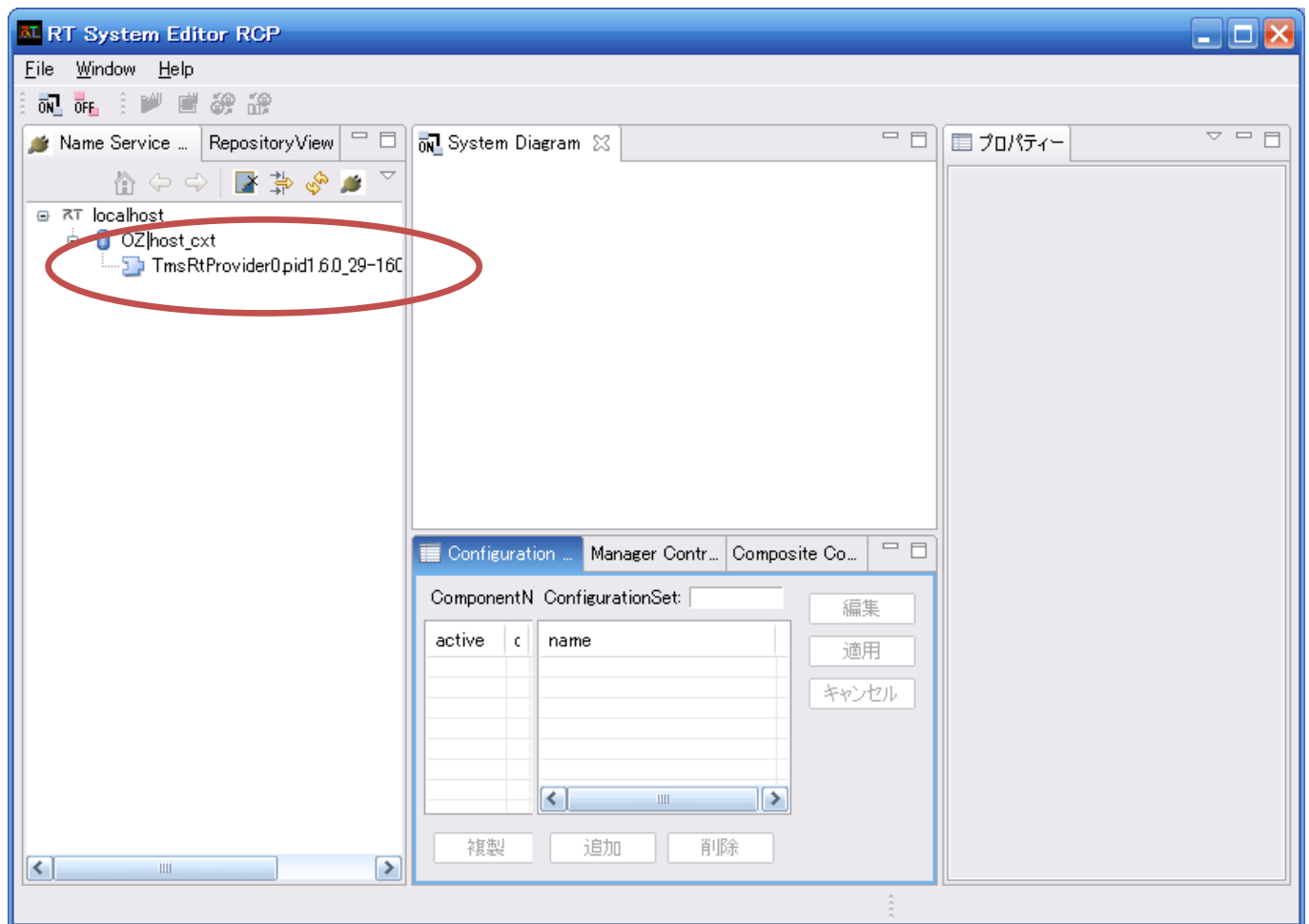


図 5. 2 TmsRtProvider の起動

4. 4. TmsRtConsumer の起動

以下のバッチファイルを実行して TmsRtConsumer を起動する。

```
C:\¥TMS2011_bin¥TmsRtConsumer¥Demo¥TmsRtConsumerComp.bat
```

起動後に図 5. 3 のように Eclipse の NameServiceView に、TmsRtConsumer が登録されていることを確認する。

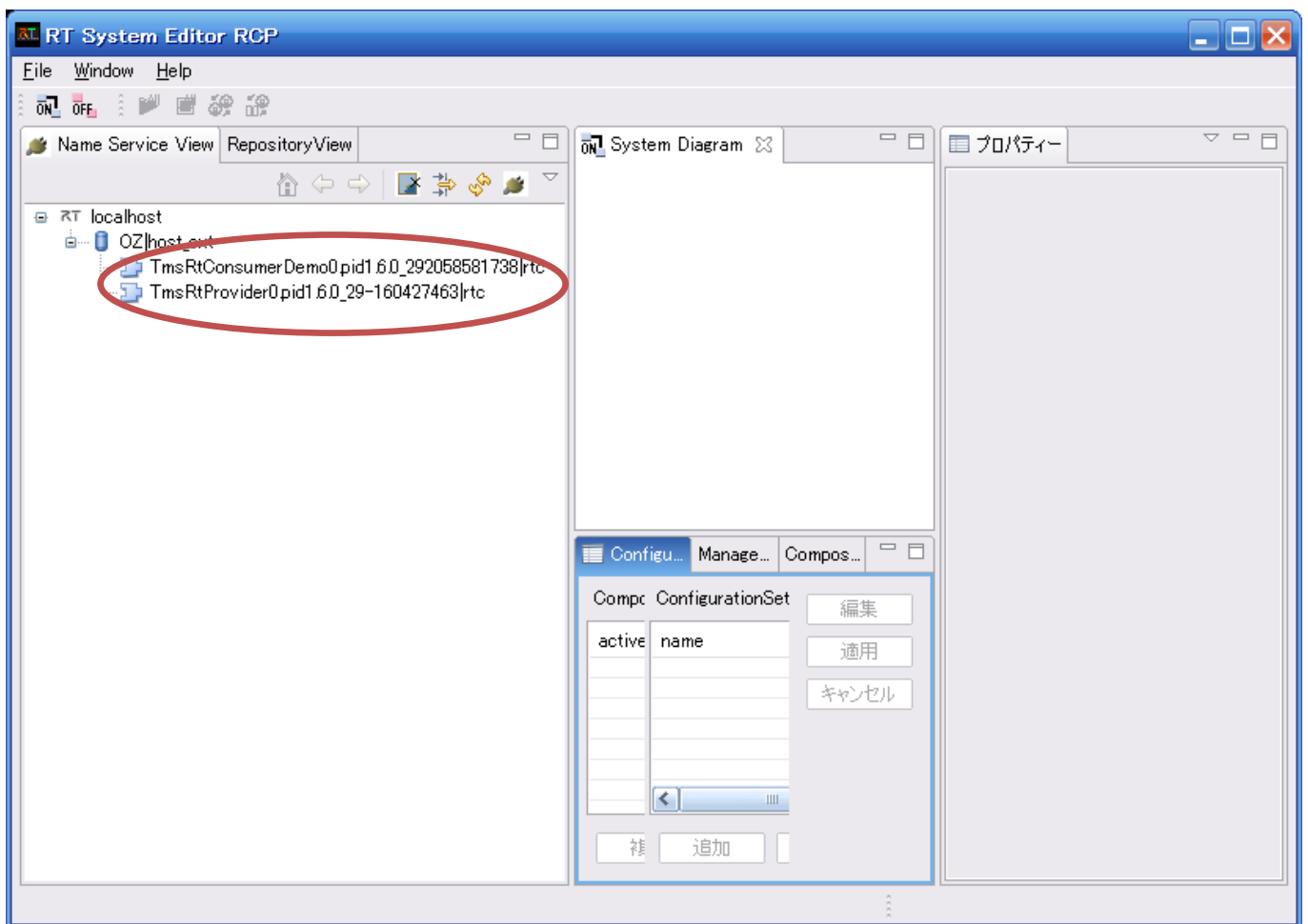


図 5. 3 TmsRtConsumer 起動

4. 5. ログイン ID の設定

TMS Server へログインする際のデバイス ID、ユーザ ID、パスワードをデフォルト値から変更する場合は、TmsRtConsumer のコンフィギュレーションで設定する (図 5. 4)。ログインするためには設定 ID がタウンマネジメントシステムに登録されている必要がある。

名称	型	デフォルト値	説明
deviceid	string	robot1	デバイス ID
userid	string	user1	ユーザ ID
password	string	abc	パスワード

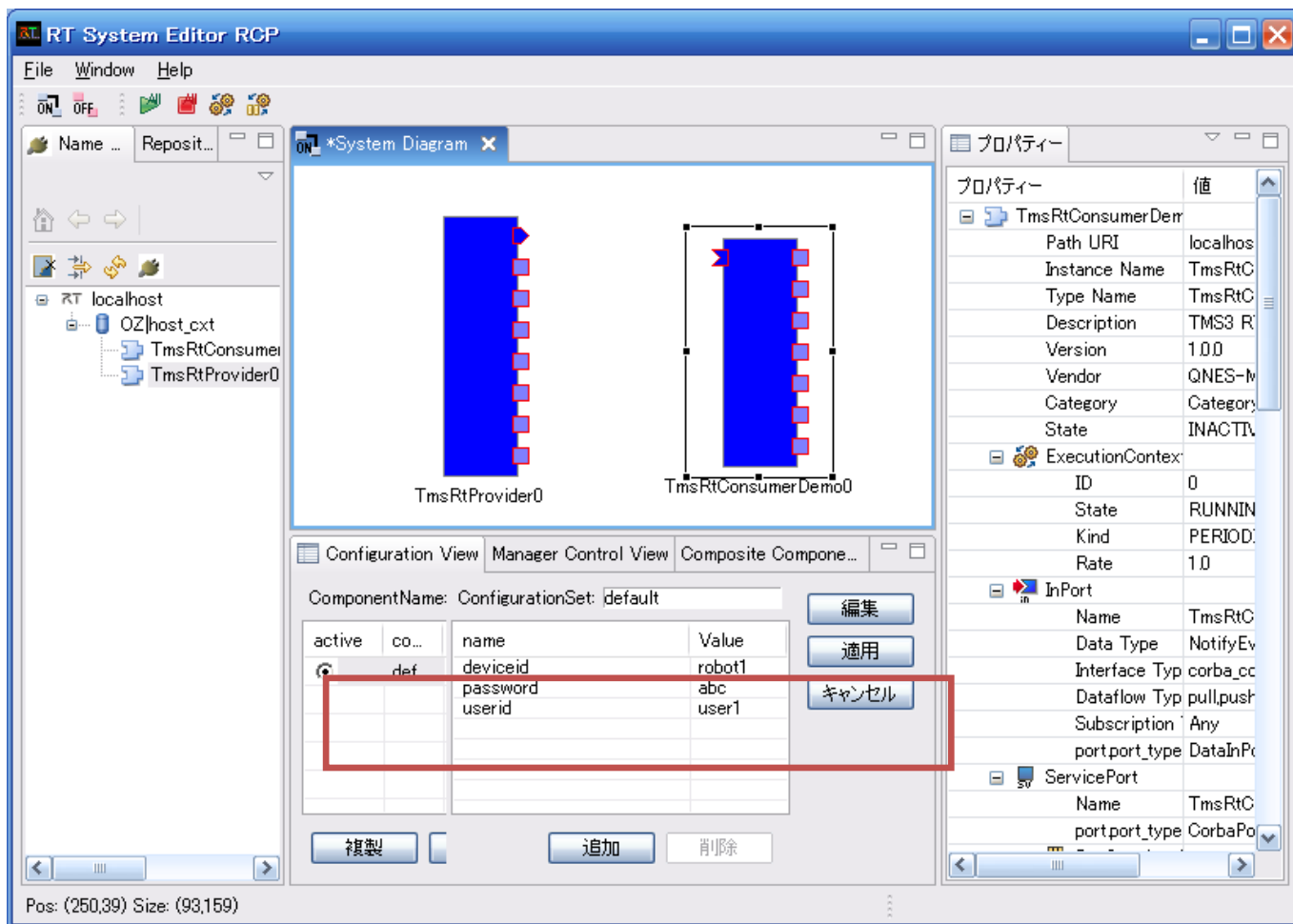


図 5. 4 コンフィギュレーションの設定

4. 6. 接続

RTSystemEditor を用いて TmsRtProvider と TmsRtConsumer を接続する。下図を参考にコンポーネントのポートを接続する。ポートの詳細は「タウンマネジメントシステム RTC 機能仕様書」を参照のこと。

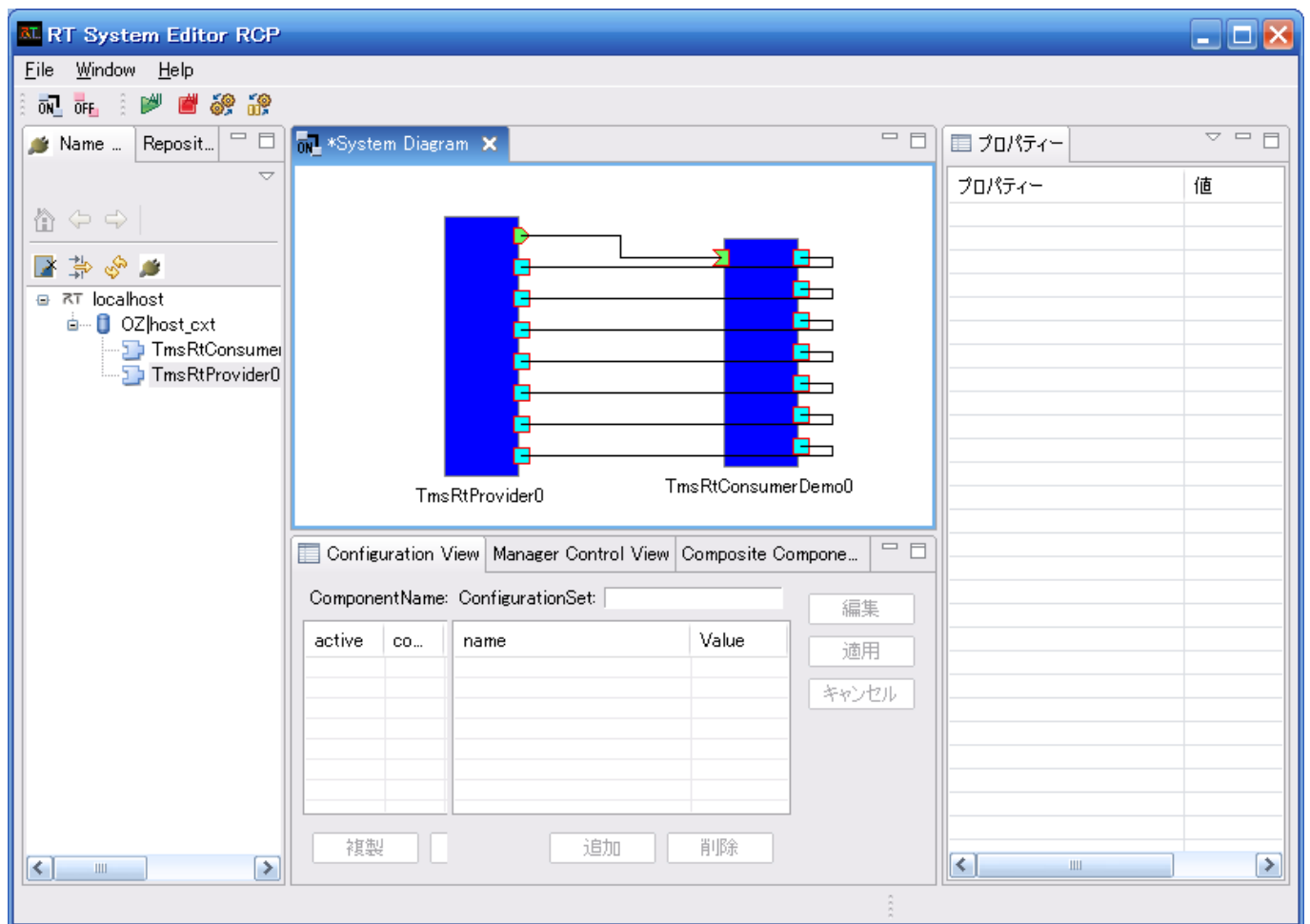


図 5. 5 RTC の接続

4. 7. 活性化

接続が完了したら、下記の順に両 RTC の活性化を行う。RTSystemEditor 上で RTC を選択し、サブメニューから”activate”を選択することで活性化する。両 RTC が活性化された状態を下図に示す。

1. TmsRtProvider
2. TmsRtConsumer

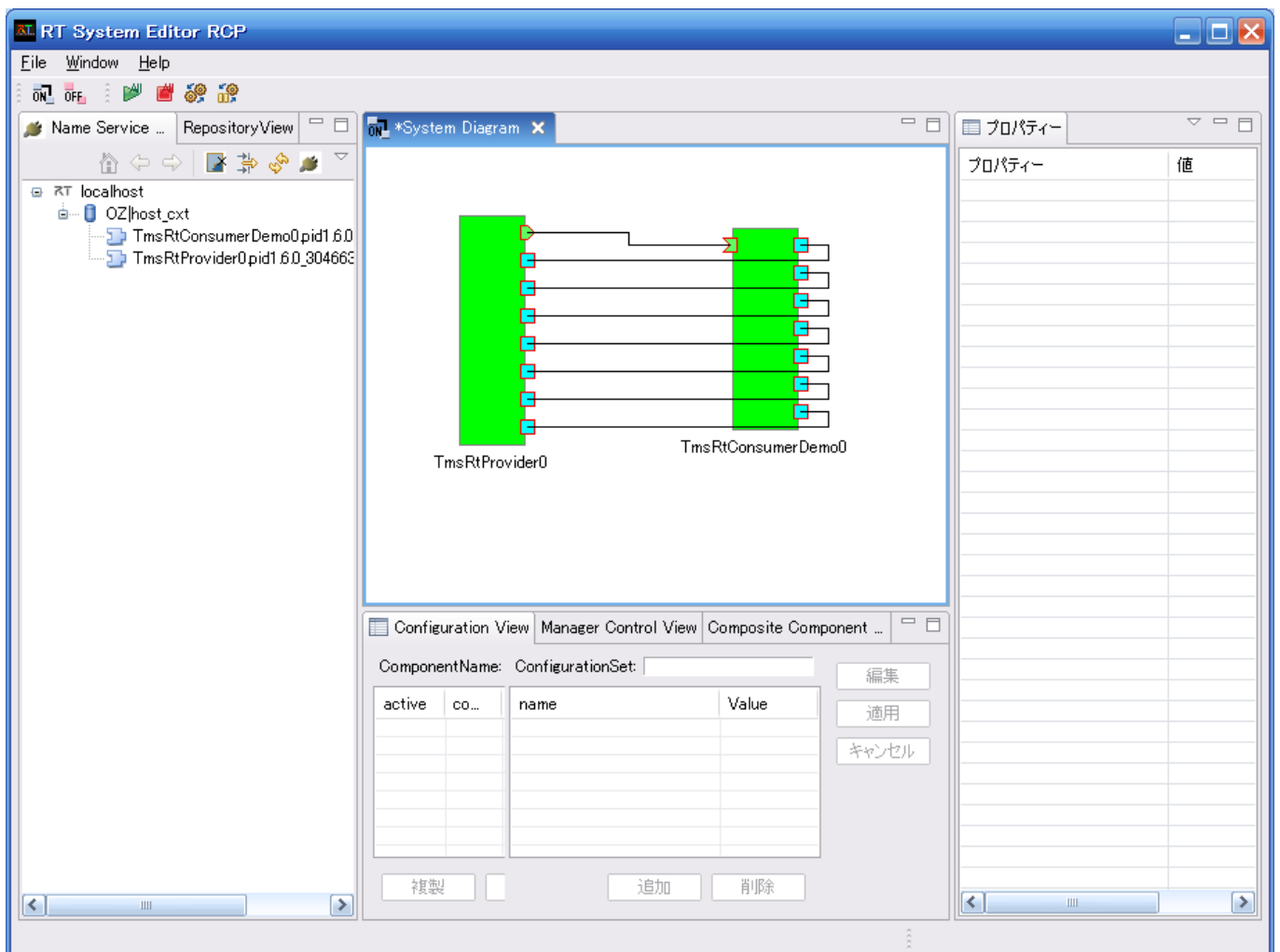
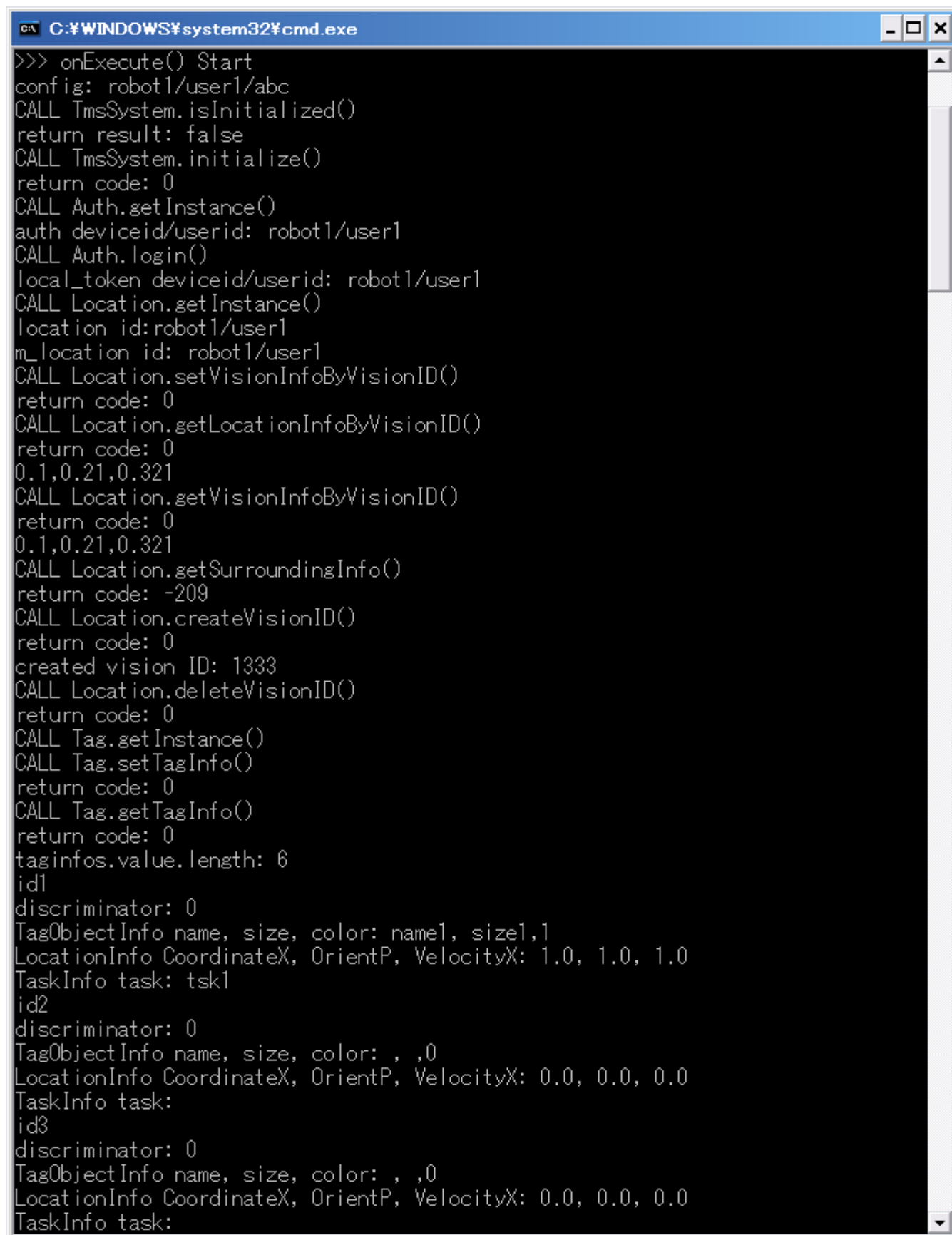


図 5. 6 RTC の活性化

4. 8. 実行

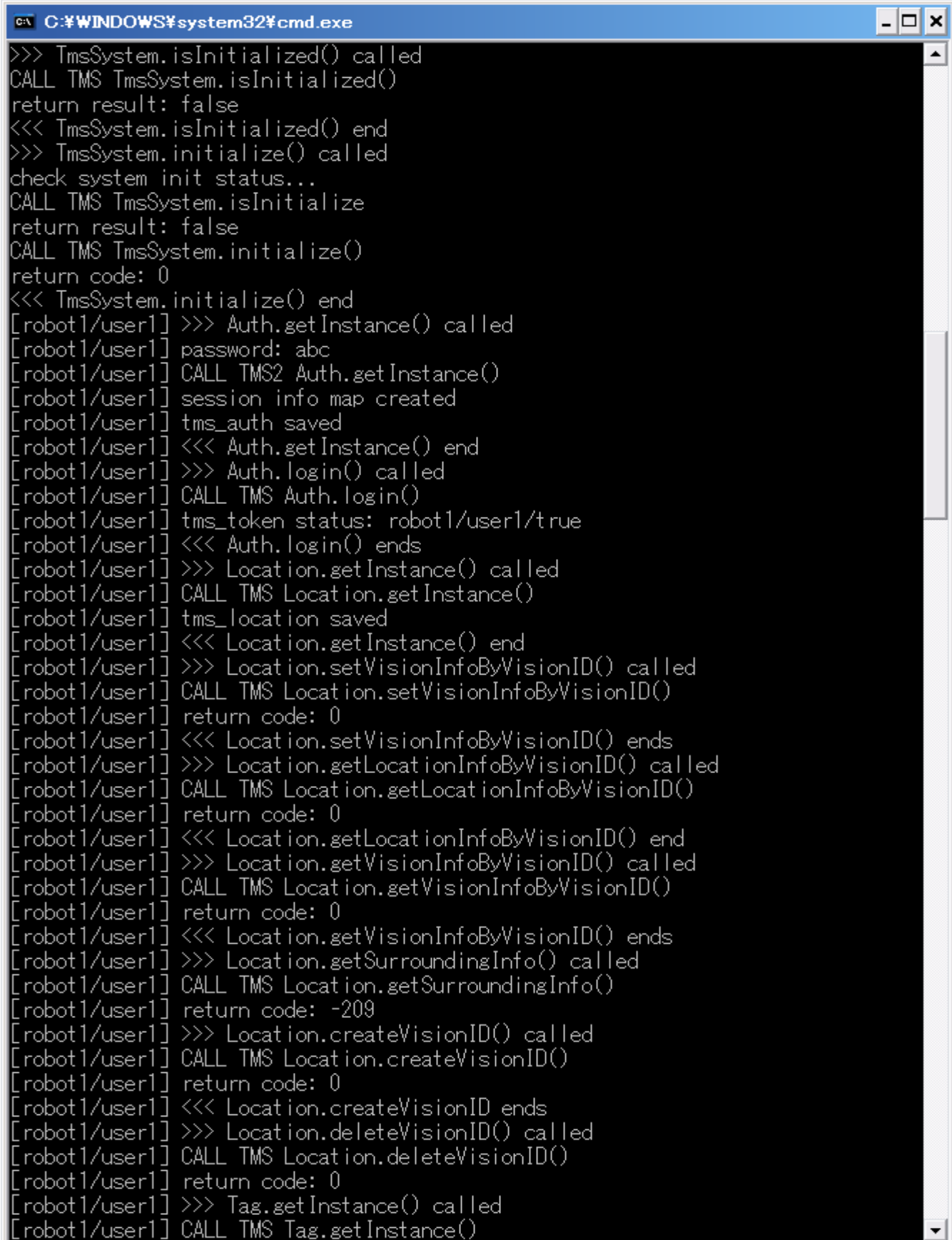
TmsRtConsumer は活性化された後、TmsRtProvider にログイン、情報取得、ログアウトを繰り返す。

返し実行する。TmsRtConsumer の実行画面では、TMS Server の DB から取得した情報を表示する (図 5. 7)。また、TmsRtProvider の実行画面では、TmsRtConsumer からの実行要求内容を表示する (図 5. 8)。



```
>>> onExecute() Start
config: robot1/user1/abc
CALL TmsSystem.isInitialized()
return result: false
CALL TmsSystem.initialize()
return code: 0
CALL Auth.getInstance()
auth deviceid/userid: robot1/user1
CALL Auth.login()
local_token deviceid/userid: robot1/user1
CALL Location.getInstance()
location id:robot1/user1
m_location id: robot1/user1
CALL Location.setVisionInfoByVisionID()
return code: 0
CALL Location.getLocationInfoByVisionID()
return code: 0
0.1,0.21,0.321
CALL Location.getVisionInfoByVisionID()
return code: 0
0.1,0.21,0.321
CALL Location.getSurroundingInfo()
return code: -209
CALL Location.createVisionID()
return code: 0
created vision ID: 1333
CALL Location.deleteVisionID()
return code: 0
CALL Tag.getInstance()
CALL Tag.setTagInfo()
return code: 0
CALL Tag.getTagInfo()
return code: 0
taginfos.value.length: 6
id1
discriminator: 0
TagObjectInfo name, size, color: name1, size1,1
LocationInfo CoordinateX, OrientP, VelocityX: 1.0, 1.0, 1.0
TaskInfo task: tsk1
id2
discriminator: 0
TagObjectInfo name, size, color: , ,0
LocationInfo CoordinateX, OrientP, VelocityX: 0.0, 0.0, 0.0
TaskInfo task:
id3
discriminator: 0
TagObjectInfo name, size, color: , ,0
LocationInfo CoordinateX, OrientP, VelocityX: 0.0, 0.0, 0.0
TaskInfo task:
```

図 5. 7 TmsRtConsumer の実行画面



```
C:\WINDOWS\system32\cmd.exe
>>> TmsSystem.isInitialized() called
CALL TMS TmsSystem.isInitialized()
return result: false
<<< TmsSystem.isInitialized() end
>>> TmsSystem.initialize() called
check system init status...
CALL TMS TmsSystem.isInitialize
return result: false
CALL TMS TmsSystem.initialize()
return code: 0
<<< TmsSystem.initialize() end
[robot1/user1] >>> Auth.getInstance() called
[robot1/user1] password: abc
[robot1/user1] CALL TMS2 Auth.getInstance()
[robot1/user1] session info map created
[robot1/user1] tms_auth saved
[robot1/user1] <<< Auth.getInstance() end
[robot1/user1] >>> Auth.login() called
[robot1/user1] CALL TMS Auth.login()
[robot1/user1] tms_token status: robot1/user1/true
[robot1/user1] <<< Auth.login() ends
[robot1/user1] >>> Location.getInstance() called
[robot1/user1] CALL TMS Location.getInstance()
[robot1/user1] tms_location saved
[robot1/user1] <<< Location.getInstance() end
[robot1/user1] >>> Location.setVisionInfoByVisionID() called
[robot1/user1] CALL TMS Location.setVisionInfoByVisionID()
[robot1/user1] return code: 0
[robot1/user1] <<< Location.setVisionInfoByVisionID() ends
[robot1/user1] >>> Location.getLocationInfoByVisionID() called
[robot1/user1] CALL TMS Location.getLocationInfoByVisionID()
[robot1/user1] return code: 0
[robot1/user1] <<< Location.getLocationInfoByVisionID() end
[robot1/user1] >>> Location.getVisionInfoByVisionID() called
[robot1/user1] CALL TMS Location.getVisionInfoByVisionID()
[robot1/user1] return code: 0
[robot1/user1] <<< Location.getVisionInfoByVisionID() ends
[robot1/user1] >>> Location.getSurroundingInfo() called
[robot1/user1] CALL TMS Location.getSurroundingInfo()
[robot1/user1] return code: -209
[robot1/user1] >>> Location.createVisionID() called
[robot1/user1] CALL TMS Location.createVisionID()
[robot1/user1] return code: 0
[robot1/user1] <<< Location.createVisionID ends
[robot1/user1] >>> Location.deleteVisionID() called
[robot1/user1] CALL TMS Location.deleteVisionID()
[robot1/user1] return code: 0
[robot1/user1] >>> Tag.getInstance() called
[robot1/user1] CALL TMS Tag.getInstance()
```


図 5. 8 TmsRtProvider の実行画面

4. 9. 終了手順

実行を停止する場合は、下記の順に両 RTC の非活性化を行う。RTSystemEditor 上で RTC を選択し、サブメニューから”deactivate”を選択することで非活性化する。

1. TmsRtConsumer
2. TmsRtProvider

5. トラブルシューティング

5 章での実行に問題がある場合は、次章に記述している連絡先まで問い合わせのこと。

6. 特記事項

本モジュールをご利用される場合には、以下の記載事項・条件にご同意いただいたものとします。

- 本モジュールは独立行政法人 新エネルギー・産業技術総合開発機構の「次世代ロボット知能化技術開発プロジェクト」内実施者向けに評価を目的として提供するものであり、商用利用など他の目的で使用することを禁じます。
- ドキュメントに情報を掲載する際には万全を期していますが、それらの情報の正確性またはお客様にとっての有用性等については一切保証いたしません。
- 利用者が本モジュールを利用することにより生じたいかなる損害についても一切責任を負いません。
- 本モジュールの変更、削除等は、原則として利用者への予告なしに行います。また、止むを得ない事由により公開を中断あるいは中止させていただくことがあります。
- 本モジュールの情報の変更、削除、公開の中断、中止により、利用者に生じたいかなる損害についても一切責任を負いません。

【連絡先】

九州大学大学院 システム情報科学研究院 情報知能工学部門 長谷川研究室

〒819-0395 福岡県福岡市西区元岡 744 ウエスト 2 号館 912 室

Tel : 092-802-3598

Fax : 092-802-3607

E-Mail : tms@irvs.ait.kyushu-u.ac.jp