

2011

TMS API 導入 マニュアル

(Ver. 3.0)

本文章では、TMS API のインストール手順及び TMS API を用いたアプリケーション開発方法について説明します



目次

1. 事前条件.....	3
2. TMS API インストール手順.....	4
3. Java Runtime Environment のインストール.....	4
4. フォルダ構成	9
5. 設定変更.....	10
6. TMS API の動作確認.....	13
7. C++アプリケーションの開発について	14
7.1 環境設定	14
7.2 アプリケーション開発毎の設定	16
8. Java アプリケーションの開発について	19

1. 事前条件

本APIは、Microsoft Windows XP Professional上で動作します。

本APIの動作には、Sun Java Runtime Environment 6（以降、JREと省略）が必要です。セキュリティのためインストール後アップデートをお願いします。

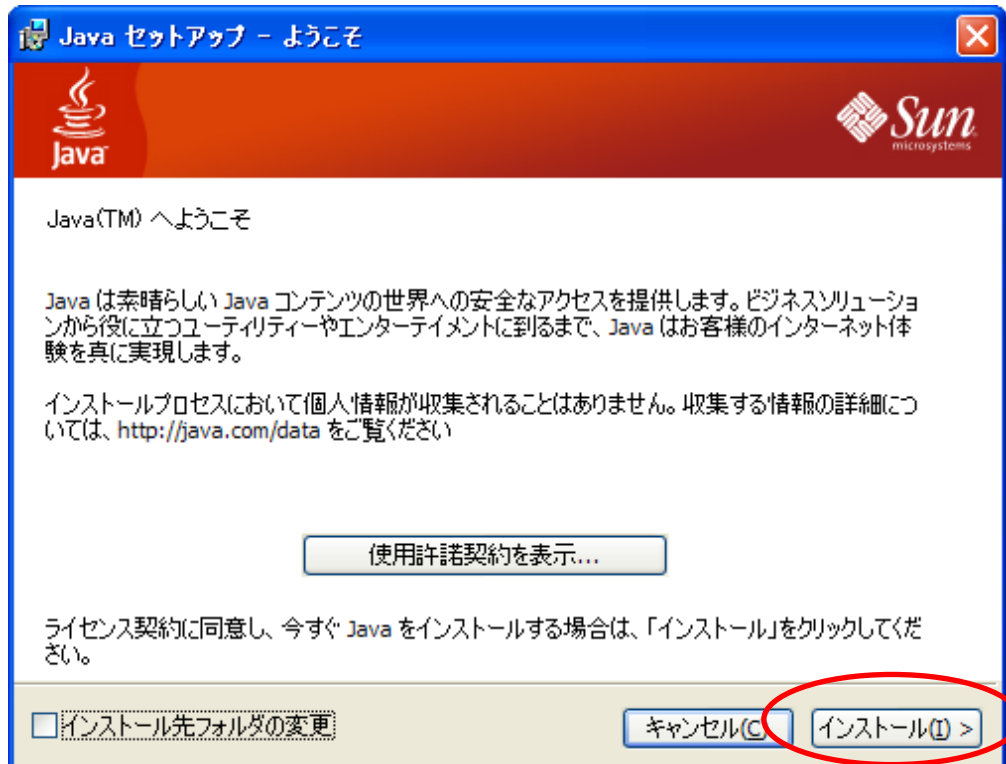
2. Java Runtime Environment のインストール

本章では、Java Runtime Environment(以下、JREと略)本体のインストールを行います。

下記サイトで、JRE 6 最新版をダウンロードします。

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

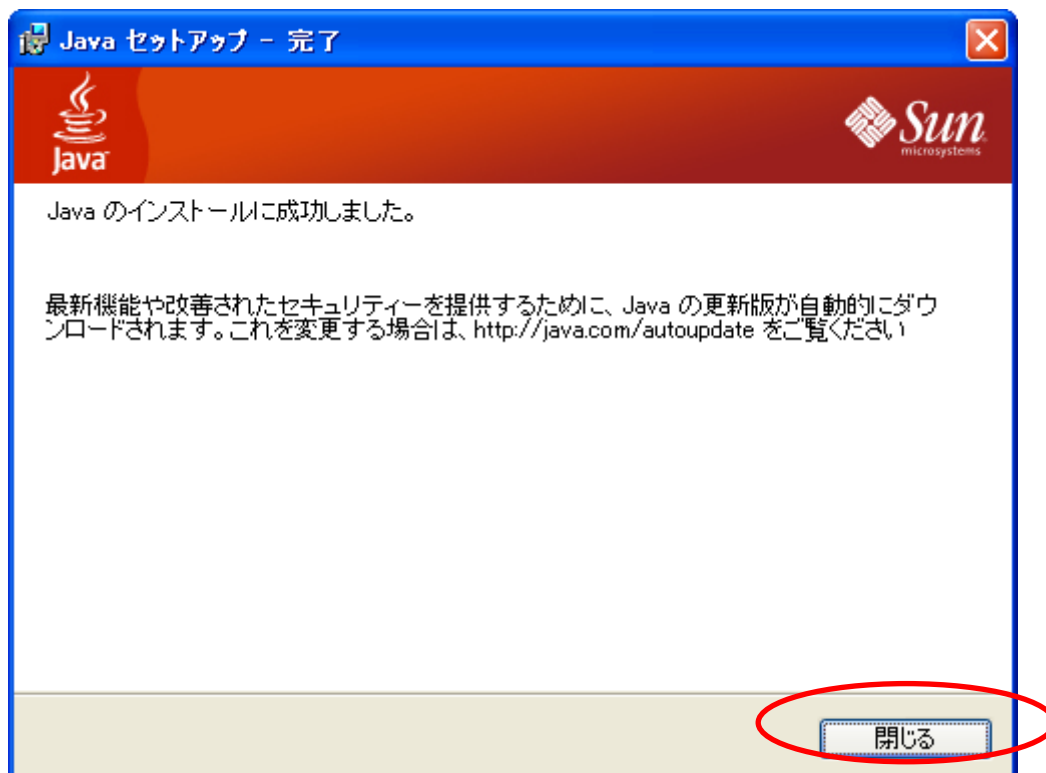
- ① JRE本体のインストールが開始します。「同意する」ボタンを押下して下さい。



- ② インストールがはじまります。



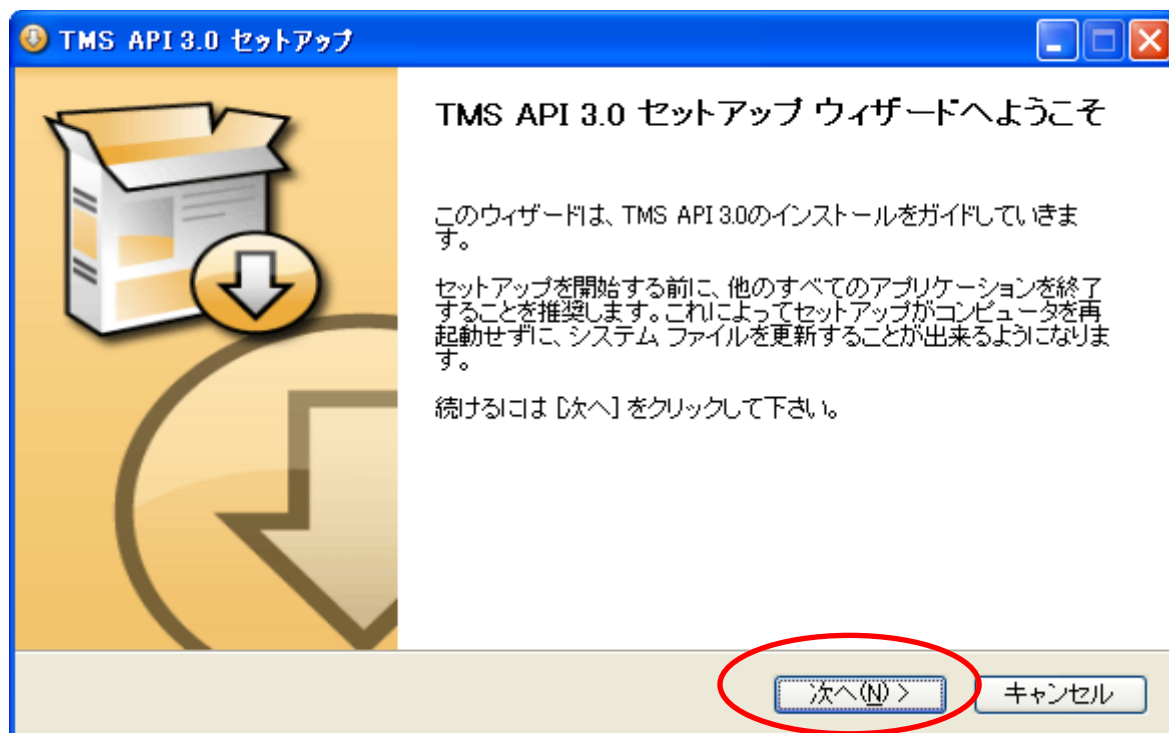
- ③ インストールが完了しました。「完了」ボタンを押下して下さい。



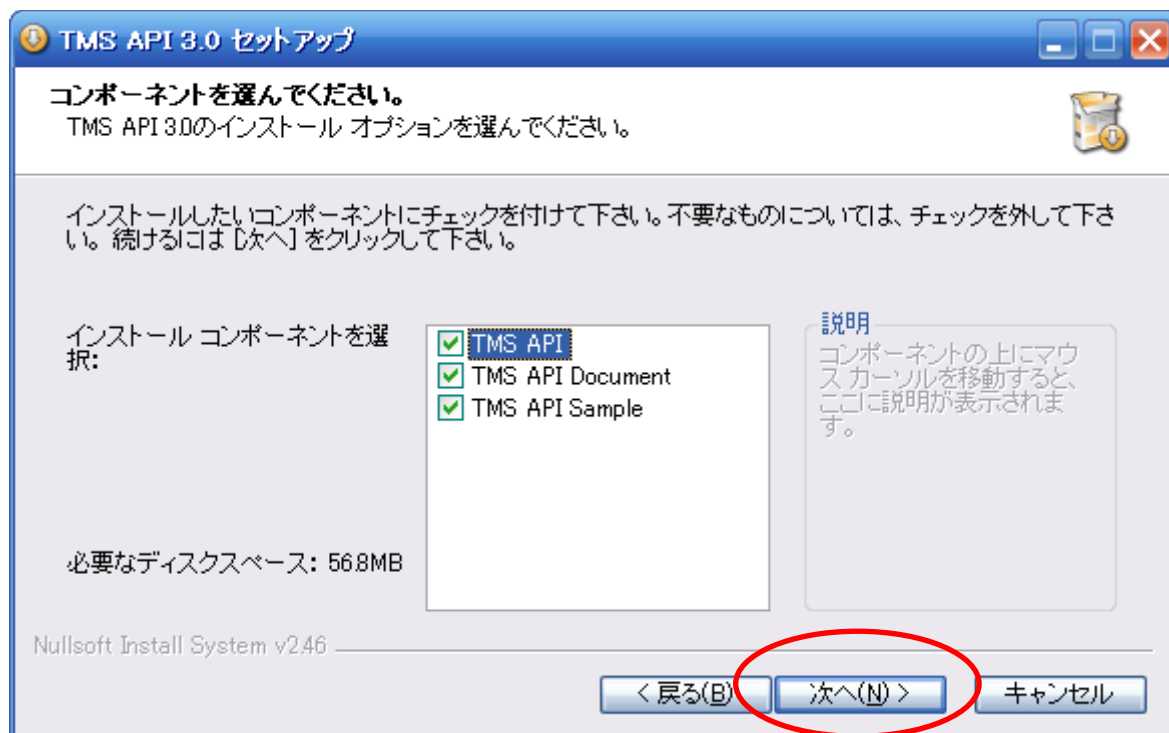
3. TMS APIインストール手順

本章では、TMS APIのインストールを行います。

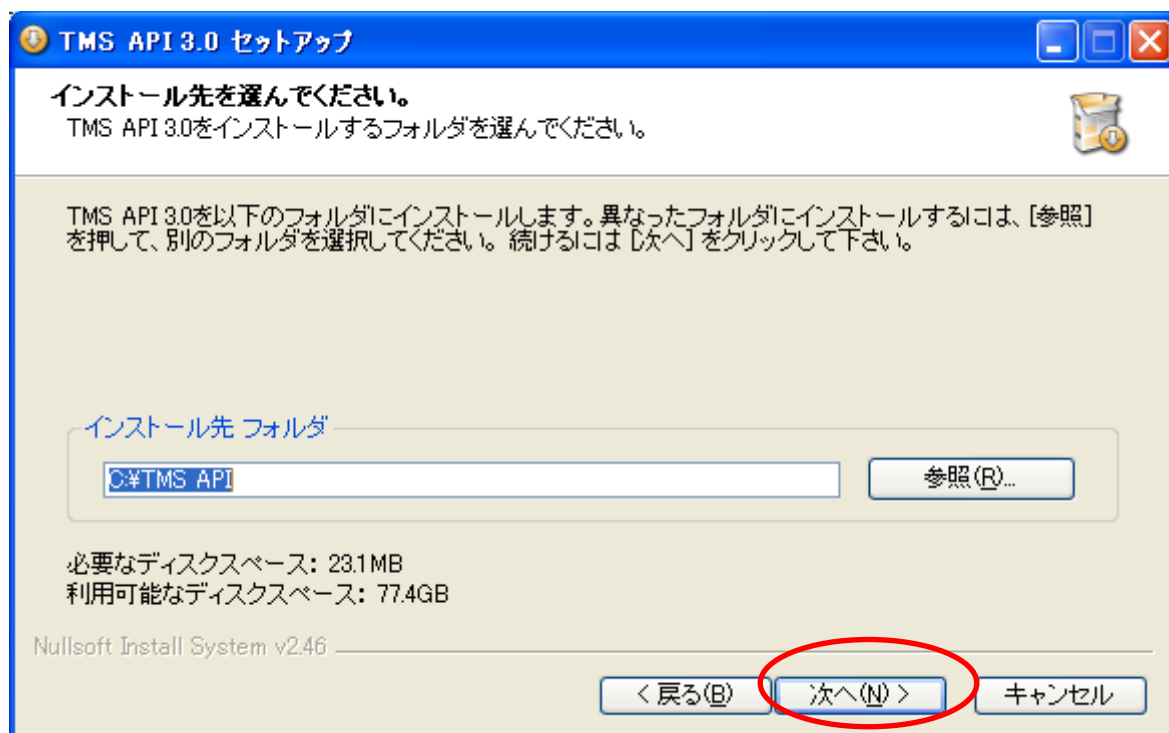
- ① 配布するインストーラの「TmsApiSetup.exe」を起動します。
- ② 「次へ」ボタンを押下して下さい。



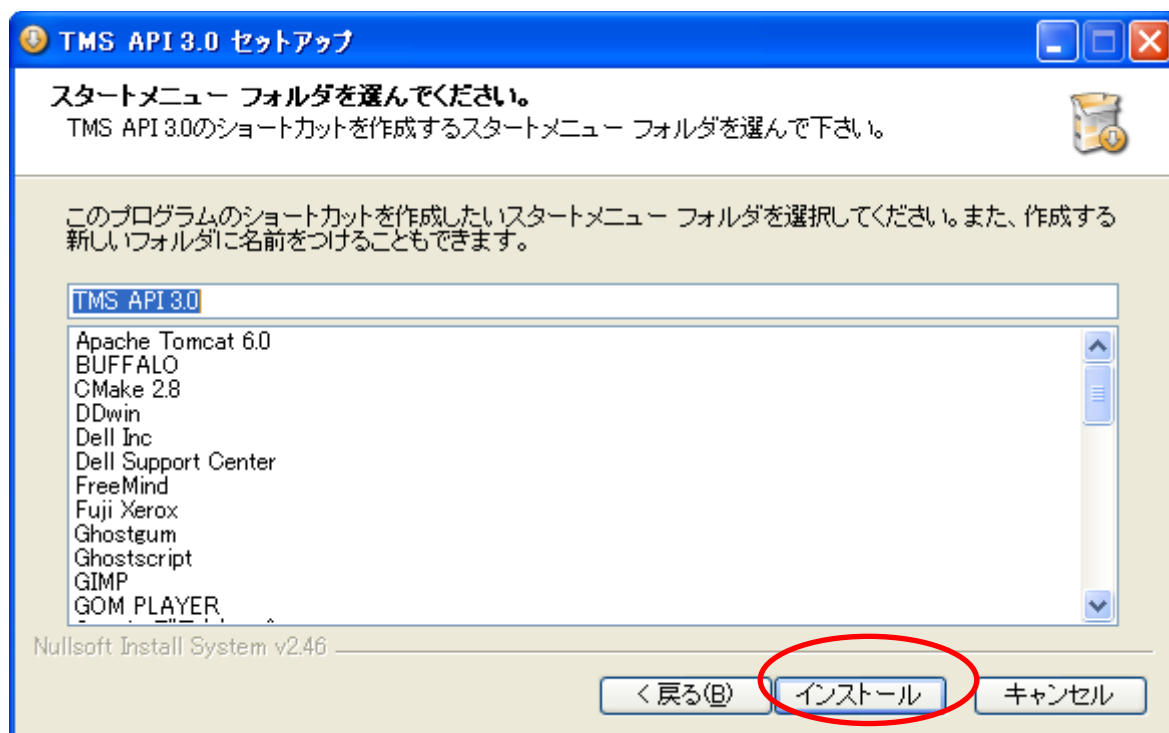
- ③ インストールを行うコンポーネントを選択し、「次へ」ボタンを押下して下さい。



- ④ インストール先のフォルダを確認し、「次へ」ボタンを押下します。



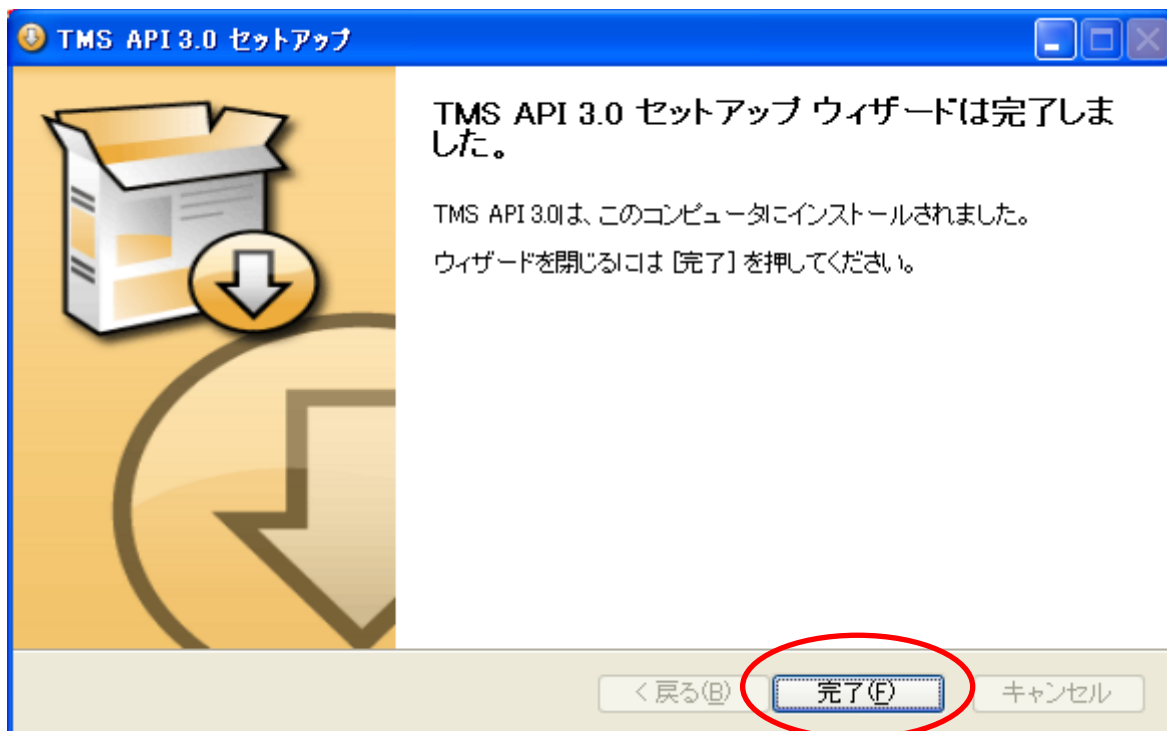
- ⑤ 「インストール」ボタンを押下して下さい。



- ⑥ 選択された物に対して順次インストールが開始されます。

※各コンポーネントのインストール手順については次章以降を参照して下さい。

- ⑦ 各コンポーネントのインストール完了後に、「完了」ボタンを押下して下さい。



※インストーラによって、以下のものが自動的に設定されます。

環境変数 : TMSAPI_HOME、 JAVA_HOME

システムパス : %JAVA_HOME%\bin¥client

4. フォルダ構成

%TMSAPI_HOME%\device の直下にいくつかのサブフォルダが存在します。それらの用途について説明します。

サブフォルダ	用途
bin	API本体と必要なライブラリファイル
cfg	TMSサーバのホスト名などを設定するtms.properties、 ログファイルを制御するlog4j.xml を格納します。
doc	インストールマニュアル、APIリファレンス
include	C++のヘッダファイルとlibファイル
sample	Java / C++ サンプルファイル
log	ログファイル格納用フォルダ

5. 設定変更

① TMS サーバ名の設定（この設定は、必ず行って下さい）

%TMSAPI_HOME%\device\cfg\tmsapi.properties (以下プロパティファイルと称す)を開き、Tms サーバのホスト名(またはIPアドレス)「**tmshost**」を設定して下さい。(図5-1参照)

② TMSサーバポート設定（この設定は、必ず行って下さい）

ロボット・センサーはTMS サーバからの通知を受け取るために8081番ポートを使用する為、**8081番ポートを開放**するようにファイアウォールを設定して下さい。また、プロパティファイルのlocalportの変更によって、デバイス側の待ち受けポートを8081番以外に設定可能です。図5-1を参照して下さい。

③ ネットワークタイムアウト設定（この設定は、必要に応じて行って下さい）

通信環境に応じて、プロパティファイルのconnection_timeoutとsocket_timeoutを修正して下さい。fconnection_timeout はサーバと接続するときのタイムアウトを意味し、物理的な断線などによって発生します。socket_timeout は、ネットワークトラフィックの混雑によって、一時的にデータが取れなくなるなどの場合に発生します。図5-1 を参照して下さい。

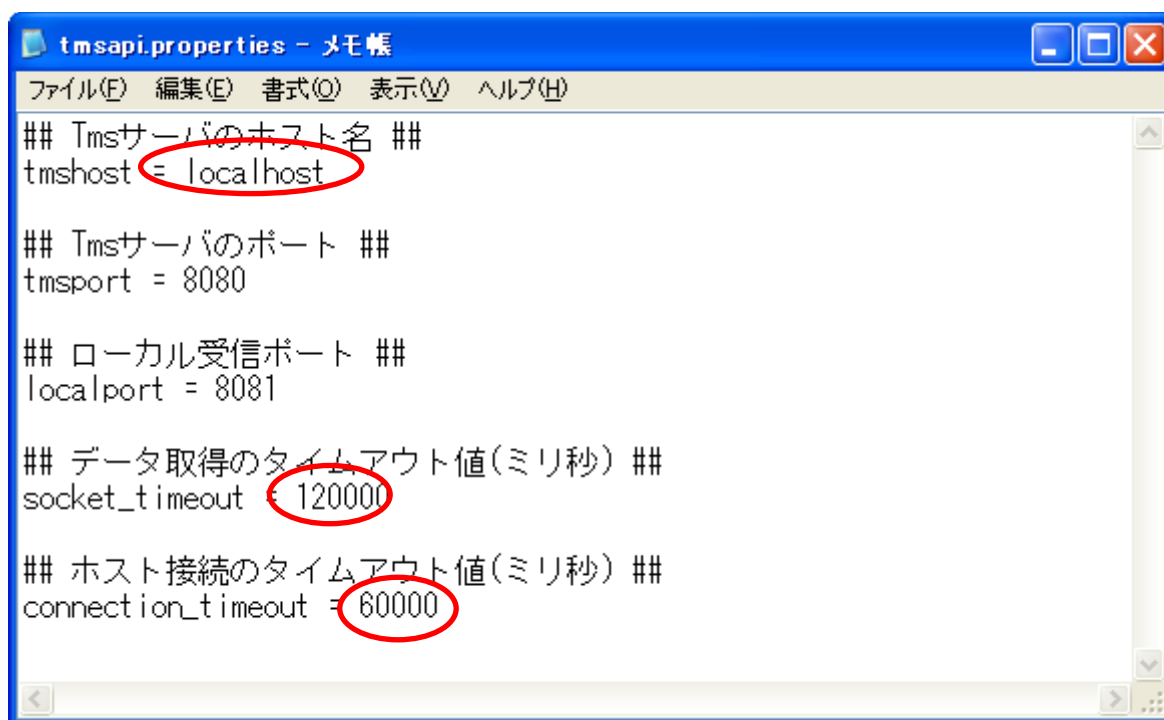
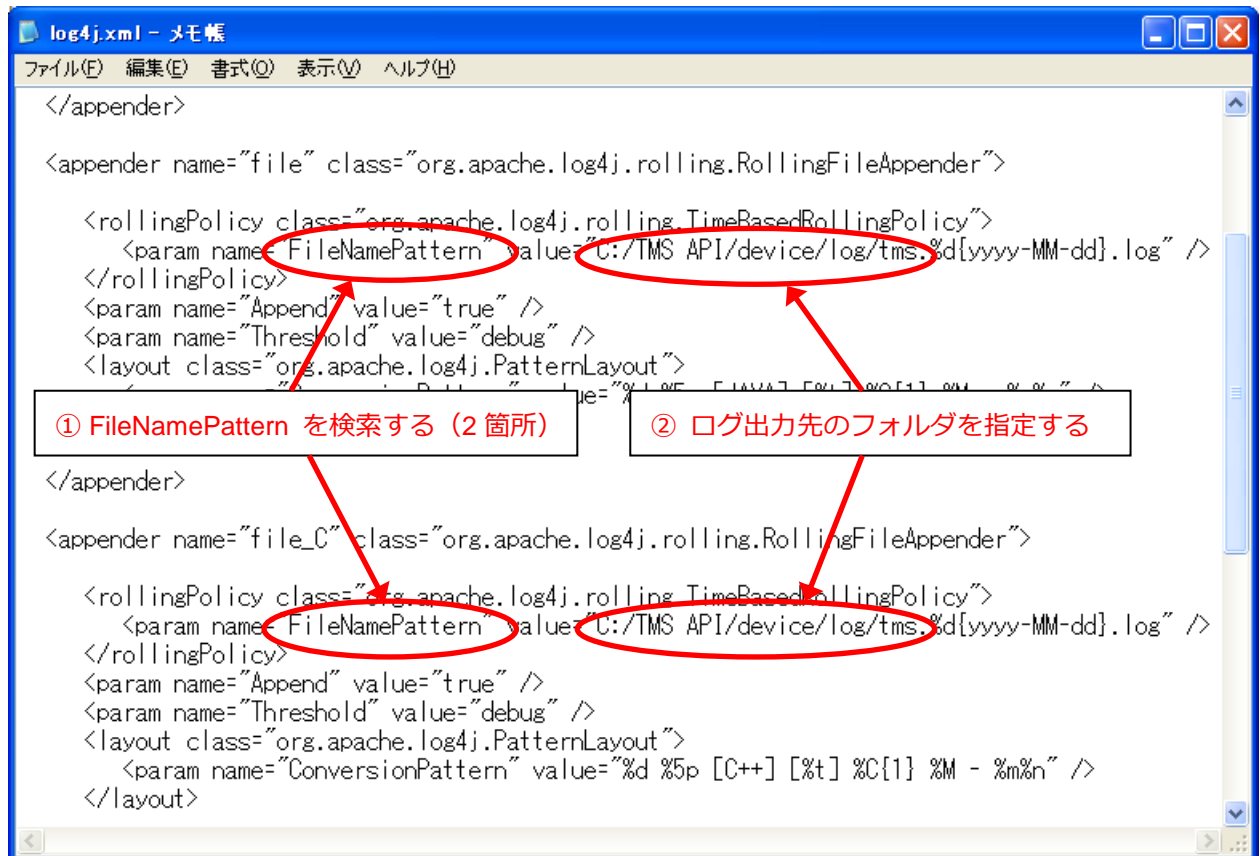


図5-1

④ ログ出力設定（この設定は、開発時、運用時にそれぞれ調整して下さい）

- ・ ログの出力先の変更

インストール直後は、ログが%TMSAPI_HOME%\device\log 配下に出力されるようになって
います。ログの出力先を変更する際は、%TMSAPI_HOME%\device\cfg\log4j.xmlを開き、ロ
グファイルの出力先を変更して下さい。



- ・ ログレベルの変更

開発時や原因調査時などにAPI の動作状態をログに出力させる場合などは、ログレベルを「DEBUG」に設定して下さい。

運用時などはログレベルを「INFO」以上に設定して下さい。不要なログが運用時に出力されないようになり、負荷を下げる事が可能になります。

```
</appender>

<logger name="com.nec.qnes.tms2" >
  <level value="info"/>
  <appender-ref ref="file" />
  <appender-ref ref="stdout" />
</logger>

<logger name="com.nec.qnes.tms2.robotapi.RobotAPI_C" >
  <level value="info"/>
  <appender-ref ref="file_C" />
</logger>

<root>
  <level value="info"/>
</root>

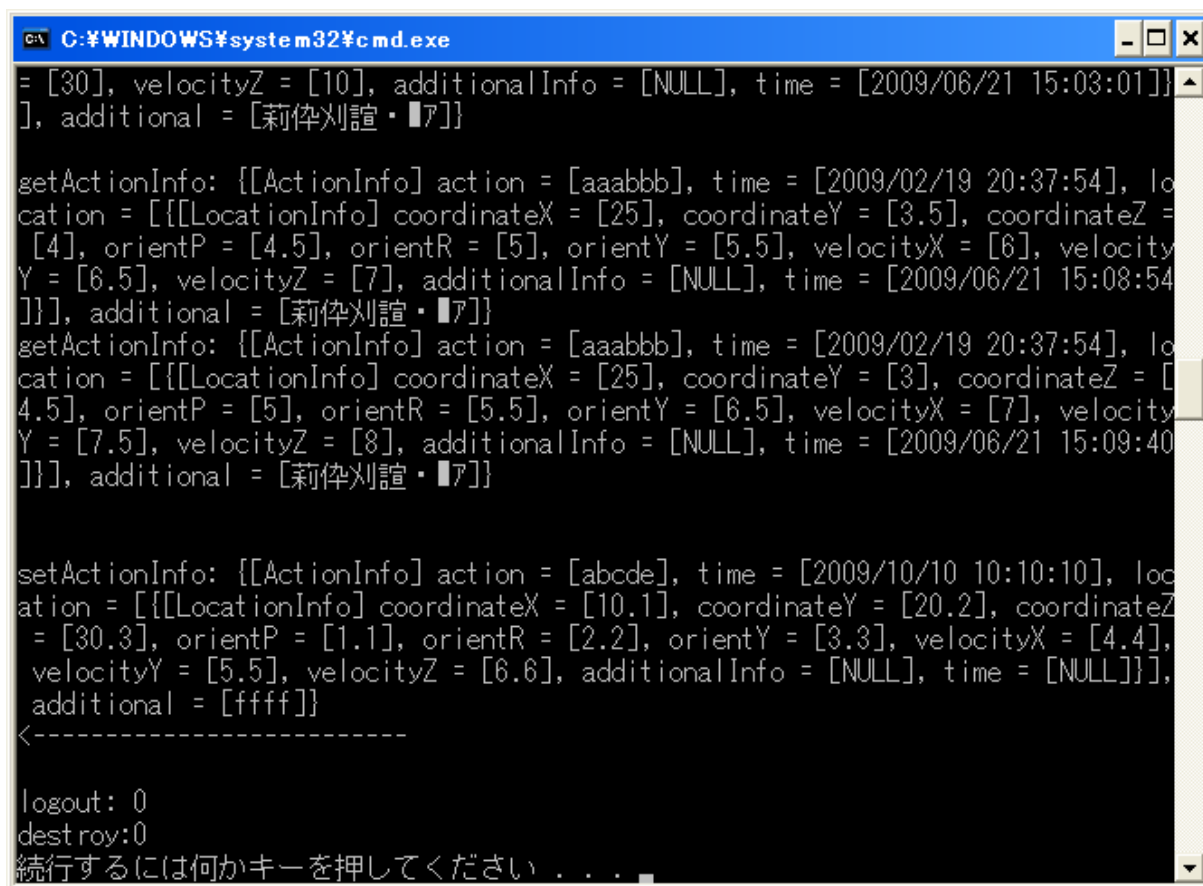
</log4j:configuration>
```

6. TMS APIの動作確認

本章では、TMS API が正しく動作することを確認するために、デモプログラムを実行します。

デモプログラムは「%TMSAPI_HOME%\device\bin\TMS2 Device Api Demo.exe」にあります。

正しく動作した場合の実行例を以下に示します。



```
C:\WINDOWS\system32\cmd.exe
= [30], velocityZ = [10], additionalInfo = [NULL], time = [2009/06/21 15:03:01]]
], additional = [莉俸刈誼・■7]]

getActionInfo: {[ActionInfo] action = [aaabbb], time = [2009/02/19 20:37:54], lo
cation = {[LocationInfo] coordinateX = [25], coordinateY = [3.5], coordinateZ =
[4], orientP = [4.5], orientR = [5], orientY = [5.5], velocityX = [6], velocity
Y = [6.5], velocityZ = [7], additionalInfo = [NULL], time = [2009/06/21 15:08:54
]}], additional = [莉俸刈誼・■7]]
getActionInfo: {[ActionInfo] action = [aaabbb], time = [2009/02/19 20:37:54], lo
cation = {[LocationInfo] coordinateX = [25], coordinateY = [3], coordinateZ = [
4.5], orientP = [5], orientR = [5.5], orientY = [6.5], velocityX = [7], velocity
Y = [7.5], velocityZ = [8], additionalInfo = [NULL], time = [2009/06/21 15:09:40
]}], additional = [莉俸刈誼・■7]]

setActionInfo: {[ActionInfo] action = [abcde], time = [2009/10/10 10:10:10], loc
ation = {[LocationInfo] coordinateX = [10.1], coordinateY = [20.2], coordinateZ
= [30.3], orientP = [1.1], orientR = [2.2], orientY = [3.3], velocityX = [4.4],
velocityY = [5.5], velocityZ = [6.6], additionalInfo = [NULL], time = [NULL]}],
additional = [fffff]}
<-----
logout: 0
destroy: 0
続行するには何かキーを押してください . . .
```

デモプログラムが正しく動作しなかった場合、以下の作業を行ってください。

- 「このアプリケーションの構成が正しくないため、アプリケーションを開始できませんでした。(以下省略)」とメッセージが出てプログラムが起動しない場合
 - ① Microsoft Visual C++ 2005 SP1 再頒布可能パッケージ「vcredist_x86.exe」をインストールする
 - ② Windows Update を実行し、PC を再起動する
- アプリケーションは起動するが「can't get reference to Tag」等と表示された場合
 - ① TMS サーバ名の設定を間違えていないかの確認(前章参照)
 - ② TMS サーバの起動確認(TMS サーバ導入マニュアルの 11 章参照)

7. C++アプリケーションの開発について

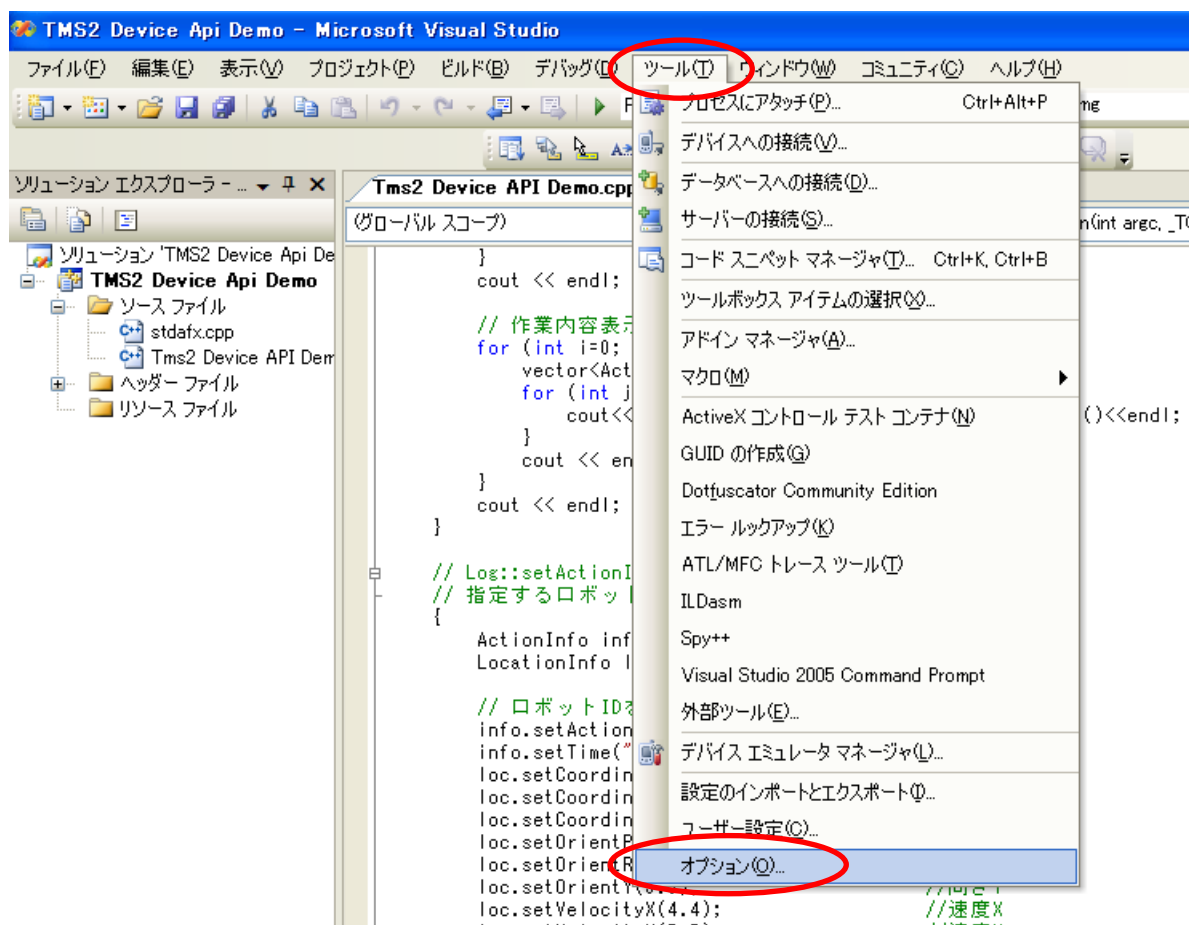
本章では、TMS APIを利用したC++アプリケーション開発を行うための設定について説明します。
設定には、初めに一度だけ行えばよい環境設定とアプリケーションの開発毎に行う設定の2種類があります。

7.1 環境設定

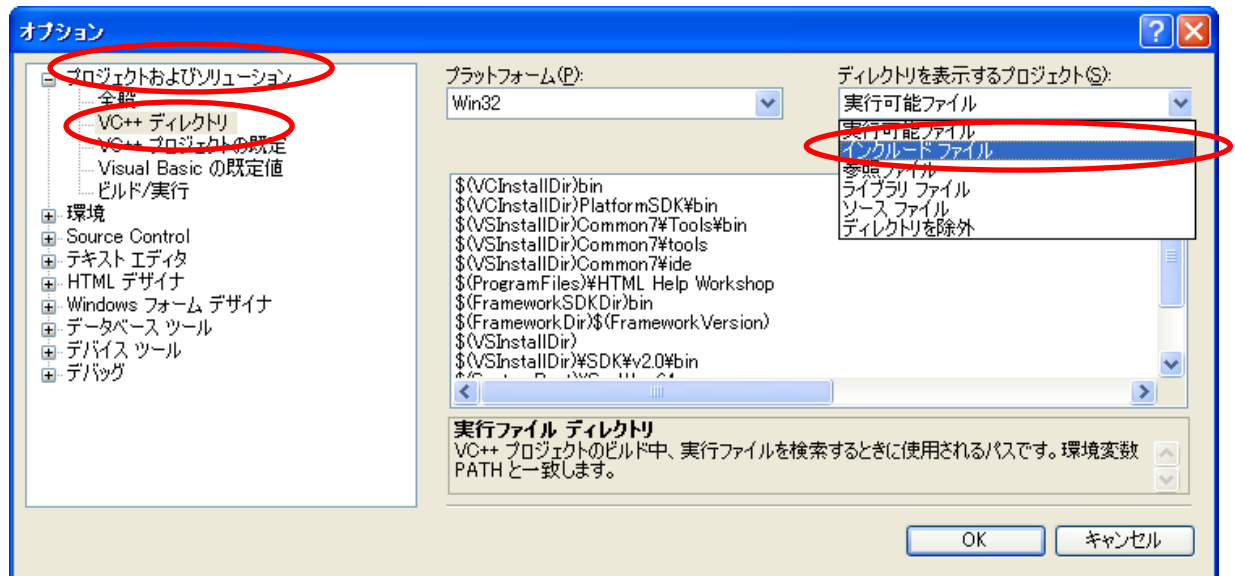
TMS API のインクルードファイルとライブラリファイルのパス設定を行います。

以下開発環境として、Microsoft Visual Studio を用いることを前提に説明します。

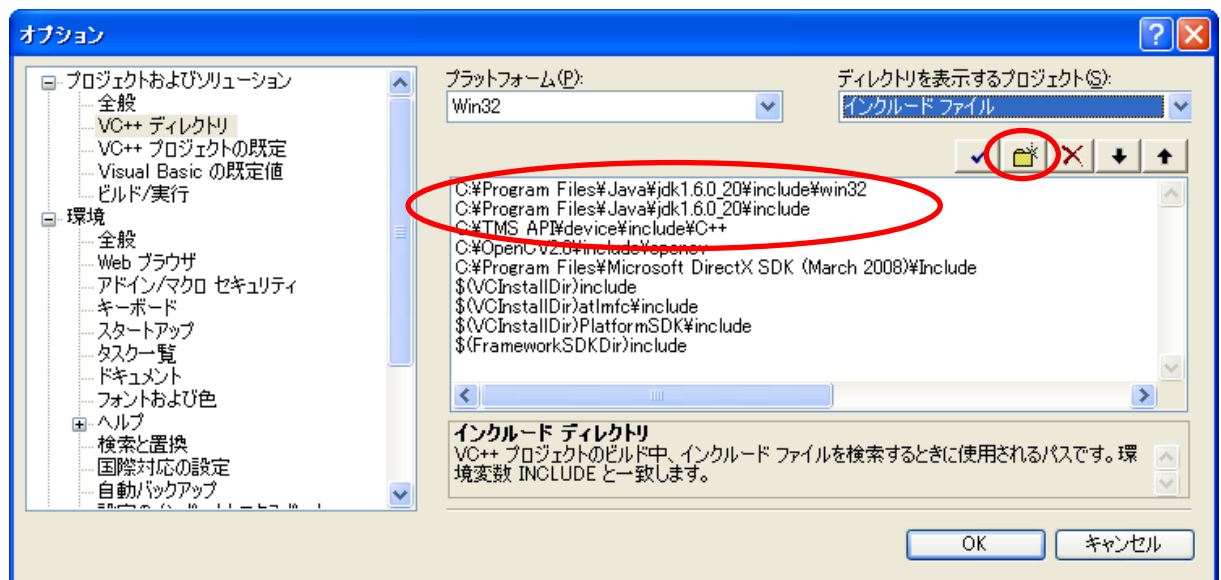
- ① Sun Java Development Kit（以下、JDK と略）をインストールします。
「jdk-6u20-windows-i586.exe」を起動して、その指示に従ってインストールを進めてください。
JDK をインストールしたフォルダを%JDK_HOME%とします。
- ② Visual Studio を起動し、「ツール」のメニューから「オプション」を開きます



- ③ 「オプション」の ウィンドウから「プロジェクトおよびソリューション」→「VC++ディレクトリ」を選びます。その後、「ディレクトリを表示するプロジェクト」から「インクルードファイル」を選びます。



- ④ 左から 2 番目のアイコン(新しい行を追加)を押し、
「%TMSAPI_HOME%¥device¥include¥C++」を入力します。
同様のやり方で、「%JDK_HOME%¥include」と「%JDK_HOME%¥include¥win32」も追加
します。



- ⑤ ④と同様に、「ディレクトリを表示するプロジェクト」から「ライブラリファイル」を選び、
「%TMSAPI_HOME%¥device¥include¥C++」を登録します。

7.2 アプリケーション開発毎の設定

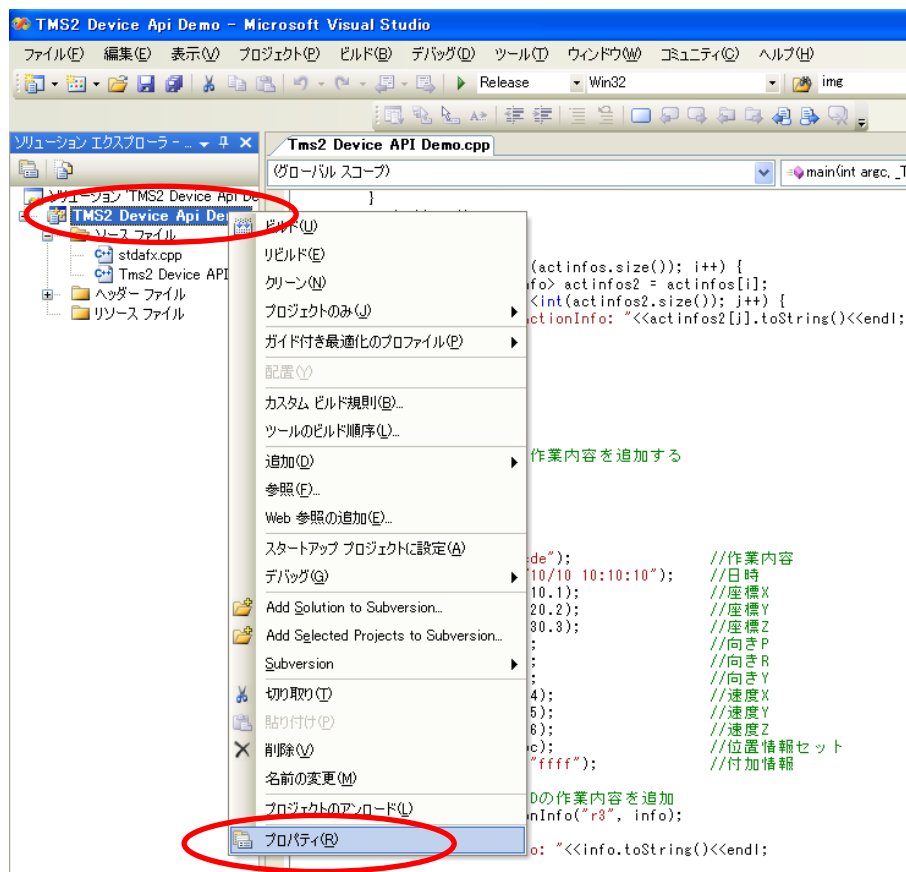
新しいアプリケーションを開発する度に、以下の設定を行います。

開発環境	アプリ構成	設定内容
Visual Studio 2003 (SP1)	Debug	コンパイルオプション /MDd を使用 Tms2APId_VS7.lib にリンク
	Release	コンパイルオプション /MD を使用 Tms2API_VS7.lib にリンク
Visual Studio 2005 (SP1)	Debug	コンパイルオプション /MD を使用 Tms2APId_VS8.lib にリンク プリプロセッサ_DEBUG を削除
	Release	コンパイルオプション /MD を使用 Tms2API_VS8.lib にリンク

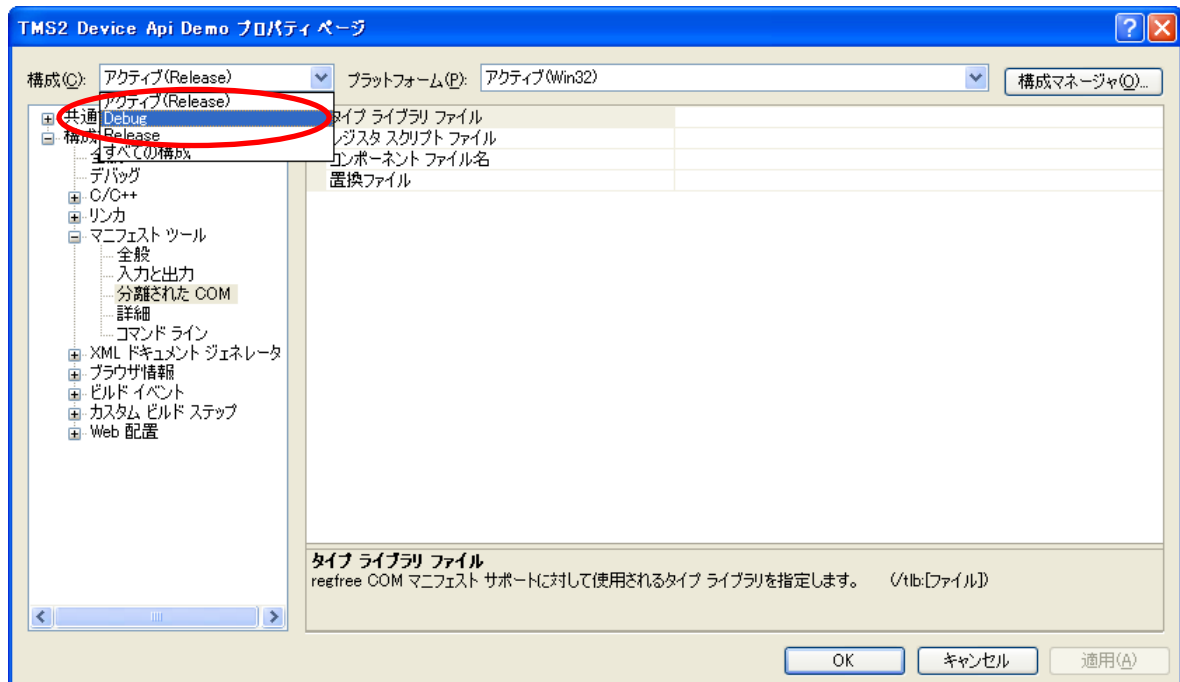
各設定の方法は、以下の通りです。

- ・ コンパイルオプション

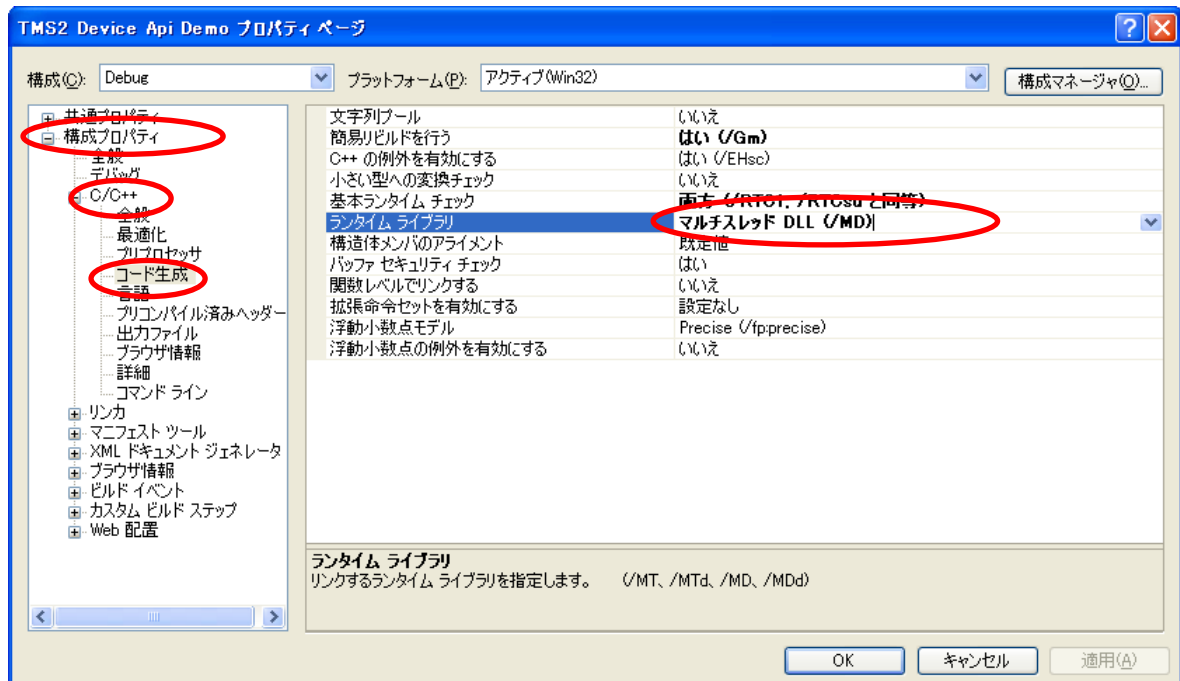
- ① ソリューションエクスプローラから設定を行いたいプロジェクト名を右クリックし、プロジェクトのプロパティを開きます



- ② 「構成」から設定を行いたいアプリ構成(例：Debug)を指定します

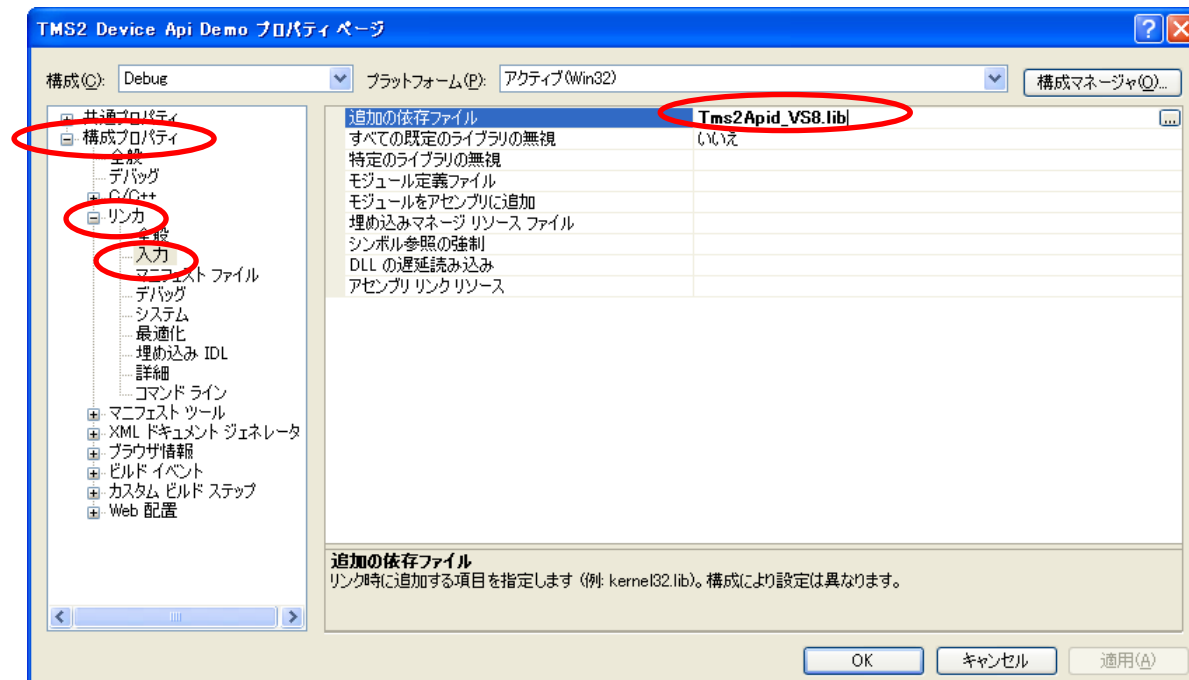


- ③ 「構成プロパティ」→「C/C++」→「コード生成」を開きます。「ランタイムライブラリ」から希望のコンパイルオプション(例：/MD)を指定します



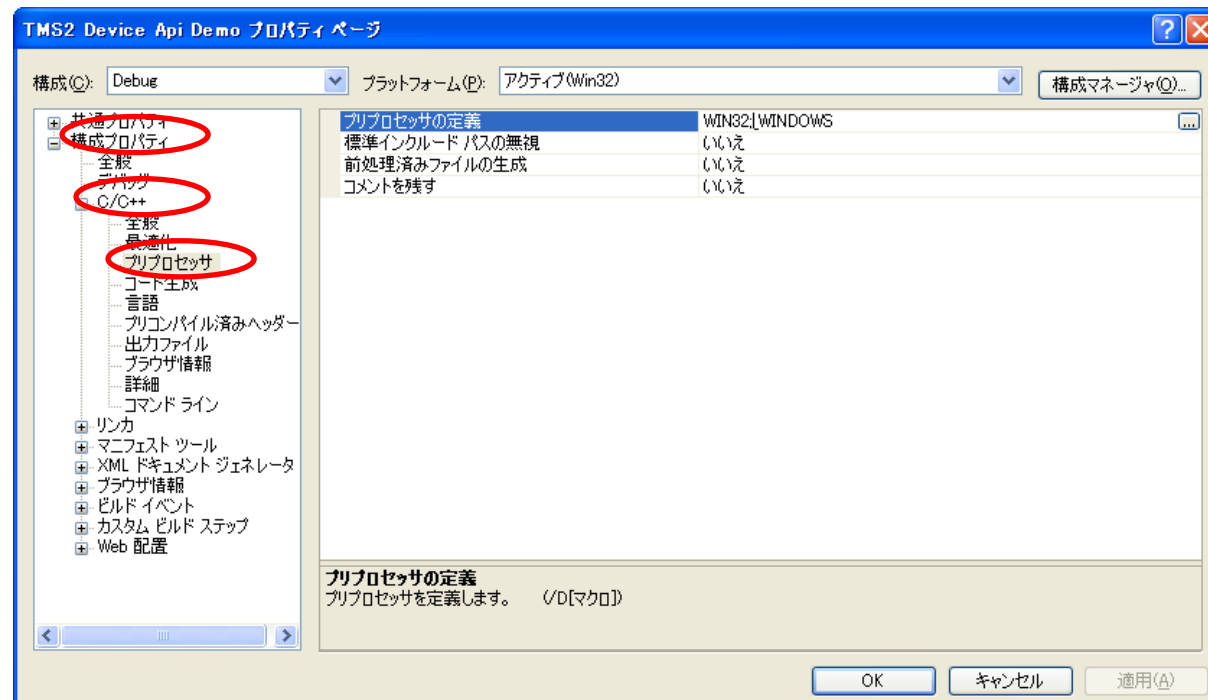
・リンク

- ④ 「構成プロパティ」→「リンカ」→「入力」を開きます。「追加の依存ファイル」にリンクを行うファイル(例：Tms2APId_VS8.lib)を指定します。



・プリプロセッサ

- 「構成プロパティ」→「C/C++」→「プリプロセッサ」を開きます。「プリプロセッサの定義」からプリプロセッサを指定します。



8. Java アプリケーションの開発について

%TMSAPI_HOME%\device\bin 内の全ての.jar ファイルをインポートする必要があります。