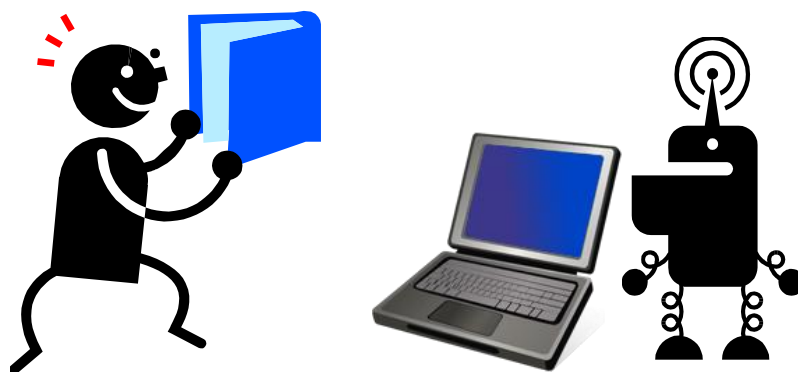


64bit 版 Windows 環境
OpenRTM-aist-1.1.0.
インストールマニュアル (v1.0)



2012 年 11 月 7 日

芝浦工業大学 機械機能工学科 知能機械システム研究室

(担当：生田目祥吾，石田真一)

～～目次～～

| | | |
|----|--------------------|-----|
| 1. | はじめに | 2 |
| 2. | ダウンロードページへの移動 | 2～3 |
| 3. | OpenRTM-aist-1.1.0 | 4～8 |

(開発環境もインストールしたい場合に必要)

| | | |
|-----|-----------------------|-------|
| 4. | Windows (64bit) 用 JRE | 9～10 |
| 5. | Python (1) | 11～13 |
| 6. | Python (2) | 14 |
| 7. | PyYAML | 15～17 |
| 8. | CMake-2.8 Visual | 18～21 |
| 9. | Doxygen | 22～26 |
| 10. | Java (32bit) | 27～28 |
| 11. | Eclipse | 29～34 |

| | | |
|-----|------------|-------|
| 12. | 動作確認 (準備編) | 35～42 |
| 13. | 動作確認 (実行編) | 43～46 |

1. はじめに

このマニュアルの内容は RT ミドルウェア初心者である私が RT ミドルウェアをインストールしていない PC に環境設定を試みた手順です。よって間違いや無駄な手順などが多数あるかもしれません。とりあえず、私の設定した PC 上ではサンプルの動作といくつかの既存コンポーネントの再利用ができたので、RT ミドルウェア初心者のインストール作業に少しでも役立てば幸いです。また、開発者および知能機械システム研究室は本マニュアルの利用によって生じたいかなる損害にも責任を負いかねますのでご了承ください。

指摘やアドバイスをして下さる方がいれば以下に示す筆者のメールアドレスへ連絡をください。喜んで修正いたします。

b09074@shibaura-it.ac.jp

それでは早速インストールを始めましょう。このマニュアルでは 2012 年 11 月 9 日現在における OpenRTM-aist1.1.0 のダウンロードから動作確認までの手順を示します。ここで解説するのは Visual C++2010 Professional をインストールした 64bit の Windows7 環境へのインストールですので、少し違う環境の場合は適宜読み替えてください。

2. ダウンロードページへの移動

- ① RT ミドルウェアを使うために必要なプログラムは OpenRTM-aist 公式 HP 上でダウンロード可能です。まずは以下の URL が示すページへ移動してみてください。

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

- ② まずは上記 HP のトップ画面からページ左側の「ナビゲーション」内にある「ダウンロード」をクリックします。



- ③ 次に OpenRTM-aist1.1.0 (C++版) と表示されている箇所の「1.1.0-RELEASE」をクリックします。



- ④ ページ中央あたりに「パッケージ」「Windows インストーラ」という項目があるので、以降の手順ではこの中から該当するものを探しましょう。(私の場合は Visual Studio 2010 (64bit) 用です)

以降の解説ではこのページ上から必要なプログラムをダウンロードします。

Visual Studio 2010 (64bit) 用

Visual C++ 2010 Express版でも(デフォルトでは)64bit バイナリのビルドができませんのでご注意ください。

| Visual Studio 2010 (64bit) 用 | | |
|--|---|----------------------------|
| Windowsインストーラ(64bit) | OpenRTM-aist-1.1.0-RELEASE_vc10_x64.msi MD5:24e2ac9673da4a3e9991e959d2d06f69 | 2012.05.25 |
| Java: Windows (64bit) 用 JRE | インストーラ | java.com |
| Python-64bit (omniidlでのみ必要) | python-2.7.3.amd64.msi | python.org |
| Python (各種ツールで必要) Python 2.x 利用可 | python-2.6.6.msi | python.org |
| PyYAML (各種ツールで必要) Pythonのバージョンに合わせて | PyYAML-3.10.win32-py2.6.exe | pyyaml.org |
| CMake-2.8 Visual (C++のプロジェクトを作成するのに必要) | cmake-2.8.8 | cmake |
| Doxygen (ビルドの過程でドキュメントを整形するのに必要) | doxygen | doxygen |

3. OpenRTM-aist-1.1.0

- ① まずは一番上にある「OpenRTM-aist-1.1.0-RELEASE_vc10_x64.msi」をクリックします。
RT コンポーネントを動作させる上でこのプログラムは必須です。

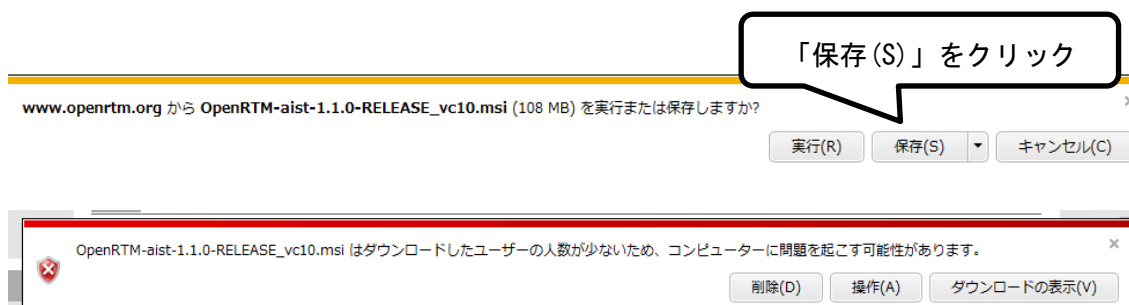
Visual Studio 2010 (64bit) 用

Visual C++ 2010 Express版では(デフォルトでは)64bitのバイナリのビルドができませんのでご注意ください。

| Visual Studio 2010 (64bit) 用 | | |
|--|---|----------------------------|
| Windowsインストーラ(64bit) | OpenRTM-aist-1.1.0-RELEASE_vc10_x64.msi MD5:24e2ac9673da4a3e9991e959d2d06f69 | 2012.05.25 |
| Java: Windows (64bit) 用 JRE | インストーラ | java.com |
| Python-64bit (omniidlでのみ必要) | python-2.7.3.amd64.msi | python.org |
| Python (各種ツールで必要) Python 2.x 利用可 | python-2.6.6.msi | python.org |
| PyYAML (各種ツールで必要) Pythonのバージョンに合わせる必要 | PyYAML-3.10.win32-py... | |
| CMake-2.8 Visual (C++のプロジェクトを 作成するのに必要) | cmake-2.8.8 | |
| Doxygen (ビルドの過程でドキュメントを 整形するのに必要) | doxygen | doxygen |

「OpenRTM-aist-1.1.0-RELEASE_vc10_x64.msi」
をクリック

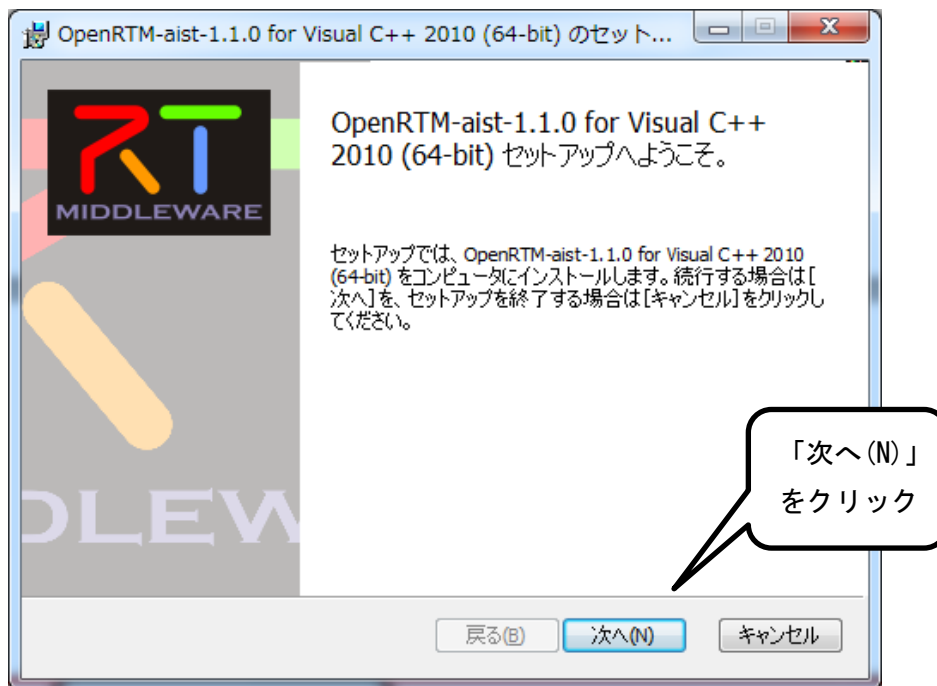
- ② 以下のような黄色の警告が出たら「保存(S)」をクリックしてダウンロードします。ダウンロード後に赤い警告が出るかもしれませんが、気にせず右上の「×」をクリックして進んでしまいましょう。



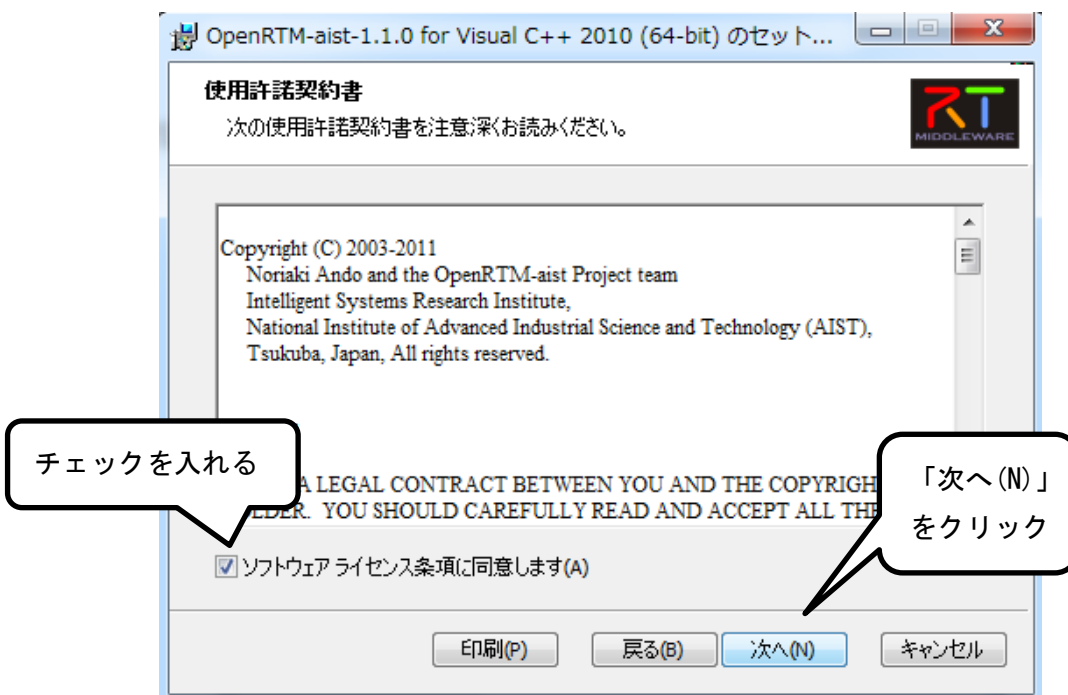
- ③ 次に、ダウンロードしたディレクトリ上で「OpenRTM-aist-1.1.0-RELEASE_vc10_x64.msi」をダブルクリックし、現れたウィンドウに対して「実行(R)」をクリックしてください。



- ④ 下図のようなウィンドウが現れるので「次へ(N)」をクリックします。



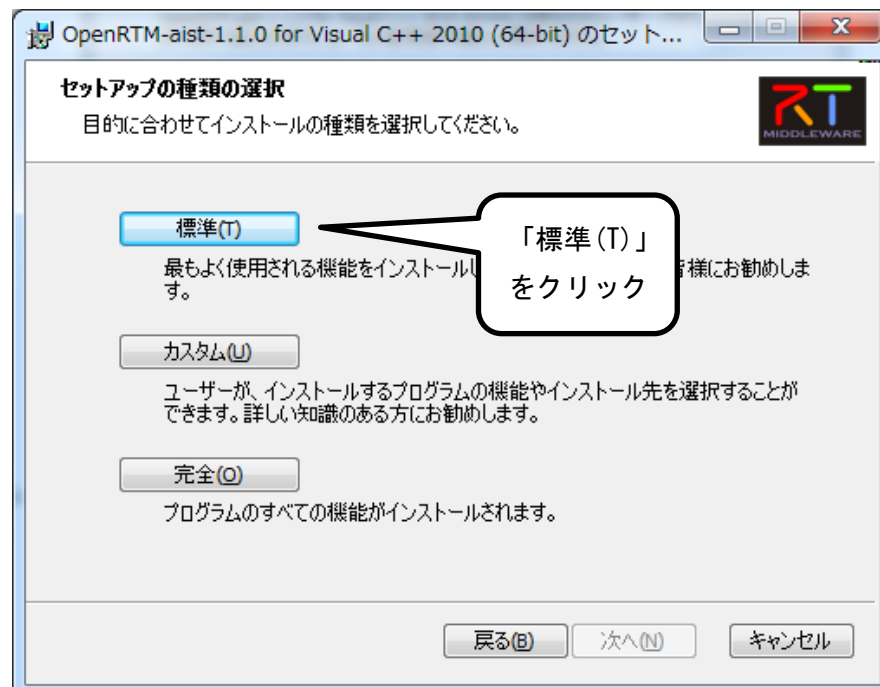
- ⑤ 次に現れた画面では「ソフトウェアライセンス条項に同意します(A)」にチェックを入れて「次へ(N)」をクリックしてください。



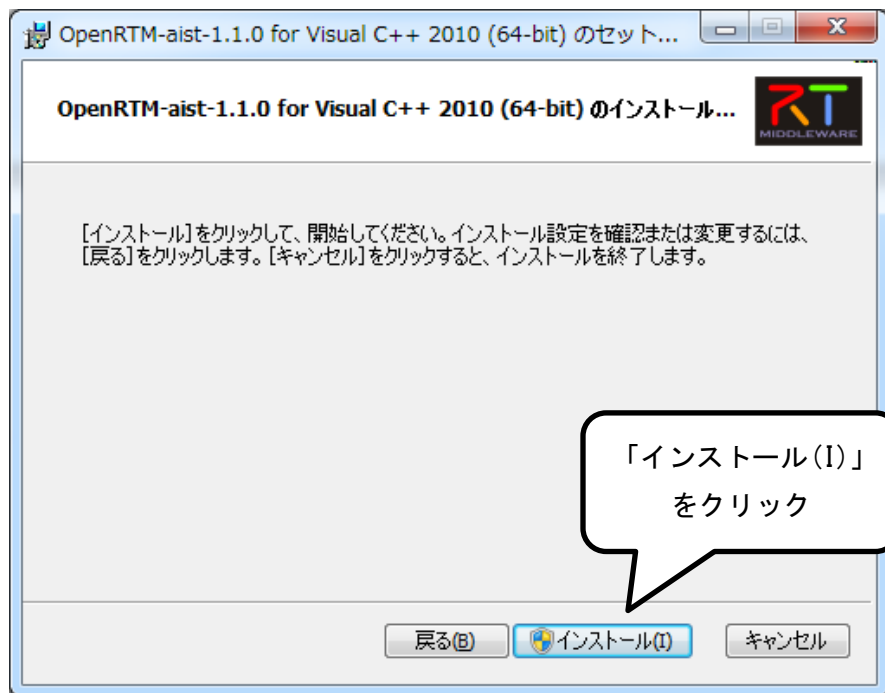
- ⑥ どちらか好きな方のインストールを選択し、「次へ(N)」をクリックします。



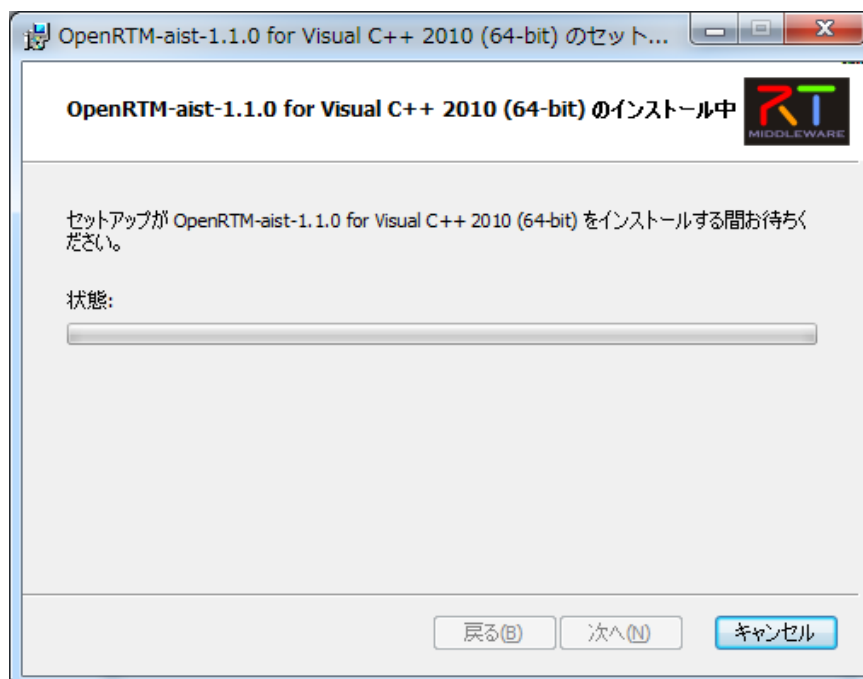
- ⑦ とりあえず「標準(T)」をクリックします。



- ⑧ 「インストール(I)」をクリックします。アカウント制御の認証ダイアログが現れた場合は許可しましょう。



- ⑨ インストールが始まりますのでしばらく待ちましょう。



- ⑩ 以下のような画面が表示されたら OpenRTM-aist1.1.0 のインストールは終了です。
「完了(F)」をクリックしてインストーラを終了してください。



以上で OpenRTM-aist1.1.0 のインストール作業に関する解説は終わりです。次に解説する 4. ～11. は RT コンポーネントの開発に必要なインストール作業ですので、単純に動作をさせたい場合は全て飛ばして 12. ～の動作確認へ進んでください。

以下の 4. ～11. は RT コンポーネントの開発環境に必要なプログラムです。
動作のみを実現させたい場合は全て読み飛ばして 12. へ進んでください。

4. Windows (64bit) 用 JRE

- ① 開発環境も構築したい場合、まずは最初に移動したダウンロードページ上で Java をインストールします。上から 2 番目の「インストーラ」をクリックしましょう。
既に Java がインストールされている場合はインストーラの実行時に警告が出ますので、そのまま閉じて次の 5. に進んでください。

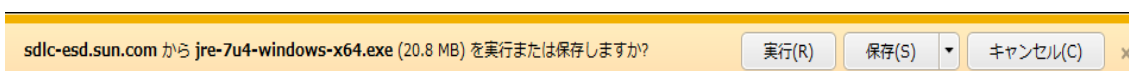
Visual Studio 2010 (64bit) 用

Visual C++ 2010 Express版では(デフォルトでは)64bit バイナリのビルドができませんのでご注意ください。

| Visual Studio 2010 (64bit) 用 | | |
|--|---|----------------------------|
| Windowsインストーラ(64bit) | OpenRTM-aist-1.1.0-RELEASE_vc10_x64.msi MD5:24e2ac9673da4a3e9991e959d2d06f69 | 2012.05.25 |
| Java: Windows (64bit) 用 JRE | インストーラ | java.com |
| Python-64bit (omniidlでのみ必要) | python-2.7.3.amd64.msi | python.org |
| Python (各種ツールで必要) Python 2.x 利用可 | python-2.6.6.msi | python.org |
| PyYAML (各種ツールで必要) Pythonのバージョンに合わせる必要 | PyYAML-3.10.win32-py2.6.exe | pyyaml.org |
| CMake-2.8 Visual (C++のプロジェクトを作成するのに必要) | cmake-2.8.8 | cmake |
| Doxygen (ビルドの過程でドキュメントを整形するのに必要) | doxygen | doxygen |

「インストーラ」
をクリック

- ② 先ほどと同様に黄色い警告が現れますが今度は「実行(R)」で構いません。アカウント制御の認証ダイアログが出現したら許可しましょう。



- ① Java のインストーラが立ち上がるので「インストール(I)」をクリックしてください。



「インストール(I)」
をクリック

- ② Java のインストールが始まります。しばらく待ちましょう。



- ③ インストールが完了すると以下の画面になりますので、「閉じる(C)」をクリックしてインストーラを終了しましょう。



5. Python(1)

- ① 今度はダウンロードページの上から 3 番目にある Python をインストールします。
「Python-2.7.3.amd64.msi」をクリックしてください。

Visual Studio 2010 (64bit) 用

Visual C++ 2010 Express版では(デフォルトでは)64bit バイナリのビルドができませんのでご注意ください。

| Visual Studio 2010 (64bit) 用 | | |
|--|---|------------|
| Windowsインストーラ(64bit) | OpenRTM-aist-1.1.0-RELEASE_vc10_x64.msi MD5:24e2ac9673da4a3e9991e959d2d06f69 | 2012.05.25 |
| Java: Windows (64bit) 用 JRE | インストーラ | java.com |
| Python-64bit (omniidlでのみ必要) | python-2.7.3.amd64.msi | python.org |
| Python (各種ツールで必要) Python 2.x 利用可 | python-2.6.6.msi | python.org |
| PyYAML (各種ツールで必要) Pythonのバージョンに合わせて | PyYAML-3.10.win32-py2.6.exe | |
| CMake-2.8 Visual (C++のプロジェクトを 作成するのに必要) | cmake-2.8.8 | |
| Doxygen (ビルドの過程でドキュメントを 整形するのに必要) | doxygen | doxygen |

「Python-2.7.3.amd64.msi」
をクリック

- ② 先ほどと同様に警告が現れたら「実行(R)」をクリックします。

www.python.org から python-2.7.3.amd64.msi (15.6 MB) を実行または保存しますか?

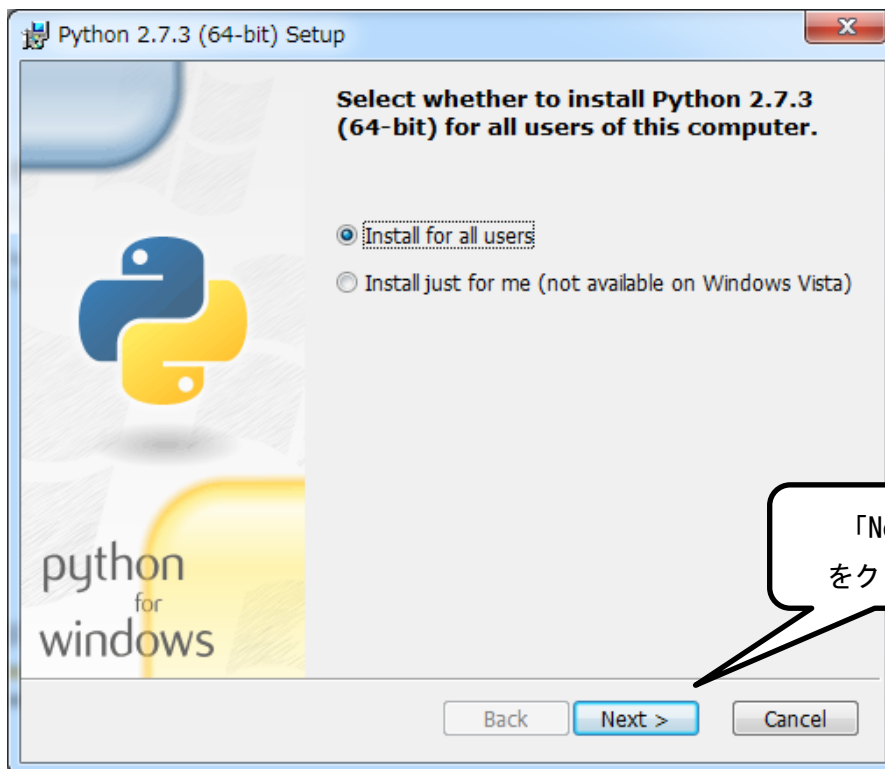
実行(R)

保存(S)

キャンセル(C)

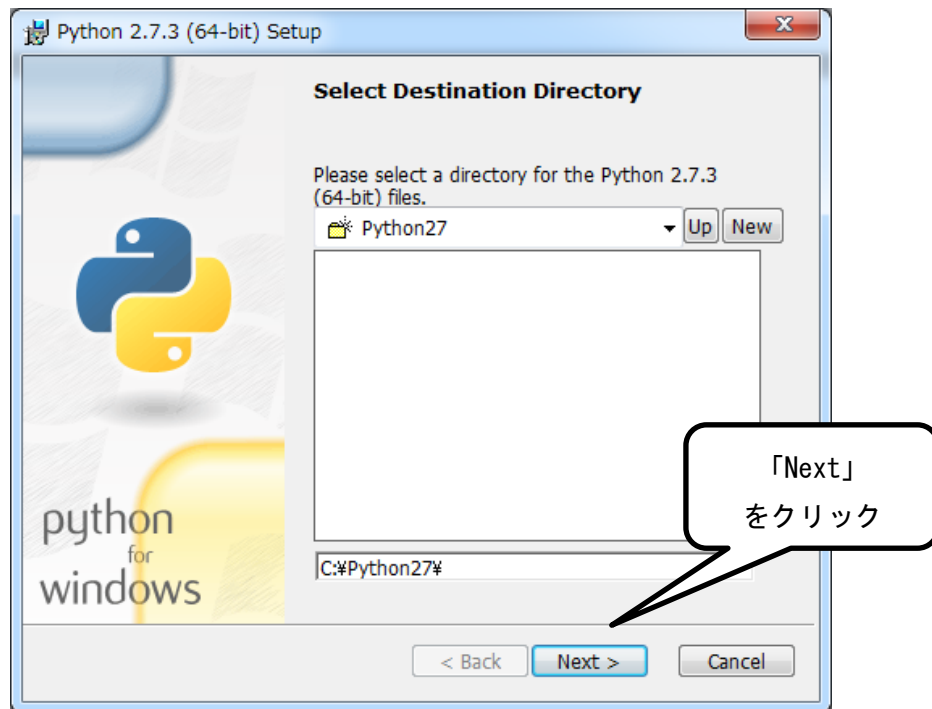
x

- ③ インストーラが起動し以下の画面が現れたら、どちらか好きな方のインストールを選択した状態で「Next」をクリックしてください。

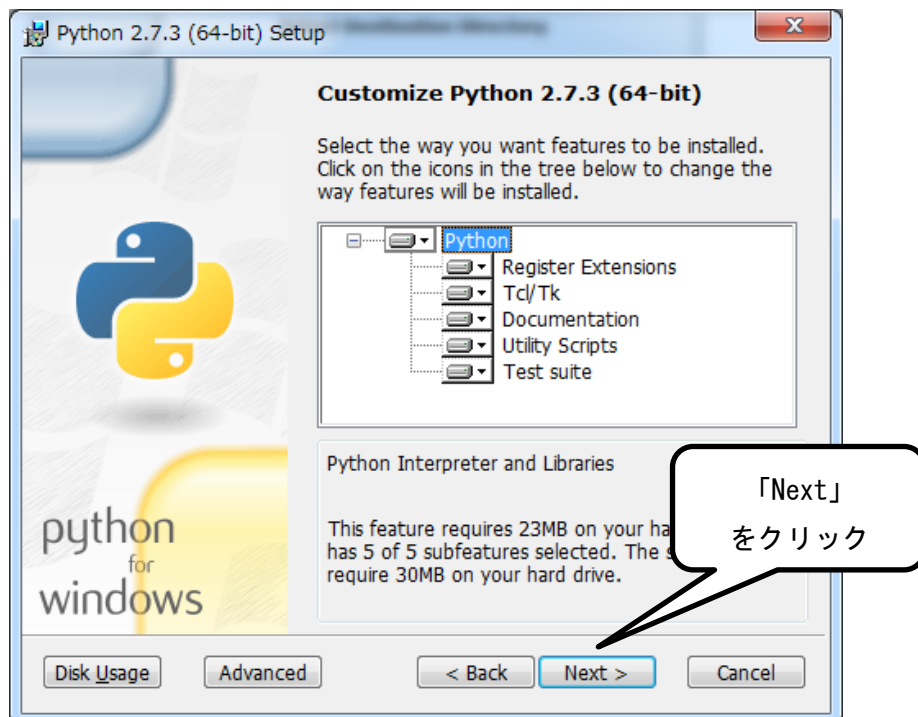


「Next」
をクリック

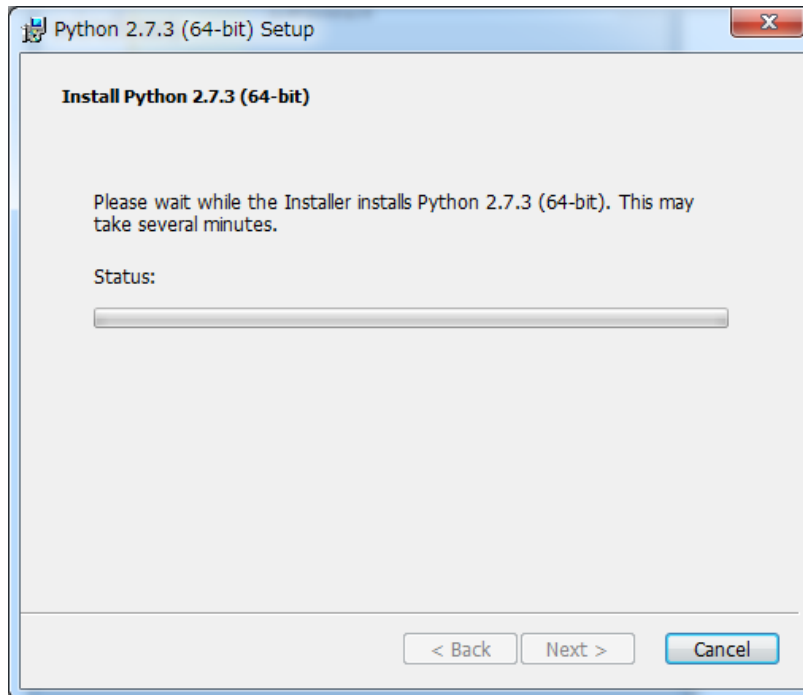
- ④ インストールする場所を指定します。今回はデフォルトで指定されているCドライブ直下に入れるので、特に変更を加えることなくそのまま「Next」をクリックしてください。



- ⑤ 次に現れた画面でもそのままの状態ですべての機能をインストールする状態で「Next」をクリックしてください。



- ⑥ 下図のようにインストールが始まります。アカウント制御の認証ダイアログが現れた場合は許可しましょう。



- ⑦ インストールが完了したら以下の画面になりますので 「Finish」をクリックしてインストーラを終了しましょう。



6. Python (2)

- ① 今度はダウンロードページで上から 4 番目の Python をインストールします。
「Python-2.6.6.msi」をクリックしてください。

Visual Studio 2010 (64bit) 用

Visual C++ 2010 Express版では(デフォルトでは)64bit バイナリのビルドができませんのでご注意ください。

| Visual Studio 2010 (64bit) 用 | | |
|--|--|----------------------------|
| Windowsインストーラ(64bit) | OpenRTM-aist-1.1.0-RELEASE_vc10_x64.msi MD5:24e2ac9673da4a3e9991e9599d2d06f69 | 2012.05.25 |
| Java: Windows (64bit) 用 JRE | インストーラ | java.com |
| Python-64bit (omniidlでのみ必要) | python-2.7.3.amd64.msi | python.org |
| Python (各種ツールで必要) Python 2.x 利用可 | python-2.6.6.msi | python.org |
| PyYAML (各種ツールで必要) Pythonのバージョンに合わせる必要 | PyYAML-3.10.win32-py2.6.exe | pyyaml.org |
| CMake-2.8 Visual (C++のプロジェクトを作成するのに必要) | cmake-2.8.8 | |
| Doxygen (ビルドの過程でドキュメントを整形するのに必要) | doxygen | |

「Python-2.6.6.msi」
をクリック

- ② 警告が現れたら「実行(R)」をクリックします。

www.python.org から python-2.6.6.msi (14.5 MB) を実行または保存しますか?

実行(R)

保存(S)

キャンセル(C)

x

- ③ 以降のインストーラの操作手順は前頁の 5. Python (1) で説明したものと全く同様ですので省略します。解説が必要な場合は 5. Python (1) の③～⑦を再び参照しましょう。

7. PyYAML

- ① ダウンロードページで上から 5 番目の PyYAML をインストールします。

「PyYAML-3.10.win32-py2.6.exe」をクリックしてください。

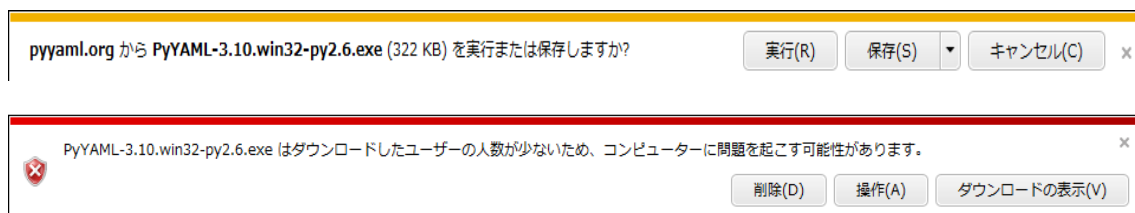
Visual Studio 2010 (64bit) 用

Visual C++ 2010 Express版では(デフォルトでは)64bit バイナリのビルドができませんのでご注意ください。

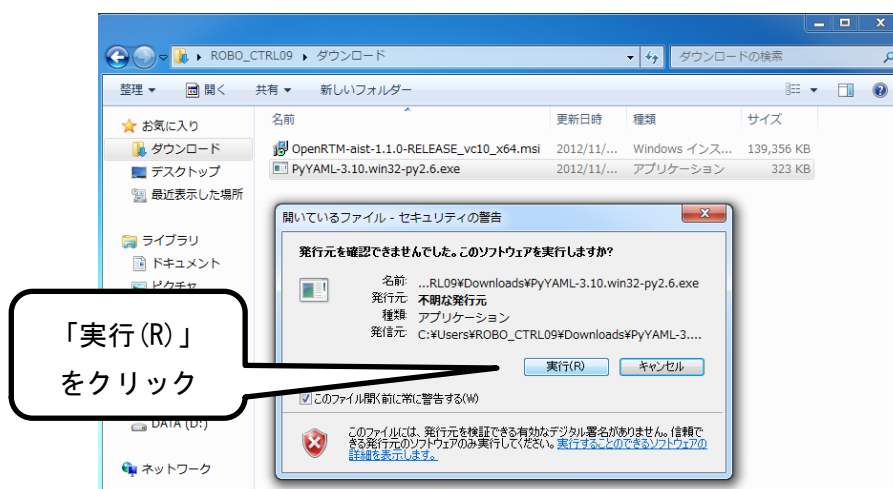
| Visual Studio 2010 (64bit) 用 | | |
|--|---|----------------------------|
| Windowsインストーラ(64bit) | OpenRTM-aist-1.1.0-RELEASE_vc10_x64.msi MD5:24e2ac9673da4a3e9991e959d2d06f69 | 2012.05.25 |
| Java: Windows (64bit) 用 JRE | インストーラ | java.com |
| Python-64bit (omniidlでのみ必要) | python-2.7.3.amd64.msi | python.org |
| Python (各種ツールで必要) Python 2.x 利用可 | python-2.6.6.msi | python.org |
| PyYAML (各種ツールで必要) Pythonのバージョンに合わせる必要 | PyYAML-3.10.win32-py2.6.exe | |
| CMake-2.8 Visual (C++のプロジェクトを作成するのに必要) | cmake-2.8.8 | |
| Doxygen (ビルドの過程でドキュメントを整形するのに必要) | doxygen | |

「PyYAML-3.10.win32-py2.6.exe」
をクリック

- ② 黄色い警告が現れたら今度は「保存」をクリックします。ダウンロード後に赤い警告が出るかもしれませんが、再び「×」をクリックして気にせず進みましょう。

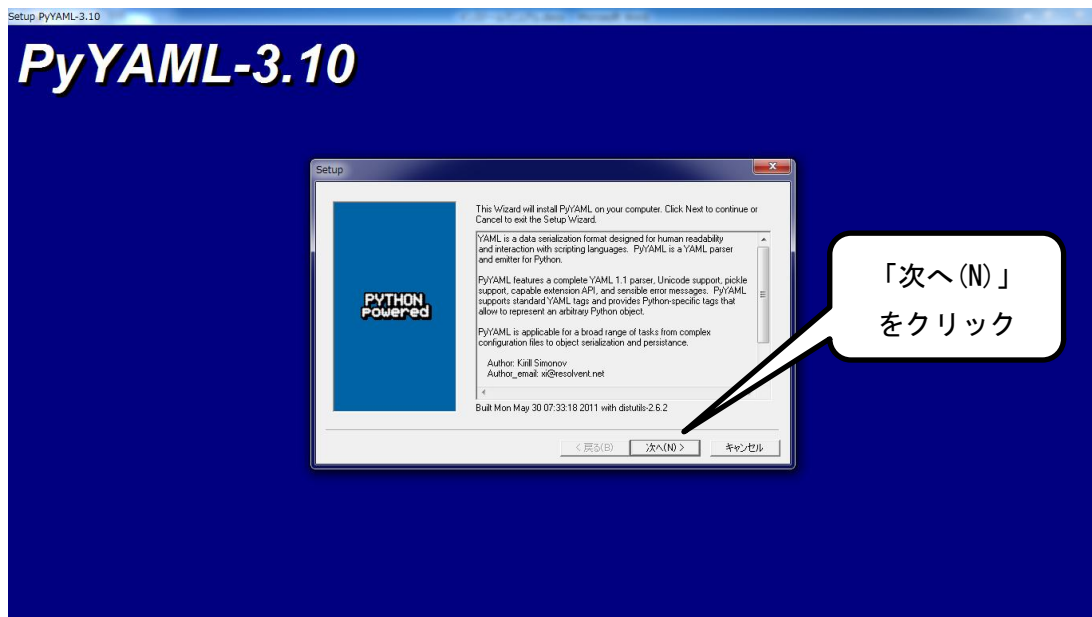


- ③ ダウンロードしたディレクトリ上で「PyYAML-3.10.win32-py2.6.exe」をダブルクリックし、現れたダイアログ上で「実行(R)」をクリックします。

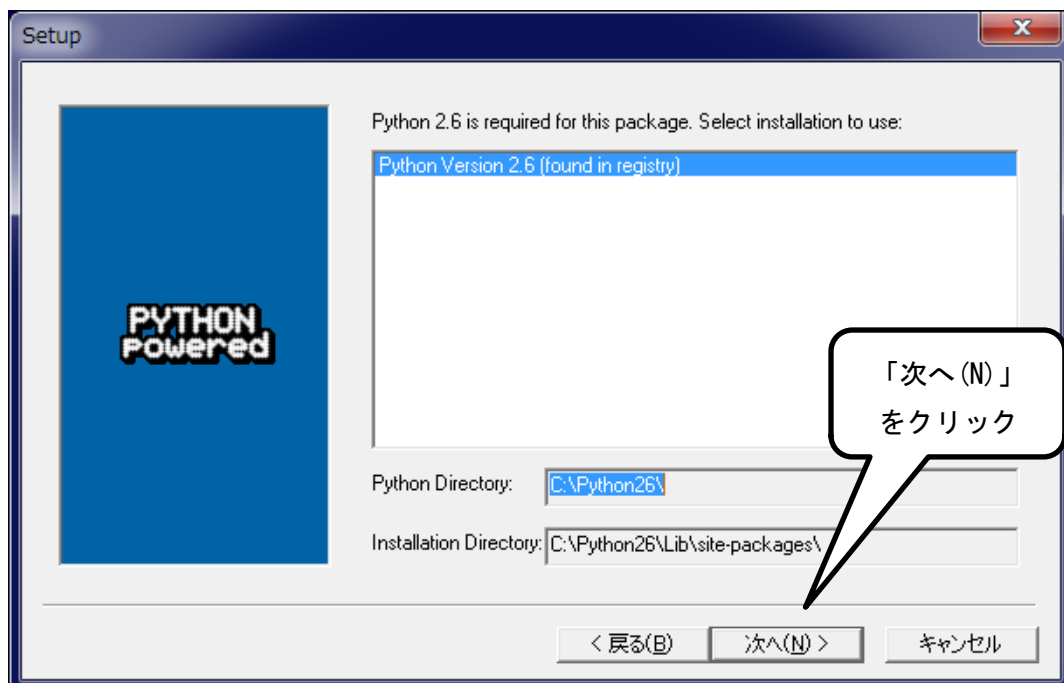


「実行(R)」
をクリック

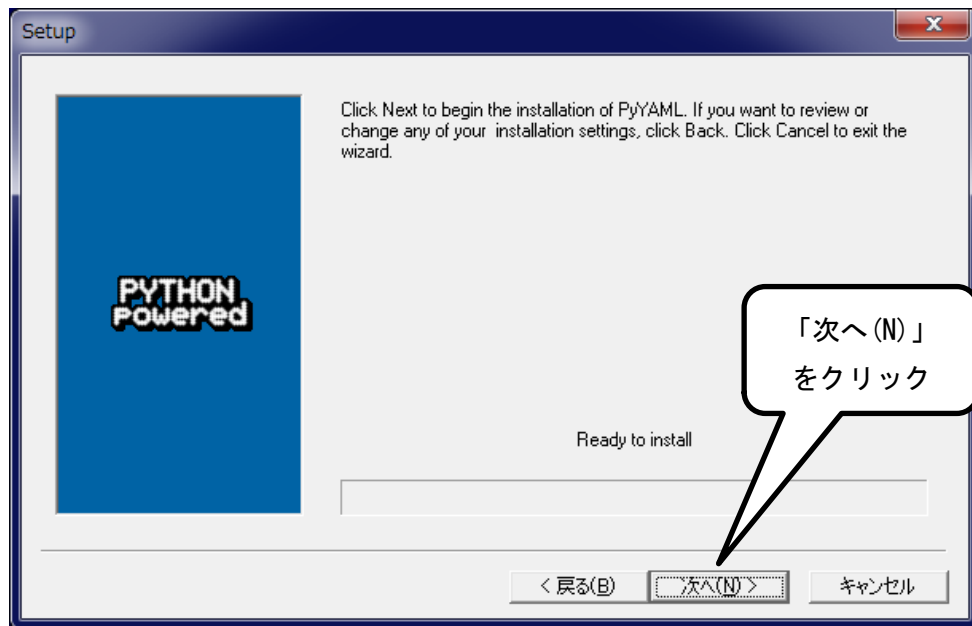
- ④ 画面全体が図のような画面に切り替わるので現れたウィンドウ上の「次へ(N)」をクリックしてください。



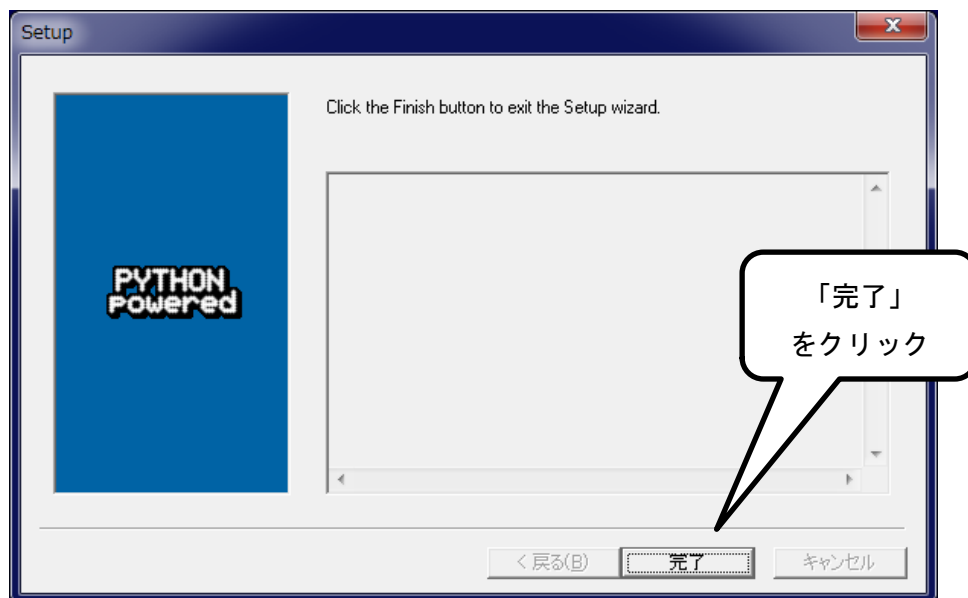
- ⑤ インストールする場所を指定します。特に理由がなければデフォルトで指定されている場所のまま変更を加えることなく、「次へ(N)」をクリックしてください。



- ⑥ 下図の画面になったら再び「次へ(N)」をクリックしてください。インストールが始まります。



- ⑦ インストールが終了すると以下のような画面となるので、そのまま「完了」をクリックしてインストーラを終了してください。



8. CMake-2.8 Visual

- ① 今度はダウンロードページで上から 6 番目の CMake-2.8 をインストールします。
「cmake-2.8.8」をクリックしてください。

Visual Studio 2010 (64bit) 用

Visual C++ 2010 Express版では(デフォルトでは)64bit バイナリのビルドができませんのでご注意ください。

| Visual Studio 2010 (64bit) 用 | | |
|---|---|----------------------------|
| Windowsインストーラ(64bit) | OpenRTM-aist-1.1.0-RELEASE_vc10_x64.msi MD5:24e2ac9673da4a3e9991e959d2d06f69 | 2012.05.25 |
| Java: Windows (64bit) 用 JRE | インストーラ | java.com |
| Python-64bit (omniidlでのみ必要) | python-2.7.3.amd64.msi | python.org |
| Python (各種ツールで必要) Python 2.x 利用可 | python-2.6.6.msi | python.org |
| PyYAML (各種ツールで必要) Pythonのバージョンに合わせることに必要 | PyYAML-3.10.win32-py2.6.exe | pyyaml.org |
| CMake-2.8 Visual (C++のプロジェクトを作成するのに必要) | cmake-2.8.8 | |
| Doxygen (ビルドの過程でドキュメントを整形するのに必要) | doxygen | |

「cmake-2.8.8」
をクリック

- ② 黄色い警告が現れたら「実行(R)」をクリックします。アカウント制御の認証ダイアログが出現したら許可しましょう。

www.cmake.org から cmake-2.8.8-win32-x86.exe (8.69 MB) を実行または保存しますか?

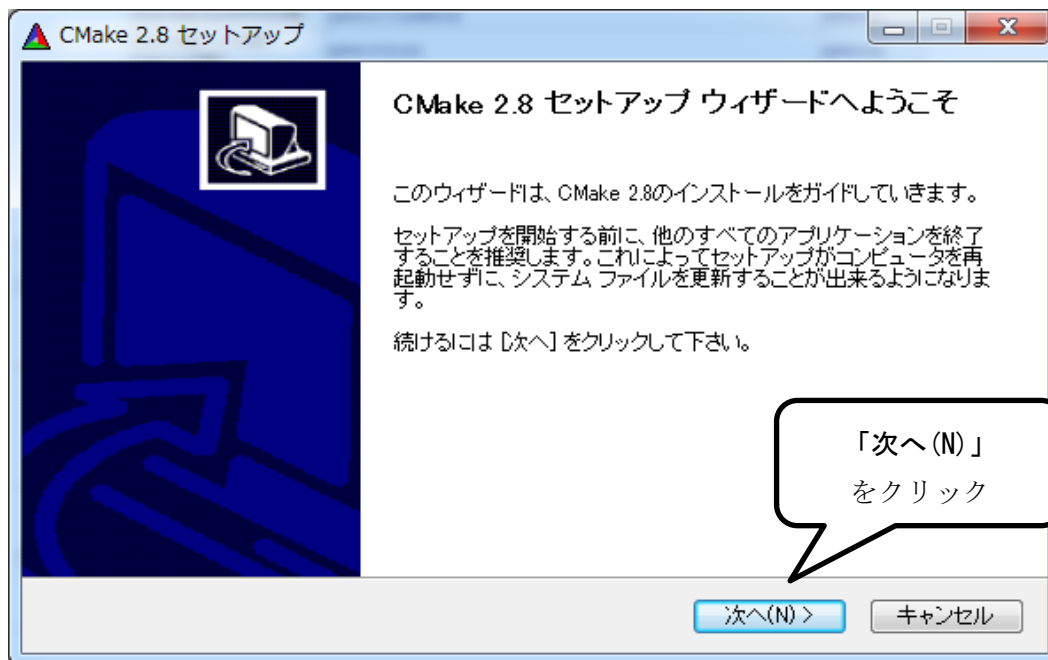
実行(R)

保存(S)

キャンセル(C)

x

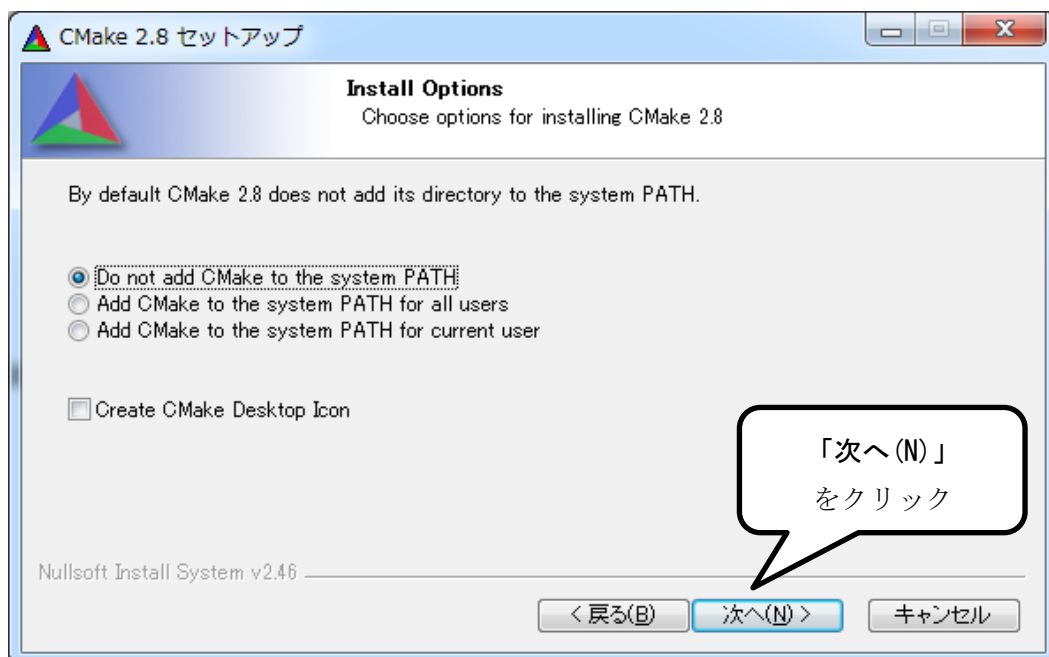
- ③ 下図のようにインストーラが起動したら「次へ(N)」をクリックしてください。



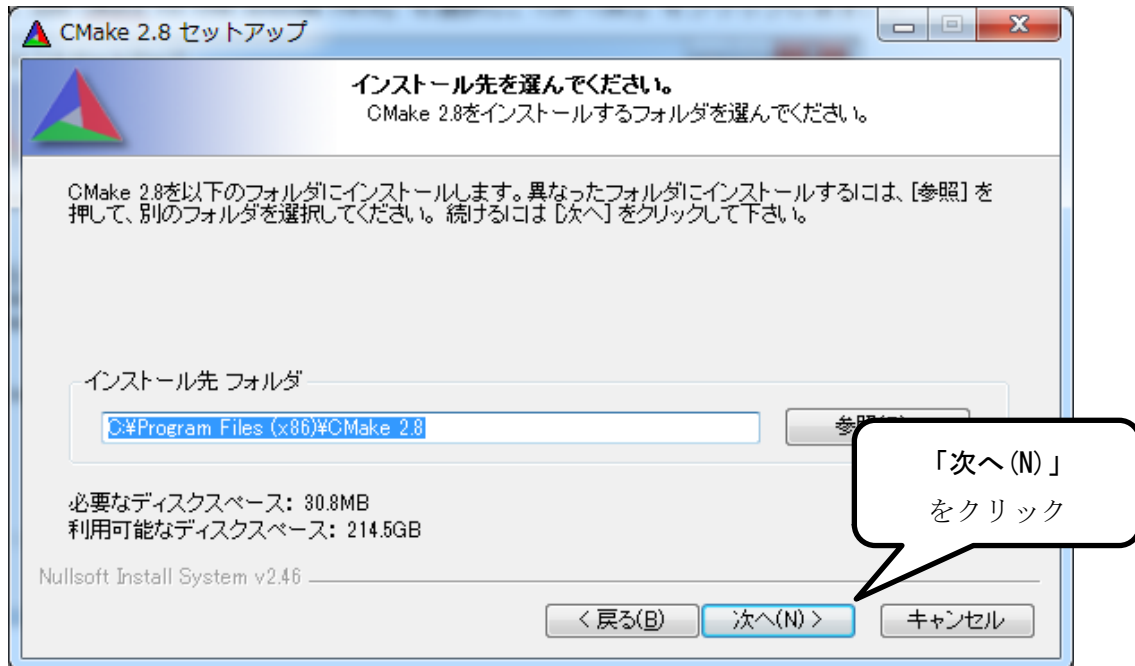
- ④ ライセンス契約書の内容を確認し、「同意する (A)」をクリックしてください。



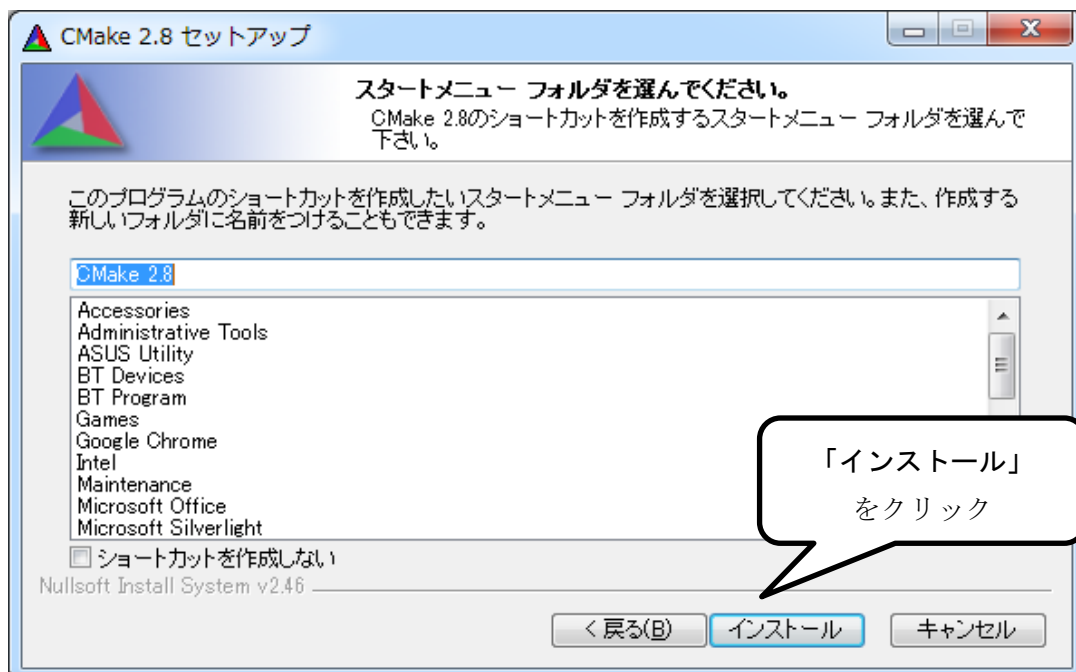
- ⑤ 「Do not add CMake to the system PATH」を選択し、「次へ (N)」をクリックします。



- ⑥ 次にインストールする場所を指定します。特に理由がなければデフォルトで指定されている場所のまま変更を加えることなく、「次へ(N)」をクリックしてください。



- ⑦ 次に以下のような画面になりますが、こちらもデフォルトのまま「インストール」をクリックしてください。



- ⑧ インストールが開始され、しばらくすると以下の画面になります。「完了(F)」をクリックしてインストーラを終了させてください。



9. Doxygen

- ① 今度はダウンロードページで一番下の Doxygen をインストールします。「doxygen」をクリックしてください。

Visual Studio 2010 (64bit) 用

Visual C++ 2010 Express版では(デフォルトでは)64bit バイナリのビルドができませんのでご注意ください。

| Visual Studio 2010 (64bit) 用 | | |
|--|---|----------------------------|
| Windowsインストーラ(64bit) | OpenRTM-aist-1.1.0-RELEASE_vc10_x64.msi MD5:24e2ac9673da4a3e9991e959d2d06f69 | 2012.05.25 |
| Java: Windows (64bit) 用 JRE | インストーラ | java.com |
| Python-64bit (omniidlでのみ必要) | python-2.7.3.amd64.msi | python.org |
| Python (各種ツールで必要) Python 2.x 利用可 | python-2.6.6.msi | python.org |
| PyYAML (各種ツールで必要) Pythonのバージョンに合わせる必要 | PyYAML-3.10.win32-py2.6.exe | pyyaml.org |
| CMake-2.8 Visual (C++のプロジェクトを作成するのに必要) | cmake-2.8.8 | cmake |
| Doxygen (ビルドの過程でドキュメントを整形するのに必要) | doxygen | doxygen |

「doxygen」
をクリック

- ② 黄色い警告が現れたら「実行(R)」をクリックします。アカウント制御の認証ダイアログが出現したら許可しましょう。

ftp.stack.nl から doxygen-1.8.1-setup.exe (15.0 MB) を実行または保存しますか?

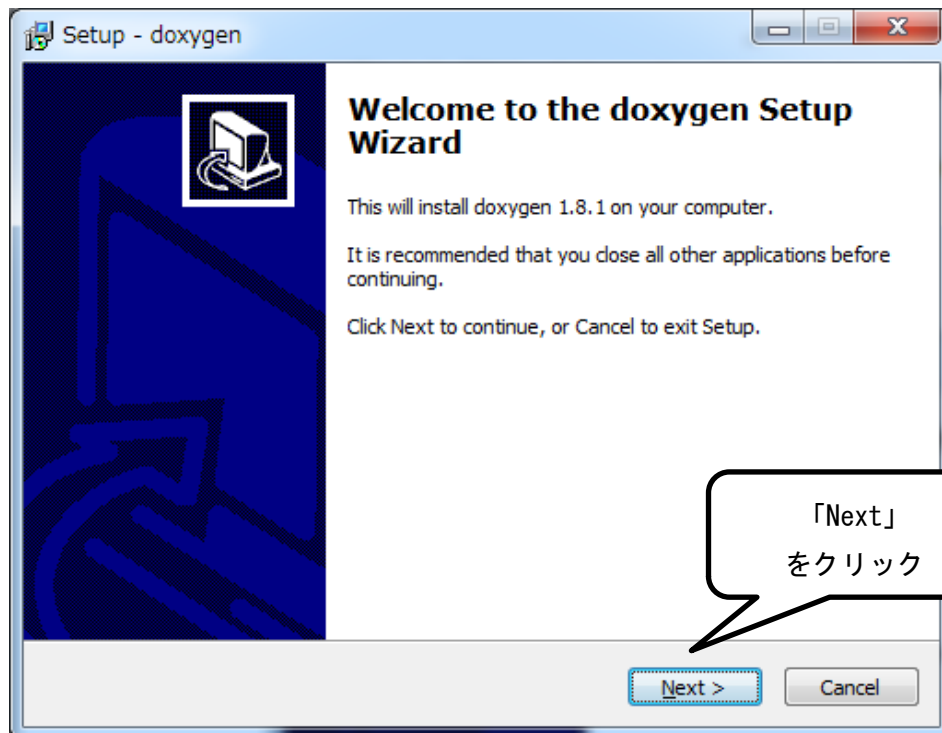
実行(R)

保存(S)

キャンセル(C)

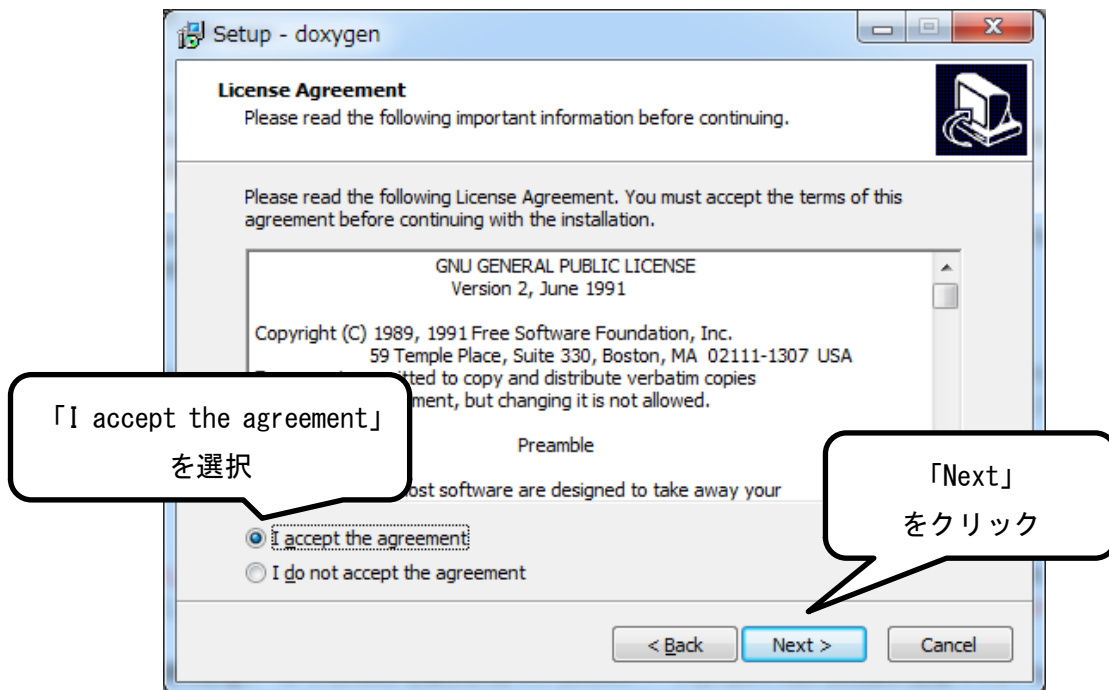


- ③ インストーラが起動し、以下のような画面が現れたら「Next」をクリックしてください。

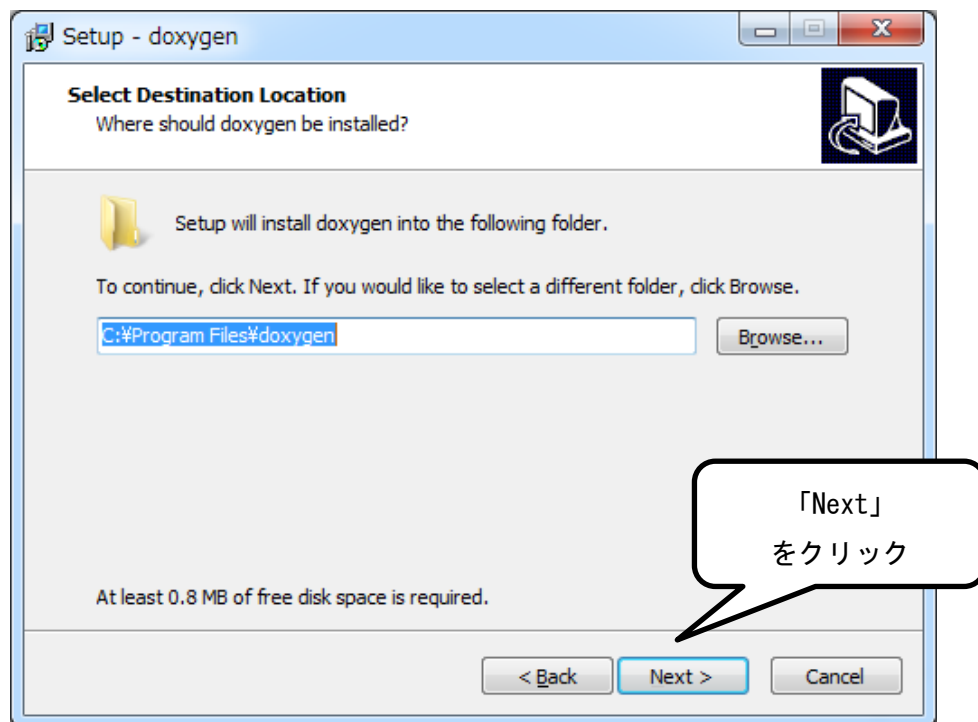


「Next」
をクリック

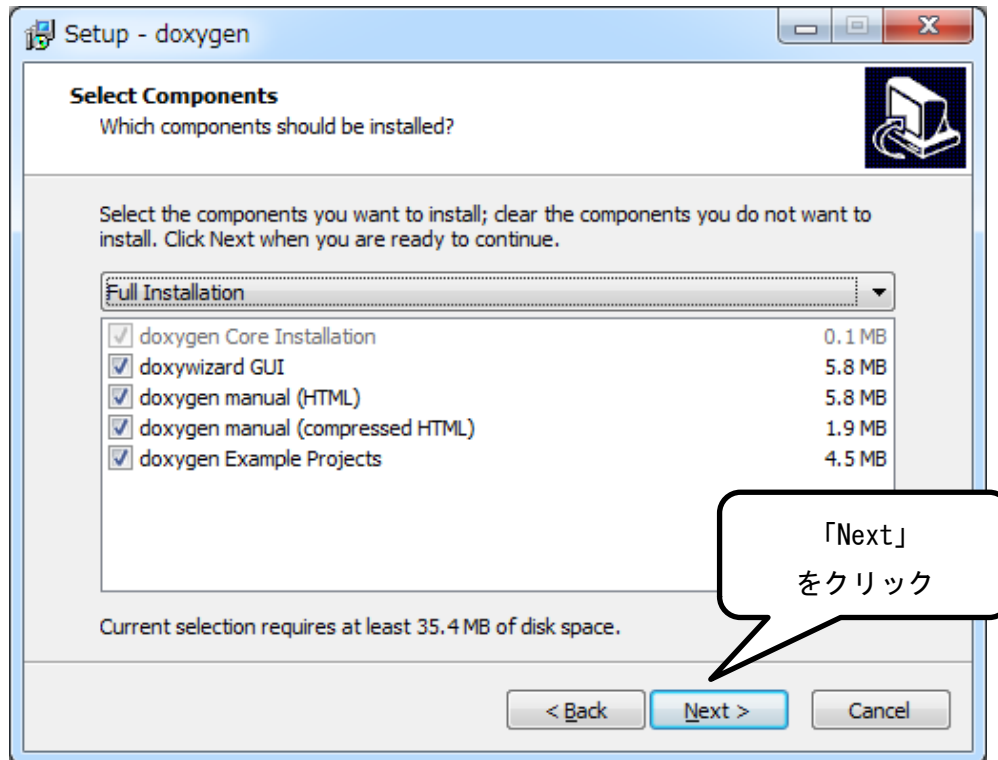
- ④ 以下のような画面が現れるので、確認したら「I accept the agreement」を選択し「Next」をクリックしてください



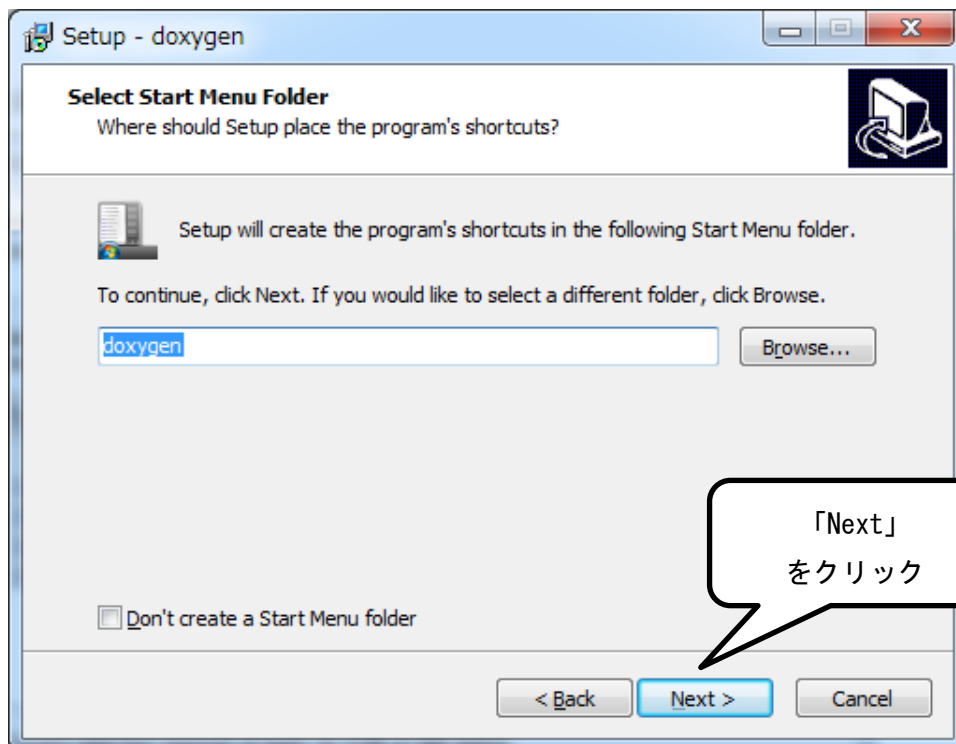
- ⑤ インストールする場所を指定します。特に理由がなければデフォルトで指定されている場所のまま変更を加えることなく、「Next」をクリックしてください。



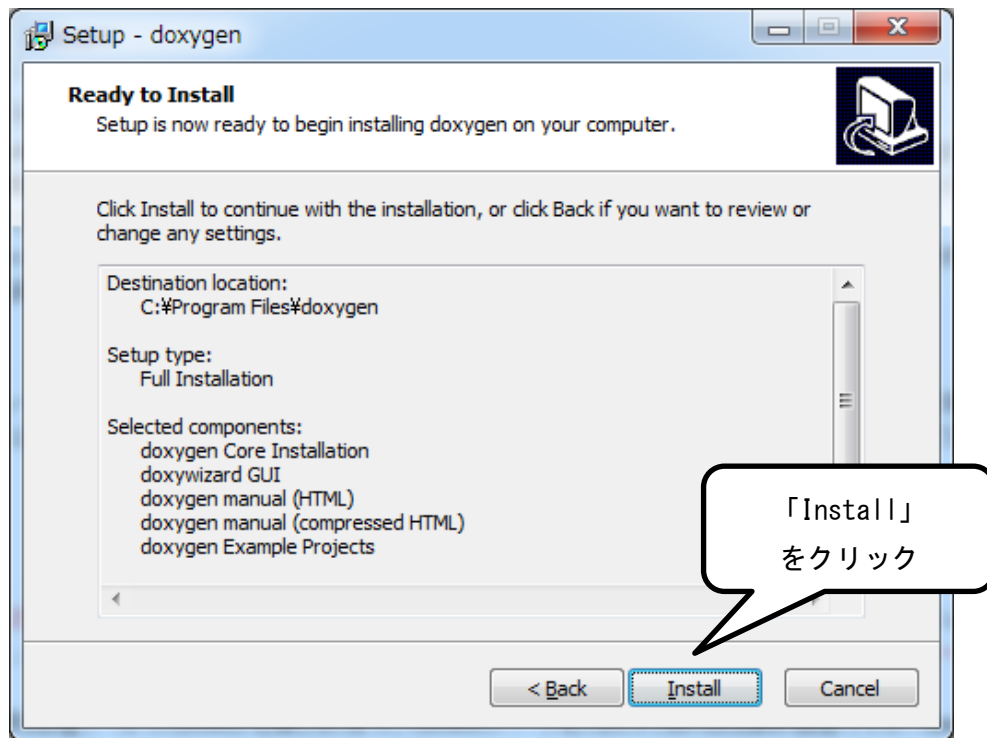
- ⑥ 次に現れる下図の画面でも同様にデフォルトのまま「Next」をクリックしてください。



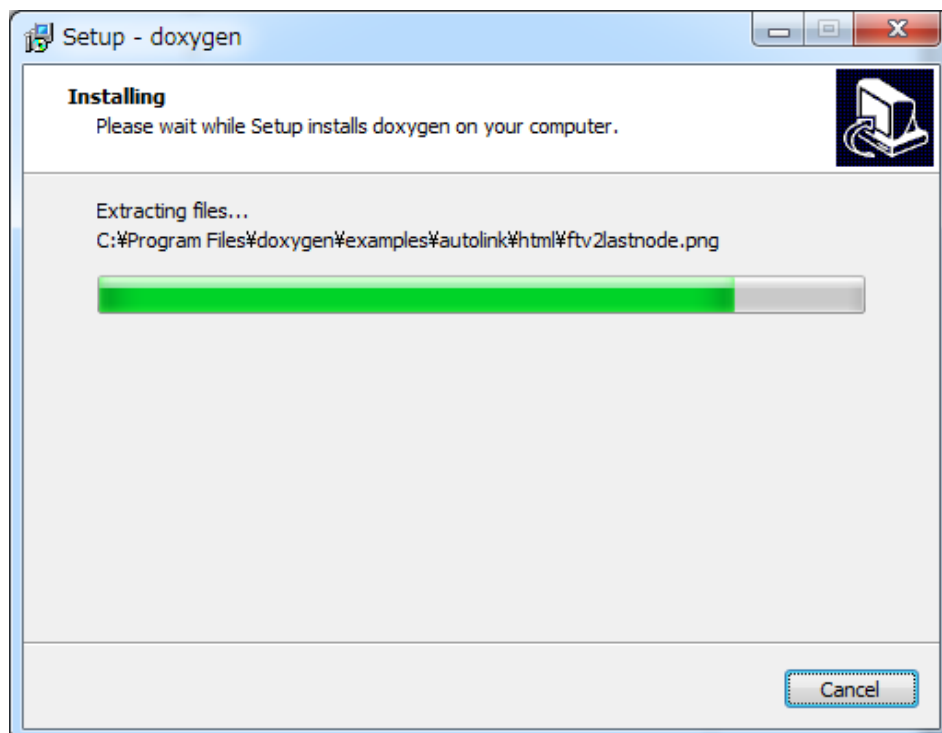
- ⑦ 次に以下のような画面になりますが、こちらもデフォルトのまま「Next」をクリックしてください。



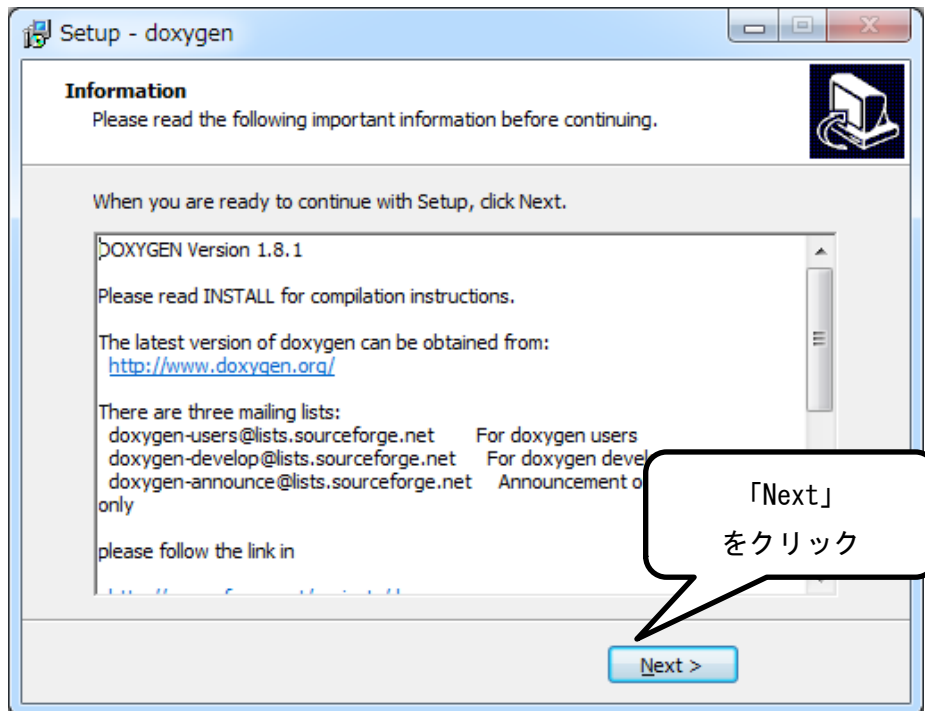
- ⑧ この画面になったら「Install」をクリックしてください。インストールが始まります。



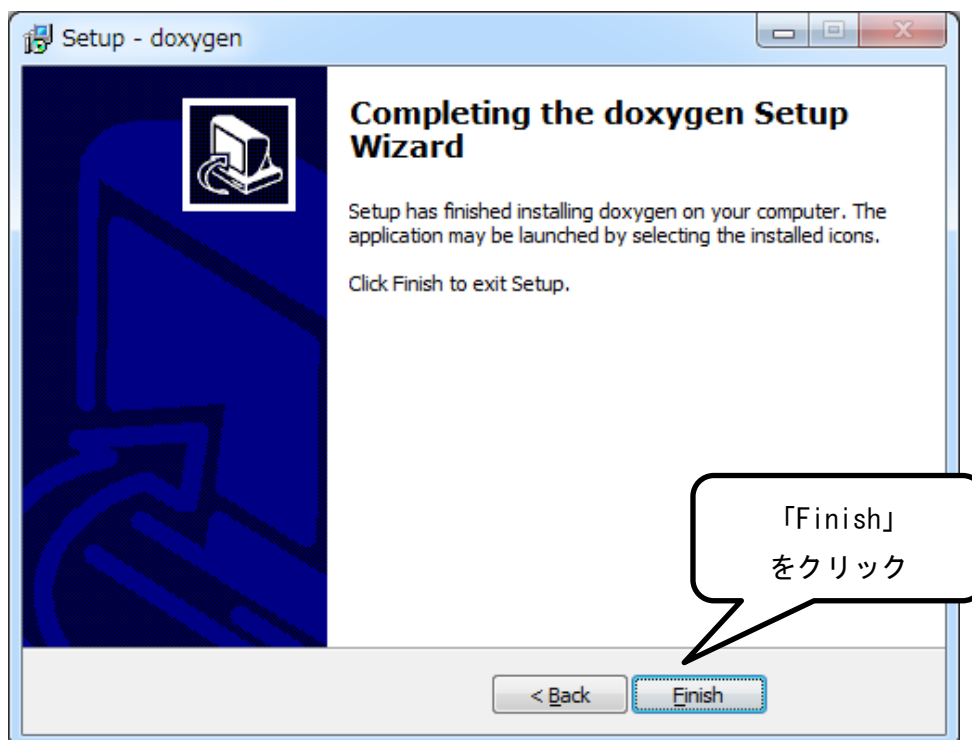
- ⑨ インストールを行っている最中の画面です。少し待ちましょう。



- ⑩ 現れた情報を確認し「Next」をクリックしてください。



- ⑪ インストールが開始され、暫くすると以下の画面になります。最後に「Finish」をクリックしてインストーラを終了させてください。



10. Java (32bit)

- ① 32bit の Java をインストールします。既にバージョン 7.0 以上がインストールされている場合は読み飛ばして 11. を参照しましょう。ダウンロード箇所は「2. ダウンロードページへの移動」で行き着いたページ上の「Windows 版インストールにあたっての注意事項」にあります。2 番目で示されている「Java のダウンロード」をクリックしましょう。

Windows版インストールにあたっての注意事項

- ※ Visual Studio 2008とVisual Studio 2010でビルドしたものは異なできません。お使いの開発環境に合わせて適切なパッケージをダウンロードしてください。
- 同梱の RTSystemEditor(RCP 版) を動作させるために、Java 動作環境 (JRE) または Java 開発環境 (JDK) が必要となります。少なくともJREは必ずインストールしてください。
 - Java のダウンロード
- RTコンポーネントの開発を行うためには、ビルドに必要な各種ツールでPythonが必要となります。Pythonのバージョンは2.xであれば比較的自由ですが、OpenRTM-aist-PythonのPythonバージョンに合わせておくといでしょう。

「Java のダウンロード」
をクリック

- ② Java のダウンロードページに飛びますので「Windows オフライン (32 ビット)」をクリックしてください。(64bit では後述する eclipse が動作しません)

全オペレーティング・システムのJavaのダウンロード一覧

推奨 Version 7 Update 9

下の一覧で、お使いのオペレーティングシステムに適した最新の Java ファイルを選択してください。

> 旧バージョンの削除 > Javaとは

Java をダウンロードすることで、エンドユーザー

Windows どれを選択すればよ

今すぐチェック

Windows オフライン (32ビット) 手順
ファイルサイズ: 29.7 MB

Windows オフライン (64ビット) 手順
ファイルサイズ: 31.1 MB

このページを32ビット版のブラウザで閲覧している可能性が検知されました。32ビットと64ビットのブラウザを切り替えて使用している場合、両方のブラウザでJava Plug-inを使用できるように、32ビットと64ビットの両方のJavaをインストールする必要があります。 » Windows版64ビットJavaに関するFAQ

- ③ 黄色い警告が現れたら「実行(R)」をクリックします。アカウント制御の認証ダイアログが出現したら許可しましょう。

sdic-esd.sun.com から jre-7u9-windows-i586.exe (29.7 MB) を実行または保存しますか?

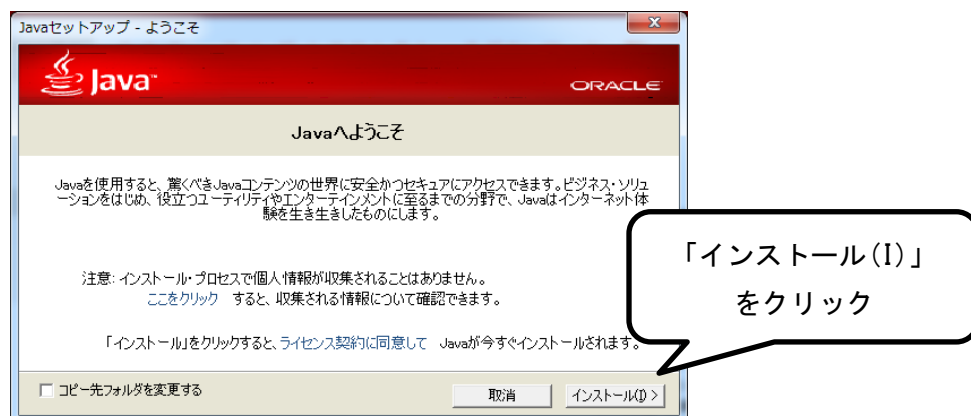
実行(R)

保存(S)

キャンセル(C)

x

- ④ 以下のような画面が現れるので「インストール(I)」をクリックしてください。



- ⑤ インストールが始まります。また少し待ちましょう。



- ⑥ インストールが終了すると以下のような画面が表示されますので「閉じる(C)」をクリックしてインストーラを終了させてください。



11. Eclipse

- ① 最後に Eclipse をインストールします。ダウンロード箇所は「2. ダウンロードページへの移動」で見つけたページの下部の「ツール」にあります。今回は一番上の「Windows 用全部入り」をダウンロードしましょう。「eclipse342_rtmtools110-rc3_win32_ja.zip」をクリックしてください。

ツール

「eclipse342_rtmtools110-rc3_win32_ja.zip」
をクリック

| Eclipse | Tool | Date |
|--|---|------------|
| Eclipse3.4.2+RTSE+RTCB Windows用全部入り | eclipse342_rtmtools110-rc3_win32_ja.zip MD5:d79780ecb24afee725ca2d9a858a3629 | 2012.05.24 |
| Eclipse3.4.2+RTSE+RTCB Linux用全部入り | eclipse342_rtmtools110-rc3_linux_ja.tar.gz MD5:f1eee7758b81a99c417d0cc26c9af118 | 2012.05.24 |
| Eclipse3.4.2+RTSE+RTCB MacOSX用全部入り | eclipse342_rtmtools110-rc3_macosx_ja.tar.gz MD5:37b28a1454ab95290d0d7431cede52ef | 2012.05.24 |
| Eclipse3.4.2+RTSE+RTCB 64bit版Linux用全部入り | eclipse342_rtmtools110-rc3_x86_64_ja.tar.gz MD5:418643f102e78845d29220b778a87477 | 2012.05.24 |

- ② 下図のようなウィンドウが表示されるので「保存(S)」あるいは「名前を付けて保存(A)」をクリックして任意の場所へ保存してください。この解説では「保存(S)」をクリックしました。

Windows Internet Explorer

eclipse342_rtmtools110-rc3_win32_ja.zip に対して行う操作を選択してください。

サイズ: 164 MB
サイト: www.openrtm.org

→ 開く(O)
ファイルは自動保存されません。

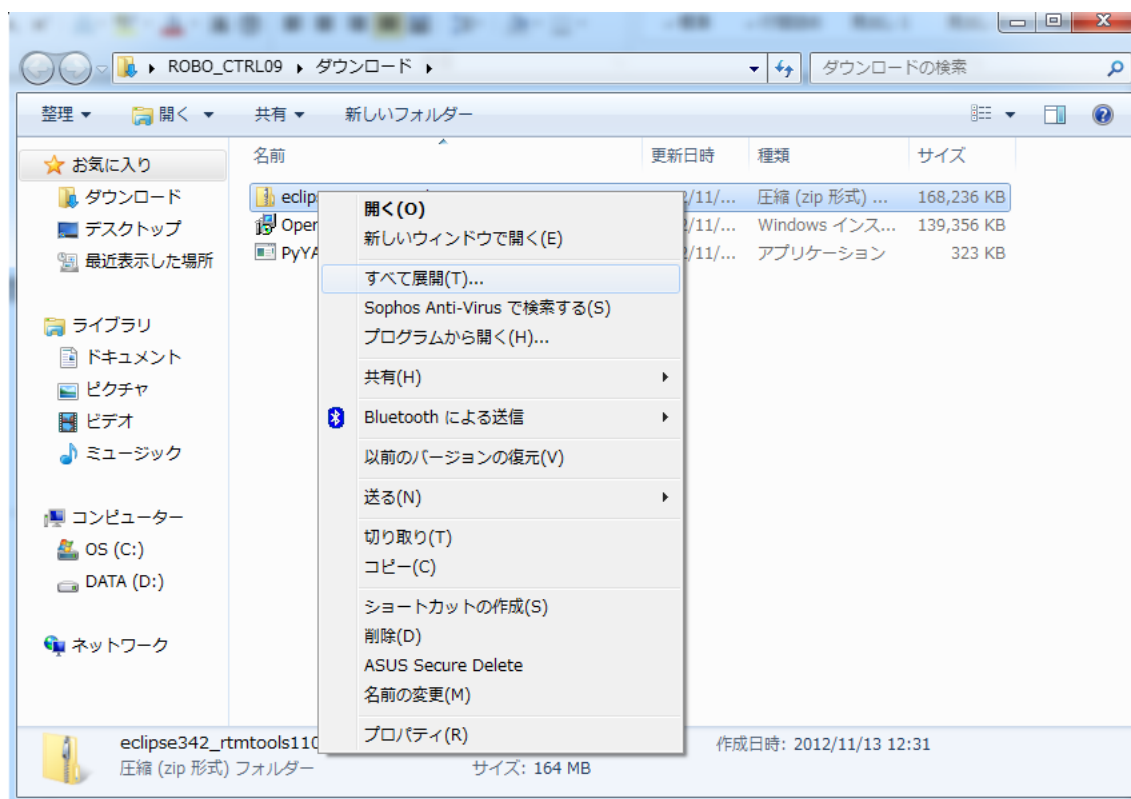
→ 保存(S)

→ 名前を付けて保存(A)

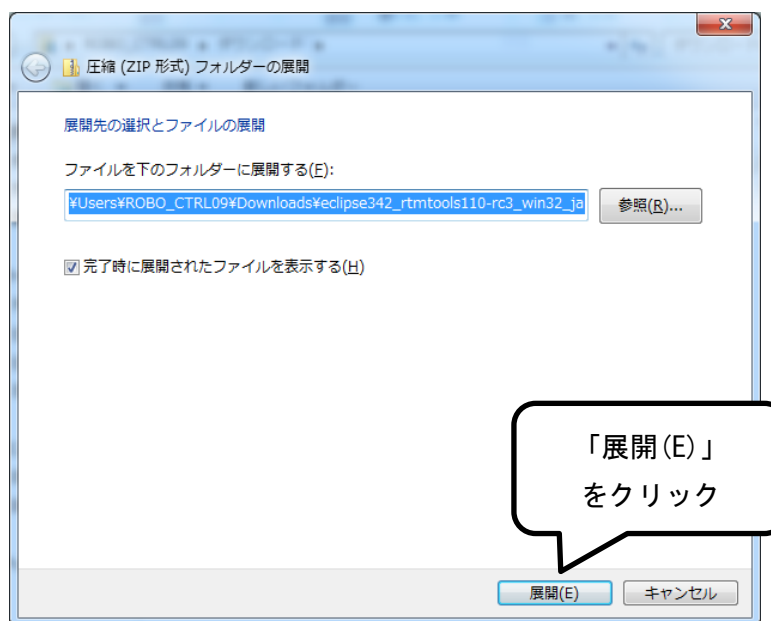
キャンセル

「保存(S)」
あるいは
「名前を付けて保存(A)」
をクリック

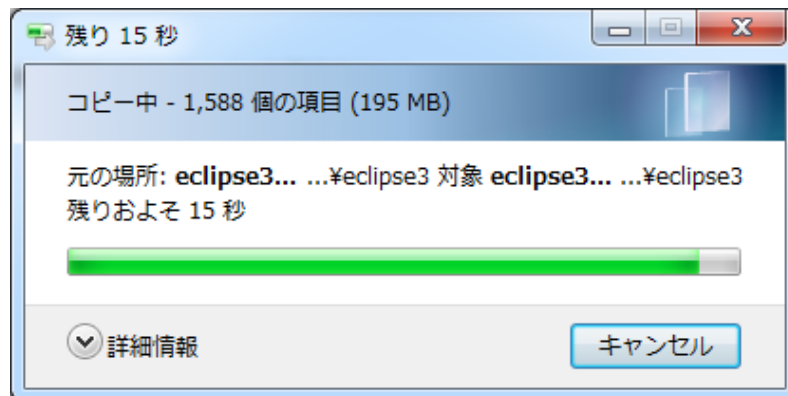
- ③ ダウンロードしたディレクトリで「eclipse342_rtmtools110-rc3_win32_ja.zip」を右クリックしてください。現れた一覧の中から「すべて展開」を選択しましょう。



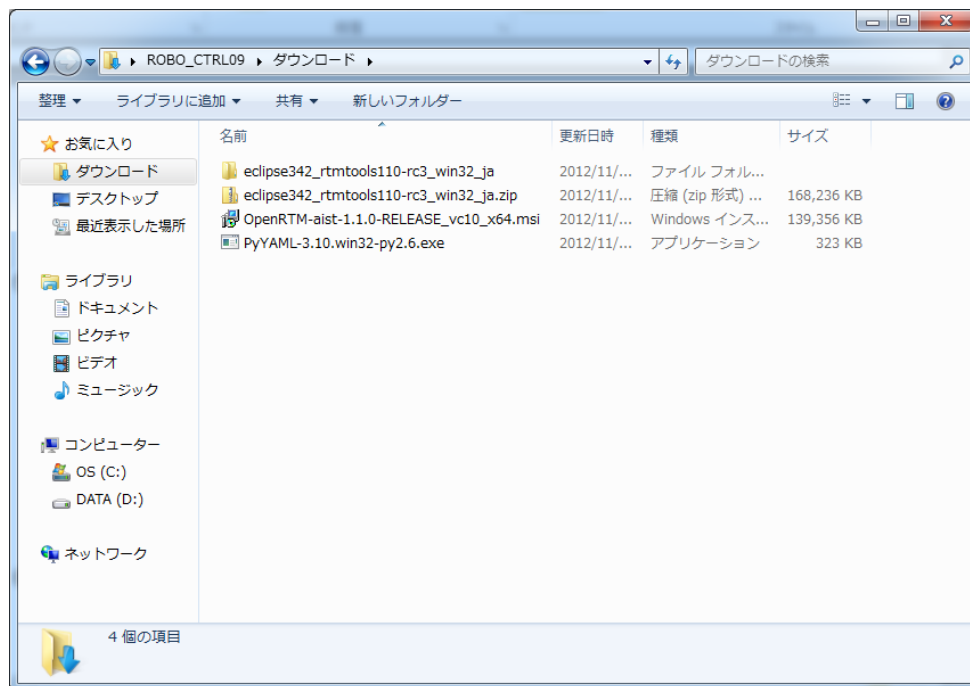
- ④ 下図のようなウィンドウが現れたら展開したいディレクトリを指定して「展開(E)」をクリックします。この解説ではデフォルトで示されているダウンロードディレクトリに展開しました。



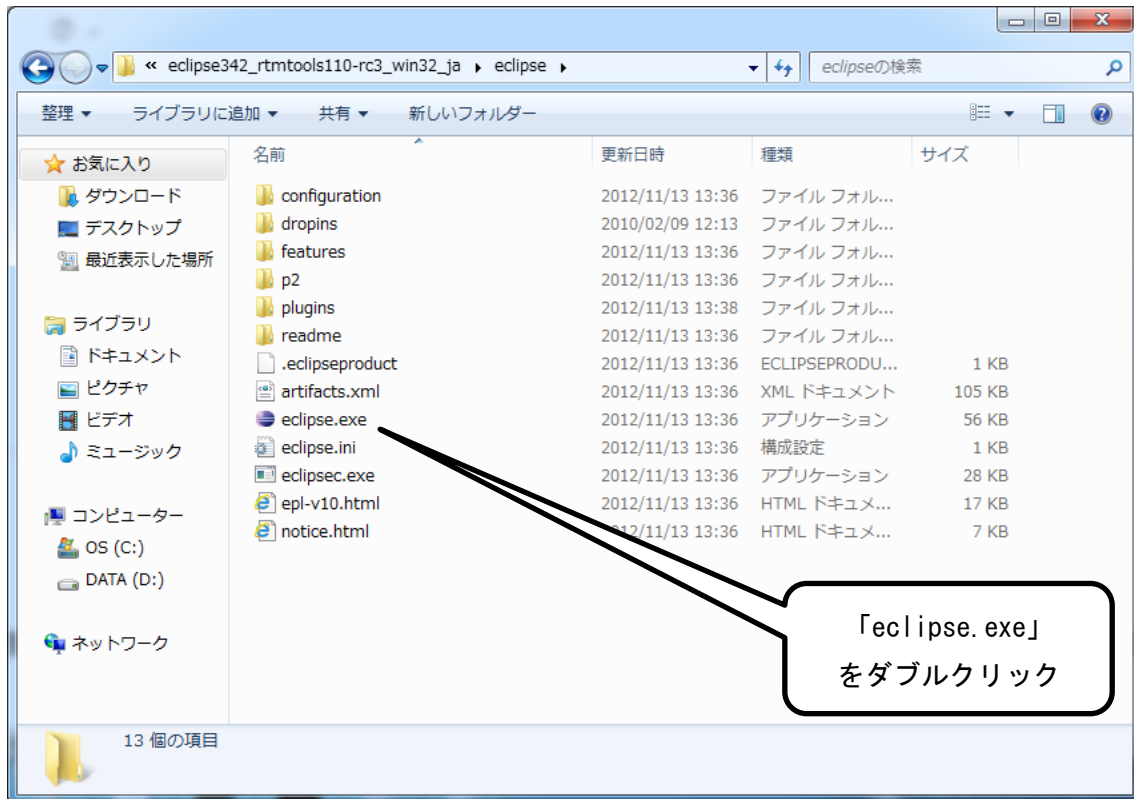
- ⑤ 下図のように展開が行われます。



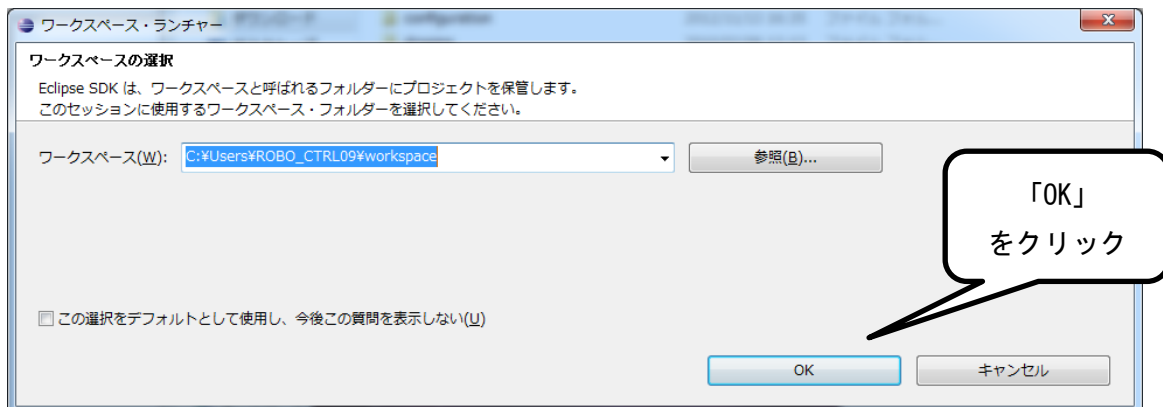
- ⑥ 展開が完了すると、新しく「eclipse342_rtmtools110-rc3_win32_ja」というフォルダが現れます。その中の「eclipse」フォルダ以内に移動してください。



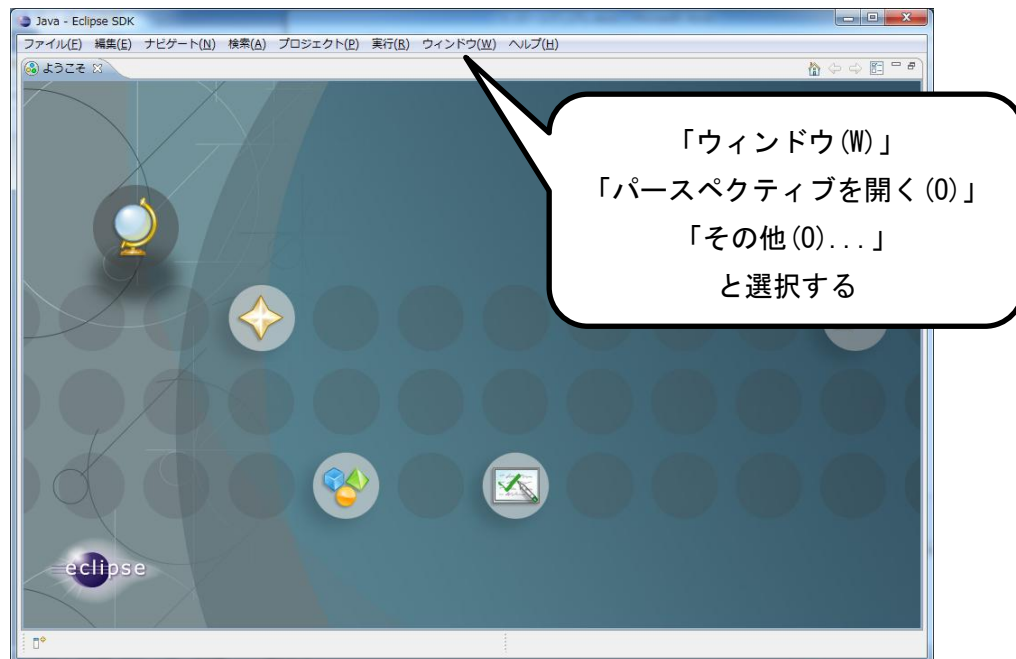
- ⑦ フォルダ内にある「eclipse.exe」をダブルクリックして起動させてください。



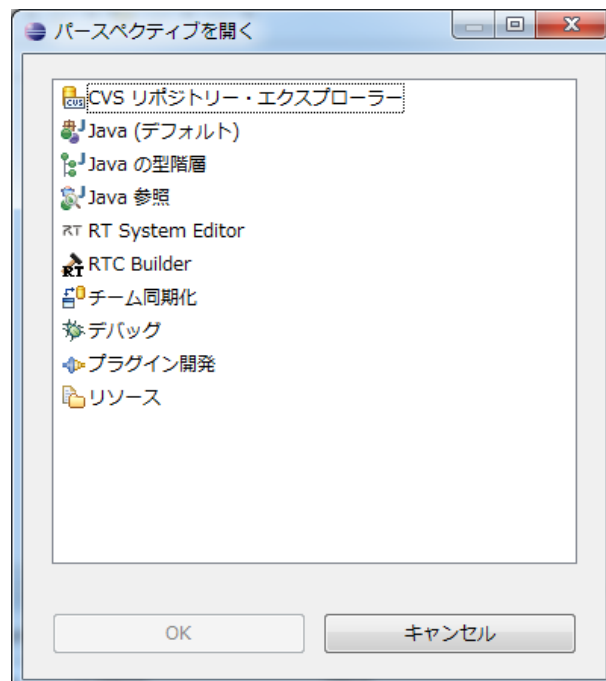
- ⑧ 下図のようなウィンドウが立ち上がるので「OK」をクリックします。



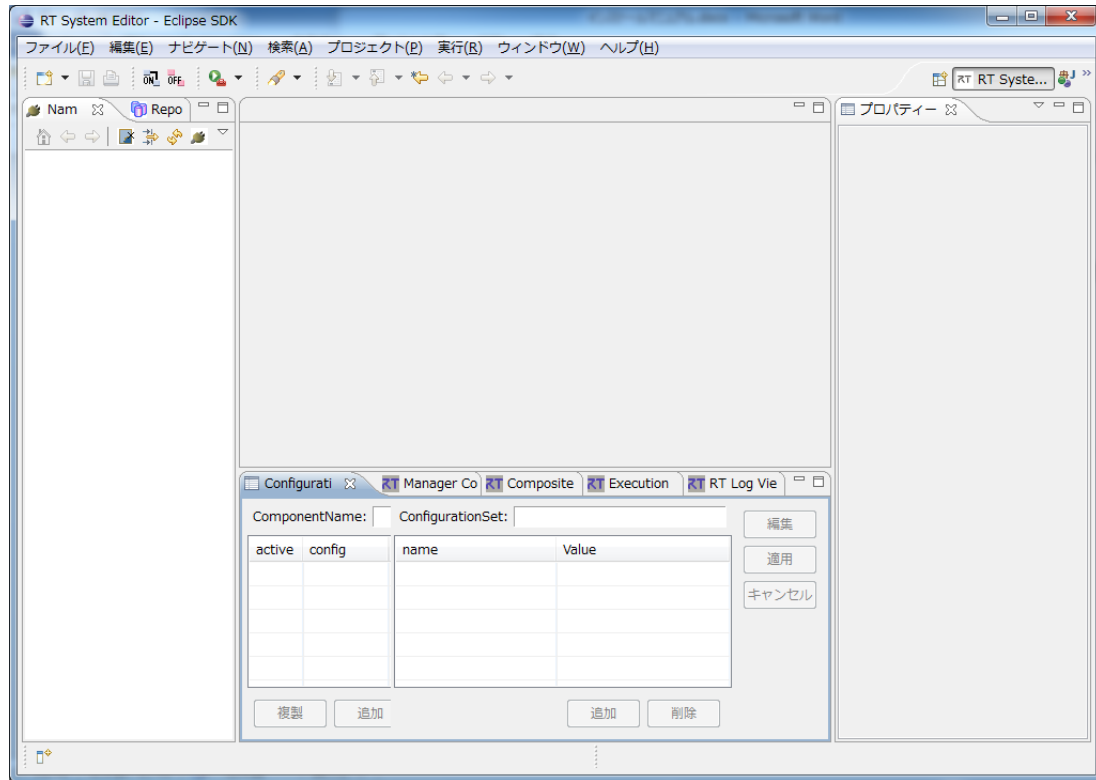
- ⑨ 下図のような画面となるので上部の「ウィンドウ (W)」から「パースペクティブを開く (O)」「その他 (O)…」と選択してゆきます。



- ⑩ 現れた一覧の中から「RT System Editor」や「RTC Builder」を選択することで RT コンポーネントの利用や開発が出来ます。まずは「RT System Editor」を立ち上げてみましょう。



- ⑪ 下図のような画面となればインストールは成功です。以降の動作確認はこの画面上で行うので、このウィンドウはそのままの状態を読み進めてください。



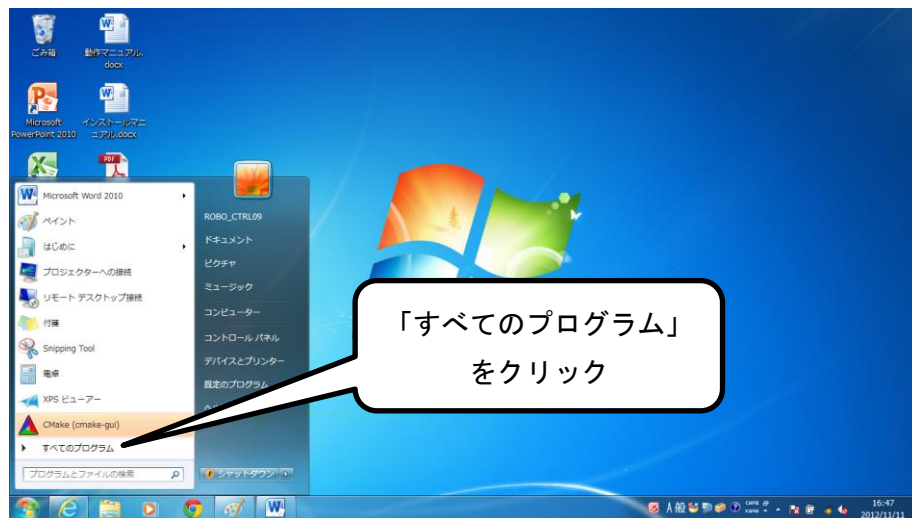
お疲れ様でした、必要なインストールは以上で終了です。

以降は実際に RT ミドルウェアを利用して、サンプルコンポーネントの動作確認をします。

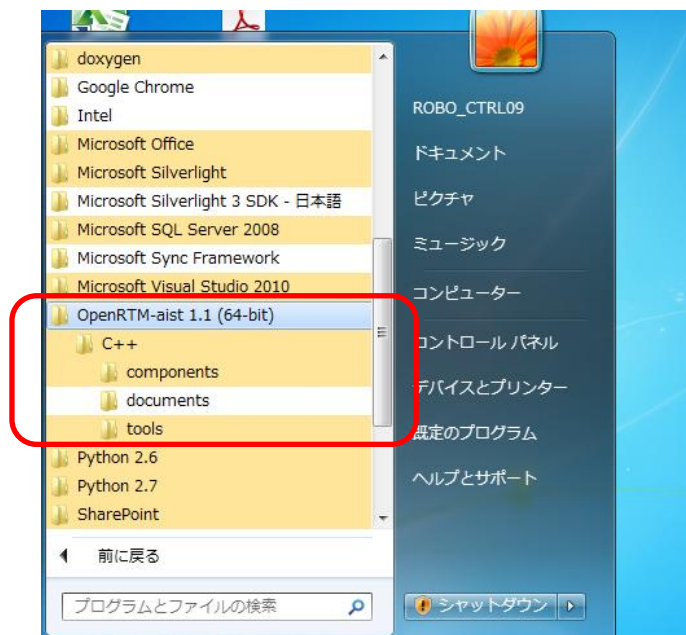
12. 動作確認（準備編）

OpenRTM-aist1.1.0 インストールすると、いくつかのサンプルコンポーネントが付属しています。そこで、このサンプルを用いて設定した環境が正常に動作するかテストしてみましょう。今回行うのは最も基本的な数字の送受信です。

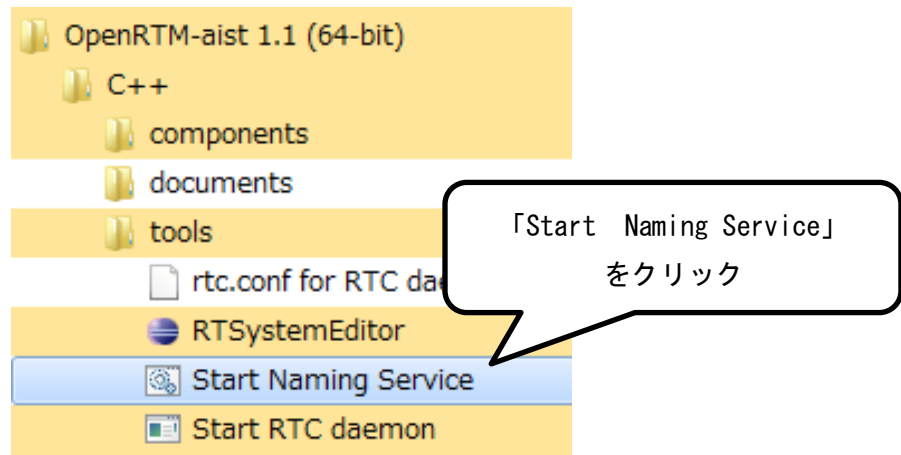
- ① デスクトップ上で画面左下にあるウィンドウズマークをクリックし、以下のような画面で「すべてのプログラム」をクリックします。



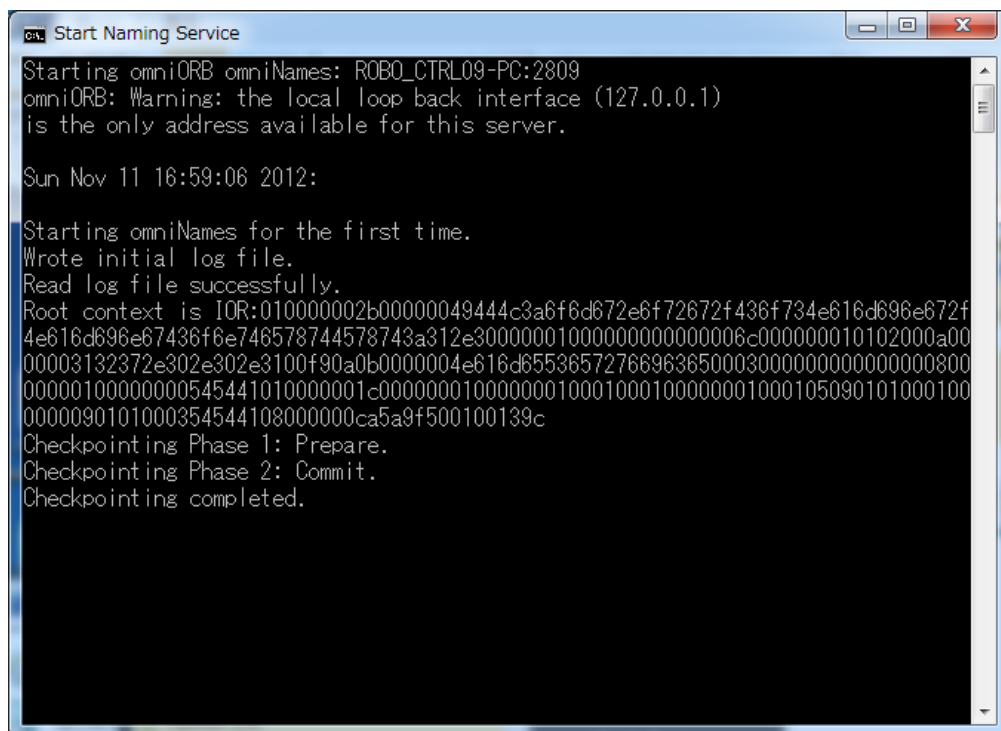
- ② 現れた一覧の中から「OpenRTM-aist 1.1(64-bit)」を選択し、その中の「C++」を選択すると以下に「components」「documents」「tools」が見つかります。



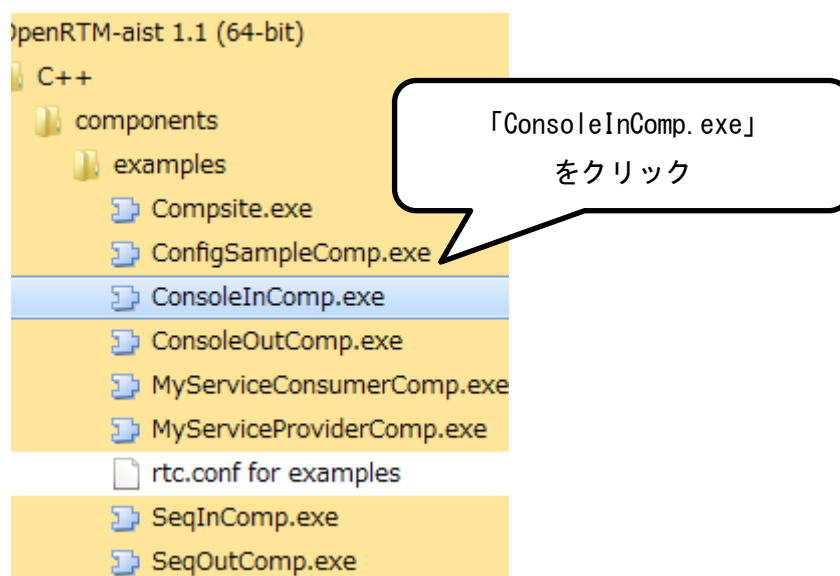
- ③ まずはネーミングサーバを立ち上げます。先ほどのフォルダから「tools」内を開き「Start Naming Service」をクリックして起動します。



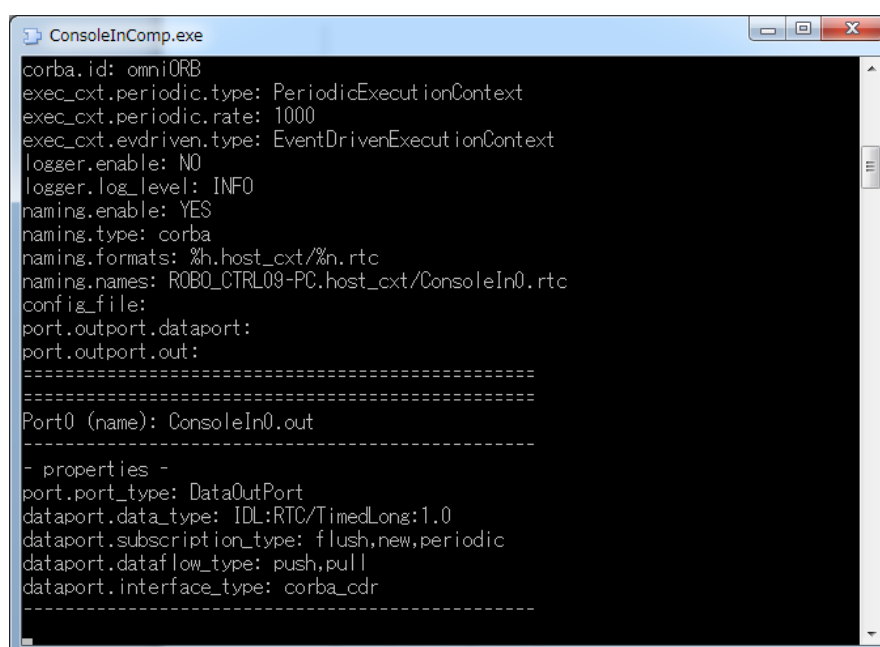
- ④ 以下のようなウィンドウが立ち上がりますので、邪魔にならないようにそのまま最小化しておきましょう。以降、RT コンポーネントを利用しているときは、このウィンドウを閉じてはいけません。



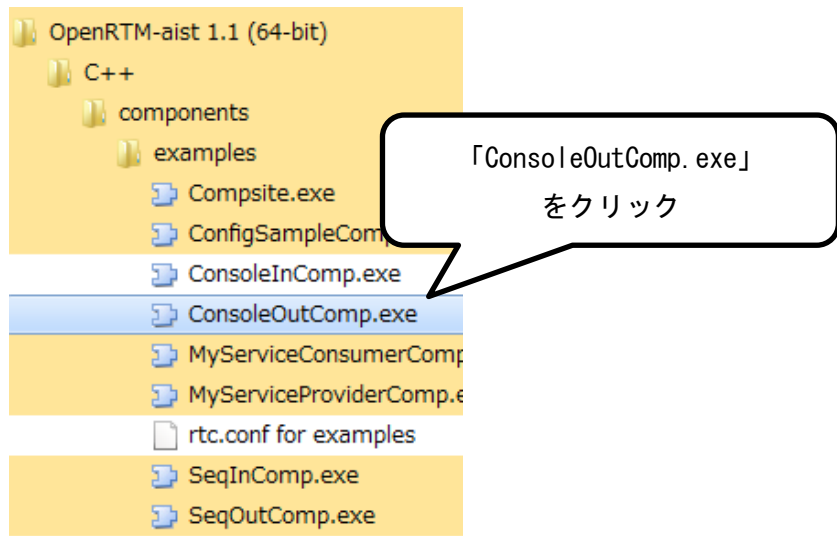
- ⑤ 次に、サンプルのプログラムである ConsoleIn を起動します。先ほどのフォルダから「components」内の「example」を開き、「ConsoleInComp.exe」をクリックして起動します。



- ⑥ プログラムを起動する際、アクセスの許可を求められる場合は許可してください。以下の画面が表示されれば起動は完了です。ウィンドウはそのままの状態にしておきましょう。



- ⑦ 次に、ConsoleOut を起動させます。以下に示す先ほど同様のディレクトリ上で「ConsoleOutComp.exe」をクリックして起動します。



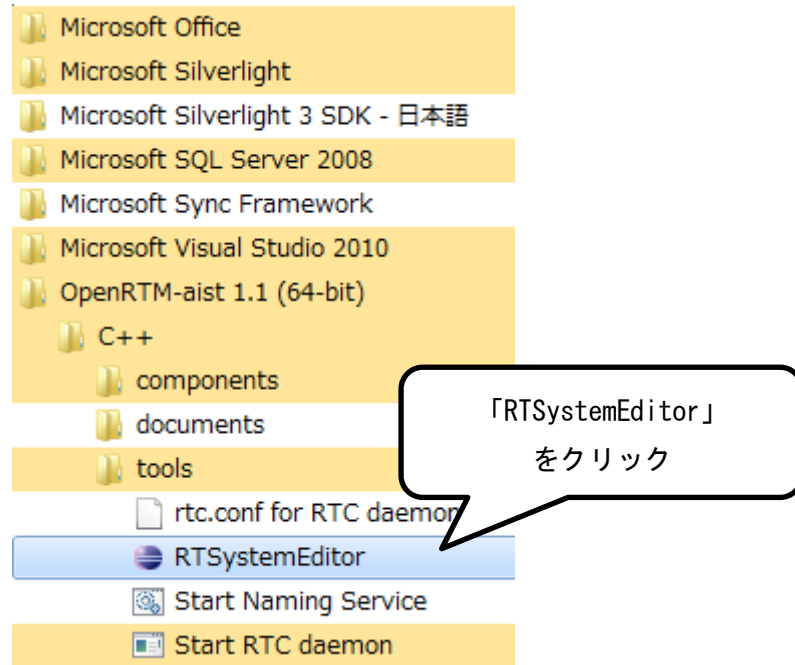
- ⑧ プログラムを起動する際、アクセスの許可を求められる場合は許可してください。以下の画面が表示されれば起動は完了です。ダイアログはそのままの状態にしておきましょう。

```

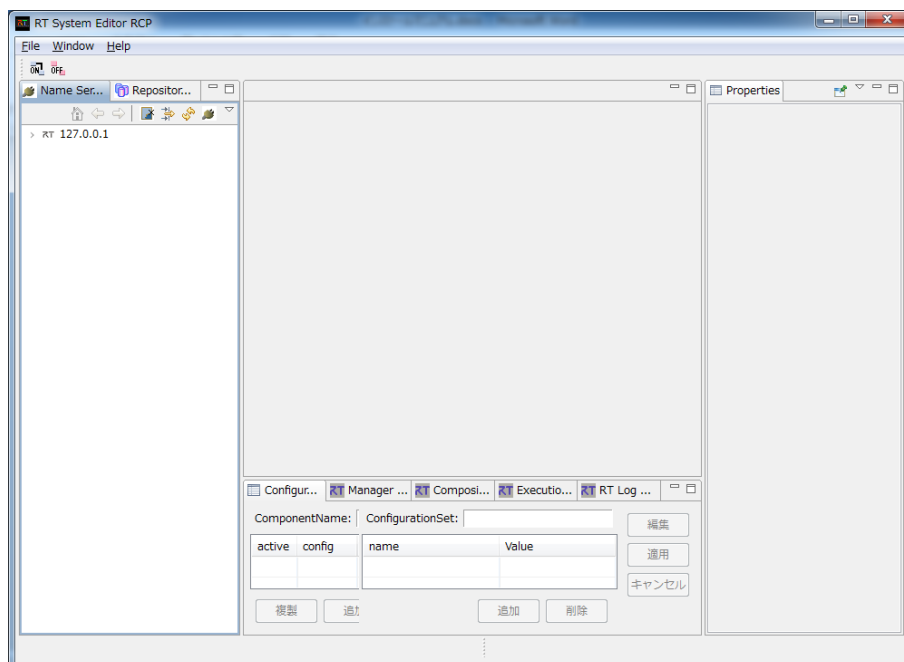
ConsoleOutComp.exe
corba.id: omniORB
exec_cxt.periodic.type: PeriodicExecutionContext
exec_cxt.periodic.rate: 1000
exec_cxt.evdriven.type: EventDrivenExecutionContext
logger.enable: NO
logger.log_level: INFO
naming.enable: YES
naming.type: corba
naming.formats: %h.host_cxt/%n.rtc
naming.names: ROBO_CTRL09-PC.host_cxt/ConsoleOut0.rtc
config_file:
port.inport.dataport:
port.inport.in:
=====
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
=====

```

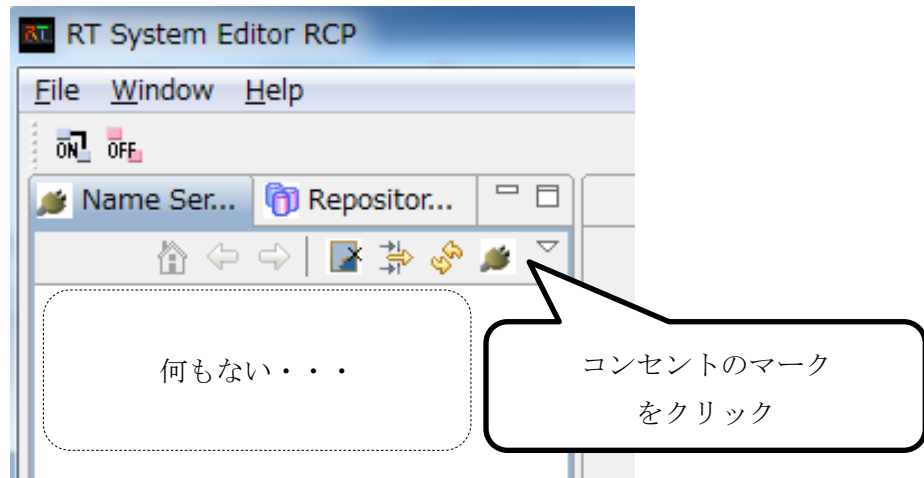
- ⑨ 次に, RT System Editor を起動させます. 開発環境をインストールする過程で RT System Editor を既に起動している場合はそのまま次の⑩に進んでください. そうでない場合は先ほどのフォルダから「tools」内の「RTSystemEditor」をクリックして起動します.



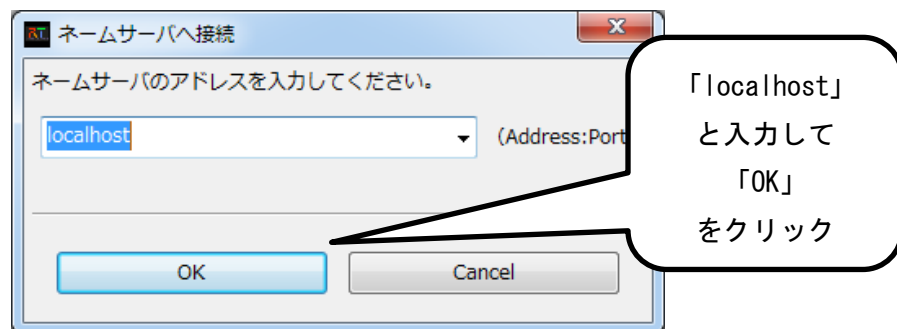
- ⑩ プログラムを起動する際, アクセスの許可を求められる場合は許可してください. RTSystemEditor が起動すると以下のようなウィンドウが立ち上がります.



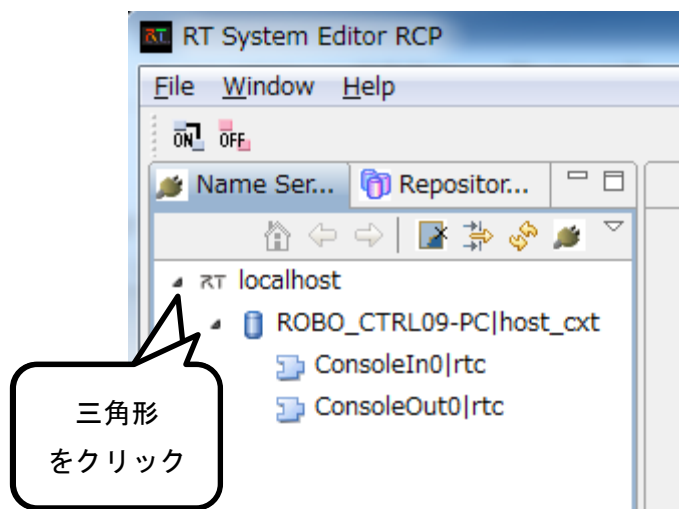
- ⑪ 下図のように、ウィンドウ左上の部分において何も表示がない場合はコンセン트의マークをクリックしてください。



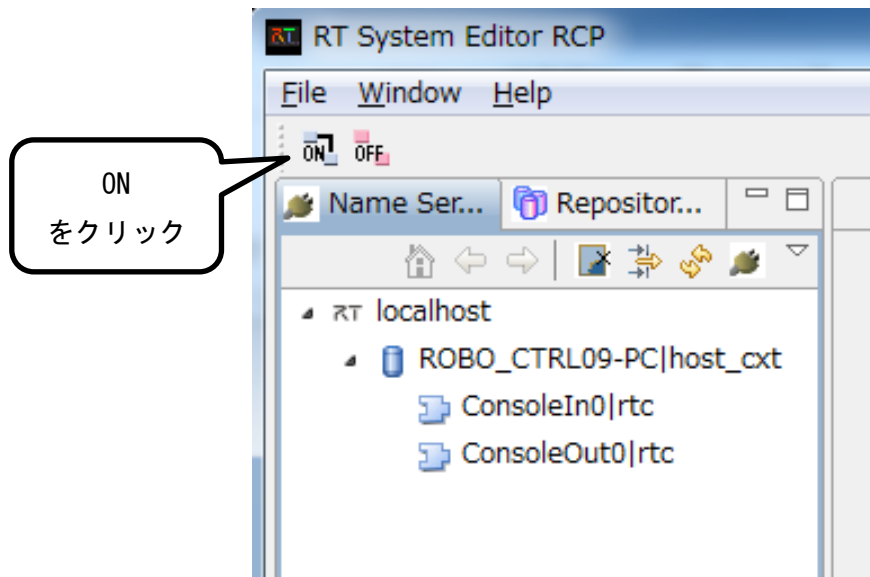
- ⑫ 現れたウィンドウ上で「localhost」と入力して「OK」をクリックしてください



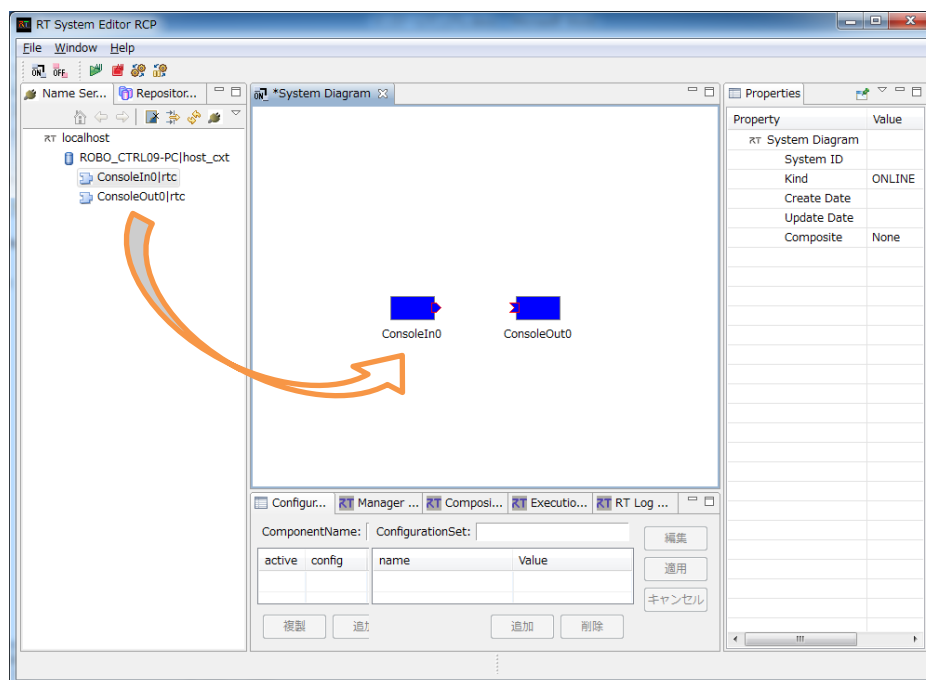
- ⑬ 現れた「localhost」左側の三角形をクリックします。先ほど立ち上げた「ConsoleIn」と「ConsoleOut」が一覧に存在することを確認してください。



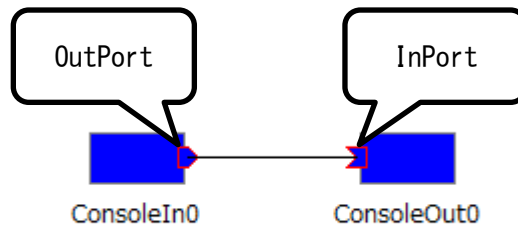
- ⑭ 次に、以下の画面で示している「ON」と書かれたマークをクリックします。



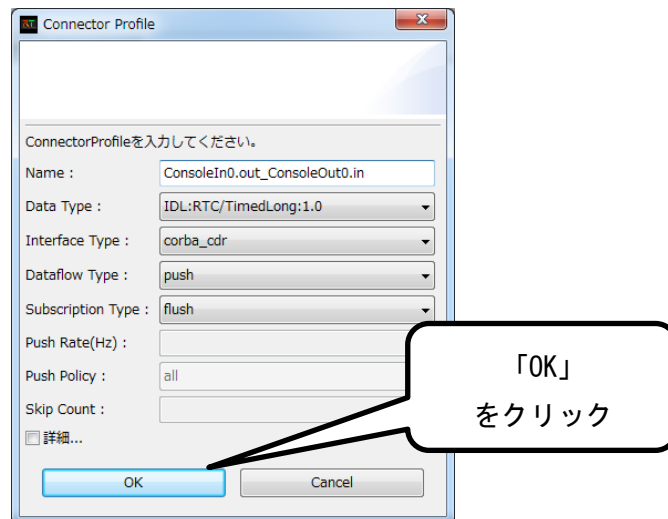
- ⑮ 画面中央に以下のような領域が現れるので、この領域に⑭で確認した各コンポーネントの名前部分をドラッグ&ドロップします。すると下図のように各 RT コンポーネントを表す青色の表示が現れます。



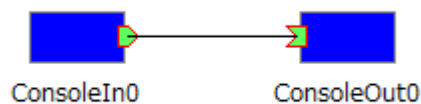
- ⑩ 現れた RT コンポーネントのうち「ConsoleIn」の右側にある OutPort 部分をドラッグして「ConsoleOut」の左側にある InPort にドロップします。すると以下のように繋がります。



- ⑪ 下図のようなダイアログが表示されますがそのまま「OK」をクリックしましょう。



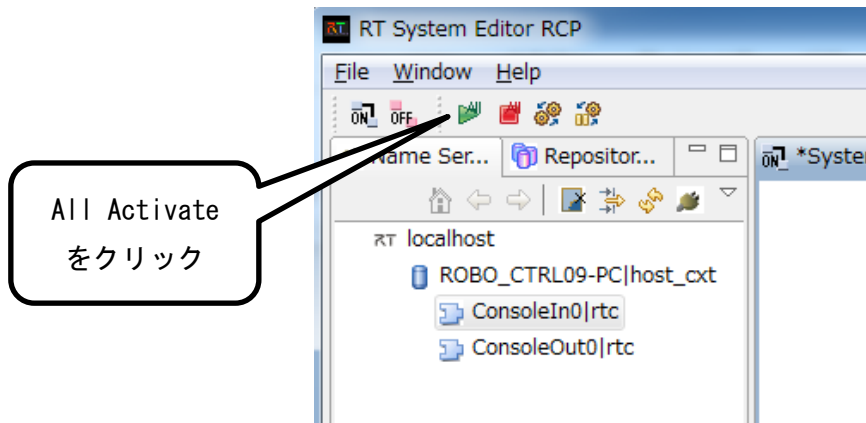
- ⑫ RT コンポーネントのポート同士が接続されると以下のように各ポートが緑色の表示になります。



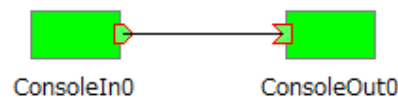
以上で準備は終了です。今度は実際に RT コンポーネントを動かしてみましょう。

13. 動作確認（実行編）

- ① まずは RT System Editor の左上部分にある緑色の三角マーク (All Activate) をクリックし、各 RT コンポーネントを活性化させます。



- ② 画面中央にある RT コンポーネントの表示が緑色に変化すれば、基本的には正常に動作しています。



- ③ 10. 動作確認（準備編）の⑥，⑧で現れたウィンドウを利用して数字の送受信を行います。各 RT コンポーネントを活性化させたので「ConsoleIn」のウィンドウ上に「Please input number」という表示が加えられています。

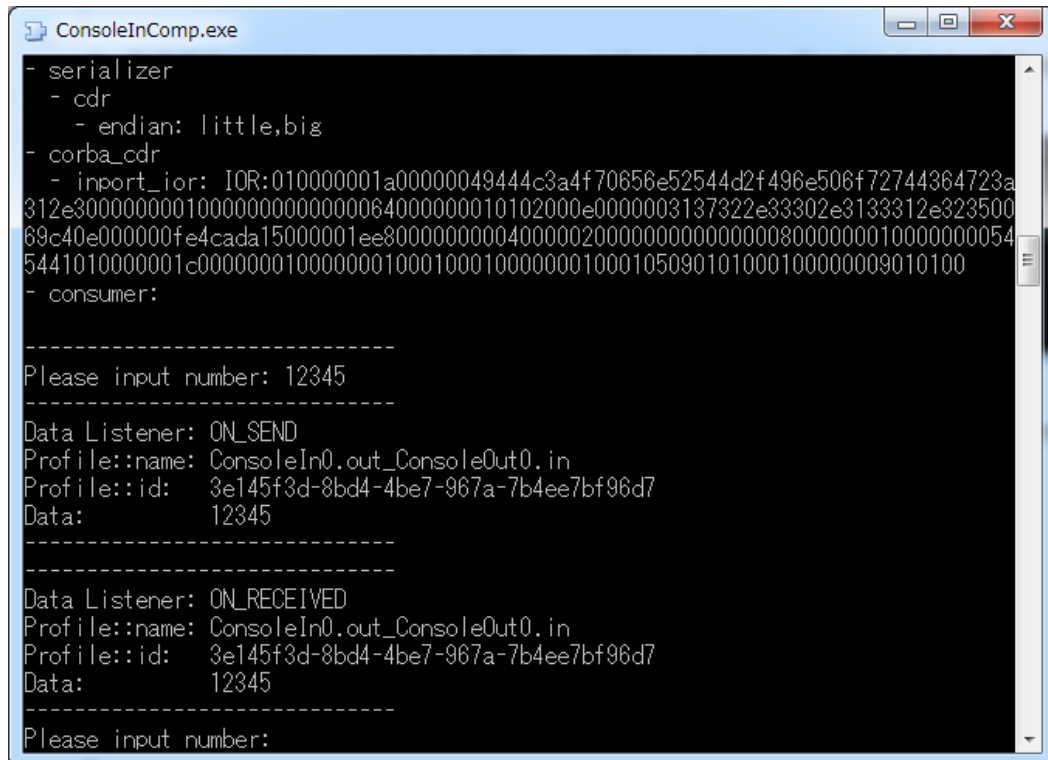
```
- corba_cdr
- inport_ior: IOR:010000001a00000049444c3a4f70656e52544d2f496e506f72744364723a
312e30000000010000000000000064000000010102000e0000003137322e33302e3133312e323500
1ac10e000000fe6525a1500000133c000000004000002000000000000008000000010000000054
5441010000001c00000001000000010001000100000001000105090101000100000009010100
- consumer:
-----
Please input number: _
```

- ④ 「ConsoleIn」のウィンドウ上の点滅している部分に「12345」等の好きな半角数字を打ち込んでEnterキーを押してください。

```
- inport_ior: IOR:010000001a00000049444c3a4f70656e52544d2f496e506f72744364723a
312e30000000010000000000000064000000010102000e0000003137322e33302e3133312e323500
69c40e000000fe4cada15000001ee80000000004000002000000000000008000000010000000054
5441010000001c00000001000000010001000100000001000105090101000100000009010100
- consumer:

-----
Please input number: 12345
```

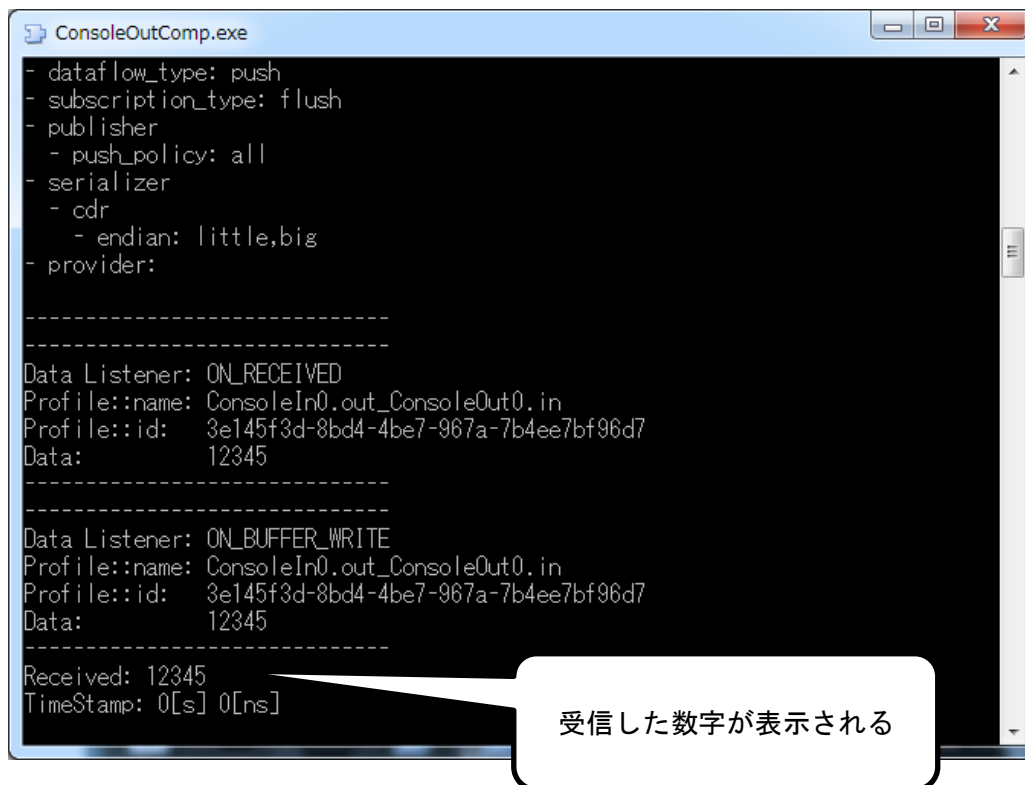
- ⑤ 「ConsoleIn」の表示が以下のように変化し、数字が正常に送信できたことを確認できます。続けて新たな数字を送信することも可能です。



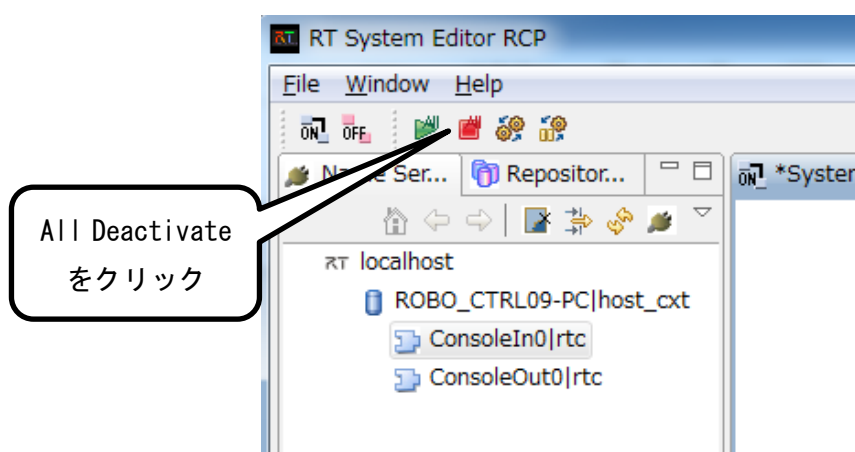
```
ConsoleInComp.exe
- serializer
- cdr
- endian: little,big
- corba_cdr
- inport_ior: IOR:010000001a00000049444c3a4f70656e52544d2f496e506f72744364723a
312e30000000010000000000000064000000010102000e0000003137322e33302e3133312e323500
69c40e000000fe4cada15000001ee80000000004000002000000000000008000000010000000054
5441010000001c00000001000000010001000100000001000105090101000100000009010100
- consumer:

-----
Please input number: 12345
-----
Data Listener: ON_SEND
Profile::name: ConsoleIn0.out_ConsoleOut0.in
Profile::id: 3e145f3d-8bd4-4be7-967a-7b4ee7bf96d7
Data: 12345
-----
Data Listener: ON_RECEIVED
Profile::name: ConsoleIn0.out_ConsoleOut0.in
Profile::id: 3e145f3d-8bd4-4be7-967a-7b4ee7bf96d7
Data: 12345
-----
Please input number:
```

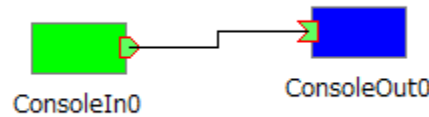
- ⑥ 「ConsoleOut」上で送信した数字と同じ値が表示できていれば OK です。以上により、RT コンポーネントを利用した簡単な通信プログラムが正常に動作していることを確認できました。④の操作を繰り返して何度か数字を送受信してみましょう。



- ⑦ 最後に終了方法を解説します。各コンポーネントを停止させるには下図の赤い四角形のマーク (All Deactivate) をクリックしてください。

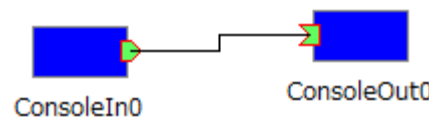


- ⑧ なんと、下図のように ConsoleIn はそのままでは停止しません。これは ConsoleIn が入力待ちを行っているために起こります。



- ⑨ これに対して、ConsoleIn 上で適当な数字を打ち込み Enter を押すと、入力待ちから解放されて正常に RT コンポーネントが停止します。

```
-----
Data Listener: ON_RECEIVED
Profile::name: ConsoleIn0.out_ConsoleOut0.in
Profile::id: 3e145f3d-8bd4-4be7-967a-7b4ee7bf96d7
Data: 1
-----
Please input number: 1_
```



- ⑩ 以上で RT コンポーネントは正常に停止しました。再び「All Activate」で活性化させることもできますし、完全に終了したい場合はそのまま各ウィンドウを閉じてしまっても構いません。

以上で解説は終わりとなります。RT ミドルウェアをインストールし、動作確認までできたでしょうか？

最後に

OpenRTM-aist1.1.0 には様々なサンプルが付属しています。今回解説した「ConsoleIn」や「ConsoleOut」以外にも Open-RTM-aist の公式 HP 上で様々なコンポーネントが手に入ります。面白いので他の解説も参考にして是非いろいろと試してみてください。