

手順 1 サンプルコンポーネントの実行

目次

1	PC で NameServer と ConsoleIn コンポーネントを起動	2
2	PC に Raspberry Pi を接続する。	3
3	Raspberry Pi に接続する。	4
4	Raspberry Pi で NameServer と ConsoleOut コンポーネントを起動	5
5	PC で RTSystemEditorRCP を起動し各コンポーネントを Active にする。	6
6	ConsoleIn と ConsoleOut が動作することを確認	10
	トラブルシューティング	12
	Raspberry Pi 側のサービスのしたに ConsoleOut が表示されない.....	12
	RTSystemEditor で接続できない、反応がなくなる等.....	12

※ 文中の「x.y」 や「x.y.z」 の表記は使用環境の OpenRTM-aist のバージョンに読み替えてください。

当ドキュメントは下記ページを参考にしています。

- ・開発環境のインストール

http://www.openrtm.org/openrtm/ja/content/raspberrypi_openrtm_installation (2016/1/20 アクセス)

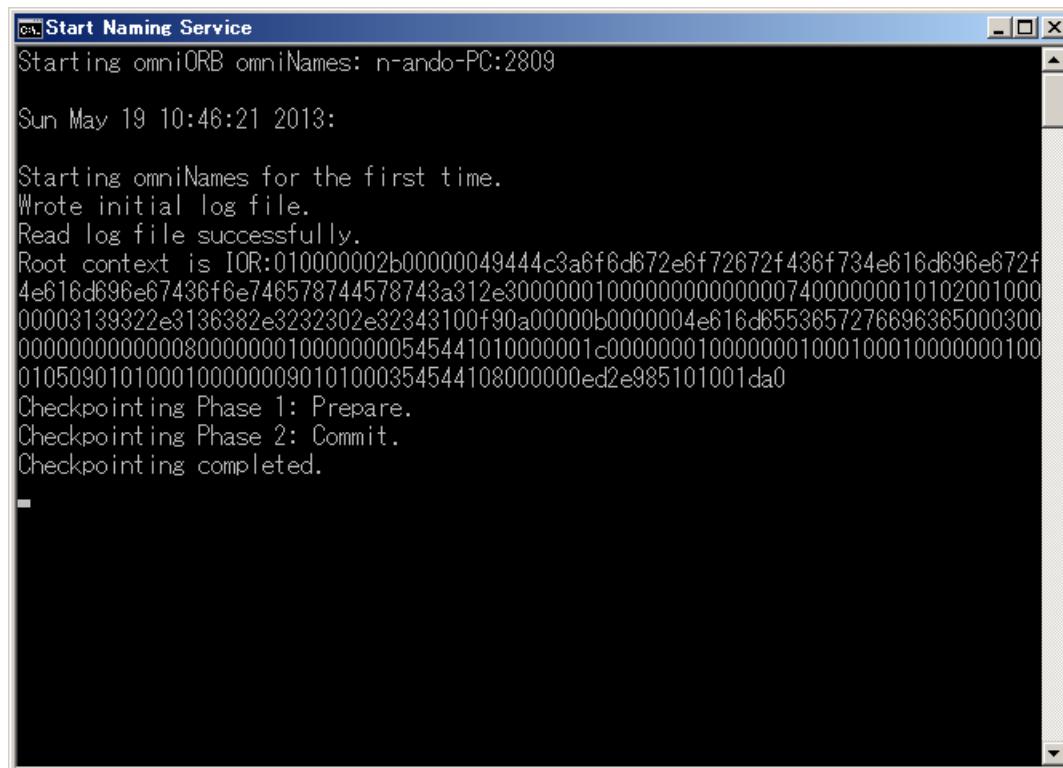
1 PC で NameServer と ConsoleIn コンポーネントを起動

ネームサーバを起動していない場合は、ネームサーバを起動します。スタートメニューから「OpenRTM-aist x.y」→「tools」の下の Start Naming Service から起動します。

※OpenRTM-aist C++ 1.1.1 使用の方は[Start C++ Naming Service]を起動してください。

※Windows8 の場合は以下のパスにあります。

C:\ProgramData\Microsoft\Windows\Start Menu\Programs\OpenRTM-aist x.y\Tools



```
c:\ Start Naming Service
Starting omniORB omniNames: n-ando-PC:2809
Sun May 19 10:46:21 2013:
Starting omniNames for the first time.
Wrote initial log file.
Read log file successfully.
Root context is IOR:010000002b00000049444c3a6f6d672e6f72672f436f734e616d696e672f
4e616d696e67436f6e746578744578743a312e300000010000000000000074000000010102001000
00003139322e3136382e3232302e32343100f90a00000b0000004e616d6553657276696365000300
00000000000000080000000100000000545441010000001c0000000100000001000100010000000100
01050901010001000000090101000354544108000000ed2e985101001da0
Checkpointing Phase 1: Prepare.
Checkpointing Phase 2: Commit.
Checkpointing completed.
```

続いて、コンポーネントの ConsoleIn を起動します。「OpenRTM-aist x.y」→「C++」→「Components」

→「Examples」の下の ConsoleInComp.exe をクリックして ConsoleIn コンポーネントを起動します。

※Windows8 の場合は以下のパスにあります。

C:\ProgramData\Microsoft\Windows\Start Menu\Programs\OpenRTM-aist x.y\C++\Components\Examples

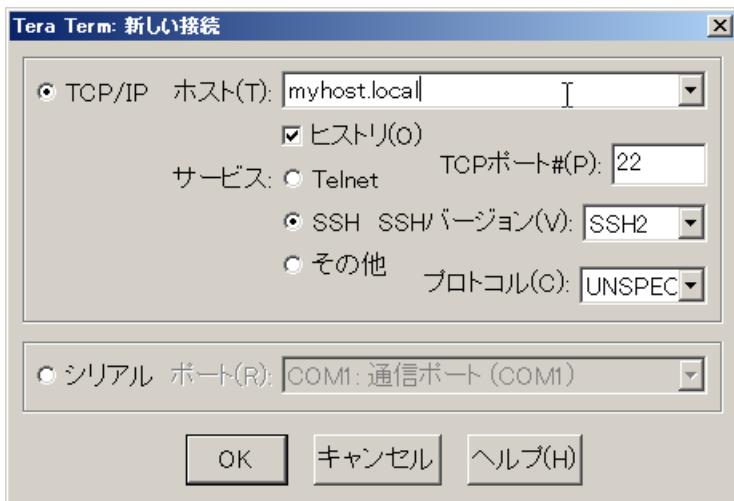
2 PC に Raspberry Pi を接続する。

PC と Raspberry Pi を USB ケーブル（A コネクタ-マイクロ B コネクタ）で接続し、Raspberry Pi に電源を供給します。

3 Raspberry Pi に接続する。

Windows から Raspberry Pi に ssh 経由でログインするためには、ssh クライアントを利用する必要があります。 今回は Tera Term を利用します。

- [Tera Term](#)



TeraTerm を起動すると接続ダイアログが現れるので、接続する Raspberry Pi のホスト名+.local 、または、Raspberry Pi の IP アドレスを「ホスト」のテキストボックスに入力し OK を押します。今回の講習会ではホスト名は箱の付箋に記載されています。

ID: pi

パスワード: raspberry

でログインできます。

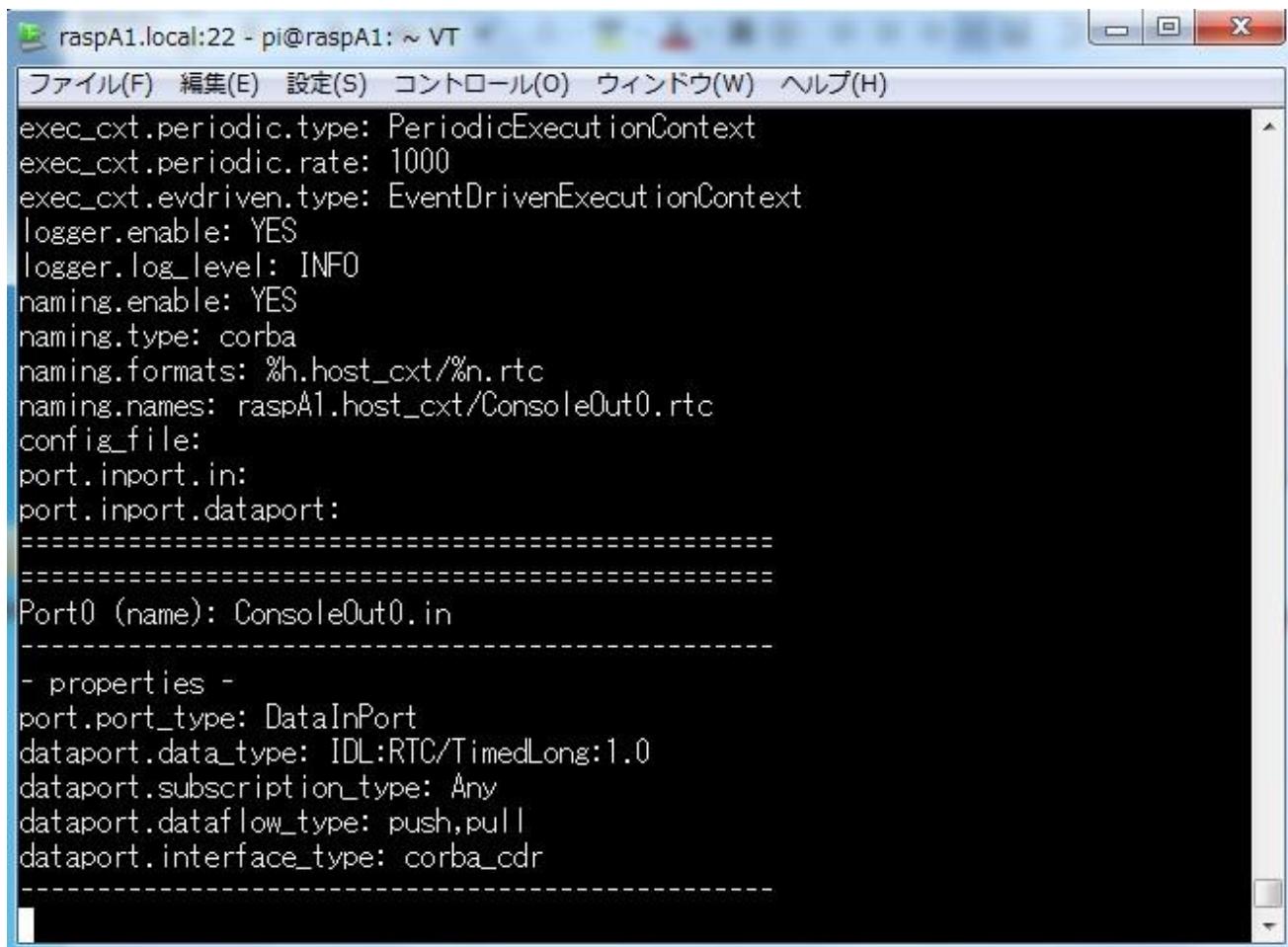
4 Raspberry Pi で NameServer と ConsoleOut コンポーネントを起動

Raspberry Pi に TeraTerm などで接続後、コンポーネントを起動します。 ネーミングサービスを起動してから ConsoleOut コンポーネントを起動します。

```
$ rtm-naming  
$ /usr/share/openrtm-1.1/example/ConsoleOutComp
```

rtm-naming : ネーミングサービスの起動

下図の様に表示されます。



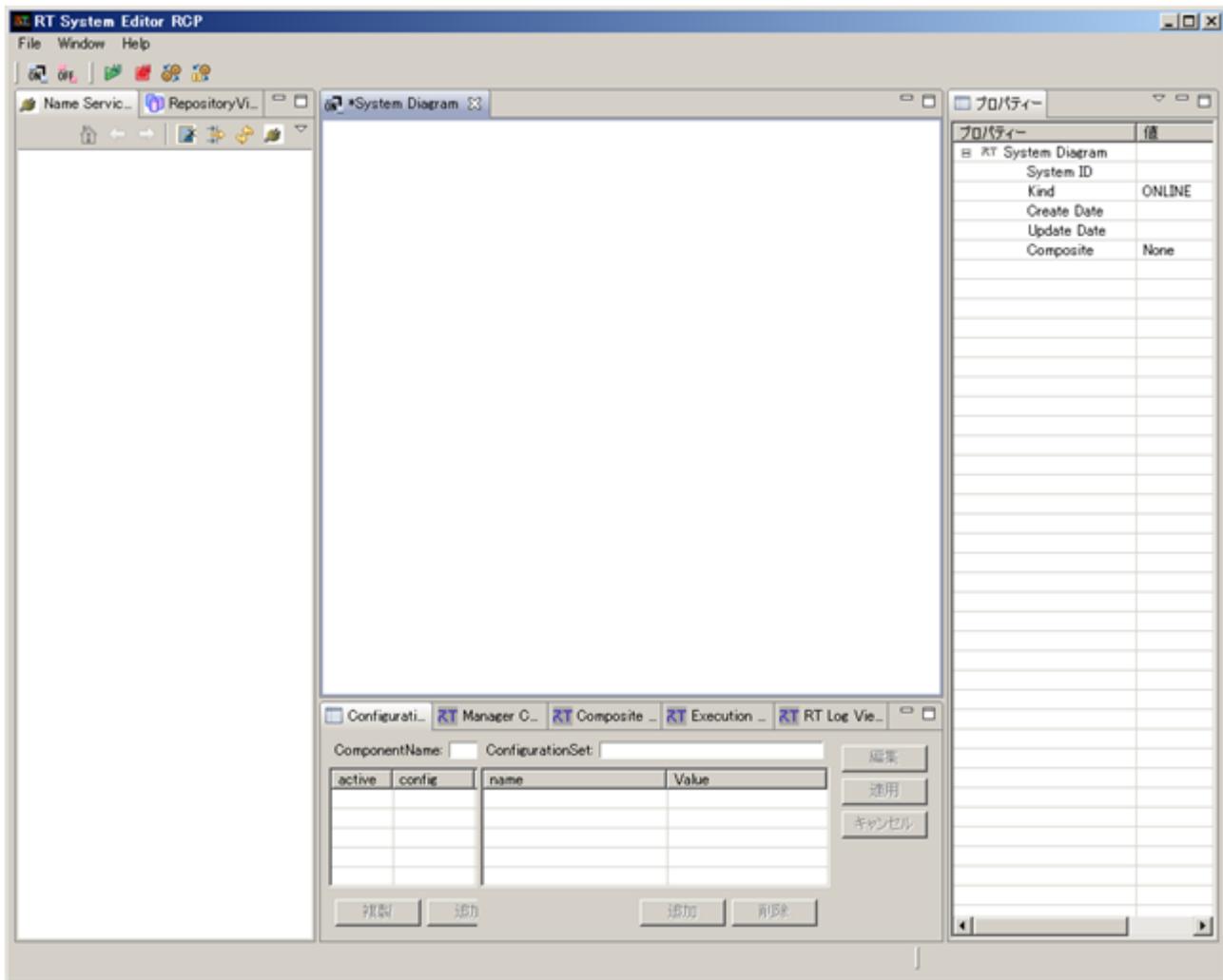
```
raspA1.local:22 - pi@raspA1: ~ VT  
ファイル(F) 編集(E) 設定(S) コントロール(O) ウィンドウ(W) ヘルプ(H)  
exec_ctxt.periodic.type: PeriodicExecutionContext  
exec_ctxt.periodic.rate: 1000  
exec_ctxt.evdriven.type: EventDrivenExecutionContext  
logger.enable: YES  
logger.log_level: INFO  
naming.enable: YES  
naming.type: corba  
naming.formats: %h.host_ctxt/%n.rtc  
naming.names: raspA1.host_ctxt/ConsoleOut0.rtc  
config_file:  
port.inport.in:  
port.inport.dataport:  
=====  
=====  
Port0 (name): ConsoleOut0.in  
-----  
- properties -  
port.port_type: DataInPort  
dataport.data_type: IDL:RTC/TimedLong:1.0  
dataport.subscription_type: Any  
dataport.dataflow_type: push,pull  
dataport.interface_type: corba_cdr  
-----
```

※ConsoleOut コンポーネントの終了方法は Raspberry Pi に接続した TeraTerm で Ctrl キー+C キーを押します。

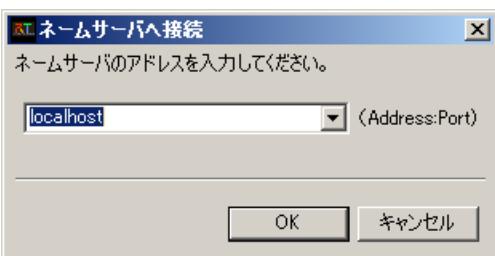
5 PC で RTSystemEditorRCP を起動し各コンポーネントを Active にする。

OpenRTP を起動していない場合、スタートメニューから 「OpenRTM-aist x.y」 → 「tools」 の下の OpenRTP から起動します。

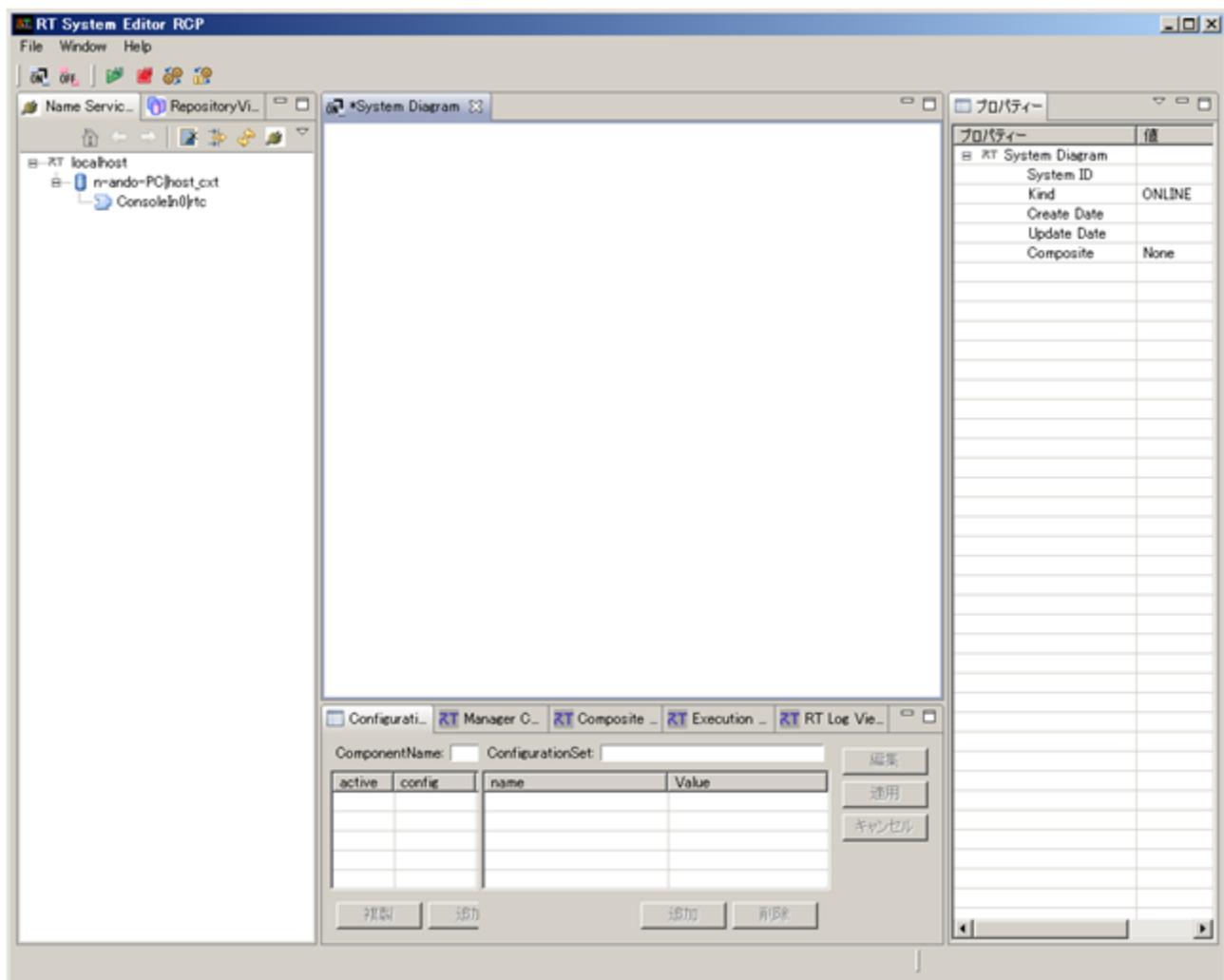
RTSystemEditor の左側の NameService View のコンセントアイコンをクリックし、ネームサーバに接続します。



まず、自ホストのネームサーバに接続します。接続ダイアログに localhost と入力します。



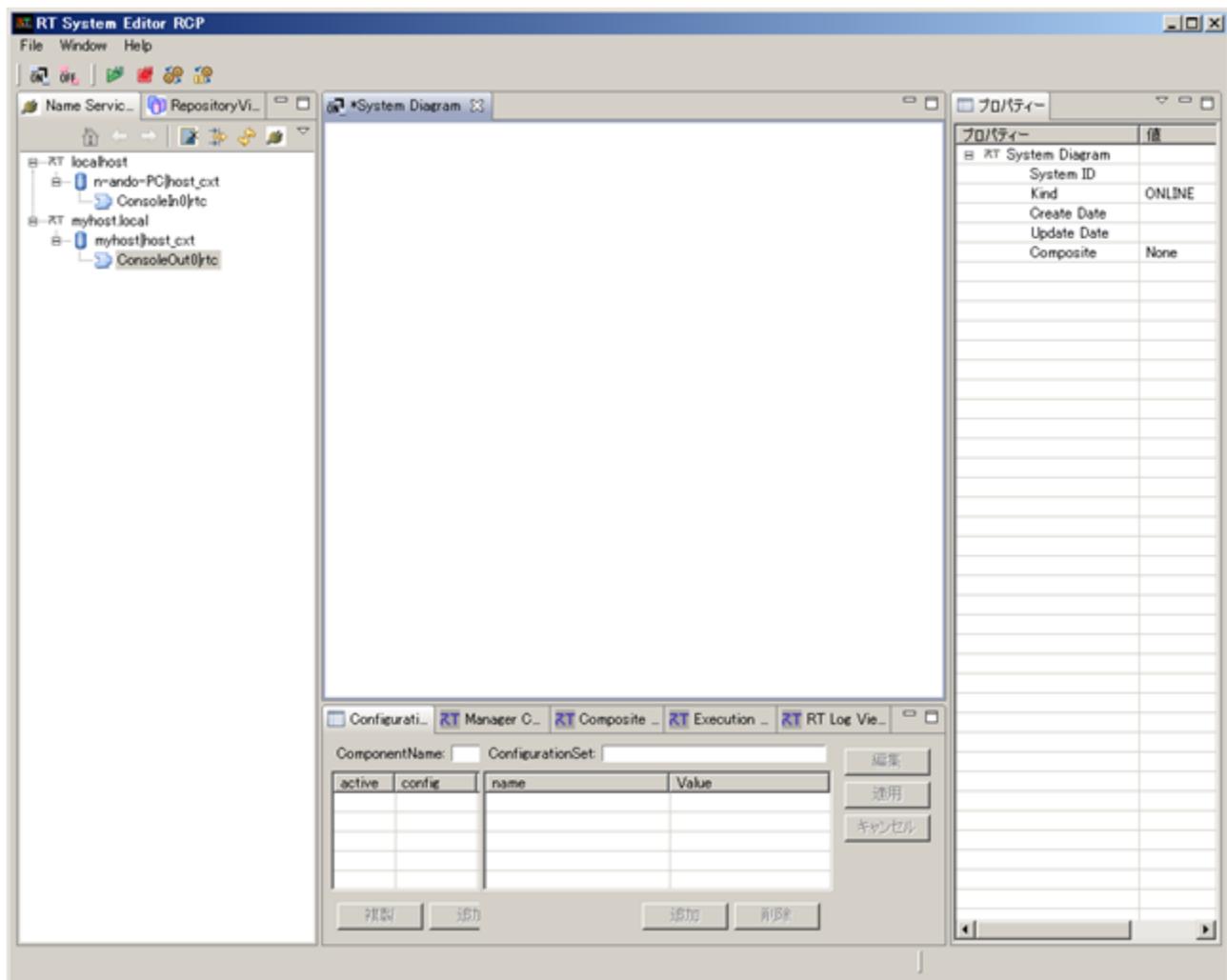
手順1 サンプルコンポーネントの実行



次に、Raspberry Pi のネームサーバへ接続します。再度 NameService View の接続アイコンをクリックし、Raspberry Pi のホスト名+.local、または、Raspberry Pi の IP アドレスをダイアログに入力します。

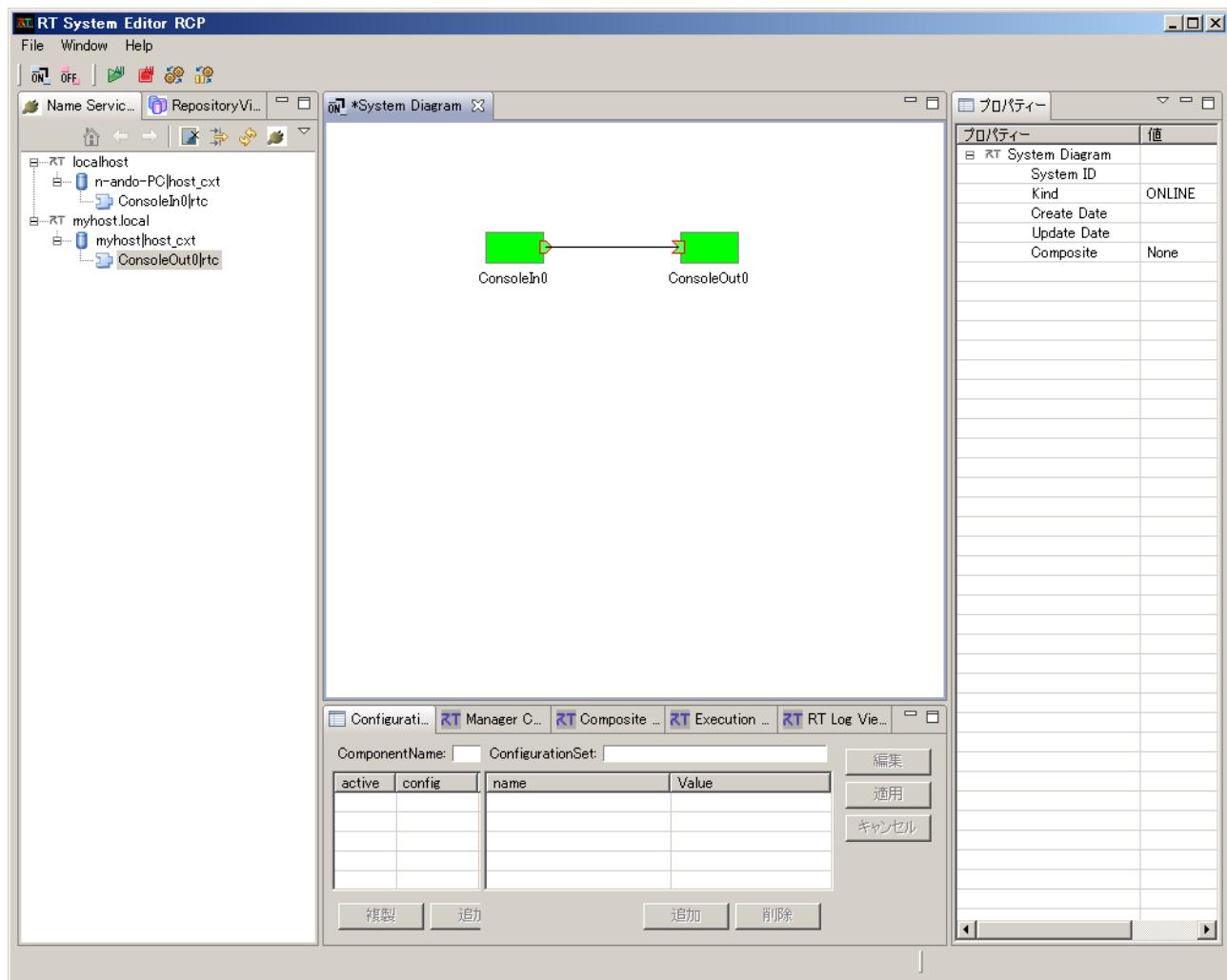


手順1 サンプルコンポーネントの実行



ネームサービスビューには 2 つのネームサーバの状態が表示され、それぞれのネームサーバの下に ConsoleIn0、ConsoleOut0 という 2 つのコンポーネントが見えているはずです。 RTSystemEditor のメニューバーの online エディタアイコン(ON と書かれたアイコン)をクリックし、 SystemEditor を開きます。 NameService View から ConsoleIn0 と ConsoleOut0 をそれぞれ SystemEditor 上に ドラッグアンドドロップし、 InPort と OutPort を接続します。

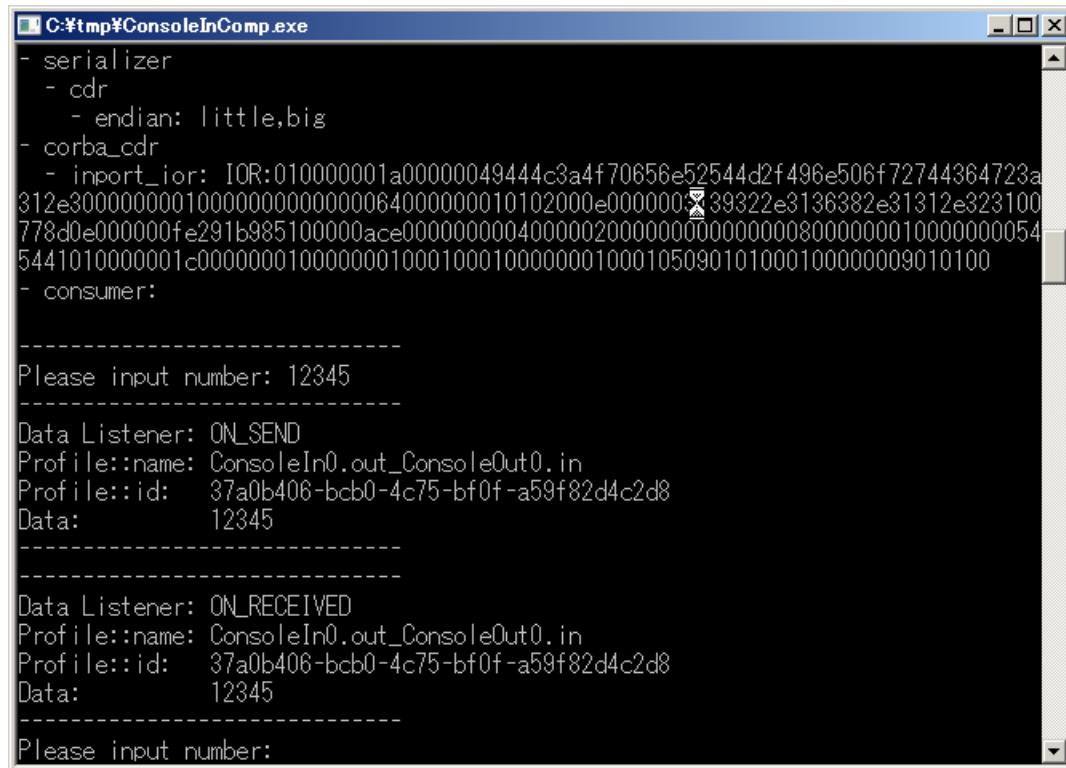
手順1 サンプルコンポーネントの実行



メニューバーの緑の再生ボタンをクリックすると、2つのコンポーネントがアクティベートされ上図のような状態になります。

6 ConsoleIn と ConsoleOut が動作することを確認

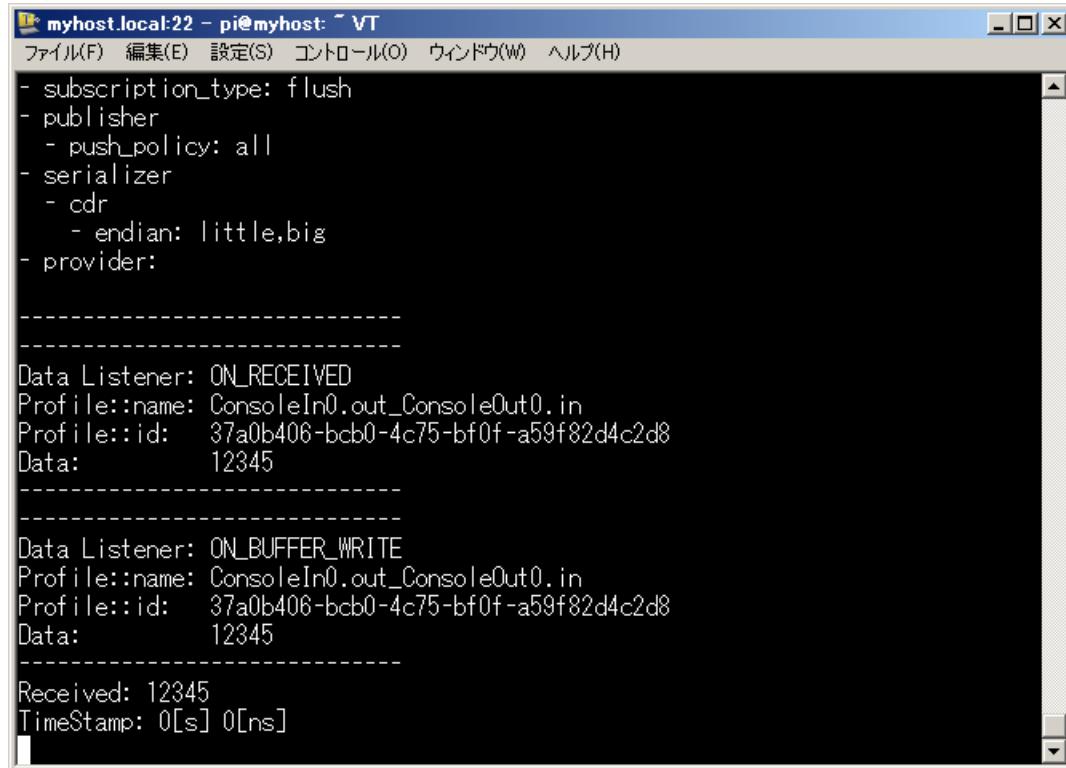
PC 側の ConsoleIn コンポーネントのウインドウから適当な数字を入力します。



```
C:\tmp\ConsoleInComp.exe
- serializer
  - cdr
    - endian: little,big
- corba_cdr
  - import_iior: IOR:010000001a00000049444c3a4f70656e52544d2f496e506f72744364723a
312e3000000001000000000000000064000000010102000e000000039322e3136382e31312e323100
778d0e000000fe291b985100000ace0000000004000002000000000000000008000000010000000054
544101000001c0000000100000001000100010509010100010000009010100
- consumer:

-----
Please input number: 12345
-----
Data Listener: ON_SEND
Profile::name: ConsoleIn0.out_ConsoleOut0.in
Profile::id: 37a0b406-bcb0-4c75-bf0f-a59f82d4c2d8
Data: 12345
-----
Data Listener: ON_RECEIVED
Profile::name: ConsoleIn0.out_ConsoleOut0.in
Profile::id: 37a0b406-bcb0-4c75-bf0f-a59f82d4c2d8
Data: 12345
-----
Please input number:
```

すると、下図のように Raspberry Pi 側の ConsoleOut の表示に ConsoleIn で入力した数値が現れます。



```
myhost.local:22 - pi@myhost: ~ VT
ファイル(F) 編集(E) 設定(S) コントロール(O) ウィンドウ(W) ヘルプ(H)
- subscription_type: flush
- publisher
  - push_policy: all
- serializer
  - cdr
    - endian: little,big
- provider:

-----
Data Listener: ON_RECEIVED
Profile::name: ConsoleIn0.out_ConsoleOut0.in
Profile::id: 37a0b406-bcb0-4c75-bf0f-a59f82d4c2d8
Data: 12345
-----
Data Listener: ON_BUFFER_WRITE
Profile::name: ConsoleIn0.out_ConsoleOut0.in
Profile::id: 37a0b406-bcb0-4c75-bf0f-a59f82d4c2d8
Data: 12345
-----
Received: 12345
TimeStamp: 0[s] 0[ns]
```

動作確認が終了しましたら、OpenRTP の画面の赤の停止ボタンをクリックしてディアクティベートします。

ディアクティベートが完了したら、ConsoleIn コンポーネントのウインドウの×ボタンをクリックし、コンポーネントを終了させます。Raspberry Pi に接続した TeraTerm で Ctrl キー+C キーを押し ConsoleOut コンポーネントを終了します。

トラブルシューティング

Raspberry Pi 側のサービスのしたに ConsoleOut が表示されない

ネームサーバの問題

ネームサーバのエンドポイントアドレスが不正な場合このような現象が起きます。rtm-naming で再度ネームサーバを起動しなおすと解決する場合があります。このほか、Raspberry Pi の有線 LAN と無線 LAN 等 2つ以上のネットワークインターフェースがある場合、PC との接続に使用するどちらかのネットワークのみを使うように設定することで解決するケースもあります。

コンポーネントの問題

コンポーネントが読み込んだ設定ファイル(rtc.conf)に localhost 以外のネームサーバが登録されている場合が考えられます。corba.nameservers: localhost と記述するなど localhost のネームサーバにコンポーネントを登録するよう設定してください。また、Raspberry Pi の有線 LAN と無線 LAN 等 2つ以上のネットワークインターフェースがある場合、PC との接続に使用するどちらかのネットワークのみを使うように設定することで解決するケースもあります。

RTSystemEditor で接続できない、反応がなくなる等

PC 側のコンポーネントの問題

PC にネットワークインターフェースが 2つ以上ある場合、Raspberry Pi で使用しない側のインターフェースアドレスがコンポーネントの参照として利用されている場合このような現象が起こります。コマンドプロンプトから ipconfig で PC の IP アドレスを調べる等して、rtc.conf に使用する方の IP アドレスを以下のように設定します。

```
corba.endpoints: 192.168.11.20
```

ただし、Vista 以降の Windows では、C:\Program Files 以下のファイルは簡単には編集ができなくなっています。c:\tmp など適当なディレクトリに ConsoleIn.exe と rtc.conf をコピー(あるいは新たに作成)するなどして、対処してください。

Raspberry Pi 側のコンポーネントの問題

Raspberry Pi の有線 LAN と無線 LAN 等 2つ以上のネットワークインターフェースがある場合で、それぞれが別のネットワークにつながっている場合、上述の PC と同様の問題が起こります。ifconfig で使用する方の IP アドレスを調べ、rtc.conf に以下のように記載します。

```
corba.endpoints: 192.168.11.21
```