MMDAgent Developer Reference

Ver. 1.01

September 29, 2016
Nagoya Institute of Technology

Contents

1. About this document

This document provides specifications for developers extending MMDAgent functionality. It describes how to build the development and run-time environments on Windows, how to develop a new plugin, and how to build development and run-time environments on Android.
This document deals with the following versions of MMDAgent.

▼ Versions

Software	Version
MMDAgent.exe	1.6.1
MMDAgent_Example	1.6

2. Building the development and run-time environments (Windows)

Overview

This section summarizes procedures to develop and run published MMDAgent source code on Windows and to develop a new plugin for MMDAgent.

This section was written based on the following environment, but development can also be done on Windows 8. In doing so, adjust the document details according to your own environment.

Software	Version
Development environment OS	Windows 7 64 bit
Development software	Visual Studio Community 2013

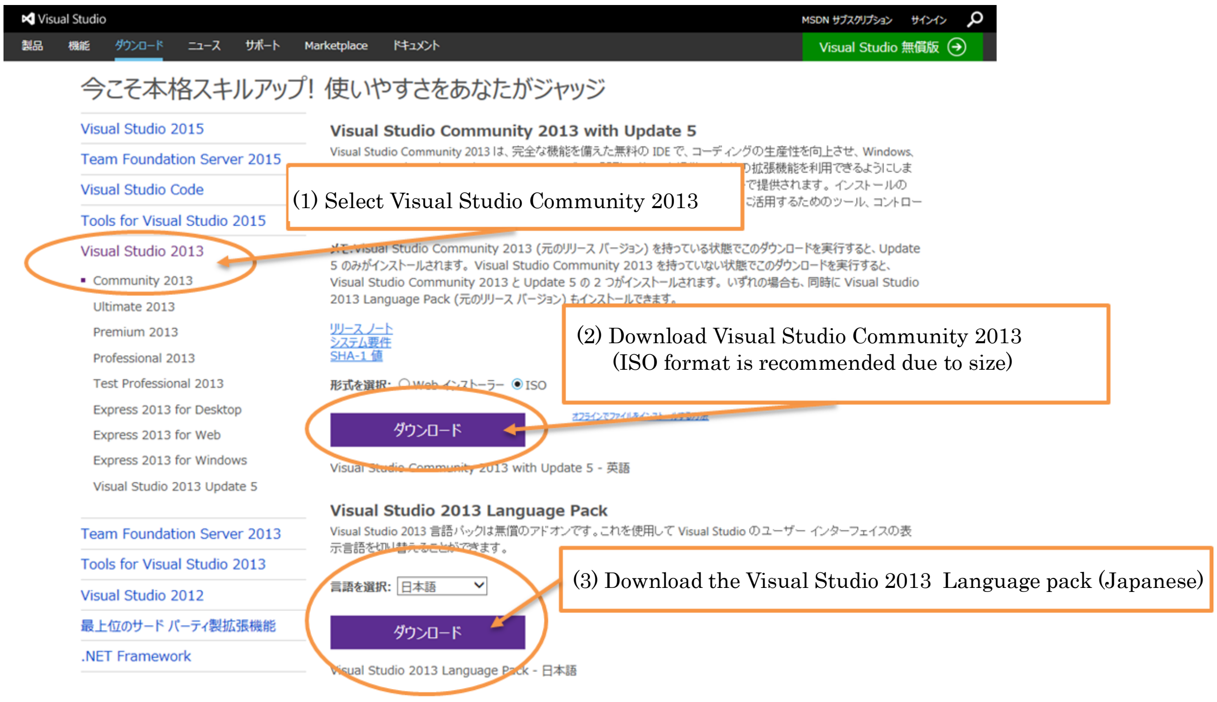
[Notes]
The published MMDAgent solution file (.sln) and project file (.vcxproj) were created for Visual Studio 2010, but later versions can be used by upgrading the project (when the solution is first opened, a request to upgrade will automatically appear).

Building the development environment

Downloading Visual Studio Community 2013

Download Visual Studio Community 2013 and the Japanese Language Pack from Microsoft.
https://www.visualstudio.com/downloads/download-visual-studio-vs

▼ Procedure



今こそ本格スキルアップ! 使いやすさをあなたがジャッジ

Visual Studio Community 2013 with Update 5

Visual Studio Community 2013 は、完全な機能を備えた無料の IDE で、コーディングの生産性を向上させ、Windows、

の拡張機能を利用できるようにしま

で提供されます。インストールの

ご活用するためのツール、コントロー

(1) Select Visual Studio Community 2013

メモ: Visual Studio Community 2013 (元のリリース バージョン) を持っている状態でこのダウンロードを実行すると、Update 5 のみがインストールされます。Visual Studio Community 2013 を持っていない状態でこのダウンロードを実行すると、Visual Studio Community 2013 と Update 5 の 2 つがインストールされます。いずれの場合も、同時に Visual Studio 2013 Language Pack (元のリリース バージョン) もインストールできます。

(2) Download Visual Studio Community 2013
(ISO format is recommended due to size)

形式を選択: ○ Web インストーラー [✓ ISO]

オフラインでファイルをインストールする方法

Visual Studio Community 2013 with Update 5 - 英語

Visual Studio 2013 Language Pack

Visual Studio 2013 言語パックは無償のアドオンです。これを使用して Visual Studio のユーザー インターフェイスの表示言語を切り替えることができます。

(3) Download the Visual Studio 2013 Language pack (Japanese)

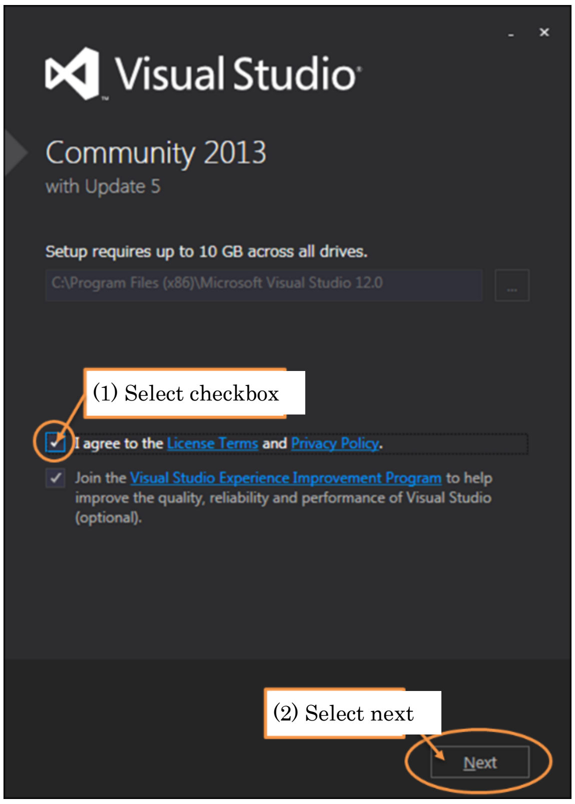
言語を選択: 日本語

Visual Studio 2013 Language Pack - 日本語

Installing Visual Studio Community 2013

Run the downloaded install file (ISO or Web installer) to install Visual Studio Community 2013.

▼ Procedure



Visual Studio
Community 2013
with Update 5

Setup requires up to 10 GB across all drives.

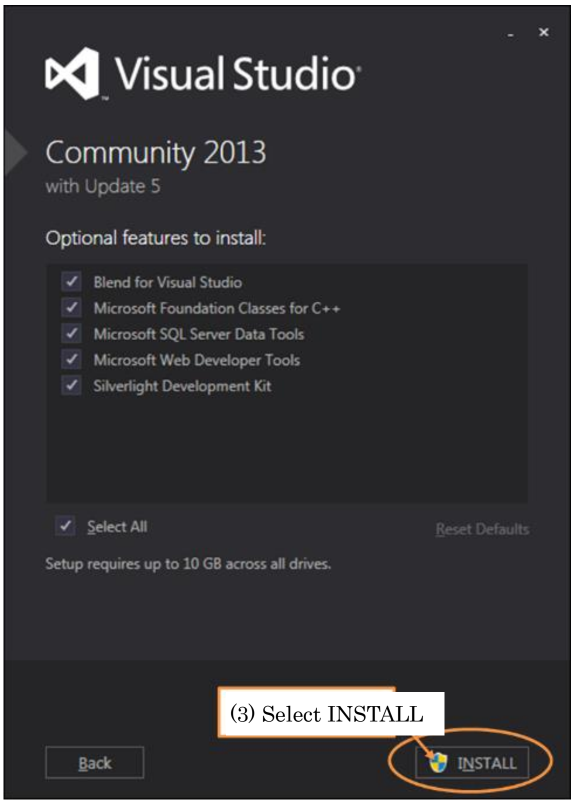
C:\Program Files (x86)\Microsoft Visual Studio 12.0

(1) Select checkbox

✓ I agree to the License Terms and Privacy Policy.

✓ Join the Visual Studio Experience Improvement Program to help improve the quality, reliability and performance of Visual Studio (optional).

(2) Select next



Visual Studio
Community 2013
with Update 5

Optional features to install:

✓ Blend for Visual Studio
✓ Microsoft Foundation Classes for C++
✓ Microsoft SQL Server Data Tools
✓ Microsoft Web Developer Tools
✓ Silverlight Development Kit

✓ Select All

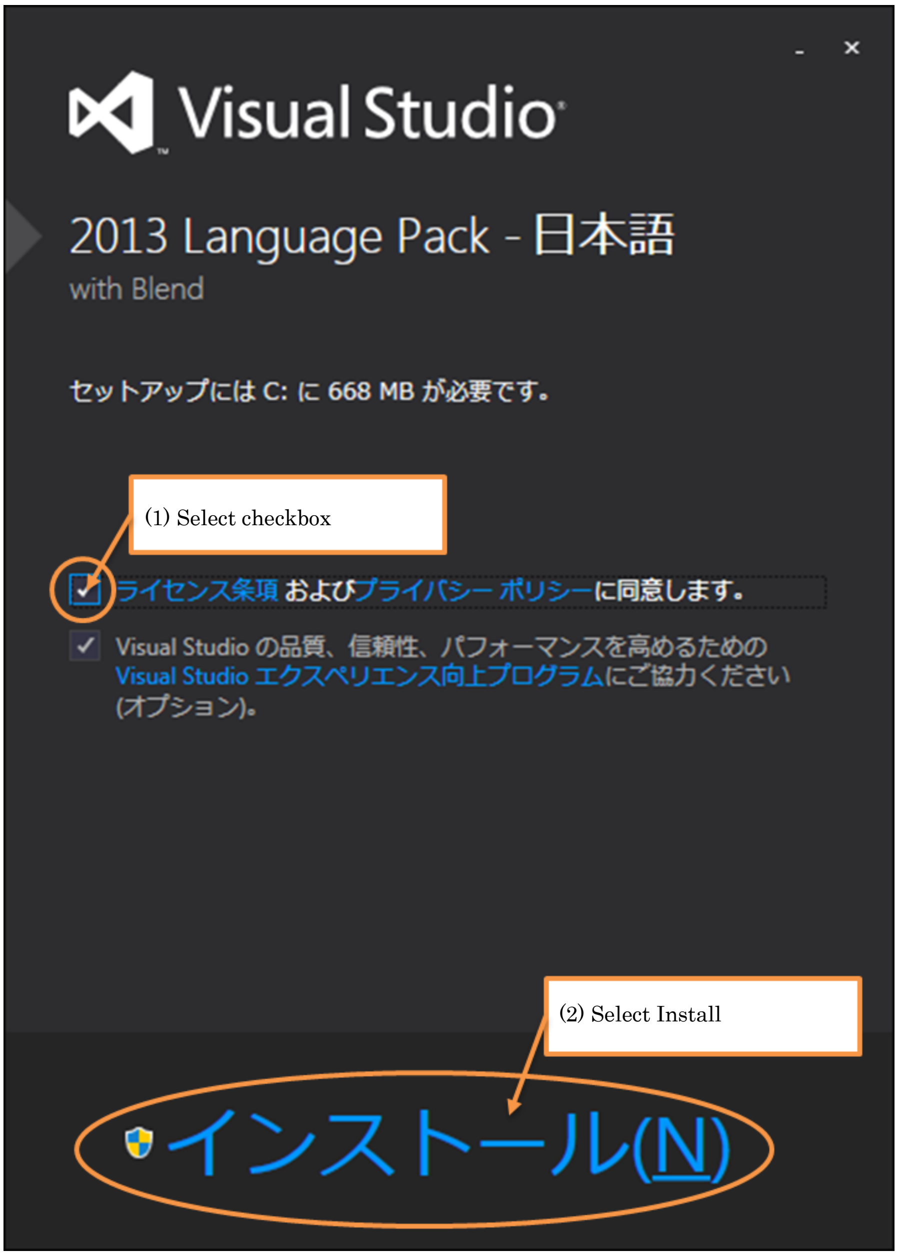
Setup requires up to 10 GB across all drives.

(3) Select INSTALL

Installing the Visual Studio 2013 Language Pack

Run the downloaded Language Pack (Japanese) installer to install it. Note that Visual Studio settings are required to complete the conversion to Japanese, and these changes are described below.

▼ Procedure



Visual Studio

2013 Language Pack - 日本語

with Blend

セットアップには C: に 668 MB が必要です。

(1) Select checkbox

✓ ライセンス条項 および プライバシー ポリシー に同意します。

✓ Visual Studio の品質、信頼性、パフォーマンスを高めるための Visual Studio エクスペリエンス向上プログラムにご協力ください (オプション)。

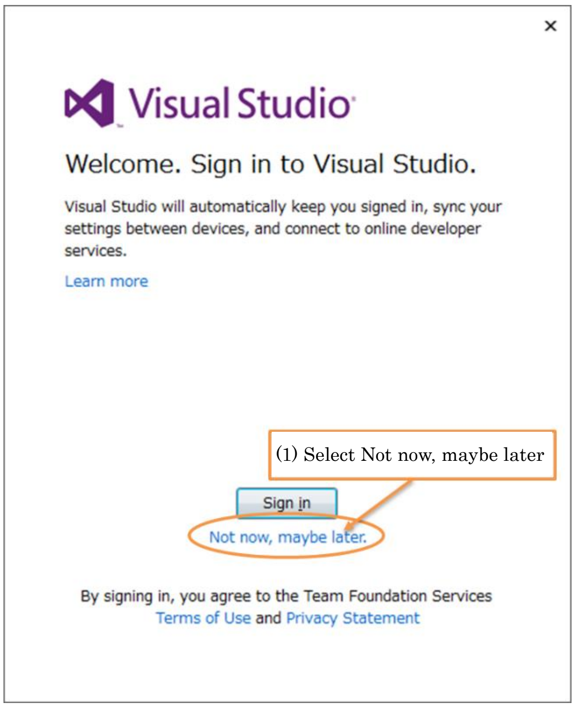
(2) Select Install

インストール(N)

Initial configuration of Visual Studio Community 2013

Run Visual Studio Community 2013 after installation to perform initial configuration.

▼ Procedure (the configuration window appears automatically upon first launch)



Visual Studio

Welcome. Sign in to Visual Studio.

Visual Studio will automatically keep you signed in, sync your settings between devices, and connect to online developer services.

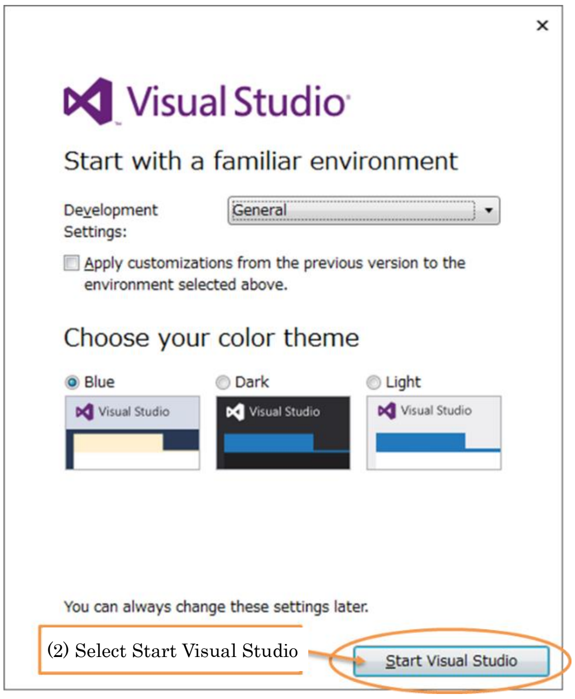
Learn more

(1) Select Not now, maybe later

Sign in

Not now, maybe later.

By signing in, you agree to the Team Foundation Services Terms of Use and Privacy Statement



Visual Studio

Start with a familiar environment

Development Settings: General

☐ Apply customizations from the previous version to the environment selected above.

Choose your color theme

[✓ Blue] Dark Light

You can always change these settings later.

(2) Select Start Visual Studio

Start Visual Studio

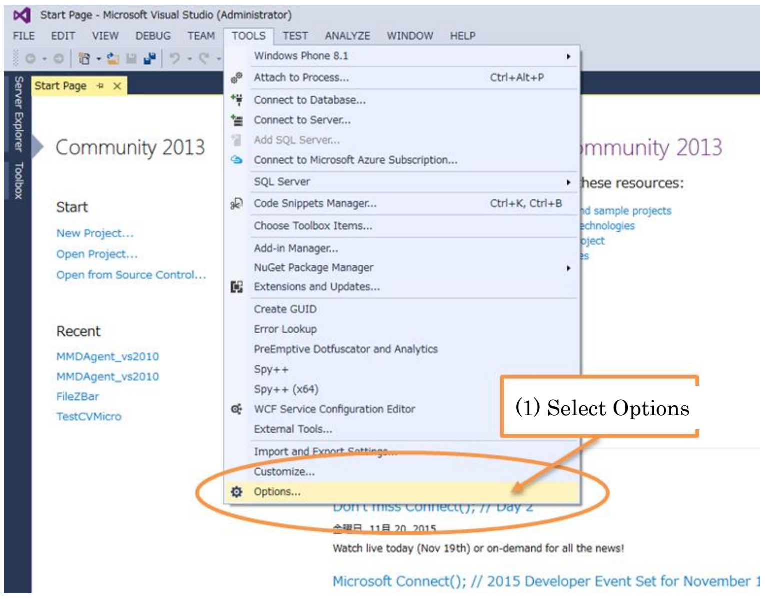
[Notes]

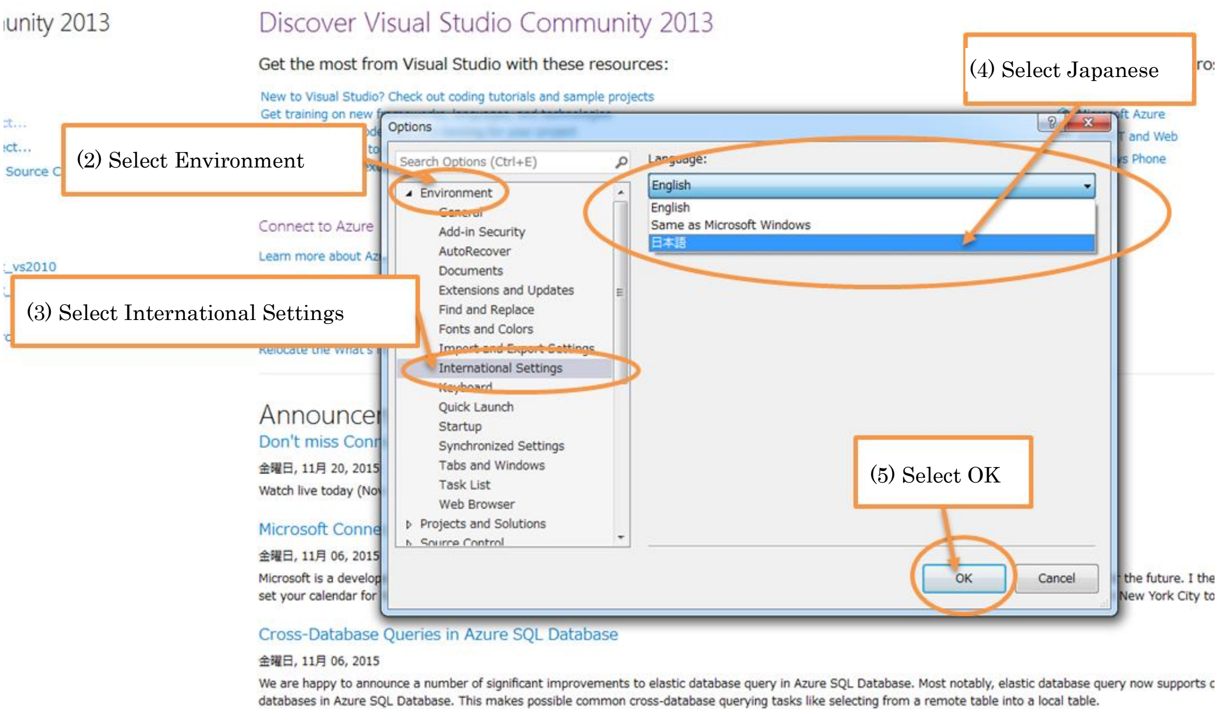
To continue using it beyond the evaluation period (30 days) requires signing in with a Microsoft account.

Setting Visual Studio Community 2013 to Japanese

Visual Studio Community 2013 can be set to Japanese in the Options settings.

▼ Procedure (the change is applied upon restarting Visual Studio)


Start Page - Microsoft Visual Studio (Administrator)
FILE EDIT VIEW DEBUG TEAM TOOLS TEST ANALYZE WINDOW HELP
Windows Phone 8.1
Attach to Process... Ctrl+Alt+P
Connect to Database...
Connect to Server...
Add SQL Server...
Connect to Microsoft Azure Subscription...
SQL Server
Code Snippets Manager... Ctrl+K, Ctrl+B
Choose Toolbox Items...
Add-in Manager...
NuGet Package Manager
Extensions and Updates...
Create GUID
Error Lookup
PreEmptive Dotfuscator and Analytics
Spy++
Spy++ (x64)
WCF Service Configuration Editor
External Tools...
Import and Export Settings...
Customize...
Options...
(1) Select Options



Building the run-time environment

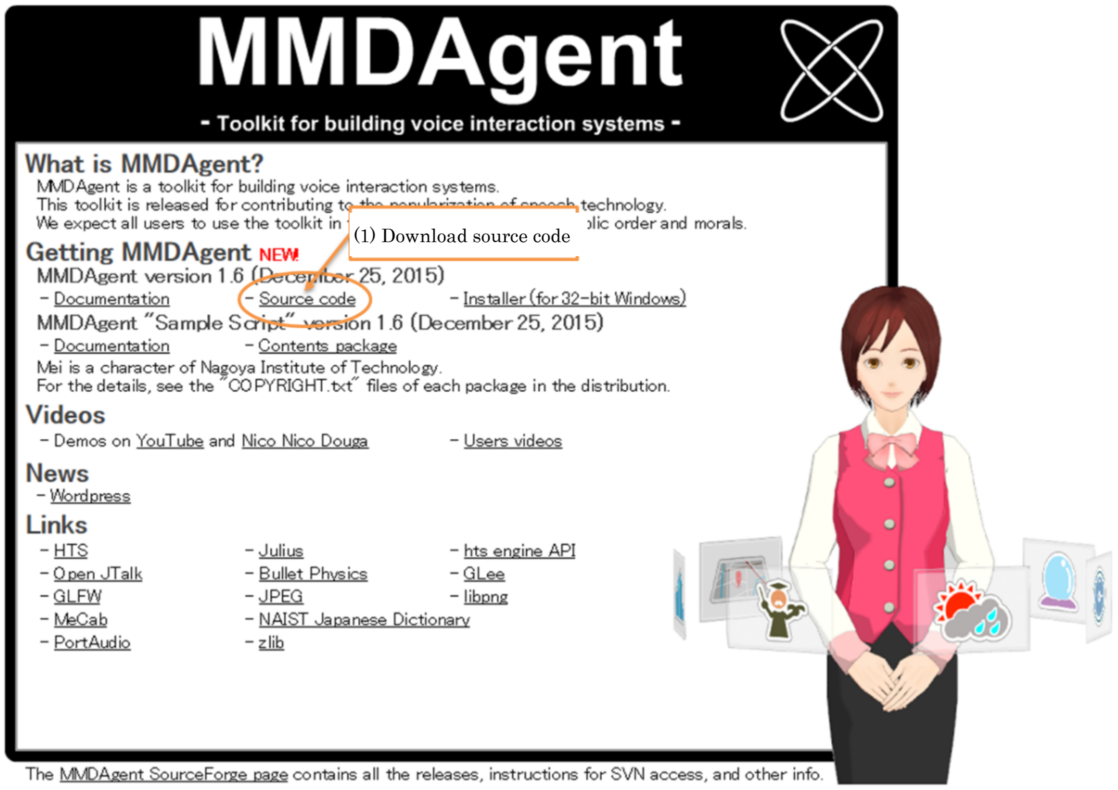
Getting the source code

Download the MMDAgent source code from the Web site and save it in a folder on your PC. The source code is archived in a Zip file, which must be extracted before performing the procedures below.

▼ MMDAgent Web site

http://www.mmdagent.jp/

▼ Procedure



MMDAgent

- Toolkit for building voice interaction systems -

What is MMDAgent?

MMDAgent is a toolkit for building voice interaction systems.
This toolkit is released for contributing to the popularization of speech technology.
We expect all users to use the toolkit in [illegible] olic order and morals.

Getting MMDAgent NEW!

MMDAgent version 1.6 (December 25, 2015)
- Documentation - Source code - Installer (for 32-bit Windows)

MMDAgent "Sample Script" version 1.6 (December 25, 2015)
- Documentation - Contents package

Mei is a character of Nagoya Institute of Technology.
For the details, see the "COPYRIGHT.txt" files of each package in the distribution.

Videos

- Demos on YouTube and Nico Nico Douga - Users videos

News

- Wordpress

Links

- HTS - Julius - hts engine API
- Open JTalk - Bullet Physics - GLee
- GLFW - JPEG - libpng
- MeCab - NAIST Japanese Dictionary
- PortAudio - zlib

The MMDAgent SourceForge page contains all the releases, instructions for SVN access, and other info.

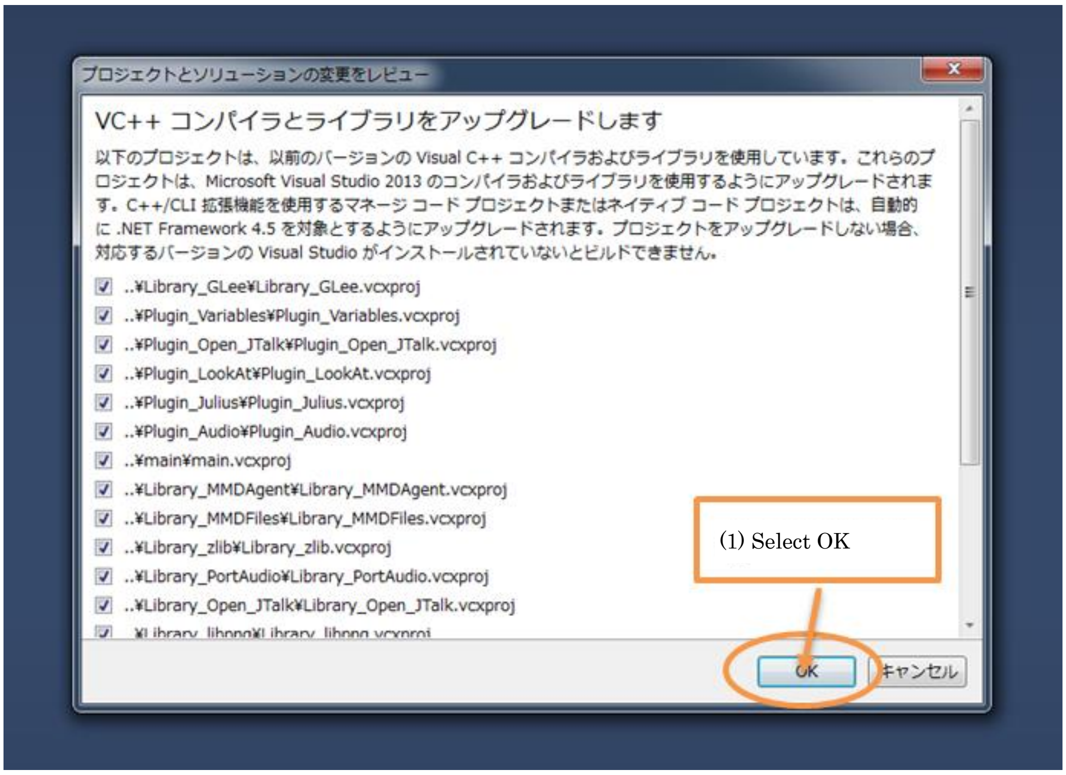
Building and running the source code

This section describes preparations to run the source code with Visual Studio Community 2013.

▼ Procedure

1. Open the solution

   Open the MMDAgent_vs2010.sln file in the extracted folder with Visual Studio Community 2013. When opening the solution, a window to upgrade to Visual Studio 2013 will appear, so select "OK" to upgrade the entire project.



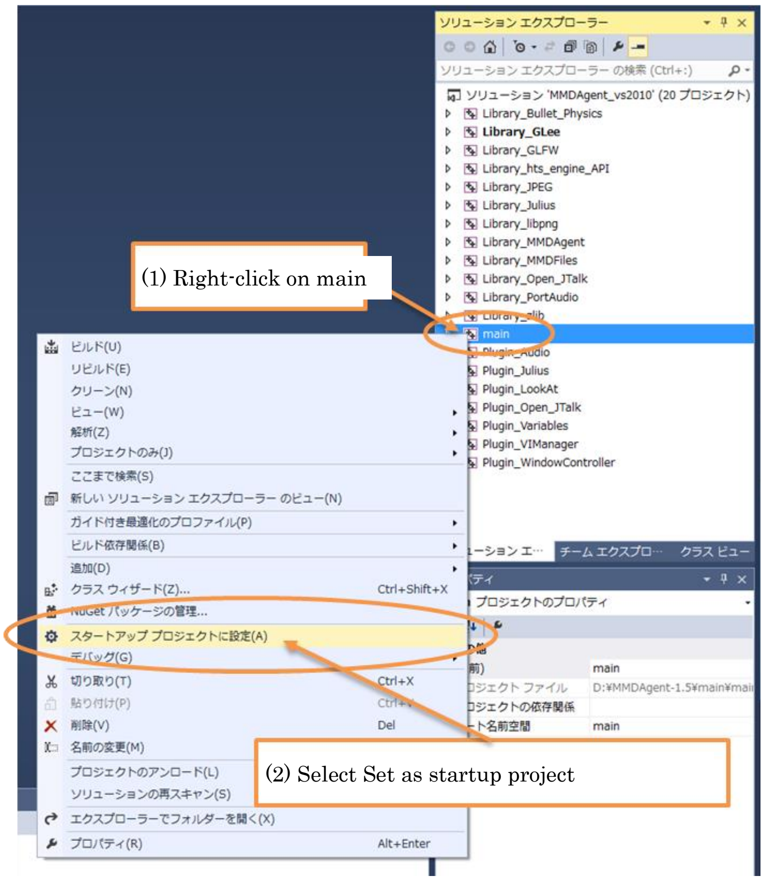
VC++ コンパイラとライブラリをアップグレードします

以下のプロジェクトは、以前のバージョンの Visual C++ コンパイラおよびライブラリを使用しています。これらのプロジェクトは、Microsoft Visual Studio 2013 のコンパイラおよびライブラリを使用するようにアップグレードされます。C++/CLI 拡張機能を使用するマネージ コード プロジェクトまたはネイティブ コード プロジェクトは、自動的に .NET Framework 4.5 を対象とするようにアップグレードされます。プロジェクトをアップグレードしない場合、対応するバージョンの Visual Studio がインストールされていないとビルドできません。

- ✓ ..¥Library_GLee¥Library_GLee.vcxproj
- ✓ ..¥Plugin_Variables¥Plugin_Variables.vcxproj
- ✓ ..¥Plugin_Open_JTalk¥Plugin_Open_JTalk.vcxproj
- ✓ ..¥Plugin_LookAt¥Plugin_LookAt.vcxproj
- ✓ ..¥Plugin_Julius¥Plugin_Julius.vcxproj
- ✓ ..¥Plugin_Audio¥Plugin_Audio.vcxproj
- ✓ ..¥main¥main.vcxproj
- ✓ ..¥Library_MMDAgent¥Library_MMDAgent.vcxproj
- ✓ ..¥Library_MMDFiles¥Library_MMDFiles.vcxproj
- ✓ ..¥Library_zlib¥Library_zlib.vcxproj
- ✓ ..¥Library_PortAudio¥Library_PortAudio.vcxproj
- ✓ ..¥Library_Open_JTalk¥Library_Open_JTalk.vcxproj
- ✓ ..¥Library_libpng¥Library_libpng.vcxproj

2. Set the startup project

   Set the startup project to main

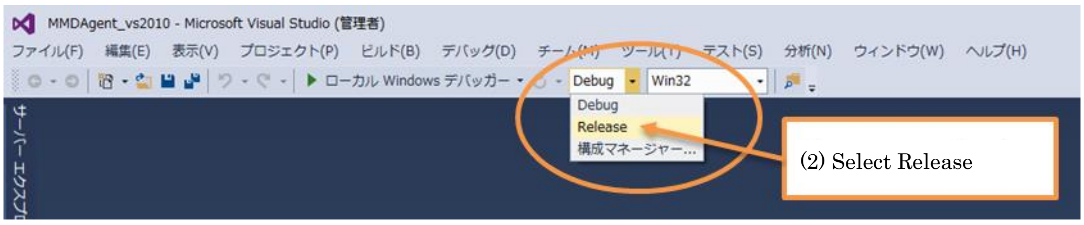


[Notes]

The name of the startup project will be shown in bold.

3. Change the solution configuration

   Change the solution configuration to Release.

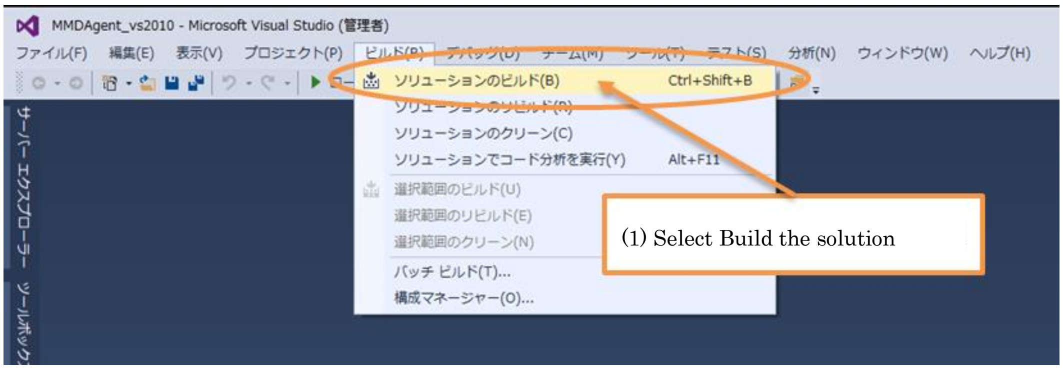

MMDAgent_vs2010 - Microsoft Visual Studio (管理者)
ファイル(F) 編集(E) 表示(V) プロジェクト(P) ビルド(B) デバッグ(D) チーム(M) ツール(T) テスト(S) 分析(N) ウィンドウ(W) ヘルプ(H)
ローカル Windows デバッガー
Debug
Win32
Debug
Release
構成マネージャー...
(2) Select Release

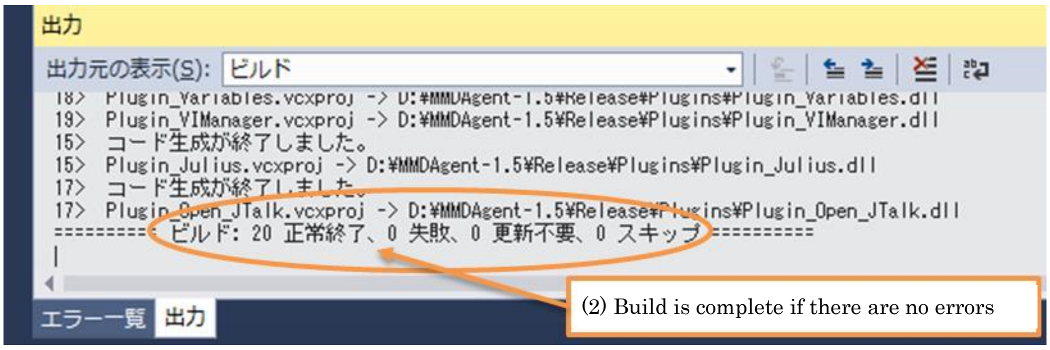
[Notes]

If executed in Debug configuration, AppData and MMDAgent.mdf will be copied from the Release folder to the Debug folder.

4. Build

Build the solution

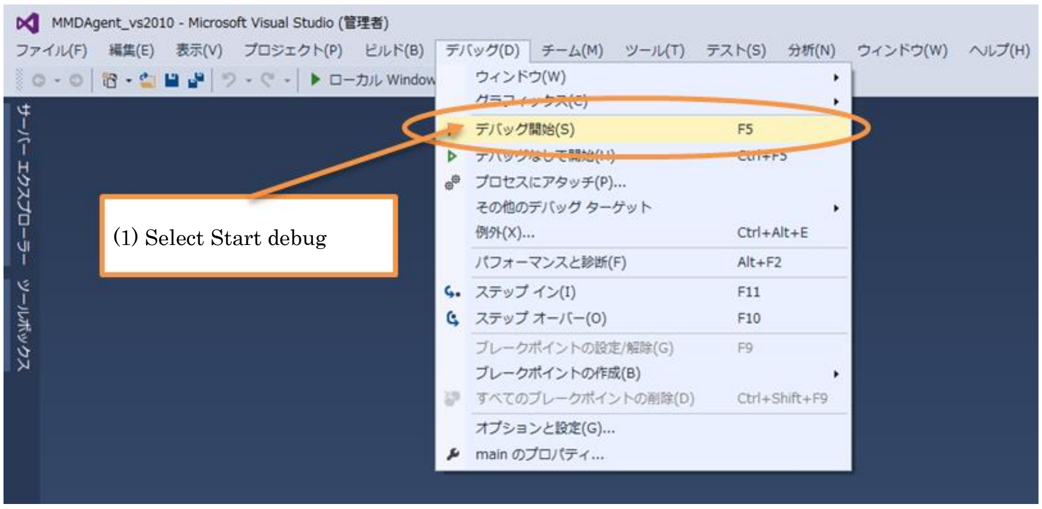

MMDAgent_vs2010 - Microsoft Visual Studio (管理者)
ファイル(F)
編集(E)
表示(V)
プロジェクト(P)
ビルド(B)
デバッグ(D)
チーム(M)
ツール(T)
テスト(S)
分析(N)
ウィンドウ(W)
ヘルプ(H)
ソリューションのビルド(B) Ctrl+Shift+B
ソリューションのリビルド(R)
ソリューションのクリーン(C)
ソリューションでコード分析を実行(Y) Alt+F11
選択範囲のビルド(U)
選択範囲のリビルド(E)
選択範囲のクリーン(N)
バッチ ビルド(T)...
構成マネージャー(O)...
サーバー エクスプローラー
ツールボックス
(1) Select Build the solution



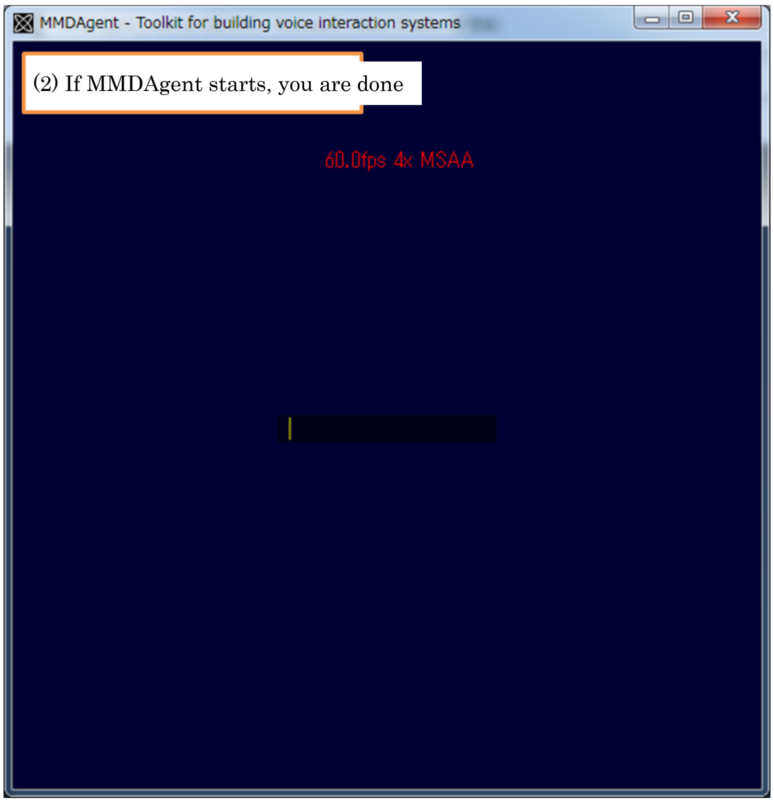
```
18>  Plugin_Variables.vcxproj -> D:¥MMDAgent-1.5¥Release¥Plugins¥Plugin_Variables.dll
19>  Plugin_VIManager.vcxproj -> D:¥MMDAgent-1.5¥Release¥Plugins¥Plugin_VIManager.dll
15>  コード生成が終了しました。
15>  Plugin_Julius.vcxproj -> D:¥MMDAgent-1.5¥Release¥Plugins¥Plugin_Julius.dll
17>  コード生成が終了しました。
17>  Plugin_Open_JTalk.vcxproj -> D:¥MMDAgent-1.5¥Release¥Plugins¥Plugin_Open_JTalk.dll
========== ビルド: 20 正常終了、0 失敗、0 更新不要、0 スキップ ==========
```

5. Run

Run the exe file generated by the build.


MMDAgent_vs2010 - Microsoft Visual Studio (管理者)
ファイル(F) 編集(E) 表示(V) プロジェクト(P) ビルド(B) デバッグ(D) チーム(M) ツール(T) テスト(S) 分析(N) ウィンドウ(W) ヘルプ(H)
ローカル Window
ウィンドウ(W)
グラフィックス(C)
デバッグ開始(S) F5
デバッグなしで開始(H) Ctrl+F5
プロセスにアタッチ(P)...
その他のデバッグ ターゲット
例外(X)... Ctrl+Alt+E
パフォーマンスと診断(F) Alt+F2
ステップ イン(I) F11
ステップ オーバー(O) F10
ブレークポイントの設定/解除(G) F9
ブレークポイントの作成(B)
すべてのブレークポイントの削除(D) Ctrl+Shift+F9
オプションと設定(G)...
main のプロパティ...
サーバー エクスプローラー
ツールボックス

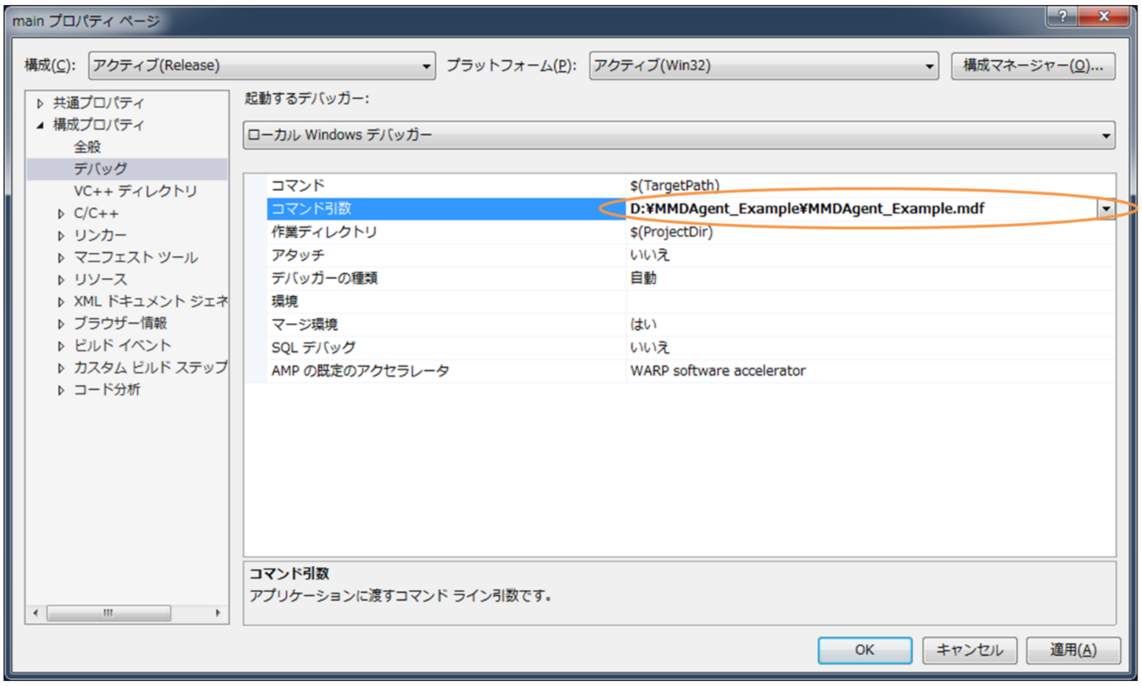
(1) Select Start debug



(2) If MMDAgent starts, you are done

[Notes]
There is a Contents Package in the extracted folder from the Web site. By giving the path of MMDAgent_Example.mdf as a parameter when running MMDAgent, development can be done from an initial configuration that displays a character called “Mei”.

▼ Configuration procedure
Open the project properties for main and under [Configuration properties] -> [Debug], set Command arguments to the path of MMDAgent_Example.mdf.

▼ Screen shot

main プロパティ ページ

構成(C): アクティブ(Release)　プラットフォーム(P): アクティブ(Win32)

起動するデバッガー: ローカル Windows デバッガー

コマンド	$(TargetPath)
コマンド引数	D:¥MMDAgent_Example¥MMDAgent_Example.mdf
作業ディレクトリ	$(ProjectDir)
アタッチ	いいえ
デバッガーの種類	自動
環境	
マージ環境	はい
SQL デバッグ	いいえ
AMP の既定のアクセラレータ	WARP software accelerator

コマンド引数
アプリケーションに渡すコマンド ライン引数です。

Developing a new plugin

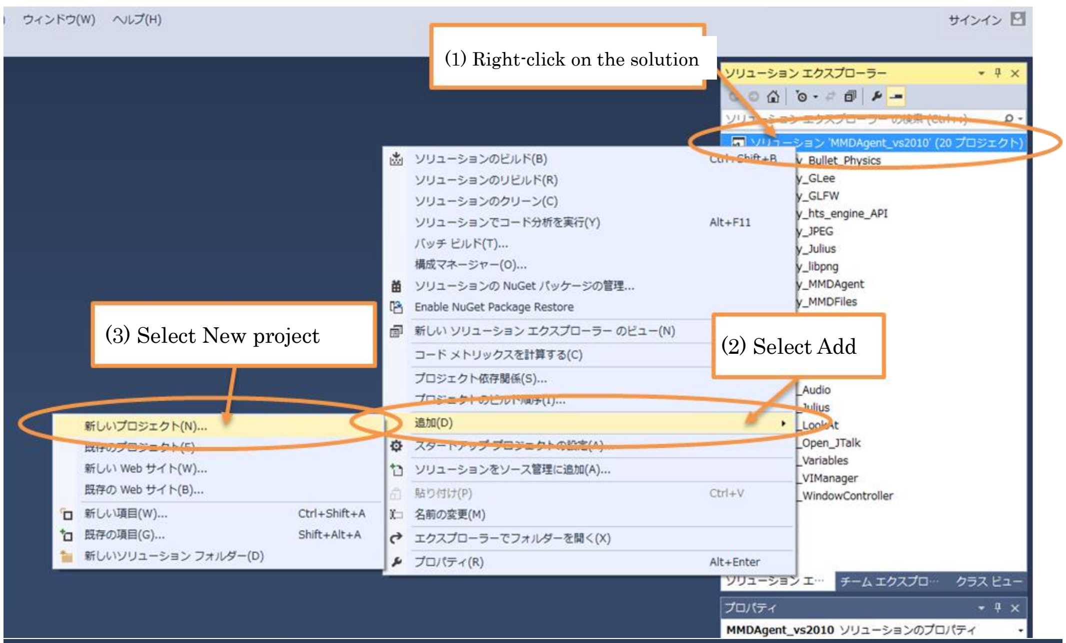
Adding and configuring a new project

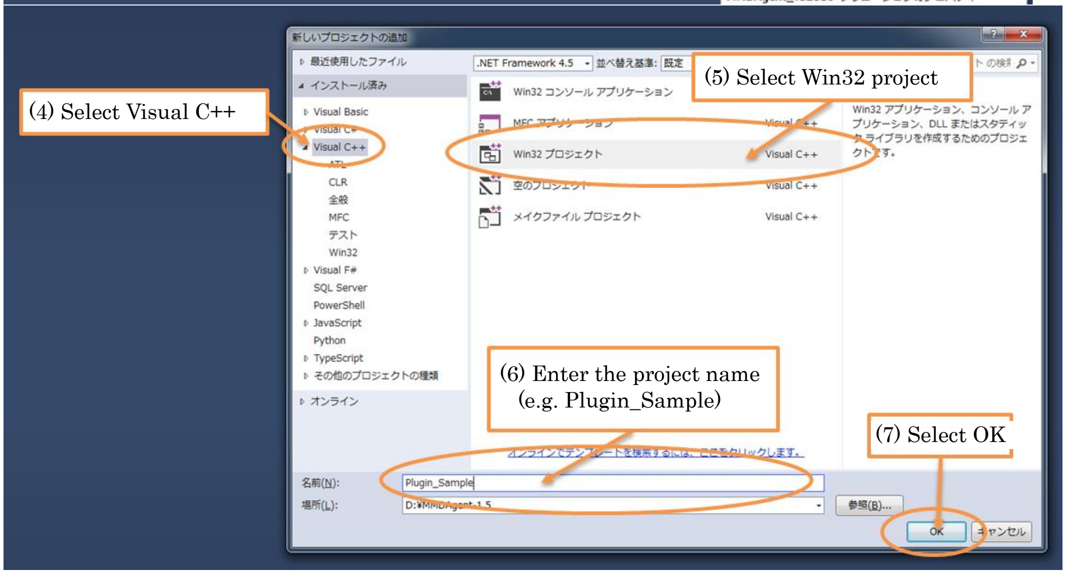
This section describes how to add a new project and prepare to develop a new plugin.

▼ Procedure

1. Add a project

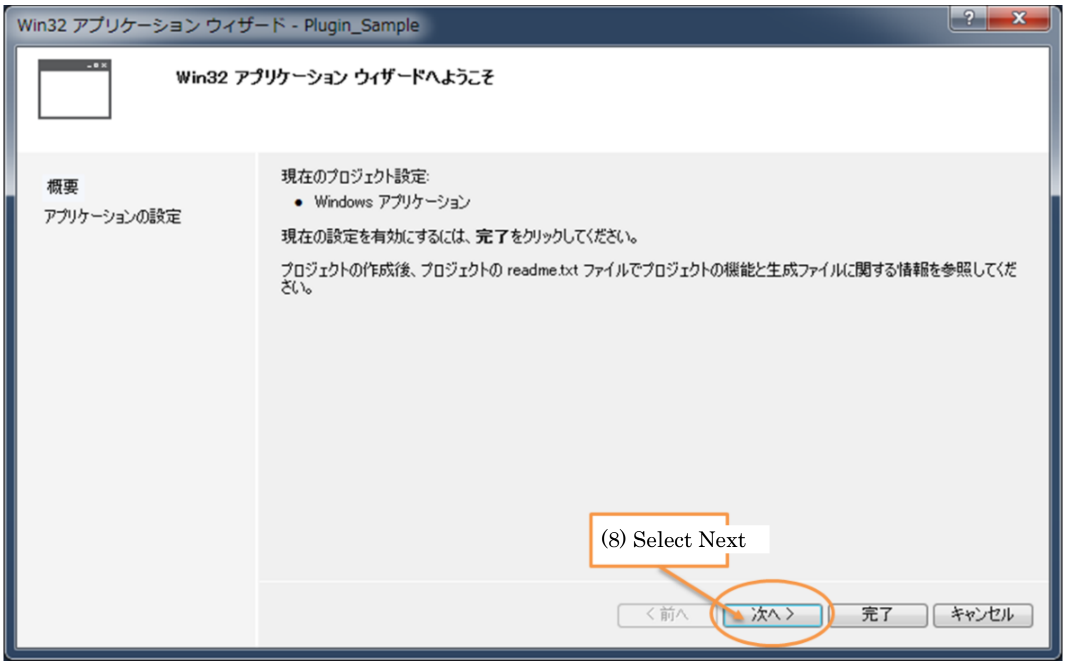
   Add a project to the solution. When a new project is added in the solution explorer, the Add Project window appears. Initialize as shown in the screen shots below.


(1) Right-click on the solution
ソリューション エクスプローラー
ソリューション 'MMDAgent_vs2010' (20 プロジェクト)
(2) Select Add
(3) Select New project
ソリューションのビルド(B)
ソリューションのリビルド(R)
ソリューションのクリーン(C)
追加(D)
新しいプロジェクト(N)...
既存のプロジェクト(E)...
プロパティ(R)



名前(N): Plugin_Sample

場所(L): D:¥MMDAgent-1.5


Win32 アプリケーション ウィザード - Plugin_Sample

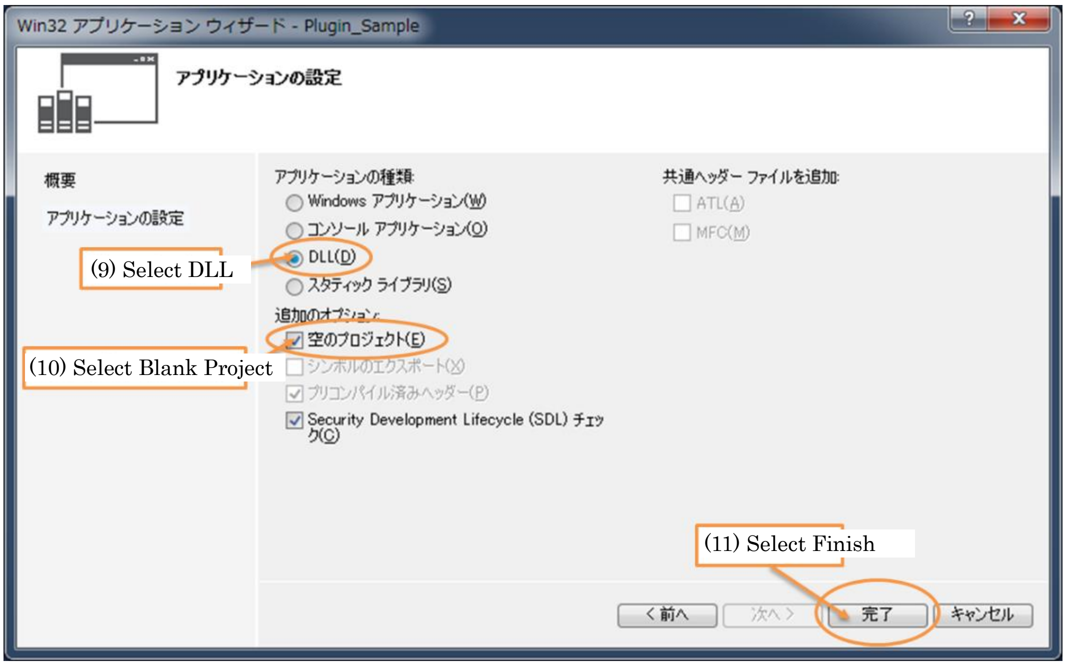
Win32 アプリケーション ウィザードへようこそ

概要
アプリケーションの設定

現在のプロジェクト設定:
- Windows アプリケーション

現在の設定を有効にするには、**完了** をクリックしてください。

プロジェクトの作成後、プロジェクトの readme.txt ファイルでプロジェクトの機能と生成ファイルに関する情報を参照してください。


Win32 アプリケーション ウィザード - Plugin_Sample

アプリケーションの設定

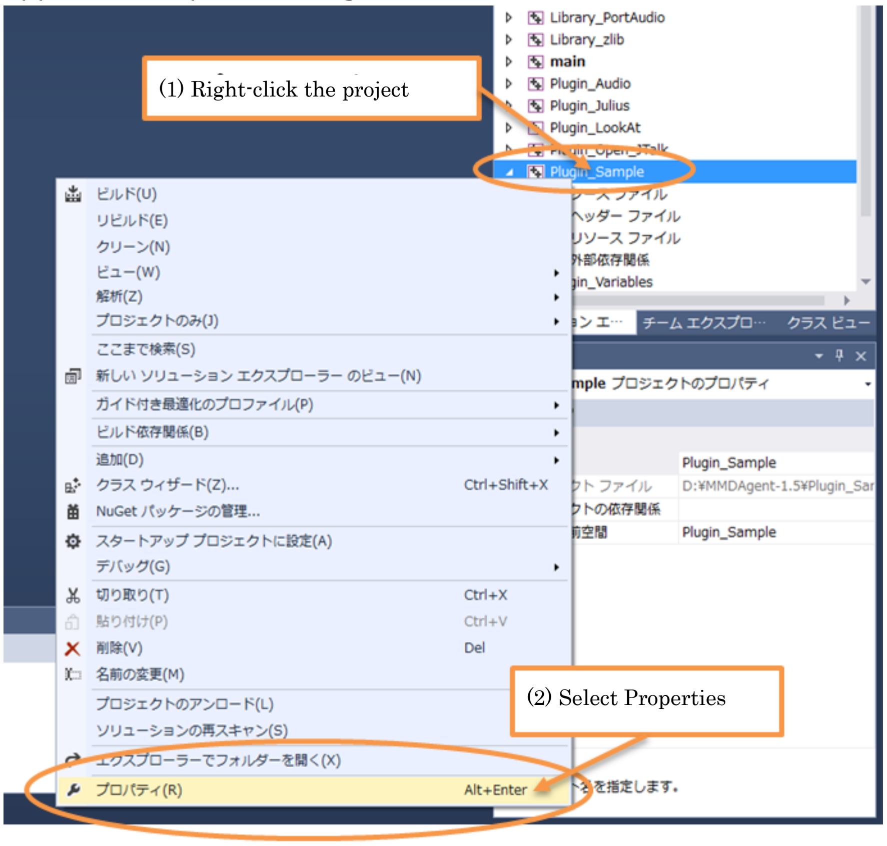
概要
アプリケーションの設定

アプリケーションの種類:
- ○ Windows アプリケーション(W)
- ○ コンソール アプリケーション(O)
- ● DLL(D)
- ○ スタティック ライブラリ(S)

追加のオプション:
- ☑ 空のプロジェクト(E)
- ☐ シンボルのエクスポート(X)
- ☑ プリコンパイル済みヘッダー(P)
- ☑ Security Development Lifecycle (SDL) チェック(C)

共通ヘッダー ファイルを追加:
- ☐ ATL(A)
- ☐ MFC(M)

2. Change the project settings
   Change the settings of the added project. After the property page window appears, complete settings I to V below.

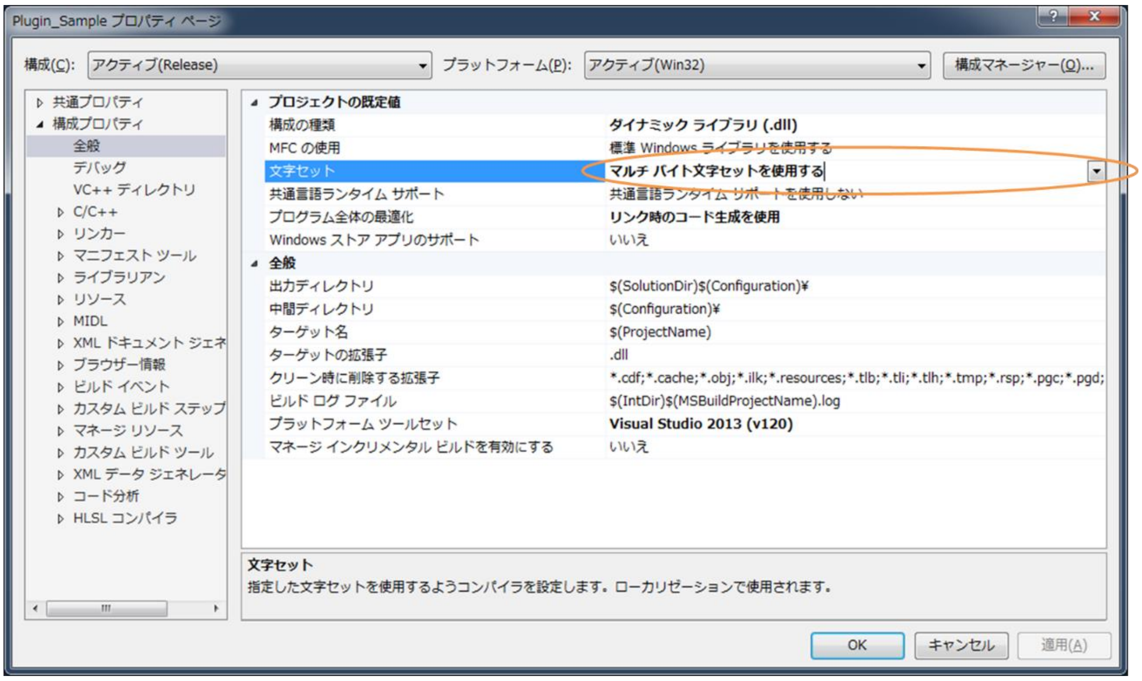


[Notes]
To run a Debug configuration, change the solution configuration to Debug and then perform these settings.

I. Character set settings

Change the [Character set] setting in [Configuration properties]->[General] to [Uses multi-byte character set].

▼ Screen shot



プロジェクトの既定値	
構成の種類	**ダイナミック ライブラリ (.dll)**
MFC の使用	標準 Windows ライブラリを使用する
文字セット	**マルチ バイト文字セットを使用する**
共通言語ランタイム サポート	共通言語ランタイム サポートを使用しない
プログラム全体の最適化	**リンク時のコード生成を使用**
Windows ストア アプリのサポート	いいえ
全般	
出力ディレクトリ	$(SolutionDir)$(Configuration)¥
中間ディレクトリ	$(Configuration)¥
ターゲット名	$(ProjectName)
ターゲットの拡張子	.dll
クリーン時に削除する拡張子	*.cdf;*.cache;*.obj;*.ilk;*.resources;*.tlb;*.tli;*.tlh;*.tmp;*.rsp;*.pgc;*.pgd;
ビルド ログ ファイル	$(IntDir)$(MSBuildProjectName).log
プラットフォーム ツールセット	**Visual Studio 2013 (v120)**
マネージ インクリメンタル ビルドを有効にする	いいえ

文字セット
指定した文字セットを使用するようコンパイラを設定します。ローカリゼーションで使用されます。

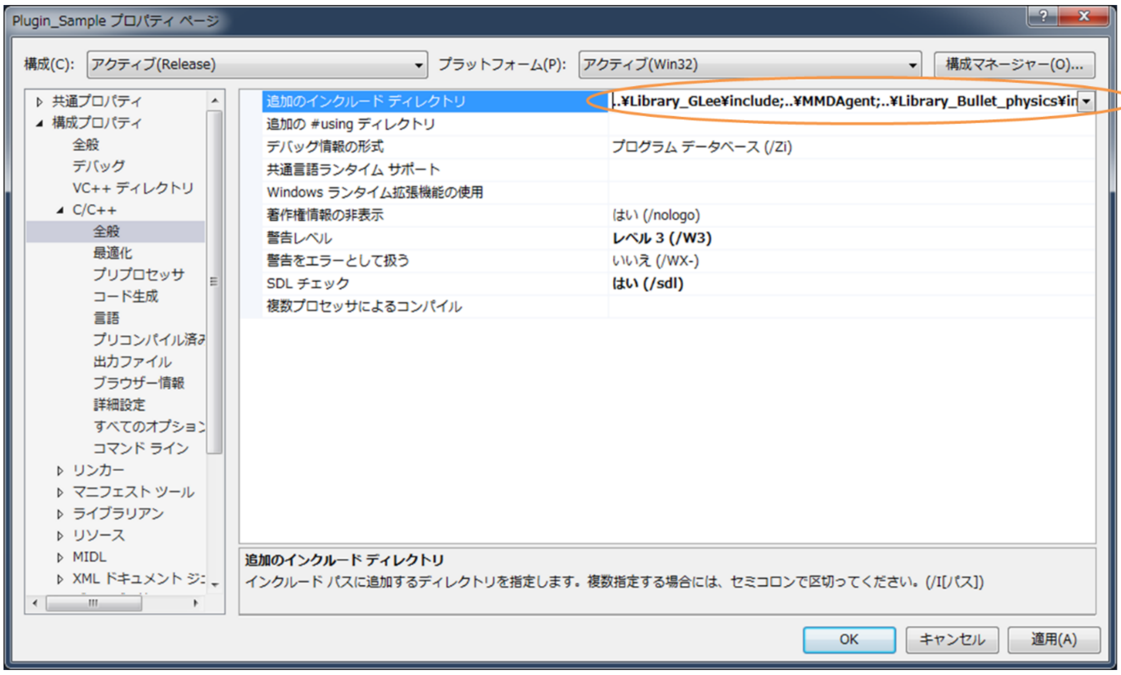
II. Set additional include directories

Enter the following values in [Additional include directories] in [Configuration properties]->[C/C++]->[General].

▼ Setting value

```
..¥Library_GLee¥include;..¥Library_Bullet_physics¥include;..¥Libra
ry_MMDFiles¥include;..¥Library_Julius¥include;..¥Library_MMDAgent¥
include;..¥Library_GLFW¥include;
```

▼ Screen shot



Plugin_Sample プロパティ ページ

構成(C): アクティブ(Release)　プラットフォーム(P): アクティブ(Win32)

追加のインクルード ディレクトリ	..¥Library_GLee¥include;..¥MMDAgent;..¥Library_Bullet_physics¥in
追加の #using ディレクトリ	
デバッグ情報の形式	プログラム データベース (/Zi)
共通言語ランタイム サポート	
Windows ランタイム拡張機能の使用	
著作権情報の非表示	はい (/nologo)
警告レベル	**レベル 3 (/W3)**
警告をエラーとして扱う	いいえ (/WX-)
SDL チェック	**はい (/sdl)**
複数プロセッサによるコンパイル	

追加のインクルード ディレクトリ
インクルード パスに追加するディレクトリを指定します。複数指定する場合には、セミコロンで区切ってください。(/I[パス])

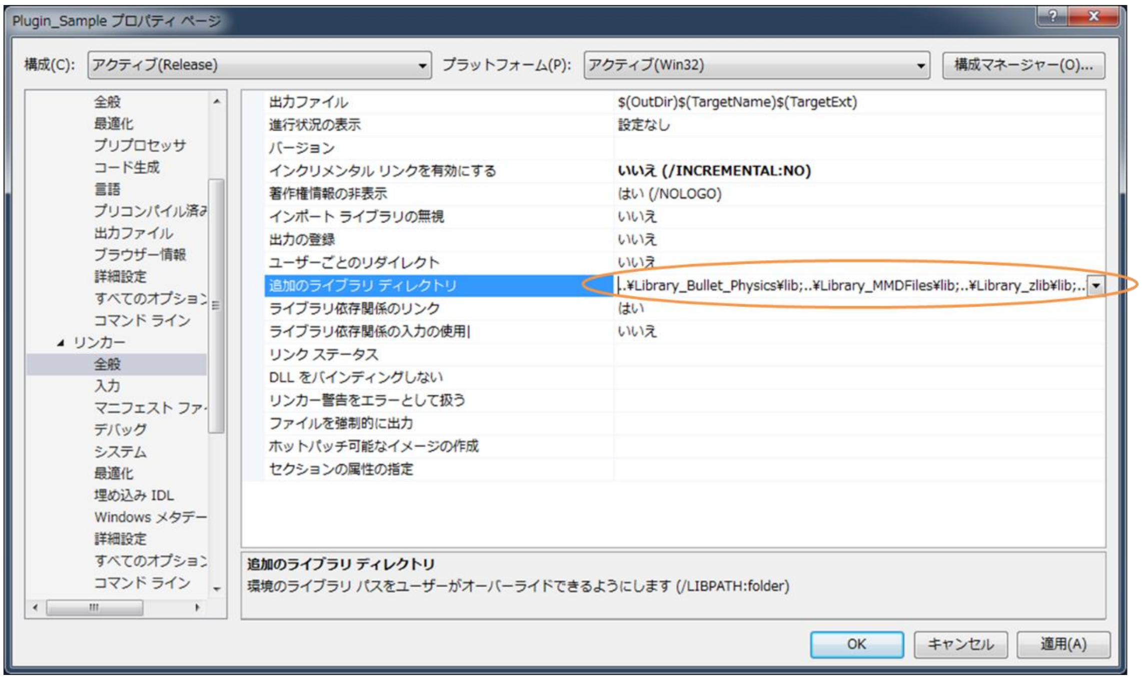
III. Set additional library directories

Set [Additional library directories] in [Configuration properties] -> [Linker] -> [General] to the following value.

▼ Setting value

```
..¥Library_Bullet_Physics¥lib;..¥Library_MMDFiles¥lib;..¥Library_z
lib¥lib;..¥Library_libpng¥lib;..¥Library_GLee¥lib;..¥Library_MMDAg
ent¥lib;..¥Library_JPEG¥lib;..¥Library_GLFW¥lib;..¥Library_FreeTyp
e¥lib
```

▼ Screen shot



構成(C): アクティブ(Release)　プラットフォーム(P): アクティブ(Win32)

出力ファイル	$(OutDir)$(TargetName)$(TargetExt)
進行状況の表示	設定なし
バージョン	
インクリメンタル リンクを有効にする	**いいえ (/INCREMENTAL:NO)**
著作権情報の非表示	はい (/NOLOGO)
インポート ライブラリの無視	いいえ
出力の登録	いいえ
ユーザーごとのリダイレクト	いいえ
追加のライブラリ ディレクトリ	..¥Library_Bullet_Physics¥lib;..¥Library_MMDFiles¥lib;..¥Library_zlib¥lib;..
ライブラリ依存関係のリンク	はい
ライブラリ依存関係の入力の使用	いいえ
リンク ステータス	
DLL をバインディングしない	
リンカー警告をエラーとして扱う	
ファイルを強制的に出力	
ホットパッチ可能なイメージの作成	
セクションの属性の指定	

追加のライブラリ ディレクトリ
環境のライブラリ パスをユーザーがオーバーライドできるようにします (/LIBPATH:folder)

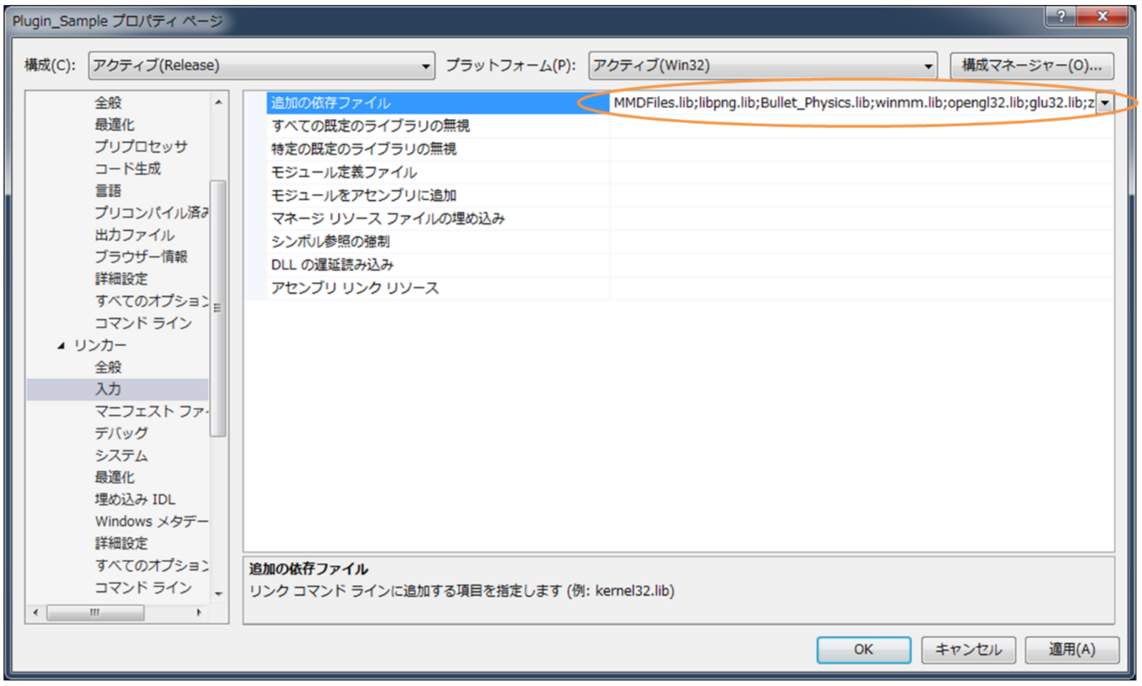
IV. Added dependency file settings

Set [Additional dependency files] in [Configuration property] -> [Linker] -> [Input] to the following value. It is okay to overwrite the value input initially.

▼ Setting value

```
MMDFiles.lib;libpng.lib;Bullet_Physics.lib;winmm.lib;opengl32.lib;
glu32.lib;zlib.lib;GLee.lib;MMDAgent.lib;JPEG.lib;GLFW.lib;FreeTyp
e.lib;%(AdditionalDependencies)
```

▼ Screen shot



Plugin_Sample プロパティ ページ

構成(C): アクティブ(Release)　プラットフォーム(P): アクティブ(Win32)　構成マネージャー(O)...

全般
最適化
プリプロセッサ
コード生成
言語
プリコンパイル済み
出力ファイル
ブラウザー情報
詳細設定
すべてのオプション
コマンド ライン
リンカー
全般
入力
マニフェスト ファ
デバッグ
システム
最適化
埋め込み IDL
Windows メタデー
詳細設定
すべてのオプション
コマンド ライン

追加の依存ファイル	MMDFiles.lib;libpng.lib;Bullet_Physics.lib;winmm.lib;opengl32.lib;glu32.lib;z
すべての既定のライブラリの無視	
特定の既定のライブラリの無視	
モジュール定義ファイル	
モジュールをアセンブリに追加	
マネージ リソース ファイルの埋め込み	
シンボル参照の強制	
DLL の遅延読み込み	
アセンブリ リンク リソース	

追加の依存ファイル
リンク コマンド ラインに追加する項目を指定します (例: kernel32.lib)

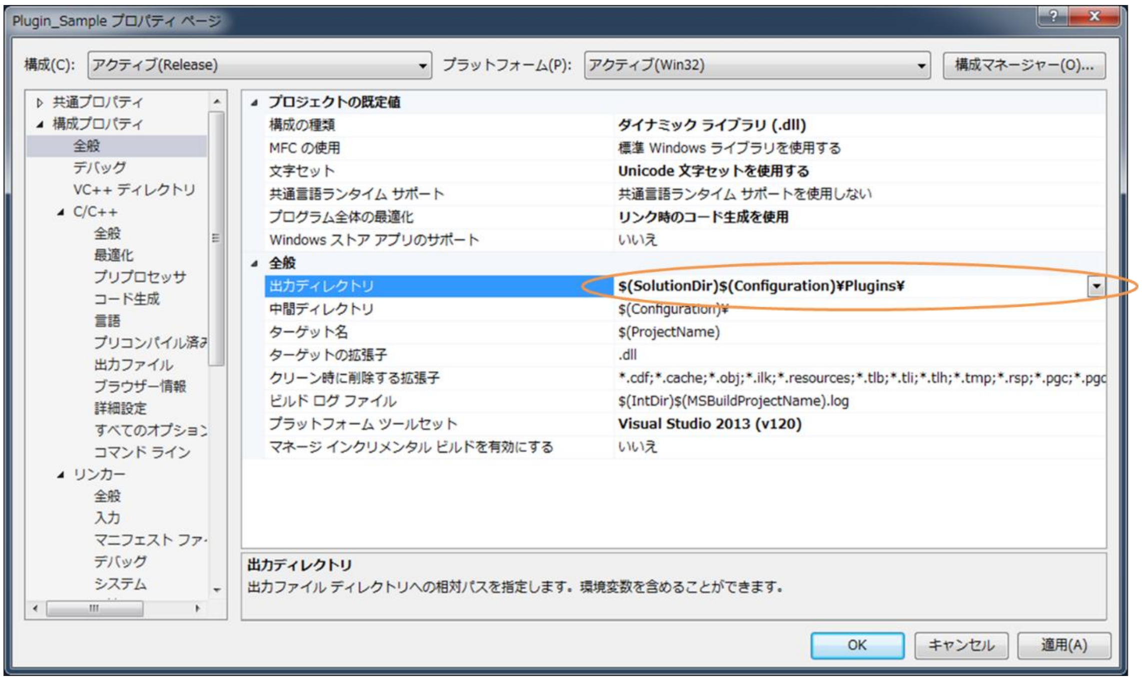
V. Set the output directory

Set [Output directory] in [Configuration properties]->[General] to the following value.

▼ Setting value

```
$(SolutionDir)$(Configuration)¥Plugins¥
```

▼ Screen shot



プロジェクトの既定値

構成の種類	ダイナミック ライブラリ (.dll)
MFC の使用	標準 Windows ライブラリを使用する
文字セット	Unicode 文字セットを使用する
共通言語ランタイム サポート	共通言語ランタイム サポートを使用しない
プログラム全体の最適化	リンク時のコード生成を使用
Windows ストア アプリのサポート	いいえ

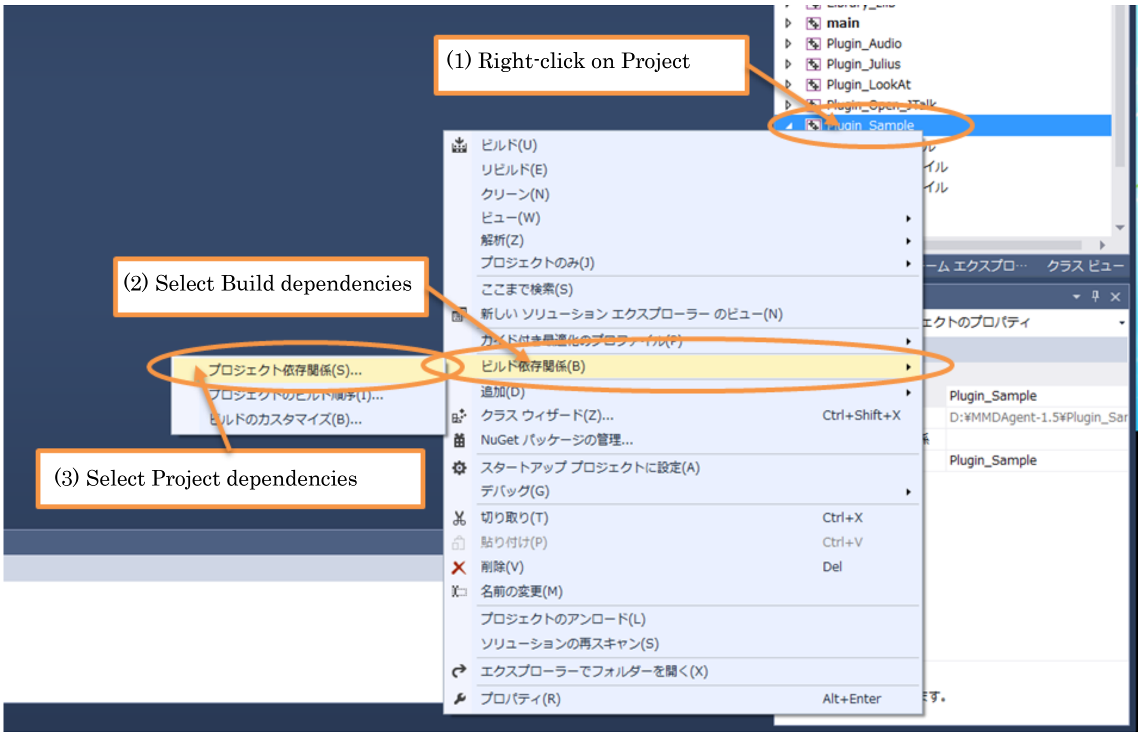
全般

出力ディレクトリ	$(SolutionDir)$(Configuration)¥Plugins¥
中間ディレクトリ	$(Configuration)¥
ターゲット名	$(ProjectName)
ターゲットの拡張子	.dll
クリーン時に削除する拡張子	*.cdf;*.cache;*.obj;*.ilk;*.resources;*.tlb;*.tli;*.tlh;*.tmp;*.rsp;*.pgc;*.pgc
ビルド ログ ファイル	$(IntDir)$(MSBuildProjectName).log
プラットフォーム ツールセット	Visual Studio 2013 (v120)
マネージ インクリメンタル ビルドを有効にする	いいえ

出力ディレクトリ
出力ファイル ディレクトリへの相対パスを指定します。環境変数を含めることができます。

3. Set the project dependency relationships

Set the dependency relationships for the added project.

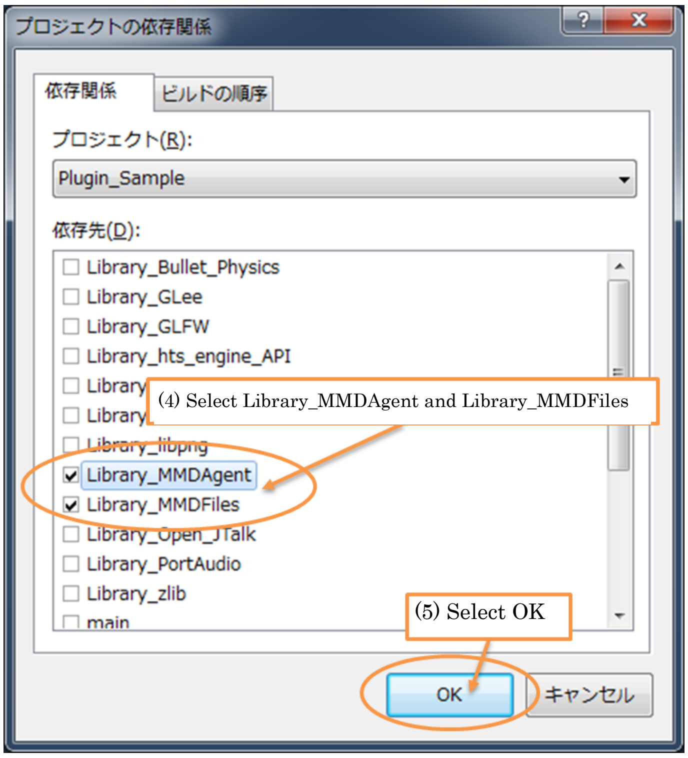


(1) Right-click on Project

(2) Select Build dependencies

(3) Select Project dependencies

main
Plugin_Audio
Plugin_Julius
Plugin_LookAt
Plugin_Open_JTalk
Plugin_Sample
ビルド(U)
リビルド(E)
クリーン(N)
ビュー(W)
解析(Z)
プロジェクトのみ(J)
ここまで検索(S)
新しい ソリューション エクスプローラー のビュー(N)
ガイド付き最適化のプロファイル(P)
ビルド依存関係(B)
追加(D)
クラス ウィザード(Z)... Ctrl+Shift+X
NuGet パッケージの管理...
スタートアップ プロジェクトに設定(A)
デバッグ(G)
切り取り(T) Ctrl+X
貼り付け(P) Ctrl+V
削除(V) Del
名前の変更(M)
プロジェクトのアンロード(L)
ソリューションの再スキャン(S)
エクスプローラーでフォルダーを開く(X)
プロパティ(R) Alt+Enter
プロジェクト依存関係(S)...
プロジェクトのビルド順序(I)...
ビルドのカスタマイズ(B)...
ーム エクスプロ… クラス ビュー
ェクトのプロパティ
Plugin_Sample
D:¥MMDAgent-1.5¥Plugin_Sar
Plugin_Sample



(4) Select Library_MMDAgent and Library_MMDFiles

(5) Select OK

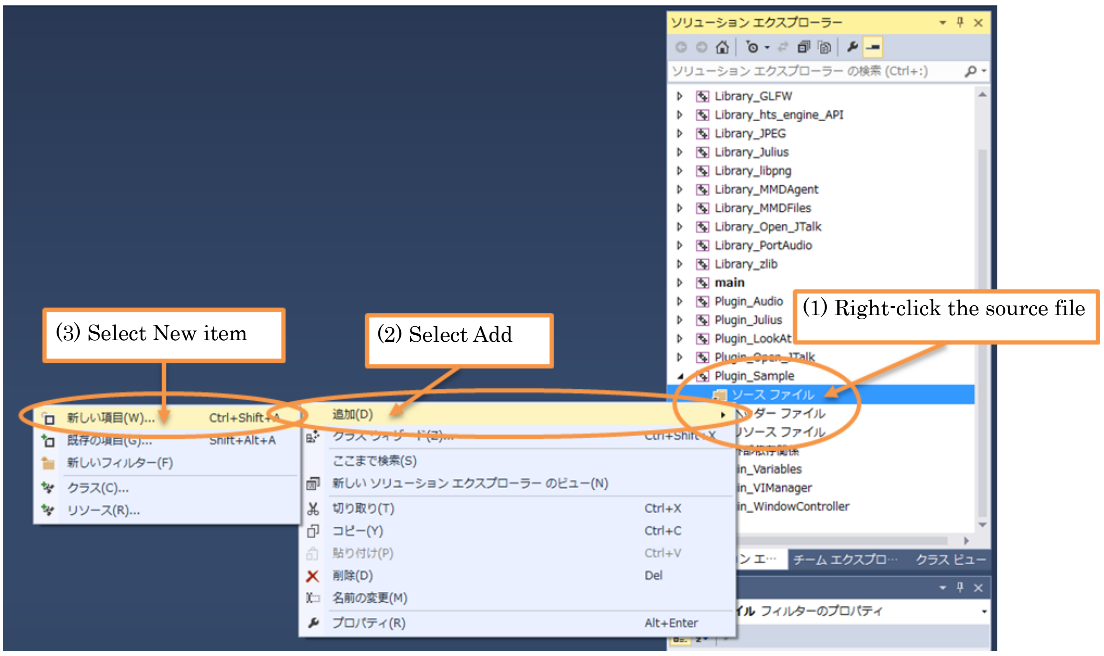
Creating, building and running a file

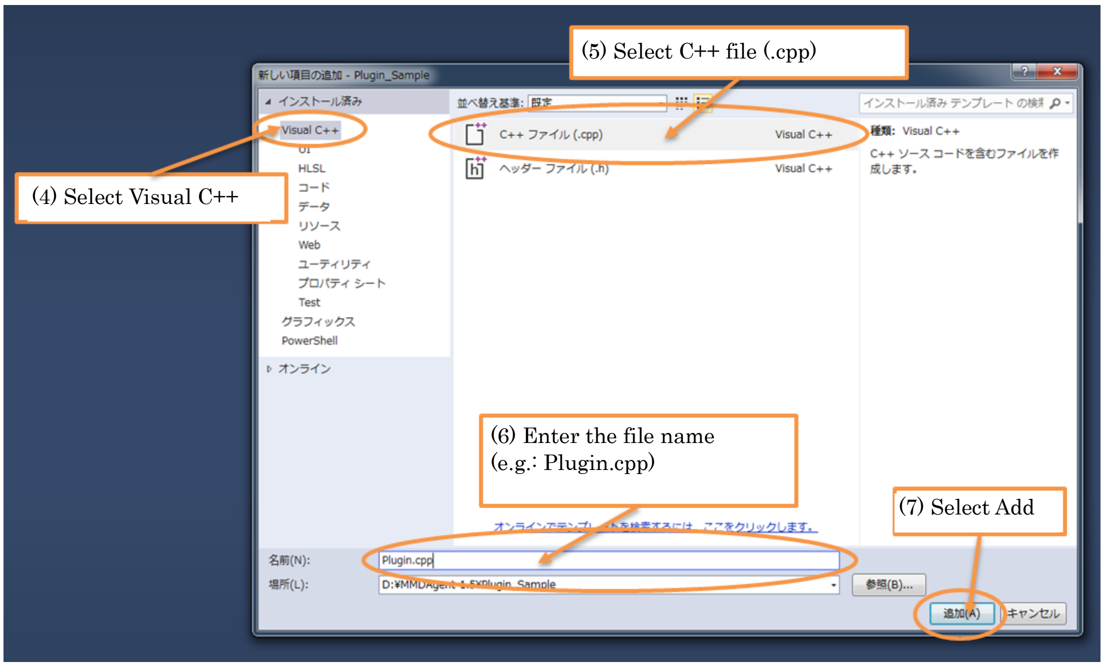
Add a source file and test run it.

▼ Procedure

1. Add the source file

   Add the source file to the project.


(1) Right-click the source file
(2) Select Add
(3) Select New item
ソリューション エクスプローラー
Plugin_Sample
ソース ファイル
追加(D)
新しい項目(W)... Ctrl+Shift+A
既存の項目(G)... Shift+Alt+A
新しいフィルター(F)
クラス(C)...
リソース(R)...



2. Enter test code

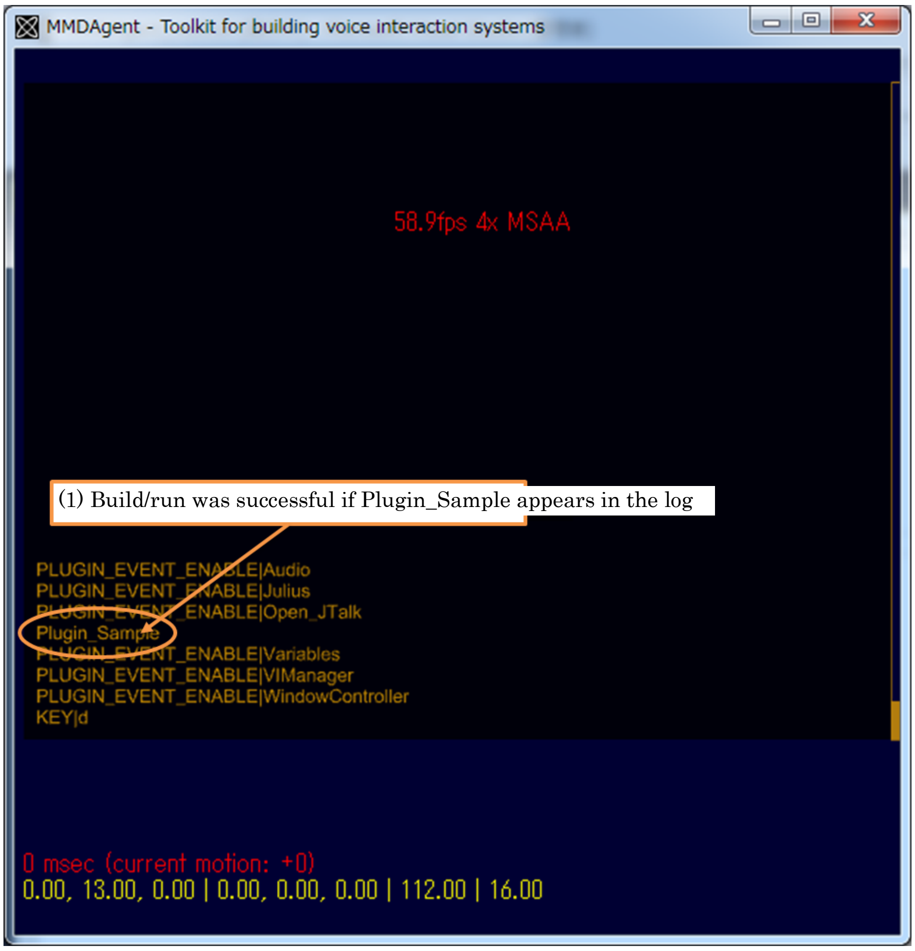
   Enter the following test code into the added source file.

```
#ifdef _WIN32
#define EXPORT extern "C" __declspec(dllexport)
#else
#define EXPORT extern "C"
#endif

#include "MMDAgent.h"

EXPORT void extAppStart(MMDAgent *mmdagent)
{
    // Log output: Plugin_Sample
    mmdagent->sendMessage("Plugin_Sample", "");
}
```

3. Build/Run

Build and Run in the same way as was done when building the run-time environment, and check that Plugin_Sample was output to the log.

▼ Screen shot



MMDAgent - Toolkit for building voice interaction systems
58.9fps 4x MSAA
(1) Build/run was successful if Plugin_Sample appears in the log

```
PLUGIN_EVENT_ENABLE|Audio
PLUGIN_EVENT_ENABLE|Julius
PLUGIN_EVENT_ENABLE|Open_JTalk
Plugin_Sample
PLUGIN_EVENT_ENABLE|Variables
PLUGIN_EVENT_ENABLE|VIManager
PLUGIN_EVENT_ENABLE|WindowController
KEY|d
```

```
0 msec (current motion: +0)
0.00, 13.00, 0.00 | 0.00, 0.00, 0.00 | 112.00 | 16.00
```

[Notes]

The plugin (.dll) is generated in the Release¥Plugins folder, which is at the same level as MMDAgent_vs2010.sln.

Implementable function set

Functions that can be implemented in an MMDAgent plugin are described here. When implementing the functions below, the following EXPORT definitions and inclusion of MMDAgent.h at the beginning of source code is required.

▼ Required code

```
#ifdef _WIN32
#define EXPORT extern "C" __declspec(dllexport)
#else
#define EXPORT extern "C"
#endif

#include "MMDAgent.h"
```

▼ The MMDAgent class

Functions that can be implemented in a plugin are passed a pointer argument that gives them access to the MMDAgent class. This argument allows use of functions published by the MMDAgent class, such as for issuing internal messages.
The syntax and an example of the method for issuing an internal message (sendMessage) are given below. Other published functions are described in Library_MMDAgent¥include¥MMDAgent.h.

・Syntax

```
void sendMessage(const char *type, const char *format, ...);
```

・Description

Issues an internal message.

Arguments

・*type*

Type of internal message.

・*format (variable length argument)*

Format specifier. Same as for printf in the C Standard Library.

Return value

None

・Example

```
mmdagent->sendMessage("Plugin_Sample", "%s", "Arg");
```

extAppStart Function

▼ Description

This function is called once when MMDAgent launches.
It is used to initialize the plugin.

▼ Syntax

```
EXPORT void extAppStart(MMDAgent *mmdagent) {}
```

[Arguments]

・*mmdagent*

Reference value to access MMDAgent functions.

[Return value]

・*void*

None

extAppEnd Function

▼ Description

This function is called once when MMDAgent exits (when the window is closed).
It is used to terminate the plugin.

▼ Syntax

```
EXPORT void extAppEnd(MMDAgent *mmdagent) {}
```

[Arguments]

・*mmdagent*

Reference value to access MMDAgent functions.

[Return value]

・*void*

None

extProcMessage Function

▼ Description

This function is called when an internal message in MMDAgent (EventMessage or CommandMessage) is issued. It is used to include any processing for the issued message.

▼ Syntax

```
EXPORT void extProcMessage(MMDAgent *mmdagent, const char *type, const char *args) {}
```

[Arguments]

・*mmdagent*

Reference value to access MMDAgent functions.

・*type*

The type of the issued internal message.

・*args*

Contents of the issued internal message.

[Return value]

・*void*

None

extUpdate Function

▼ Description

This function is called when MMDAgent performs an update. It is used to perform update processing as time progresses within a scene.

▼ Syntax

```
EXPORT void extUpdate(MMDAgent *mmdagent, double deltaFrame) {}
```

[Arguments]

・*mmdagent*

Reference value to access MMDAgent functions.

・*deltaFrame*

The frame difference elapsed since the previous update process. Units of 1/30 s.

[Return value]

・*void*

None

extRender Function

▼ Description

This function is called whenever MMDAgent performs rendering. It is used to perform processing synchronized with vertical refresh.

▼ Syntax

```
EXPORT void extRender(MMDAgent *mmdagent) {}
```

[Arguments]

・*mmdagent*

Reference value to access MMDAgent functions.

[Return value]

・*void*

None

Simple implementation example

The following simple implementation example is a plugin that outputs MMDAgent messages to a file.

▼Plugin_LogMessage.cpp

```cpp
/* definitions */
#ifdef _WIN32
#define EXPORT extern "C" __declspec(dllexport)
#else
#define EXPORT extern "C"
#endif /* _WIN32 */
#define LOGFILENAME            "MessageLog.txt" /* Log file name */
#define PLUGINLOGMESSAGE_NAME "LogMessage"     /* Plugin name */

/* Message types related to the log file (Any internal message types can be defined) */
#define MMDAGENT_EVENT_FILEOPEN  "LOGMESSAGE_EVENT_FILEOPEN"
#define MMDAGENT_EVENT_FILECLOSE "LOGMESSAGE_EVEMT_FILECLOSE"

/* headers */
#include "MMDAgent.h"
#include <fstream>
#include <ctime>

/* variables */
static bool enable;
static std::ofstream ofs;
static time_t t;
static tm *x;

/* extAppStart: load models and start thread */
EXPORT void extAppStart(MMDAgent *mmdagent)
{
    enable = true;
    mmdagent->sendMessage(MMDAGENT_EVENT_PLUGINENABLE, "%s", PLUGINLOGMESSAGE_NAME);

    /* File name used to create the log */
    const char *fileName = LOGFILENAME;

    /* Open the log file (append) */
    ofs.open(fileName, std::ios::out | std::ios::app);
    if (!ofs) { /* if opening the file fails */
        /* Issue a message indicating file open failure */
        mmdagent->sendMessage(MMDAGENT_EVENT_FILEOPEN, "%s can not be opened!", fileName);
    }
    else { /* File opened successfully */
        /* Issue a message that file opened successfully */
        mmdagent->sendMessage(MMDAGENT_EVENT_FILEOPEN, "%s can be opened", fileName);

        /* Get the current time */
        t = time(0);
        char buf[32];
        ctime_s(buf, sizeof(buf), &t);

        /* Write the time */
        ofs << buf;
        ofs << "[[Start]]" << std::endl;
    }
}

/* extProcMessage: process message */
EXPORT void extProcMessage(MMDAgent *mmdagent, const char *type, const char *args)
{
    if (enable == true) {
        /* Output to output stream (1 line) */
        /* By removing the comments from the following, the log file will only be written*/
        /* for a particular message type (speech input) */
        // if (MMDAgent_strequal(type, "RECOG_EVENT_STOP"))
        {
            ofs << type << "|" << args << std::endl;
        }
    }
}
```

```

/* extAppEnd: stop and free thread */
EXPORT void extAppEnd(MMDAgent *mmdagent)
{
    /* Show the end of the MMDAgent log */
    ofs << "[[End]]" << std::endl;
    ofs << std::endl;

    /* Close the log file when MMDAgent terminates */
    ofs.close();
    mmdagent->sendMessage(MMDAGENT_EVENT_FILECLOSE, "%s was closed", LOGFILENAME);
}

/* execUpdate: run when motion is updated */
EXPORT void extUpdate(MMDAgent *mmdagent, double deltaFrame)
{
}

/* execRender: run when scene is rendered */
EXPORT void extRender(MMDAgent *mmdagent)
{
}

```

▼ The output log file
When the plugin runs correctly, the file MessageLog.txt is output in the current directory (normally the same folder as MMDAgent.exe).

Plugin template

The following is a template containing empty functions that can be used in a plugin.

▼Plugin_Template.cpp

```cpp
#ifdef _WIN32
#define EXPORT extern "C" __declspec(dllexport)
#else
#define EXPORT extern "C"
#endif /* _WIN32 */

/* definitions */

/* headers */
#include "MMDAgent.h"

/* variables */

/* extAppStart: load models and start thread */
EXPORT void extAppStart(MMDAgent *mmdagent) {}

/* extAppEnd: stop and free thread */
EXPORT void extAppEnd(MMDAgent *mmdagent) {}

/* extProcMessage: process message */
EXPORT void extProcMessage(MMDAgent *mmdagent, const char *type, const char *args) {}

/* execUpdate: run when motion is updated */
EXPORT void extUpdate(MMDAgent *mmdagent, double deltaFrame) {}

/* execRender: run when scene is rendered */
EXPORT void extRender(MMDAgent *mmdagent) {}

```

3. Building the development and run-time environments (Android)

Overview

This section summarizes procedures for running the published MMDAgent source code on Android. Note that the development machine OS is windows.

This section is written for the environment given below, but the software can be developed and run on other environments (e.g. Windows 8 or greater, other versions of Android Studio, etc.). In such cases, adjust the descriptions in this document as necessary for your environment.
The version of the JDK required will depend on the version of Android Studio being used, so check the system requirements on the “Android Studio and SDK Tools Downloads” page on the Android developer site (https://developer.android.com).

Software	Version
Dev. Environment OS	Windows 7 64 bit
Run-time environment OS	Android 5.0.2（4.0 or greater recommended）
Development software	Android Studio 1.4
JDK Version	Java SE Development Kit 7u80
Android SDK version	r24.3.4
Android NDK version	r10e

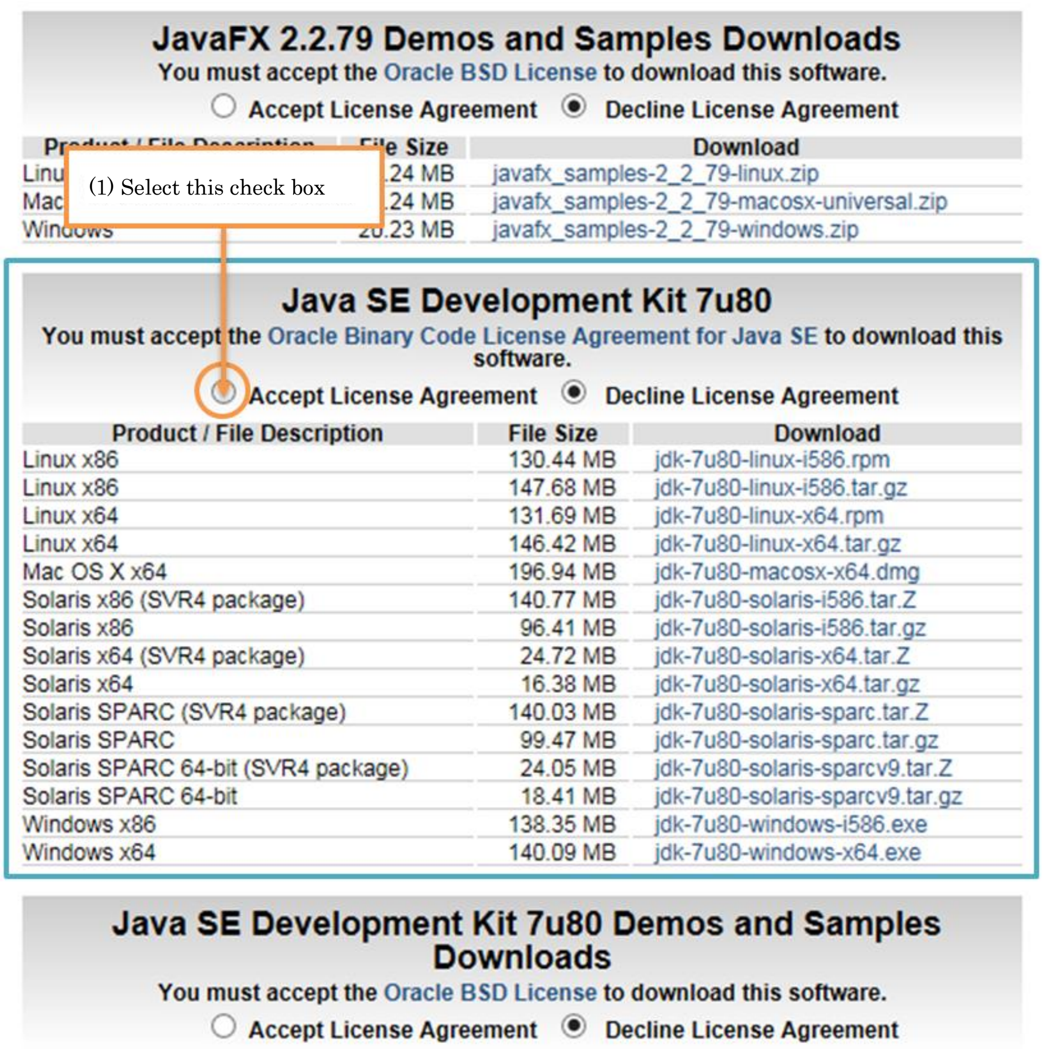
[Notes]
It will be necessary to enable developer options on the Android terminal used as the run-time environment.

Building the development environment

Downloading Java SE Development Kit 7

Download the Java SE Development Kit 7 installer from the Web site.
http://www.oracle.com/technetwork/jp/java/javase/downloads/jdk7-downloads-1880260.html

▼ Procedure



JavaFX 2.2.79 Demos and Samples Downloads

You must accept the Oracle BSD License to download this software.

Accept License Agreement [✓ Decline License Agreement]

Product / File Description	File Size	Download
Linu	.24 MB	javafx_samples-2_2_79-linux.zip
Mac	.24 MB	javafx_samples-2_2_79-macosx-universal.zip
Windows	20.23 MB	javafx_samples-2_2_79-windows.zip

Java SE Development Kit 7u80

You must accept the Oracle Binary Code License Agreement for Java SE to download this software.

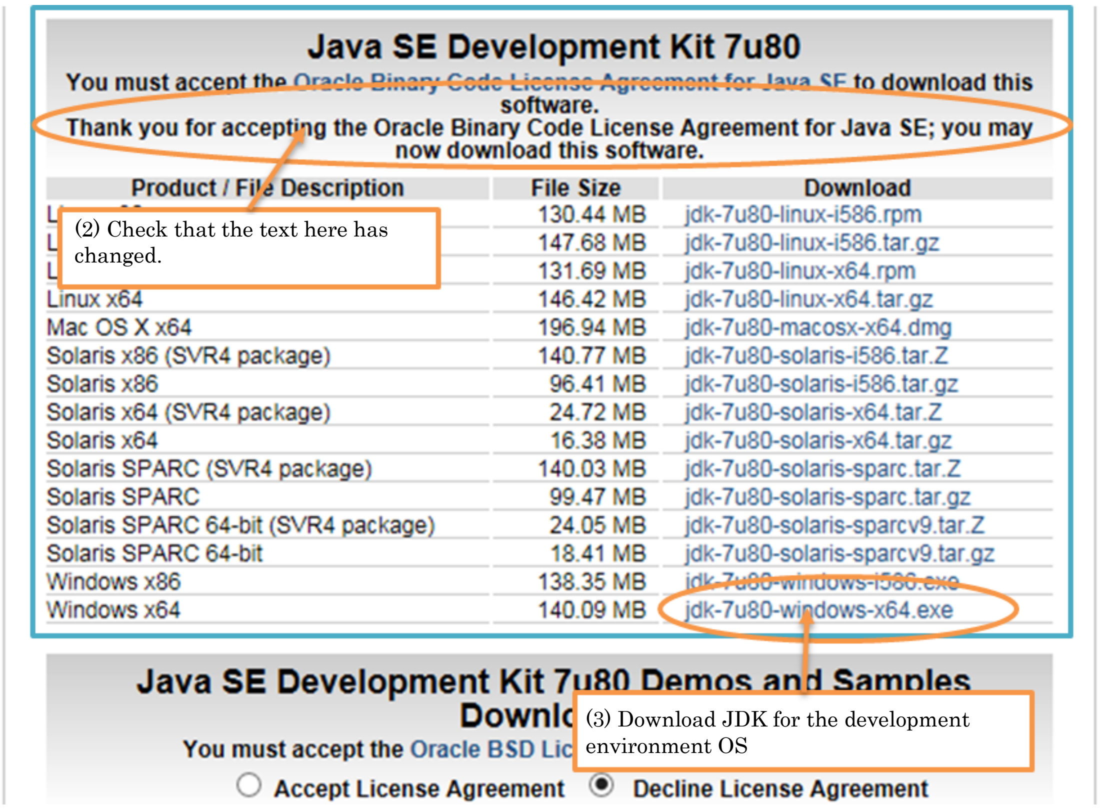
Accept License Agreement [✓ Decline License Agreement]

Product / File Description	File Size	Download
Linux x86	130.44 MB	jdk-7u80-linux-i586.rpm
Linux x86	147.68 MB	jdk-7u80-linux-i586.tar.gz
Linux x64	131.69 MB	jdk-7u80-linux-x64.rpm
Linux x64	146.42 MB	jdk-7u80-linux-x64.tar.gz
Mac OS X x64	196.94 MB	jdk-7u80-macosx-x64.dmg
Solaris x86 (SVR4 package)	140.77 MB	jdk-7u80-solaris-i586.tar.Z
Solaris x86	96.41 MB	jdk-7u80-solaris-i586.tar.gz
Solaris x64 (SVR4 package)	24.72 MB	jdk-7u80-solaris-x64.tar.Z
Solaris x64	16.38 MB	jdk-7u80-solaris-x64.tar.gz
Solaris SPARC (SVR4 package)	140.03 MB	jdk-7u80-solaris-sparc.tar.Z
Solaris SPARC	99.47 MB	jdk-7u80-solaris-sparc.tar.gz
Solaris SPARC 64-bit (SVR4 package)	24.05 MB	jdk-7u80-solaris-sparcv9.tar.Z
Solaris SPARC 64-bit	18.41 MB	jdk-7u80-solaris-sparcv9.tar.gz
Windows x86	138.35 MB	jdk-7u80-windows-i586.exe
Windows x64	140.09 MB	jdk-7u80-windows-x64.exe

Java SE Development Kit 7u80 Demos and Samples Downloads

You must accept the Oracle BSD License to download this software.

Accept License Agreement [✓ Decline License Agreement]



Java SE Development Kit 7u80

You must accept the Oracle Binary Code License Agreement for Java SE to download this software.

Thank you for accepting the Oracle Binary Code License Agreement for Java SE; you may now download this software.

Product / File Description	File Size	Download
L	130.44 MB	jdk-7u80-linux-i586.rpm
L	147.68 MB	jdk-7u80-linux-i586.tar.gz
L	131.69 MB	jdk-7u80-linux-x64.rpm
Linux x64	146.42 MB	jdk-7u80-linux-x64.tar.gz
Mac OS X x64	196.94 MB	jdk-7u80-macosx-x64.dmg
Solaris x86 (SVR4 package)	140.77 MB	jdk-7u80-solaris-i586.tar.Z
Solaris x86	96.41 MB	jdk-7u80-solaris-i586.tar.gz
Solaris x64 (SVR4 package)	24.72 MB	jdk-7u80-solaris-x64.tar.Z
Solaris x64	16.38 MB	jdk-7u80-solaris-x64.tar.gz
Solaris SPARC (SVR4 package)	140.03 MB	jdk-7u80-solaris-sparc.tar.Z
Solaris SPARC	99.47 MB	jdk-7u80-solaris-sparc.tar.gz
Solaris SPARC 64-bit (SVR4 package)	24.05 MB	jdk-7u80-solaris-sparcv9.tar.Z
Solaris SPARC 64-bit	18.41 MB	jdk-7u80-solaris-sparcv9.tar.gz
Windows x86	138.35 MB	jdk-7u80-windows-i586.exe
Windows x64	140.09 MB	jdk-7u80-windows-x64.exe

(2) Check that the text here has changed.

Java SE Development Kit 7u80 Demos and Samples Downl

(3) Download JDK for the development environment OS

You must accept the Oracle BSD Lic

○ Accept License Agreement ● Decline License Agreement

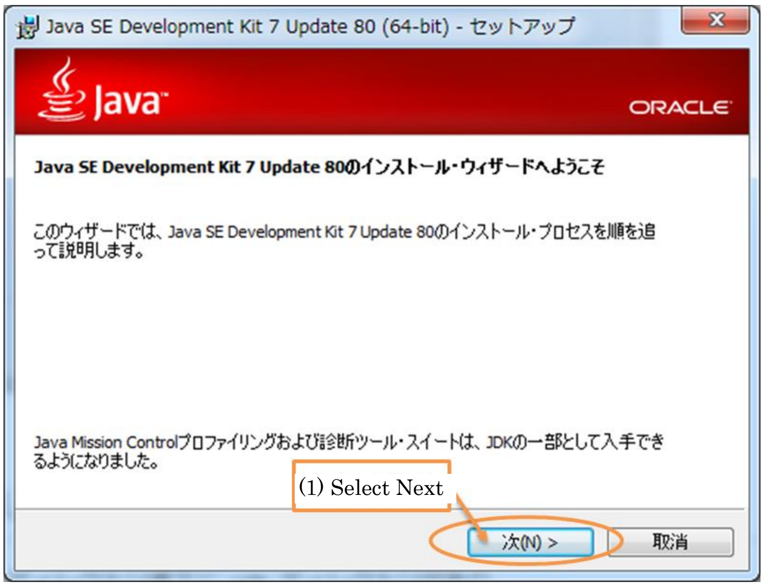
[Notes]

JDK is the Java SE Development Kit, and JRE is the Java Runtime Environment.

Installing the Java SE Development Kit 7

Install the Java SE Development Kit 7 using the downloaded installer. During the installation, the JRE install screen will appear, so follow the instructions to install the JRE as well.

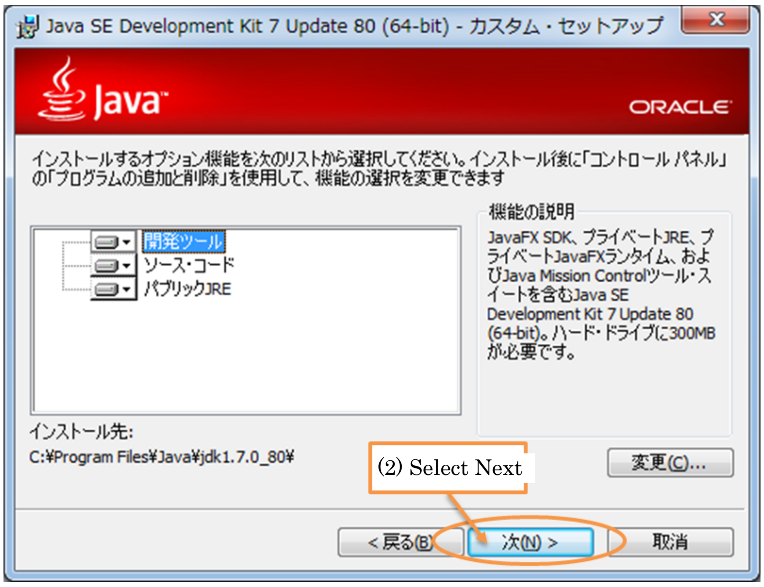
▼ Procedure (JDK install)


Java SE Development Kit 7 Update 80 (64-bit) - セットアップ

Java SE Development Kit 7 Update 80のインストール・ウィザードへようこそ

このウィザードでは、Java SE Development Kit 7 Update 80のインストール・プロセスを順を追って説明します。

Java Mission Controlプロファイリングおよび診断ツール・スイートは、JDKの一部として入手できるようになりました。


Java SE Development Kit 7 Update 80 (64-bit) - カスタム・セットアップ

インストールするオプション機能を次のリストから選択してください。インストール後に「コントロール パネル」の「プログラムの追加と削除」を使用して、機能の選択を変更できます

開発ツール
ソース・コード
パブリックJRE

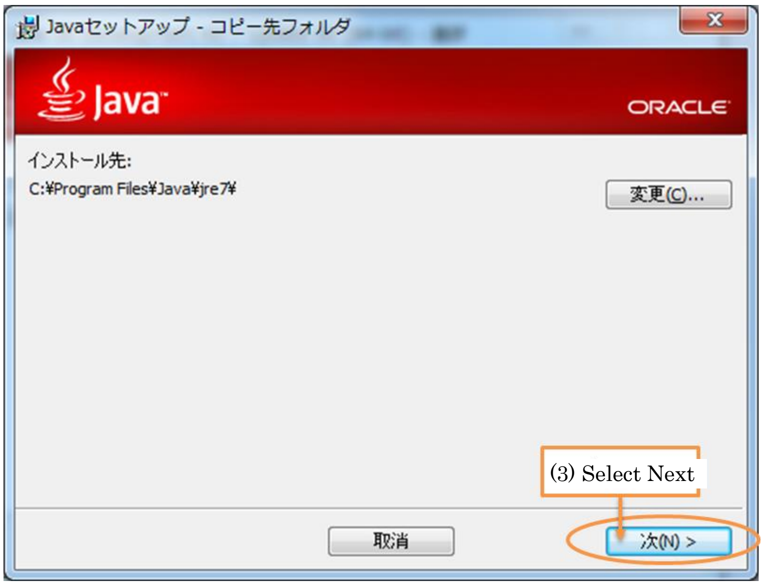
機能の説明

JavaFX SDK、プライベートJRE、プライベートJavaFXランタイム、およびJava Mission Controlツール・スイートを含むJava SE Development Kit 7 Update 80 (64-bit)。ハード・ドライブに300MBが必要です。

インストール先:
C:¥Program Files¥Java¥jdk1.7.0_80¥

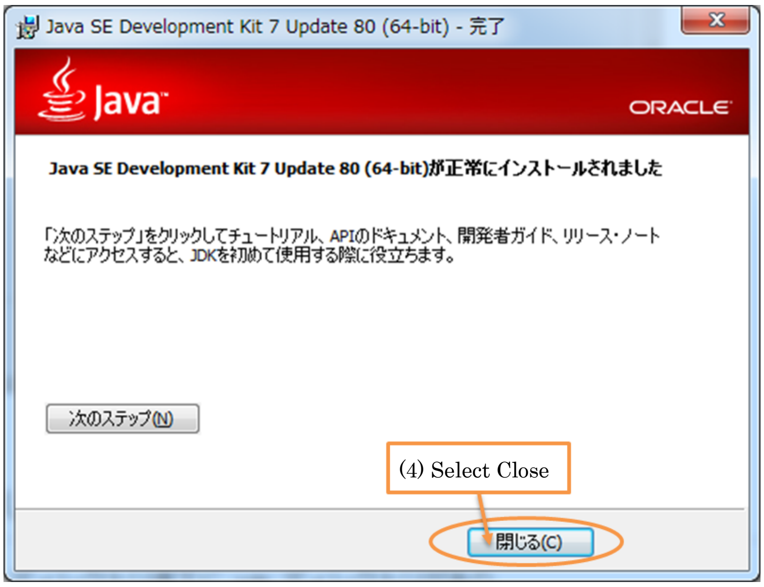
* The install path can be changed, but this will affect settings later on, so it is not recommended.

▼ Procedure (JRE install, window appears automatically)



Javaセットアップ - コピー先フォルダ

インストール先:
C:¥Program Files¥Java¥jre7¥



Java SE Development Kit 7 Update 80 (64-bit) - 完了

Java SE Development Kit 7 Update 80 (64-bit)が正常にインストールされました

「次のステップ」をクリックしてチュートリアル、APIのドキュメント、開発者ガイド、リリース・ノートなどにアクセスすると、JDKを初めて使用する際に役立ちます。

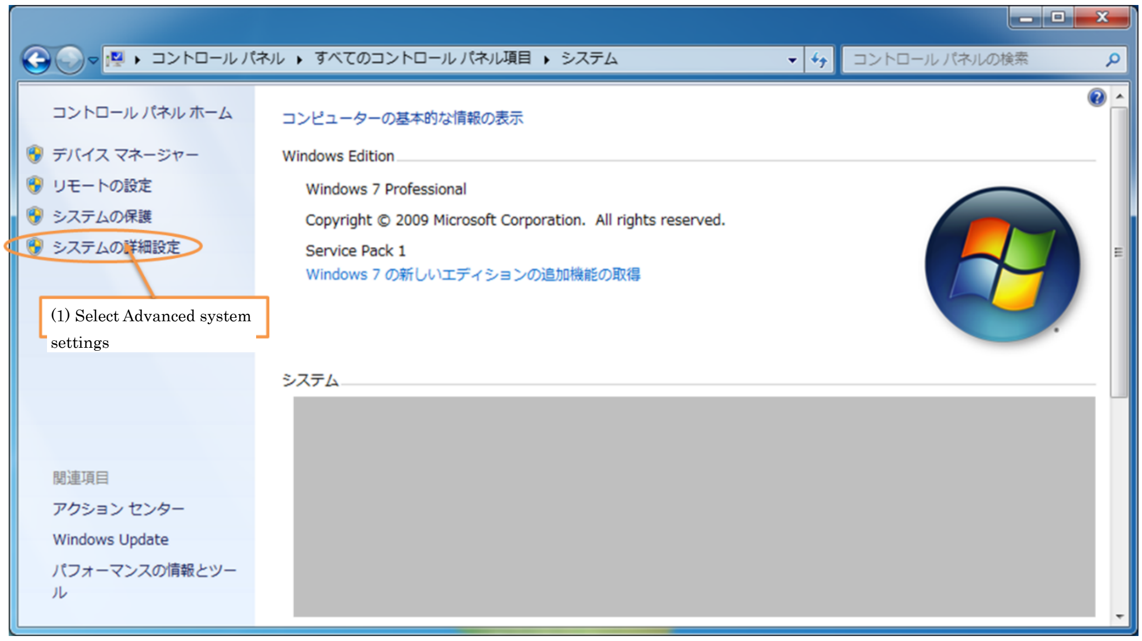
Configuring environment variables (JAVA_HOME)

Register the path of the installed JDK in an environment variable.
The environment variable to be registered is as follows.

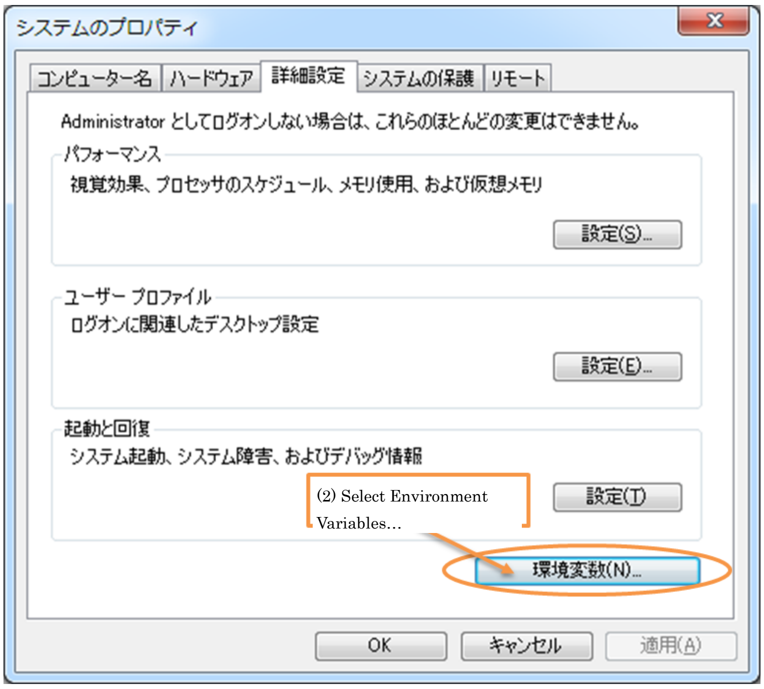
▼ Environment variable

Item	Value
Variable name	JAVA_HOME
Variable value	C:¥Program Files¥Java¥jdk1.7.0_80

▼ Procedure


コントロール パネル ▸ すべてのコントロール パネル項目 ▸ システム
コントロール パネルの検索
コントロール パネル ホーム
デバイス マネージャー
リモートの設定
システムの保護
システムの詳細設定
(1) Select Advanced system settings
関連項目
アクション センター
Windows Update
パフォーマンスの情報とツール
コンピューターの基本的な情報の表示
Windows Edition
Windows 7 Professional
Copyright © 2009 Microsoft Corporation. All rights reserved.
Service Pack 1
Windows 7 の新しいエディションの追加機能の取得
システム

* The window above can be displayed with [Control panel] -> [System].



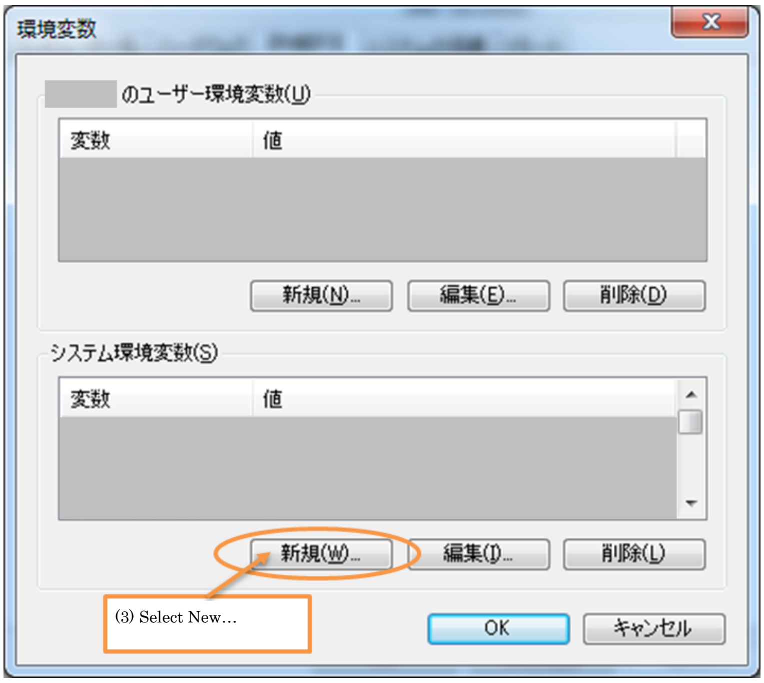
Administrator としてログオンしない場合は、これらのほとんどの変更はできません。

視覚効果、プロセッサのスケジュール、メモリ使用、および仮想メモリ

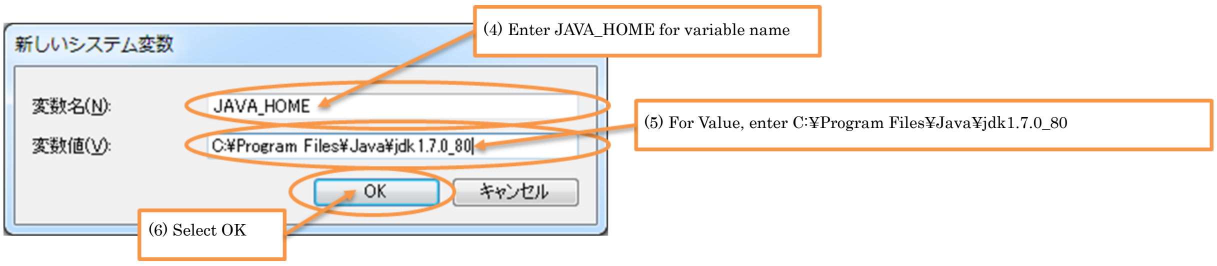
ログオンに関連したデスクトップ設定

システム起動、システム障害、およびデバッグ情報

(2) Select Environment Variables…



(3) Select New…



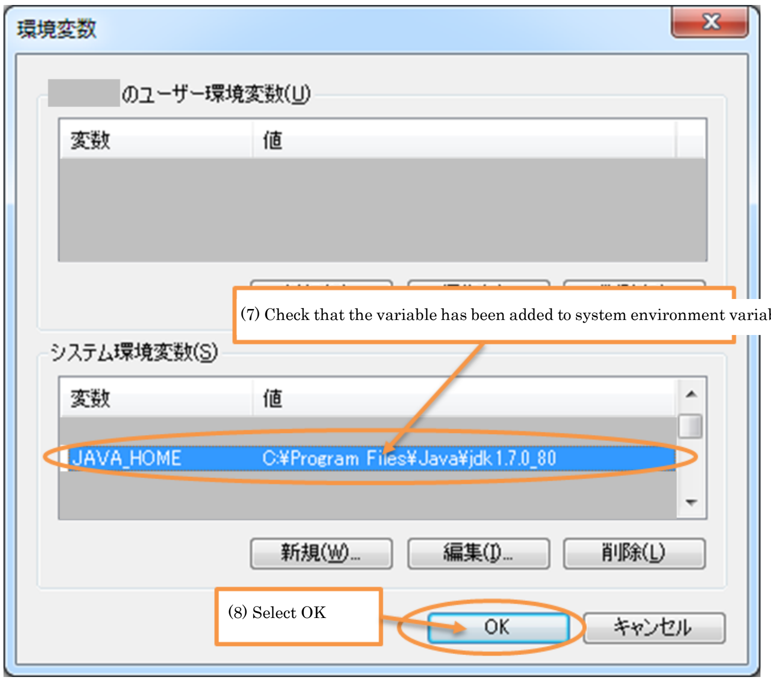
変数名(N): JAVA_HOME

変数値(V): C:¥Program Files¥Java¥jdk1.7.0_80

(4) Enter JAVA_HOME for variable name

(5) For Value, enter C:¥Program Files¥Java¥jdk1.7.0_80

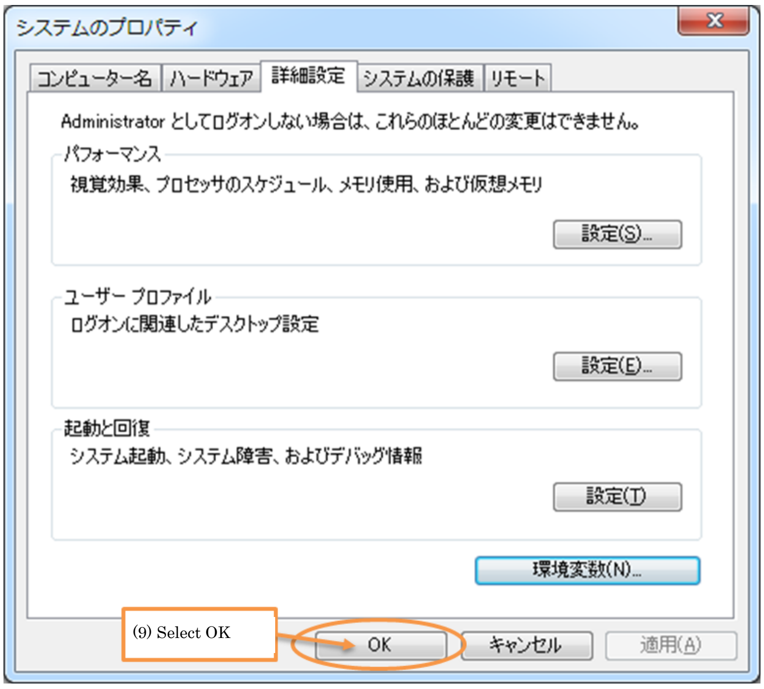
(6) Select OK



(7) Check that the variable has been added to system environment variables

変数	値
JAVA_HOME	C:¥Program Files¥Java¥jdk1.7.0_80

(8) Select OK



Administrator としてログオンしない場合は、これらのほとんどの変更はできません。

視覚効果、プロセッサのスケジュール、メモリ使用、および仮想メモリ

ログオンに関連したデスクトップ設定

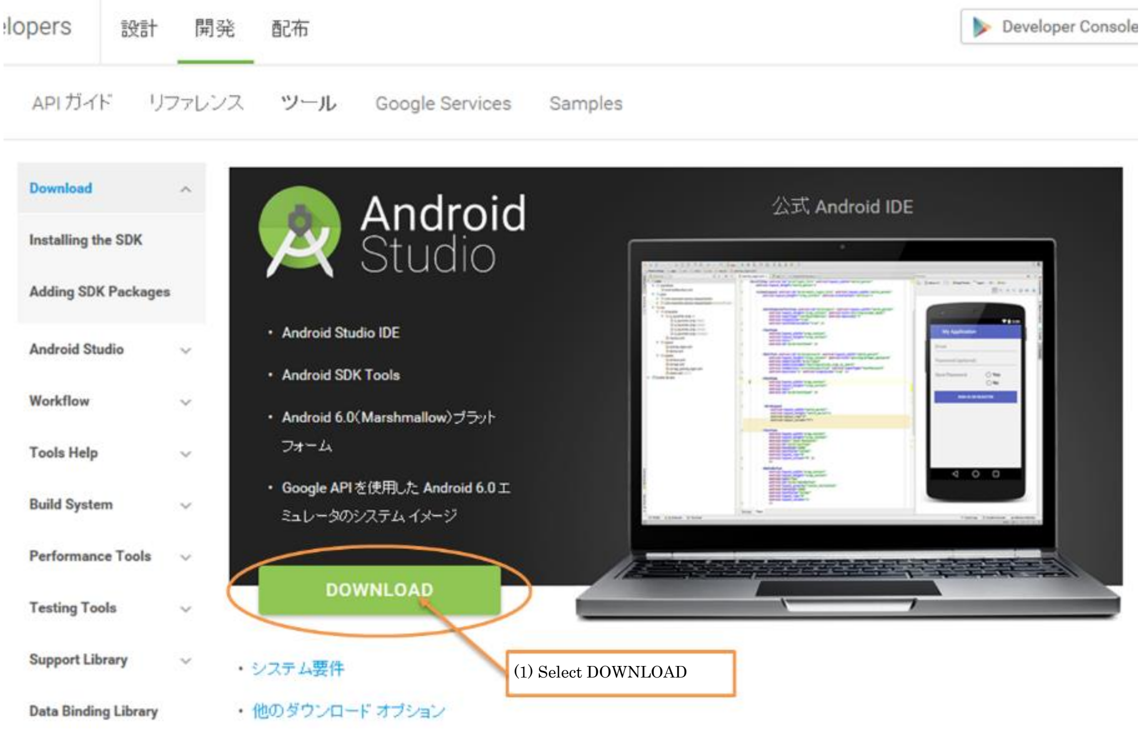
システム起動、システム障害、およびデバッグ情報

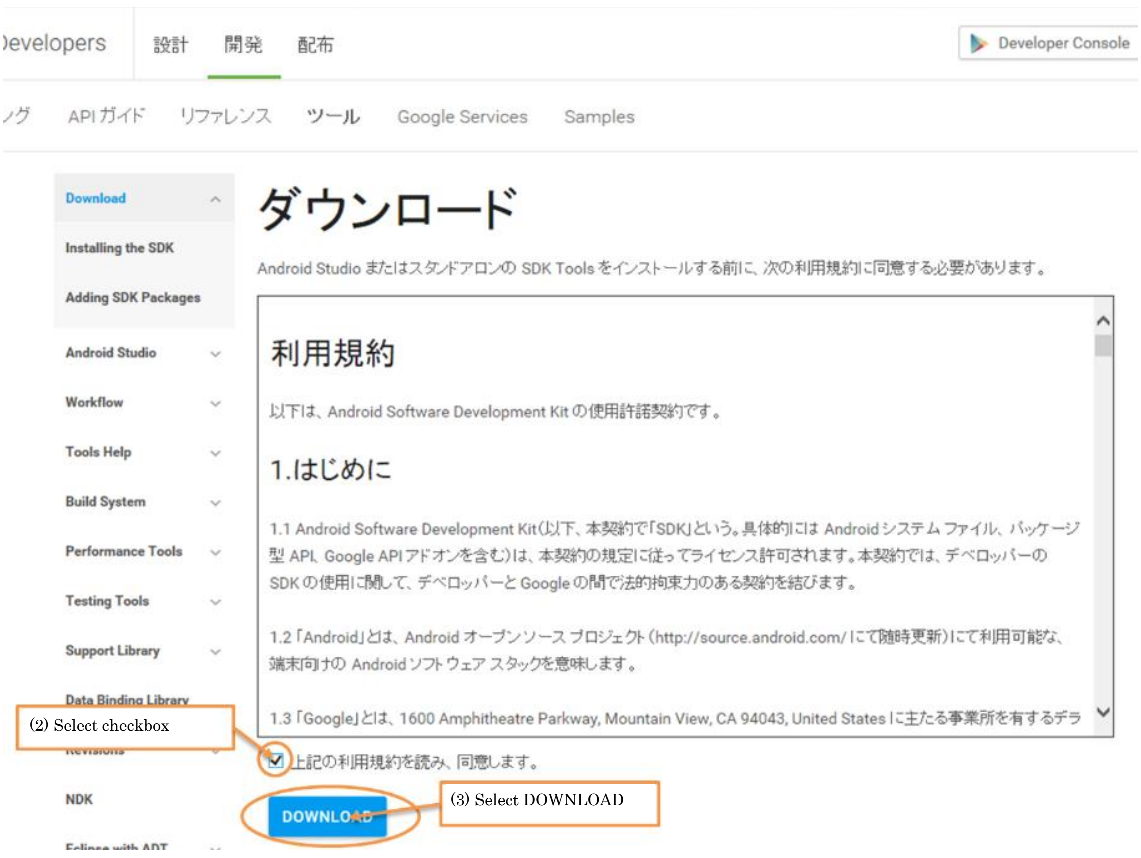
(9) Select OK

Downloading Android Studio

Download Android Studio from the Web site.
http://developer.android.com/intl/ja/sdk/index.html

▼ Procedure





ダウンロード

Android Studio またはスタンドアロンの SDK Tools をインストールする前に、次の利用規約に同意する必要があります。

利用規約

以下は、Android Software Development Kit の使用許諾契約です。

1.はじめに

1.1 Android Software Development Kit(以下、本契約で「SDK」という。具体的には Android システム ファイル、パッケージ型 API、Google API アドオンを含む)は、本契約の規定に従ってライセンス許可されます。本契約では、デベロッパーの SDK の使用に関して、デベロッパーと Google の間で法的拘束力のある契約を結びます。

1.2 「Android」とは、Android オープンソース プロジェクト(http://source.android.com/ にて随時更新)にて利用可能な、端末向けの Android ソフトウェア スタックを意味します。

1.3 「Google」とは、1600 Amphitheatre Parkway, Mountain View, CA 94043, United States に主たる事業所を有するデラ

☑ 上記の利用規約を読み、同意します。

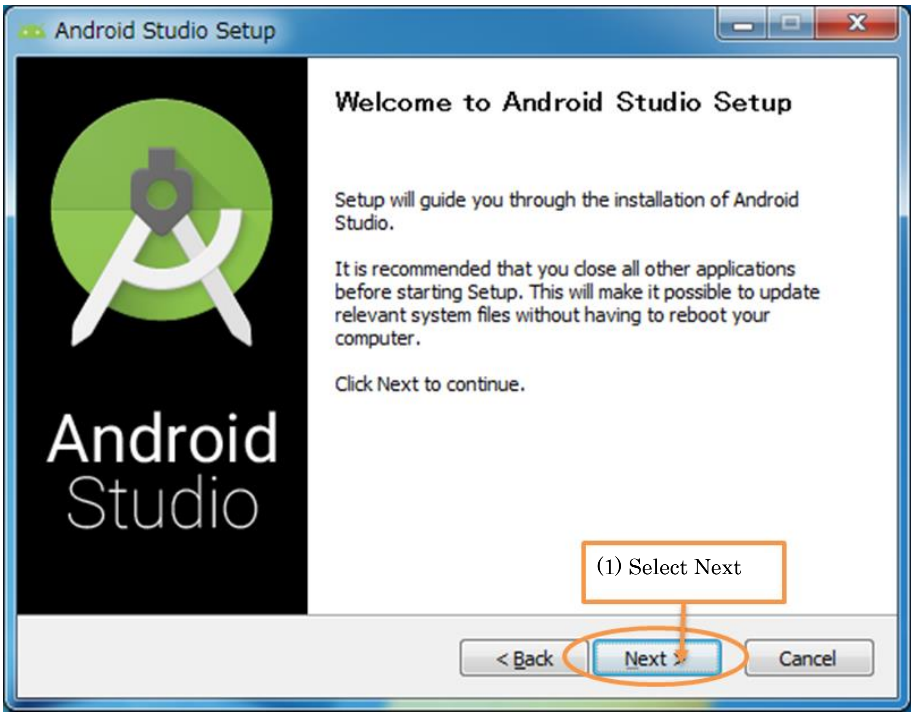
[Notes]
The version of Android Studio on the Web site is always the latest version, but past versions can be obtained at the following site.

http://tools.android.com/download/studio/canary

Installing Android Studio

Install Android Studio with the downloaded installer.

▼ Procedure



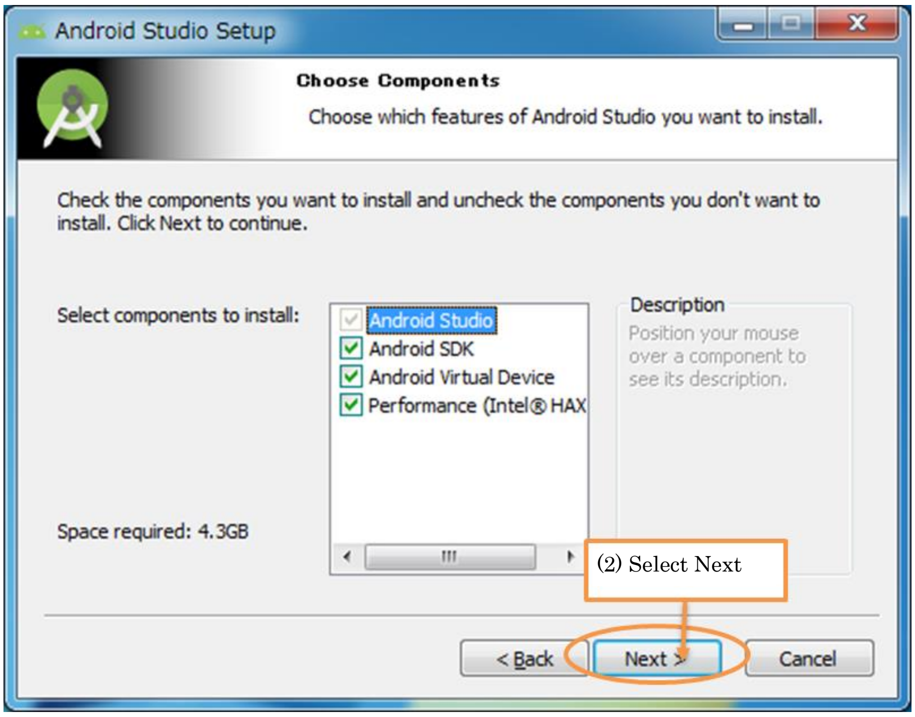
Android Studio Setup

Welcome to Android Studio Setup

Setup will guide you through the installation of Android Studio.

It is recommended that you close all other applications before starting Setup. This will make it possible to update relevant system files without having to reboot your computer.

Click Next to continue.



Android Studio Setup

Choose Components

Choose which features of Android Studio you want to install.

Check the components you want to install and uncheck the components you don't want to install. Click Next to continue.

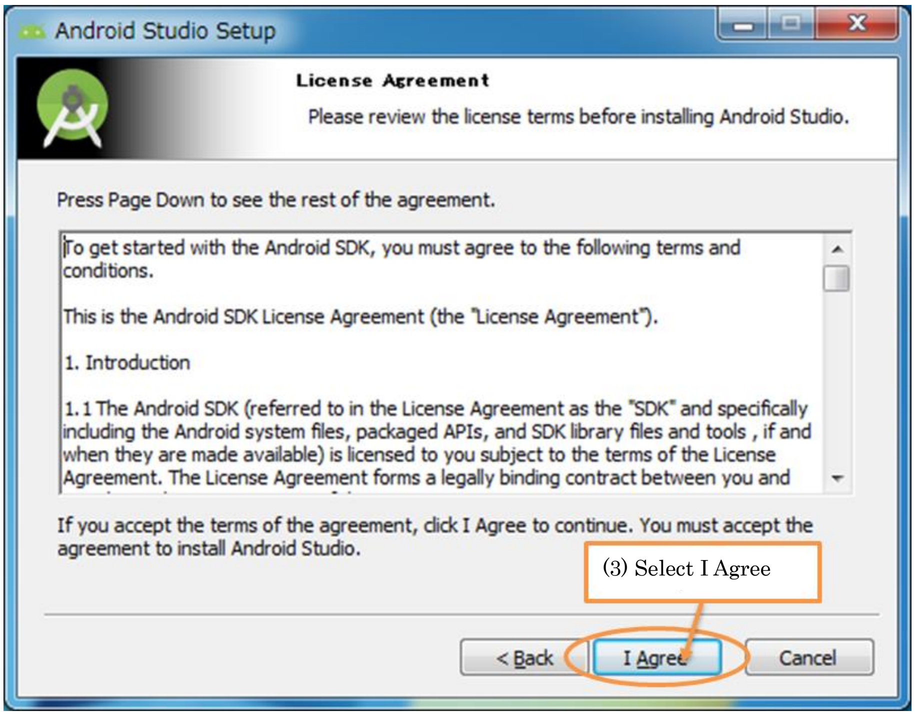
Select components to install:

- [✓] Android Studio
- [✓] Android SDK
- [✓] Android Virtual Device
- [✓] Performance (Intel® HAX

Description

Position your mouse over a component to see its description.

Space required: 4.3GB



Android Studio Setup
License Agreement
Please review the license terms before installing Android Studio.

Press Page Down to see the rest of the agreement.

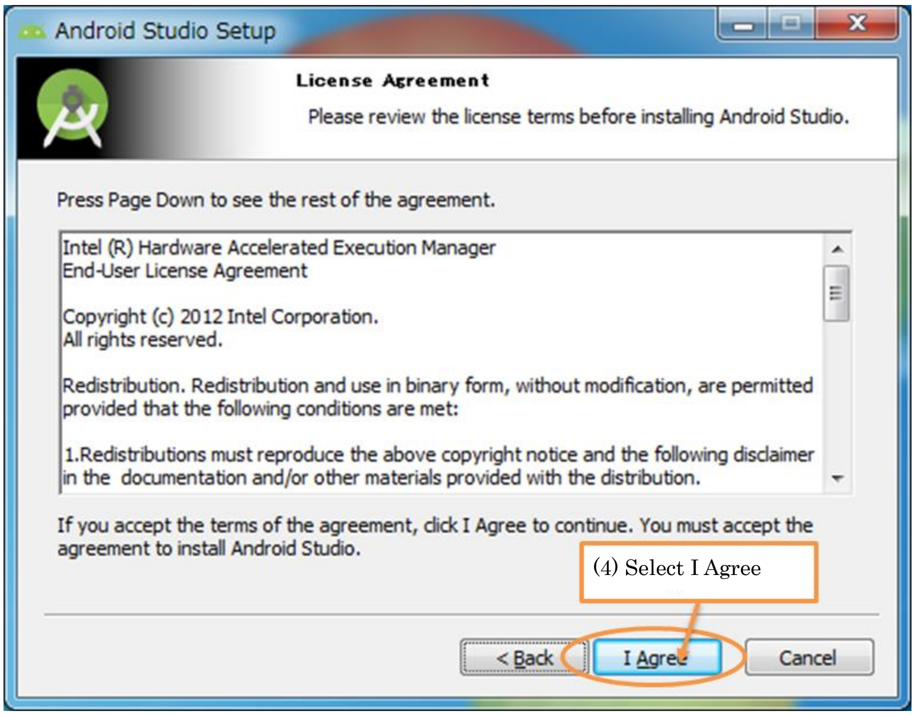
To get started with the Android SDK, you must agree to the following terms and conditions.

This is the Android SDK License Agreement (the "License Agreement").

1. Introduction

1.1 The Android SDK (referred to in the License Agreement as the "SDK" and specifically including the Android system files, packaged APIs, and SDK library files and tools , if and when they are made available) is licensed to you subject to the terms of the License Agreement. The License Agreement forms a legally binding contract between you and

If you accept the terms of the agreement, click I Agree to continue. You must accept the agreement to install Android Studio.



Android Studio Setup
License Agreement
Please review the license terms before installing Android Studio.

Press Page Down to see the rest of the agreement.

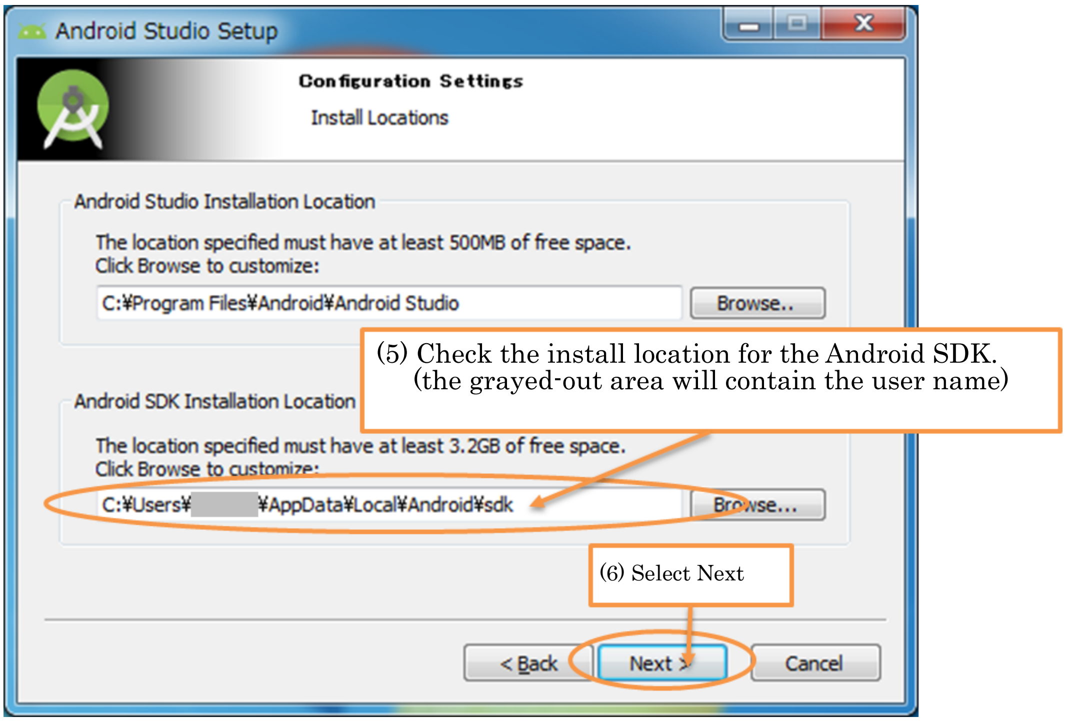
Intel (R) Hardware Accelerated Execution Manager
End-User License Agreement

Copyright (c) 2012 Intel Corporation.
All rights reserved.

Redistribution. Redistribution and use in binary form, without modification, are permitted provided that the following conditions are met:

1.Redistributions must reproduce the above copyright notice and the following disclaimer in the documentation and/or other materials provided with the distribution.

If you accept the terms of the agreement, click I Agree to continue. You must accept the agreement to install Android Studio.



Configuration Settings
Install Locations

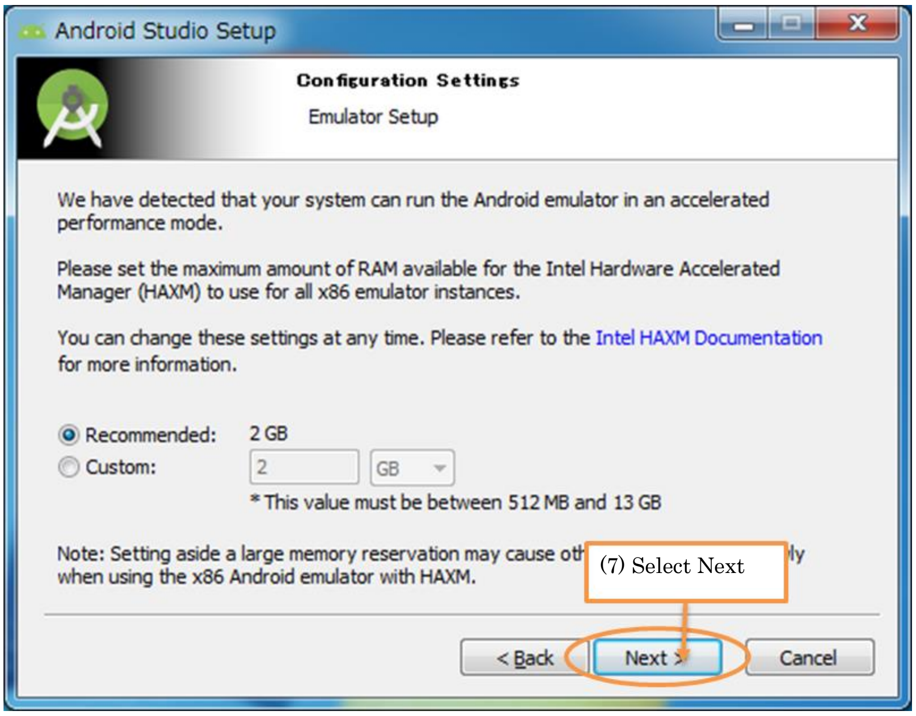
Android Studio Installation Location
The location specified must have at least 500MB of free space.
Click Browse to customize:
C:¥Program Files¥Android¥Android Studio

Android SDK Installation Location
The location specified must have at least 3.2GB of free space.
Click Browse to customize:
C:¥Users¥　　　¥AppData¥Local¥Android¥sdk

(5) Check the install location for the Android SDK.
(the grayed-out area will contain the user name)

(6) Select Next

* Any path can be used for the installation, but changing it is not recommended because it will affect other settings described below.



Configuration Settings
Emulator Setup

We have detected that your system can run the Android emulator in an accelerated performance mode.

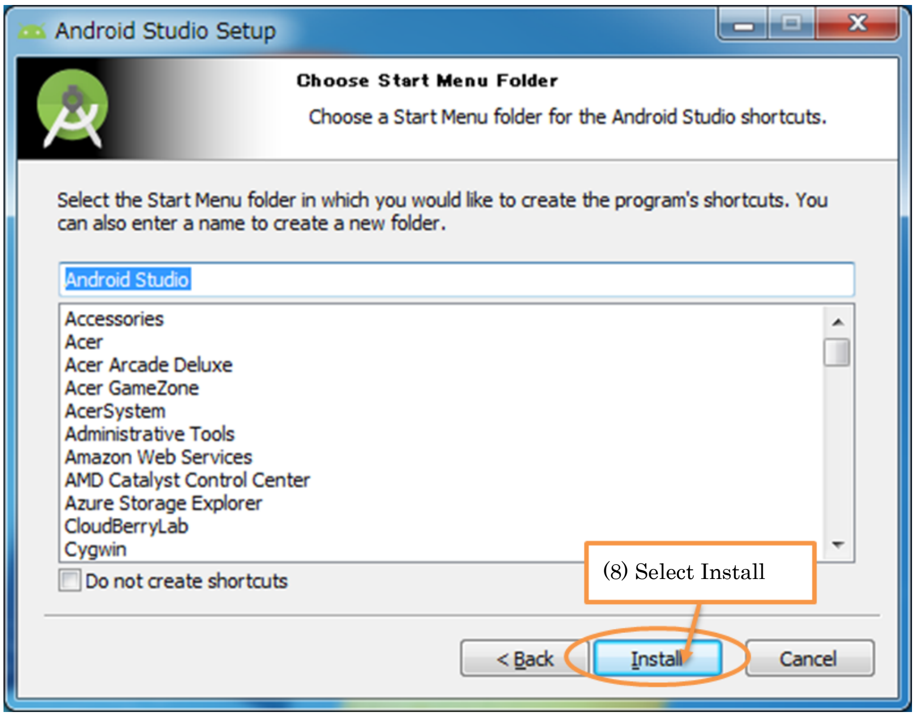
Please set the maximum amount of RAM available for the Intel Hardware Accelerated Manager (HAXM) to use for all x86 emulator instances.

You can change these settings at any time. Please refer to the Intel HAXM Documentation for more information.

● Recommended: 2 GB
○ Custom: 2 GB
* This value must be between 512 MB and 13 GB

Note: Setting aside a large memory reservation may cause oth…ly when using the x86 Android emulator with HAXM.

(7) Select Next



Select the Start Menu folder in which you would like to create the program's shortcuts. You can also enter a name to create a new folder.

Android Studio

Accessories
Acer
Acer Arcade Deluxe
Acer GameZone
AcerSystem
Administrative Tools
Amazon Web Services
AMD Catalyst Control Center
Azure Storage Explorer
CloudBerryLab
Cygwin

☐ Do not create shortcuts

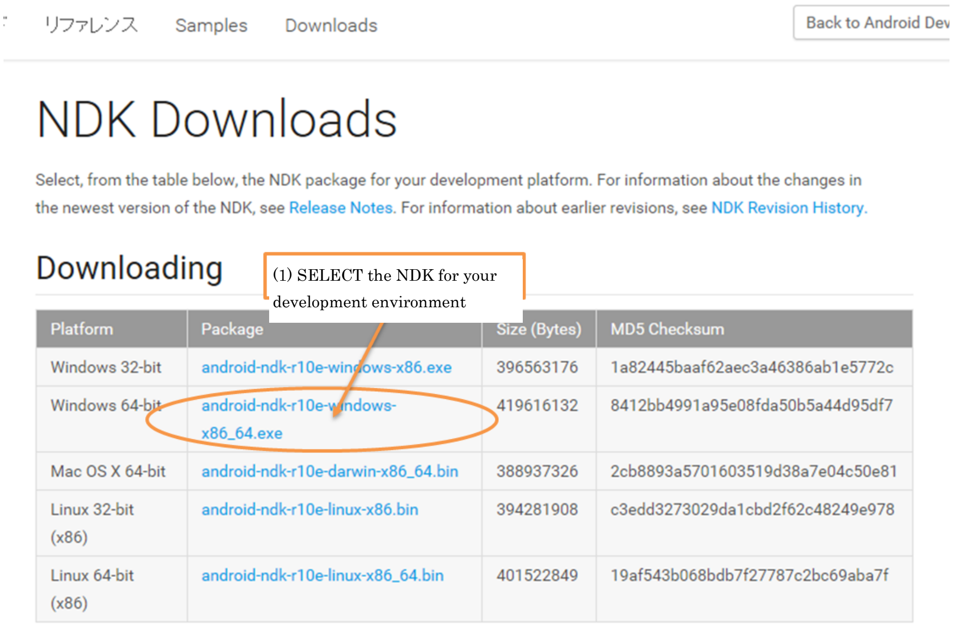
[Notes]
The Android SDK is also required for Android development, but if the full version of Android Studio was used, the Android SDK is included and does not need to be installed separately.

Downloading Android NDK

Download the Android NDK from the Web site.

http://developer.android.com/intl/ja/ndk/downloads/index.html

▼ Procedure

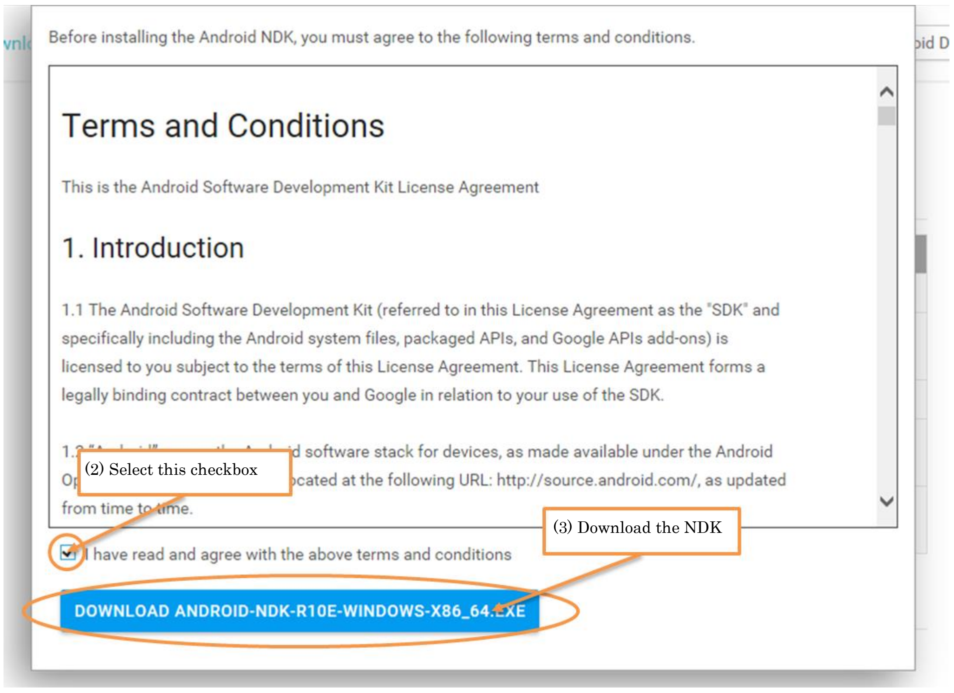


NDK Downloads

Select, from the table below, the NDK package for your development platform. For information about the changes in the newest version of the NDK, see Release Notes. For information about earlier revisions, see NDK Revision History.

Downloading

Platform	Package	Size (Bytes)	MD5 Checksum
Windows 32-bit	android-ndk-r10e-windows-x86.exe	396563176	1a82445baaf62aec3a46386ab1e5772c
Windows 64-bit	android-ndk-r10e-windows-x86_64.exe	419616132	8412bb4991a95e08fda50b5a44d95df7
Mac OS X 64-bit	android-ndk-r10e-darwin-x86_64.bin	388937326	2cb8893a5701603519d38a7e04c50e81
Linux 32-bit (x86)	android-ndk-r10e-linux-x86.bin	394281908	c3edd3273029da1cbd2f62c48249e978
Linux 64-bit (x86)	android-ndk-r10e-linux-x86_64.bin	401522849	19af543b068bdb7f27787c2bc69aba7f



Before installing the Android NDK, you must agree to the following terms and conditions.

Terms and Conditions

This is the Android Software Development Kit License Agreement

1. Introduction

1.1 The Android Software Development Kit (referred to in this License Agreement as the "SDK" and specifically including the Android system files, packaged APIs, and Google APIs add-ons) is licensed to you subject to the terms of this License Agreement. This License Agreement forms a legally binding contract between you and Google in relation to your use of the SDK.

1.2 "Android" ... the Android software stack for devices, as made available under the Android Op... located at the following URL: http://source.android.com/, as updated from time to time.

☑ I have read and agree with the above terms and conditions

Installing Android NDK

Run the downloaded installer to install the Android NDK.

▼ Procedure

1. Run the installer

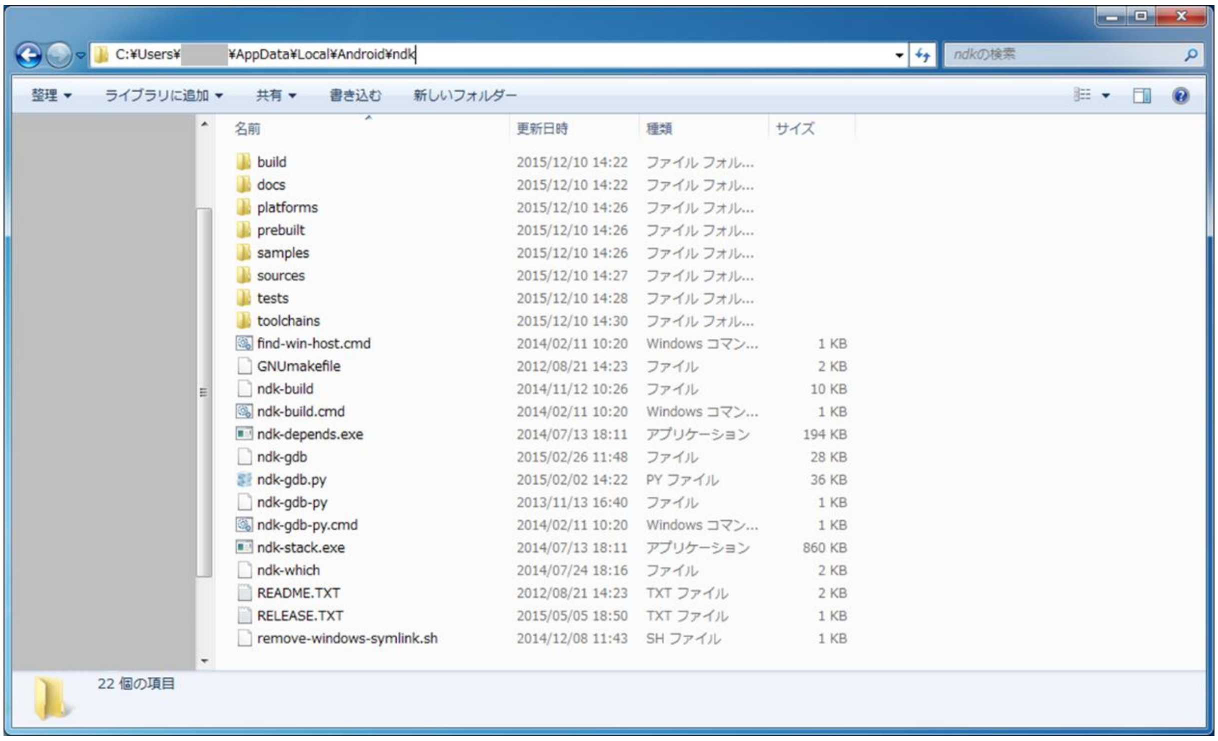
   When the installer is run, compressed folders are extracted automatically, and a folder with the same name as the installer is created in the current folder. This installer is self-extracting, so no user operation is necessary.

2. Move the folder

   Change the name of the extracted folder to ndk and move it to the following folder.

   [Folder destination]
   C:¥Users¥<user name>¥AppData¥Local¥Android
   * Any path can be used for the destination, but changing it is not recommended because it will affect other settings described below.

   [Screen shot]



C:¥Users¥[redacted]¥AppData¥Local¥Android¥ndk

名前	更新日時	種類	サイズ
build	2015/12/10 14:22	ファイル フォル...	
docs	2015/12/10 14:22	ファイル フォル...	
platforms	2015/12/10 14:26	ファイル フォル...	
prebuilt	2015/12/10 14:26	ファイル フォル...	
samples	2015/12/10 14:26	ファイル フォル...	
sources	2015/12/10 14:27	ファイル フォル...	
tests	2015/12/10 14:28	ファイル フォル...	
toolchains	2015/12/10 14:30	ファイル フォル...	
find-win-host.cmd	2014/02/11 10:20	Windows コマン...	1 KB
GNUmakefile	2012/08/21 14:23	ファイル	2 KB
ndk-build	2014/11/12 10:26	ファイル	10 KB
ndk-build.cmd	2014/02/11 10:20	Windows コマン...	1 KB
ndk-depends.exe	2014/07/13 18:11	アプリケーション	194 KB
ndk-gdb	2015/02/26 11:48	ファイル	28 KB
ndk-gdb.py	2015/02/02 14:22	PY ファイル	36 KB
ndk-gdb-py	2013/11/13 16:40	ファイル	1 KB
ndk-gdb-py.cmd	2014/02/11 10:20	Windows コマン...	1 KB
ndk-stack.exe	2014/07/13 18:11	アプリケーション	860 KB
ndk-which	2014/07/24 18:16	ファイル	2 KB
README.TXT	2012/08/21 14:23	TXT ファイル	2 KB
RELEASE.TXT	2015/05/05 18:50	TXT ファイル	1 KB
remove-windows-symlink.sh	2014/12/08 11:43	SH ファイル	1 KB

22 個の項目

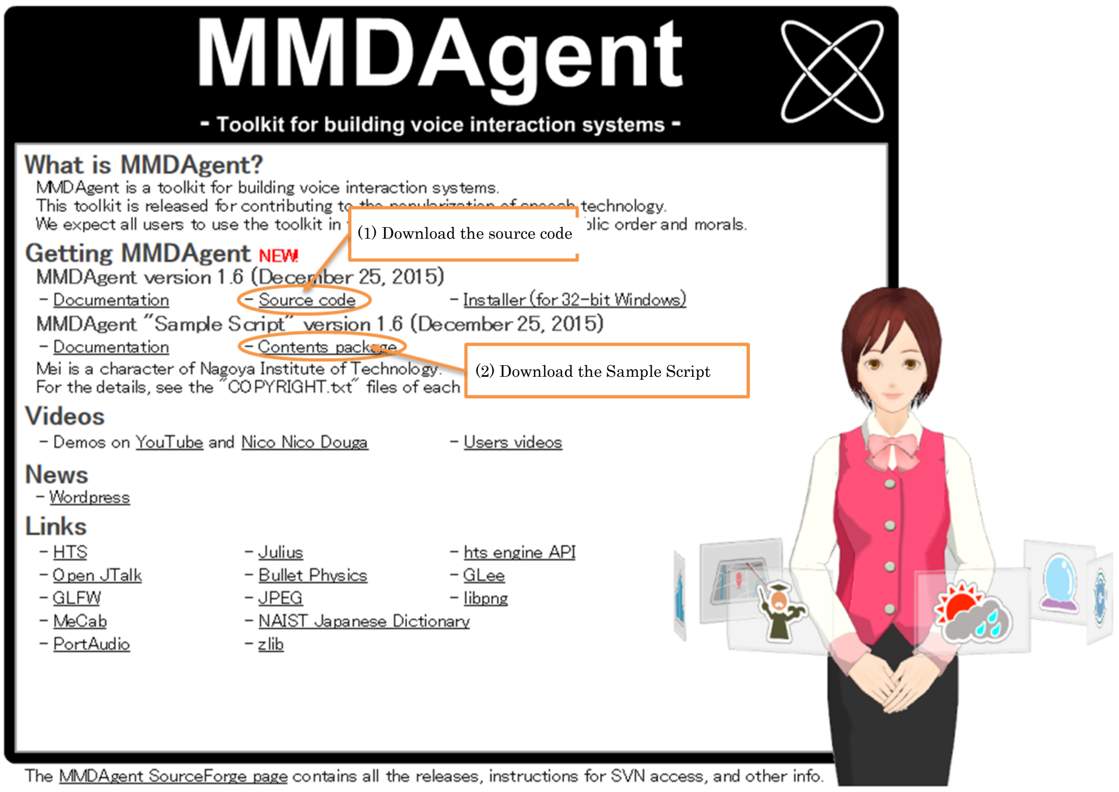
Building the run-time environment

Getting the source code and sample script

Download the MMDAgent source code and Sample Script from the Web site and save it in a suitable location on the PC. The download is a compressed Zip file that must be extracted before performing the following procedures.

▼MMDAgent Web site
http://www.mmdagent.jp/

▼ Procedure



MMDAgent

- Toolkit for building voice interaction systems -

What is MMDAgent?

MMDAgent is a toolkit for building voice interaction systems.
This toolkit is released for contributing to the popularization of speech technology.
We expect all users to use the toolkit in ... blic order and morals.

Getting MMDAgent NEW!

MMDAgent version 1.6 (December 25, 2015)
- Documentation - Source code - Installer (for 32-bit Windows)
MMDAgent "Sample Script" version 1.6 (December 25, 2015)
- Documentation - Contents package
Mei is a character of Nagoya Institute of Technology.
For the details, see the "COPYRIGHT.txt" files of each

Videos

- Demos on YouTube and Nico Nico Douga - Users videos

News

- Wordpress

Links

- HTS - Julius - hts engine API
- Open JTalk - Bullet Physics - GLee
- GLFW - JPEG - libpng
- MeCab - NAIST Japanese Dictionary
- PortAudio - zlib

The MMDAgent SourceForge page contains all the releases, instructions for SVN access, and other info.

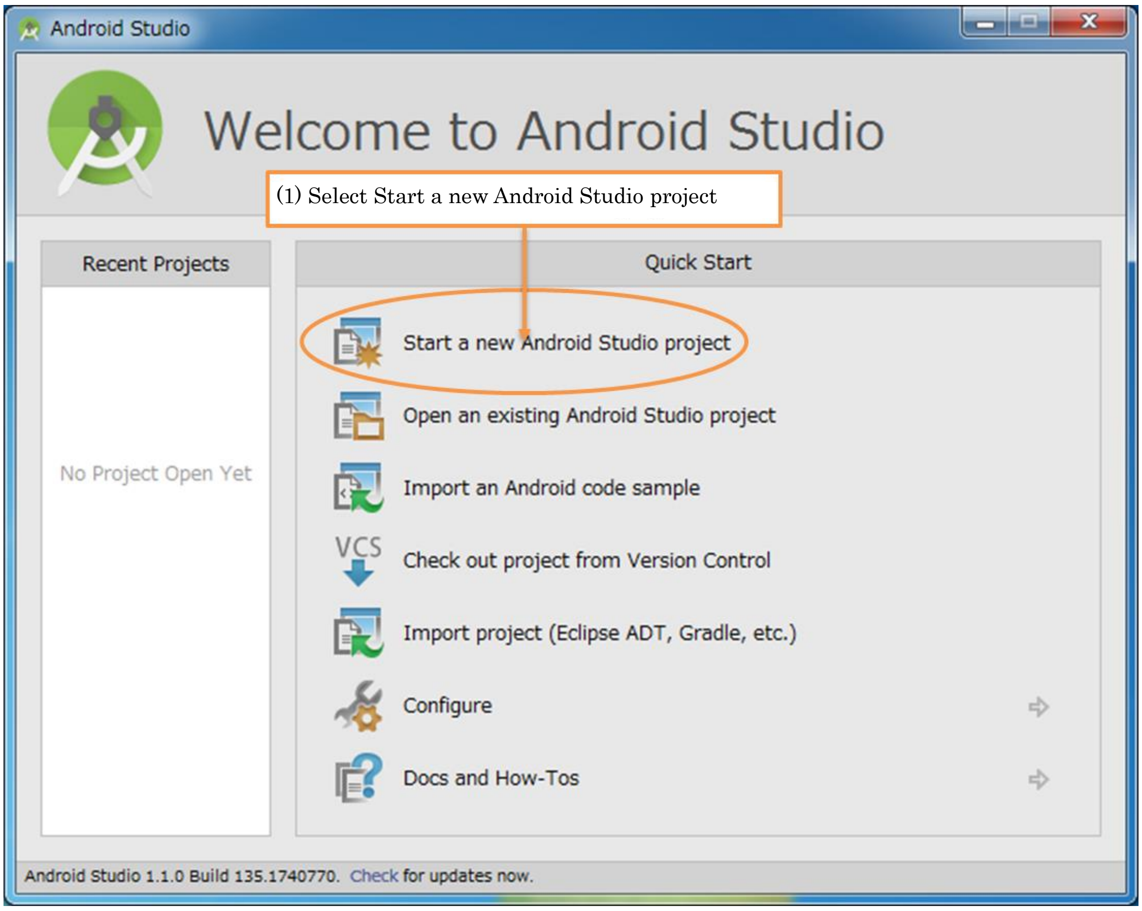
(1) Download the source code

(2) Download the Sample Script

Creating a new project

Create a new development project.

▼ Procedure

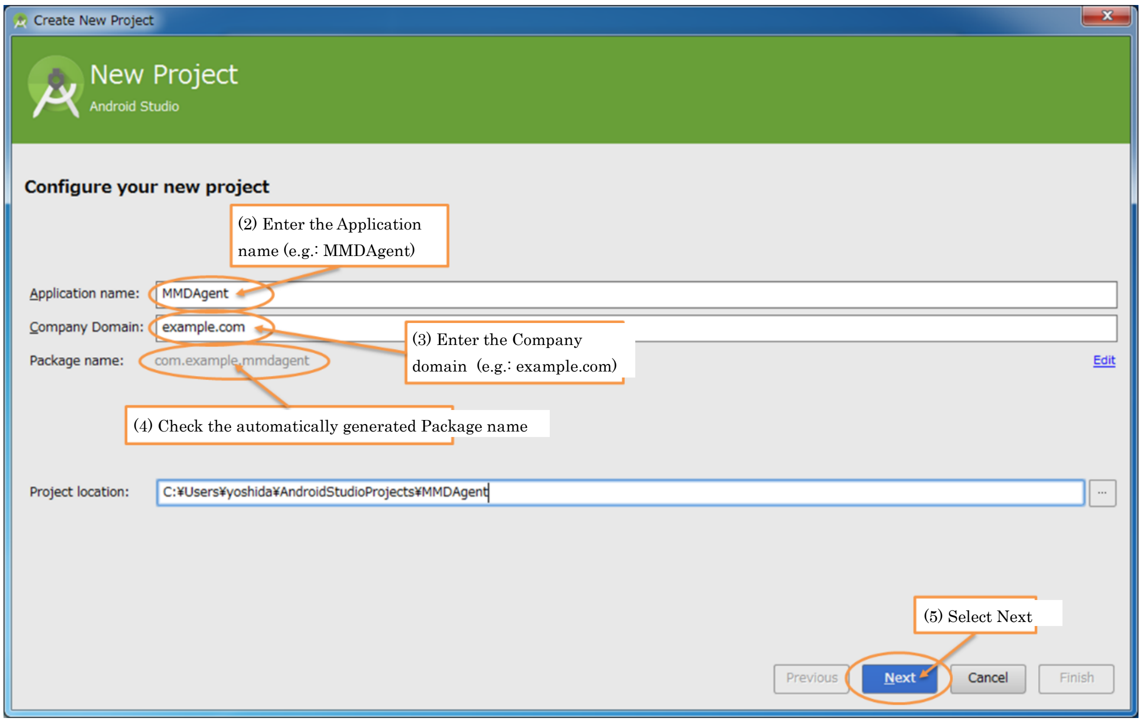


Android Studio
Welcome to Android Studio
Recent Projects
No Project Open Yet
Quick Start
Start a new Android Studio project
Open an existing Android Studio project
Import an Android code sample
VCS
Check out project from Version Control
Import project (Eclipse ADT, Gradle, etc.)
Configure
Docs and How-Tos

(1) Select Start a new Android Studio project

Android Studio 1.1.0 Build 135.1740770. Check for updates now.

* The above window is displayed automatically when Android Studio launches.



Create New Project

New Project
Android Studio

Configure your new project

(2) Enter the Application name (e.g.: MMDAgent)

Application name: MMDAgent

Company Domain: example.com

(3) Enter the Company domain (e.g.: example.com)

Package name: com.example.mmdagent

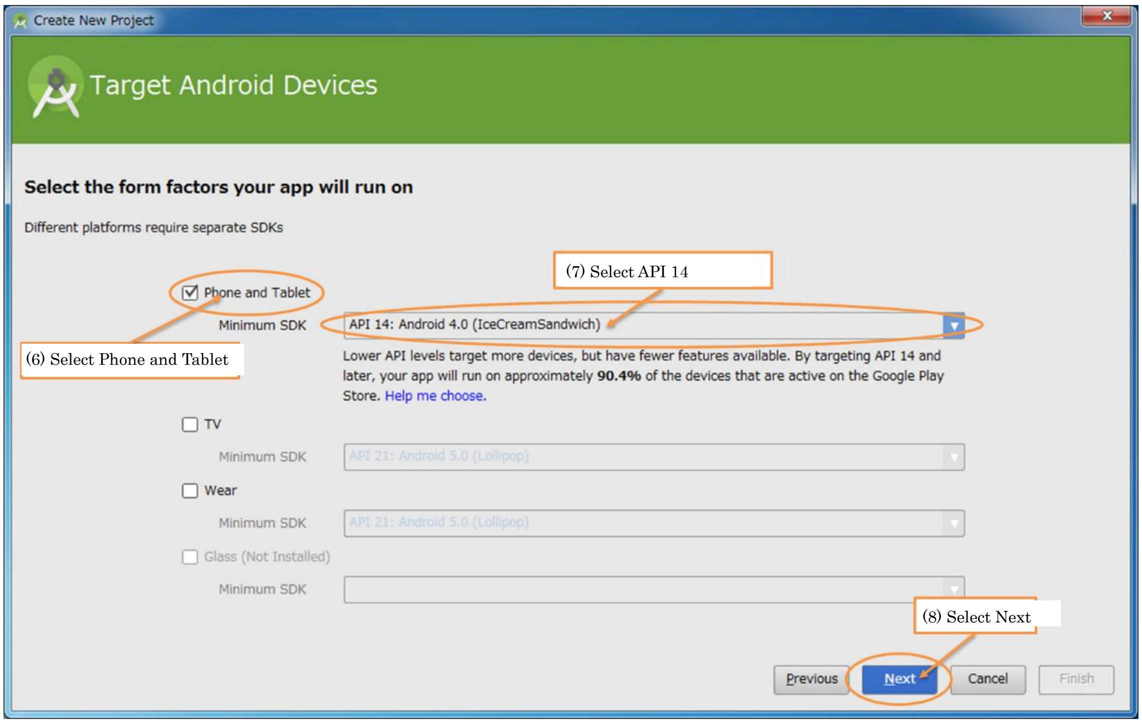
Edit

(4) Check the automatically generated Package name

Project location: C:¥Users¥yoshida¥AndroidStudioProjects¥MMDAgent

(5) Select Next

＊ Package name is used as an ID for the application, so select a name that does not overwrite any existing applications.
＊ The Package name entered automatically here is used in later settings.



Create New Project

Target Android Devices

Select the form factors your app will run on

Different platforms require separate SDKs

(7) Select API 14

☑ Phone and Tablet
Minimum SDK: API 14: Android 4.0 (IceCreamSandwich)

(6) Select Phone and Tablet

Lower API levels target more devices, but have fewer features available. By targeting API 14 and later, your app will run on approximately **90.4%** of the devices that are active on the Google Play Store. Help me choose.

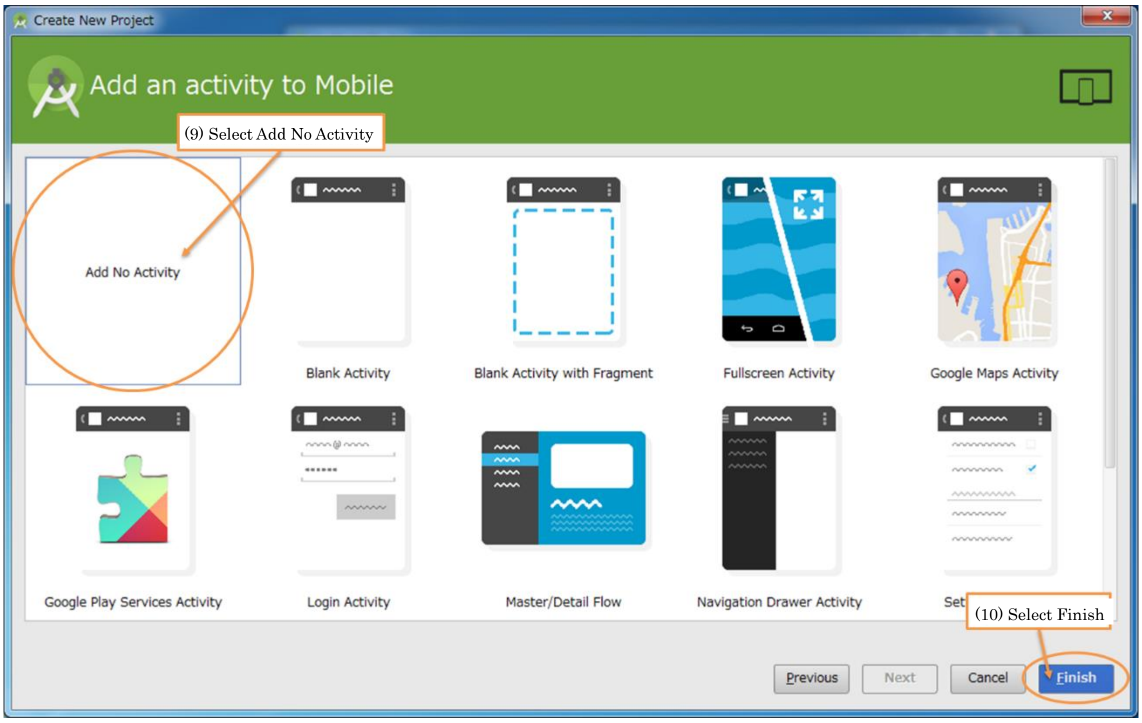
☐ TV
Minimum SDK: API 21: Android 5.0 (Lollipop)

☐ Wear
Minimum SDK: API 21: Android 5.0 (Lollipop)

☐ Glass (Not Installed)
Minimum SDK

(8) Select Next

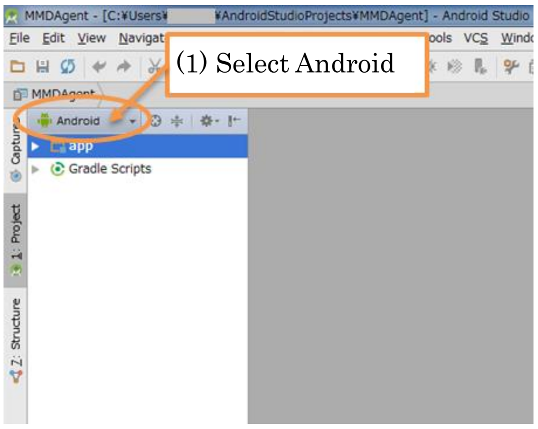
＊ Compiling and running on earlier than API 14 is not guaranteed.

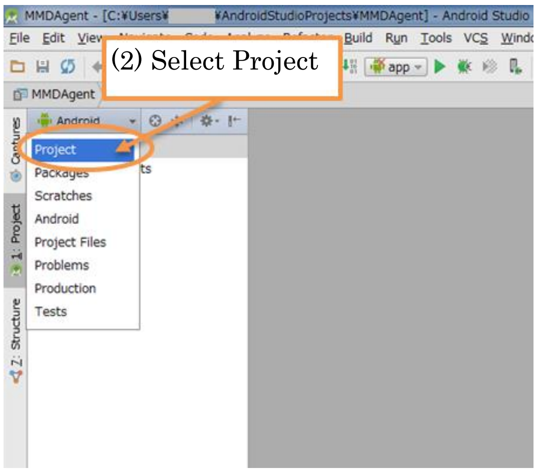

Create New Project
Add an activity to Mobile
(9) Select Add No Activity
Add No Activity
Blank Activity
Blank Activity with Fragment
Fullscreen Activity
Google Maps Activity
Google Play Services Activity
Login Activity
Master/Detail Flow
Navigation Drawer Activity
Set
(10) Select Finish
Previous
Next
Cancel
Finish

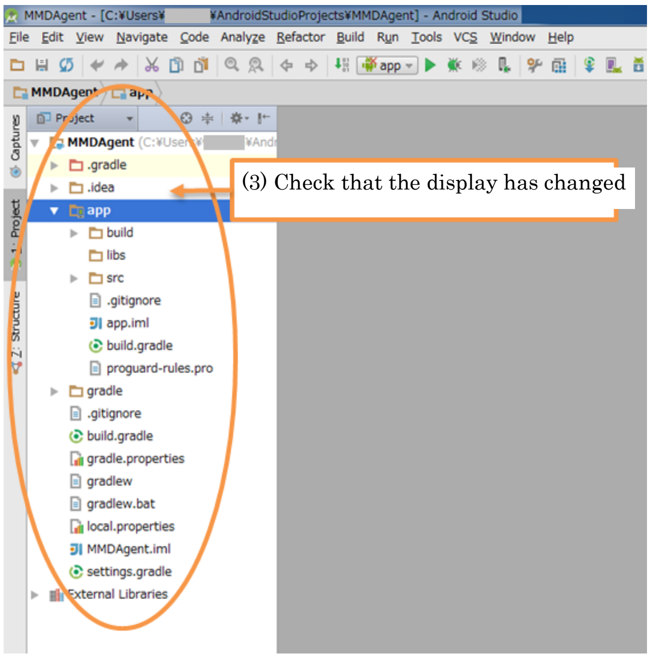
Changing how projects are displayed

Change the folder display so that it shows the actual project folder structure.

▼ Procedure



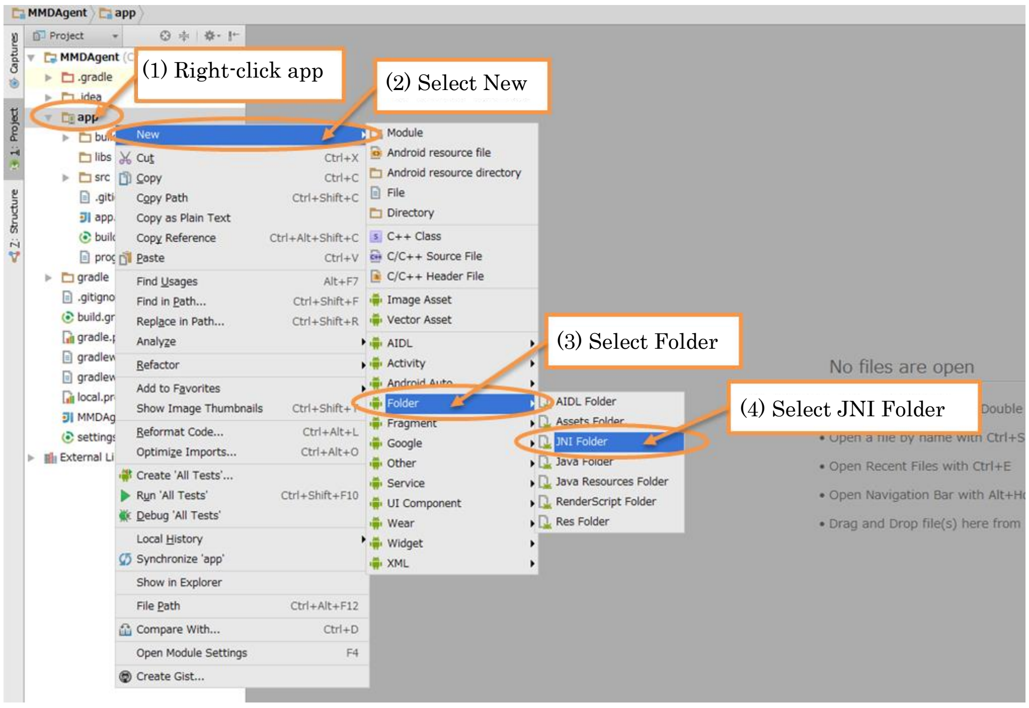


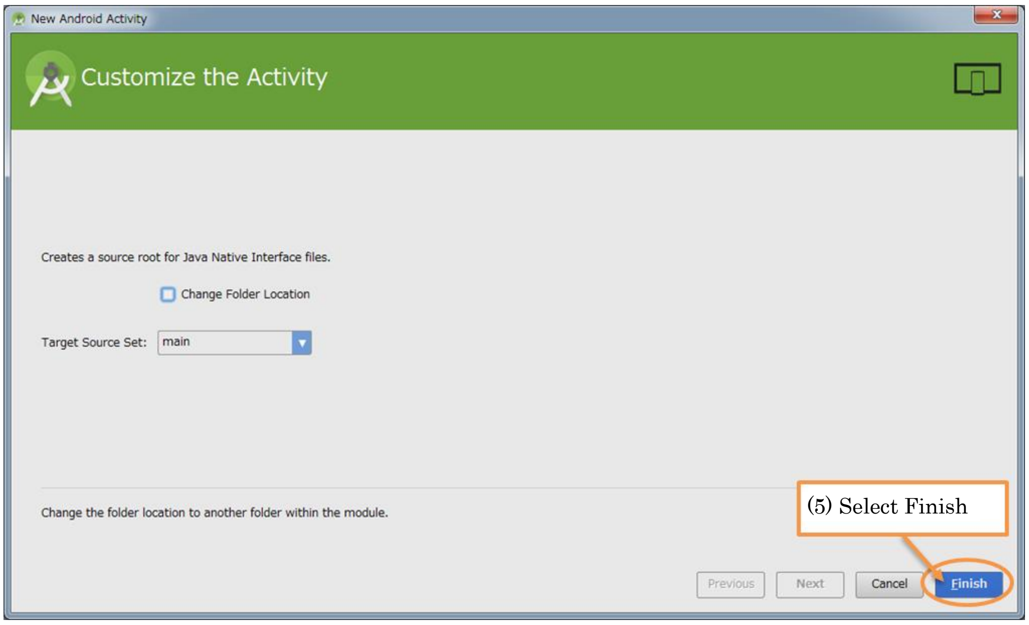


Creating a JNI folder

Create a jni folder to store the MMDAgent source code.

▼ Procedure





Creates a source root for Java Native Interface files.

Change Folder Location

Target Source Set: main

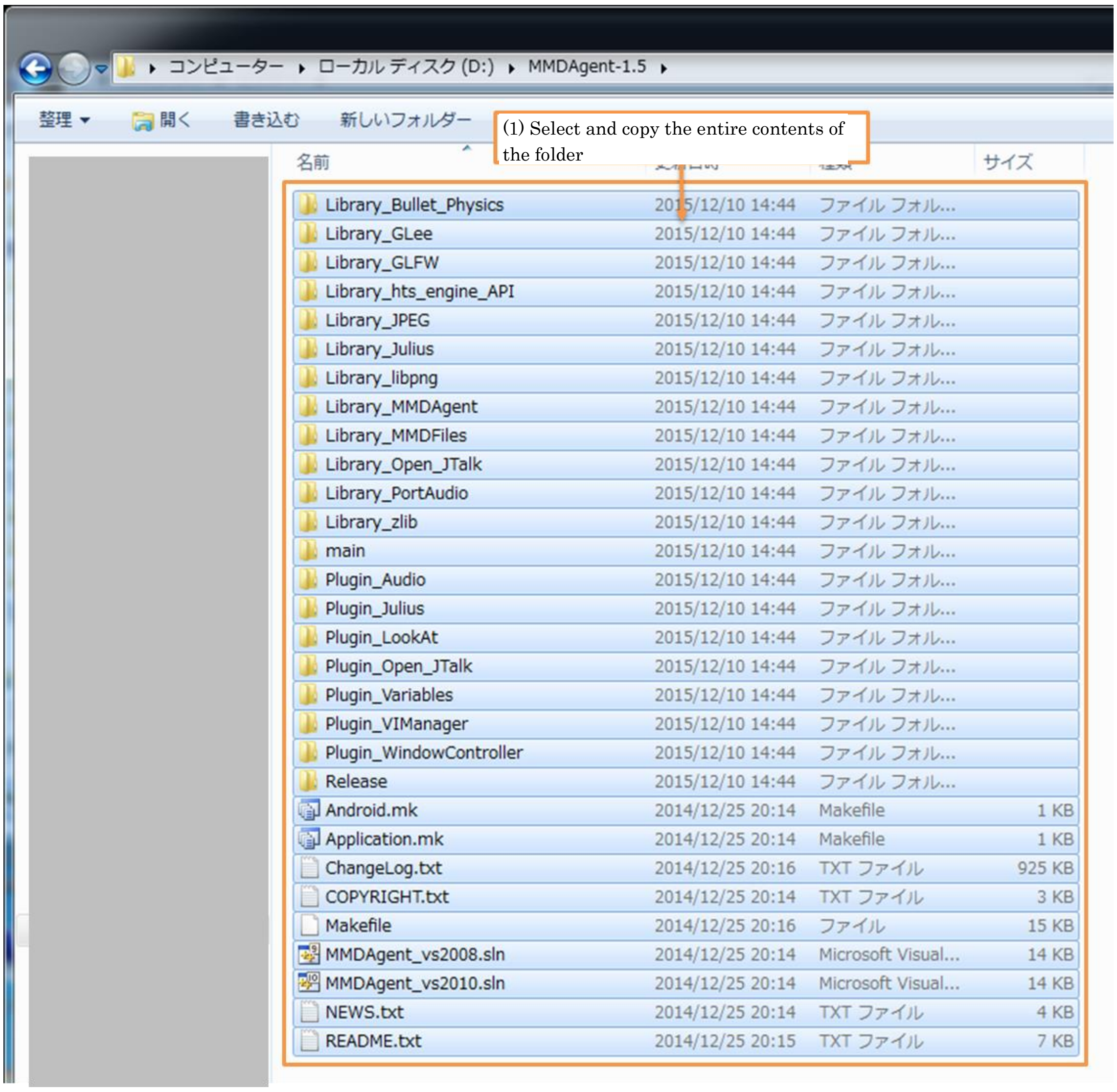
Change the folder location to another folder within the module.

* The JNI folder is created in the MMDAgent¥app¥src¥main folder.

Importing source code

Copy the downloaded source code to the jni folder.

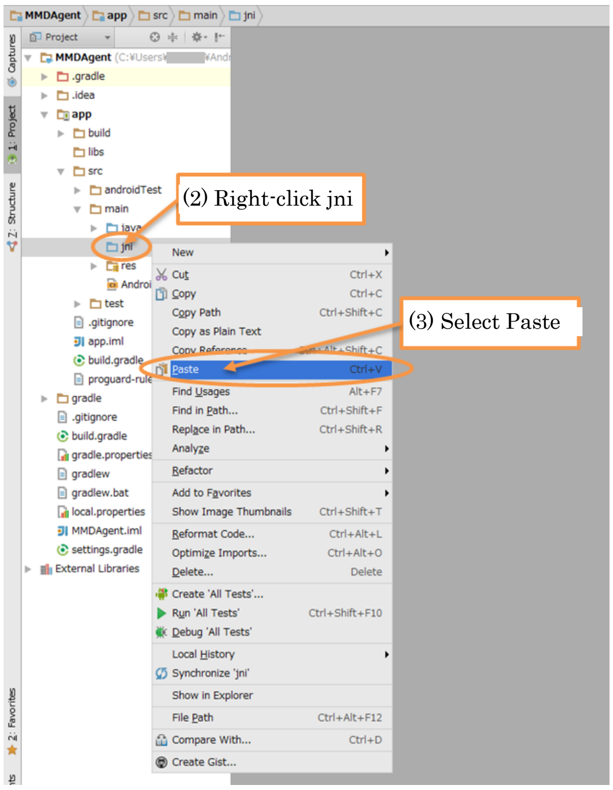
▼ Procedure (when using the standard explorer)

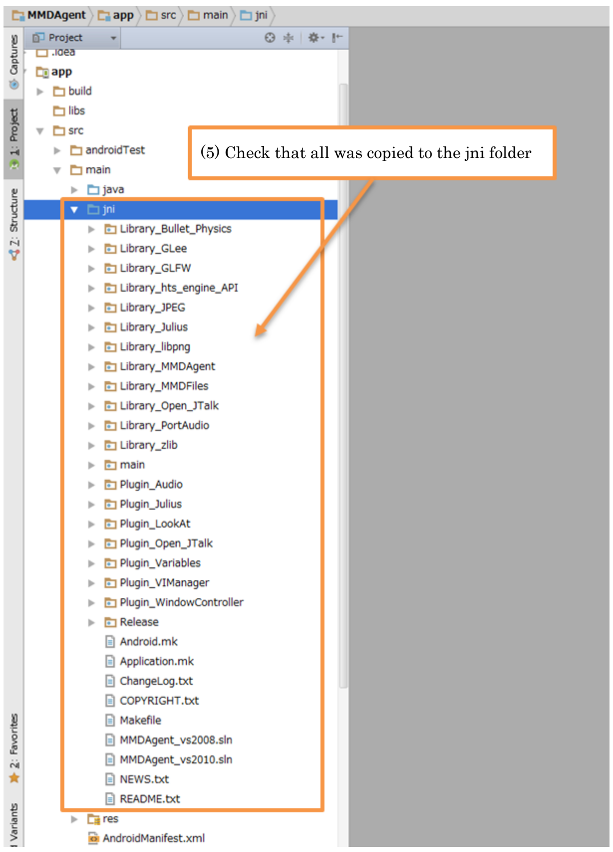


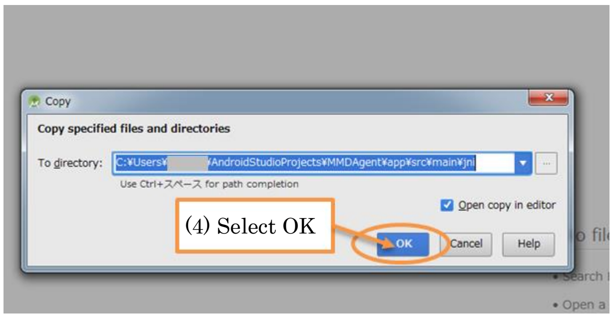
(1) Select and copy the entire contents of the folder

名前			サイズ
Library_Bullet_Physics	2015/12/10 14:44	ファイル フォル...	
Library_GLee	2015/12/10 14:44	ファイル フォル...	
Library_GLFW	2015/12/10 14:44	ファイル フォル...	
Library_hts_engine_API	2015/12/10 14:44	ファイル フォル...	
Library_JPEG	2015/12/10 14:44	ファイル フォル...	
Library_Julius	2015/12/10 14:44	ファイル フォル...	
Library_libpng	2015/12/10 14:44	ファイル フォル...	
Library_MMDAgent	2015/12/10 14:44	ファイル フォル...	
Library_MMDFiles	2015/12/10 14:44	ファイル フォル...	
Library_Open_JTalk	2015/12/10 14:44	ファイル フォル...	
Library_PortAudio	2015/12/10 14:44	ファイル フォル...	
Library_zlib	2015/12/10 14:44	ファイル フォル...	
main	2015/12/10 14:44	ファイル フォル...	
Plugin_Audio	2015/12/10 14:44	ファイル フォル...	
Plugin_Julius	2015/12/10 14:44	ファイル フォル...	
Plugin_LookAt	2015/12/10 14:44	ファイル フォル...	
Plugin_Open_JTalk	2015/12/10 14:44	ファイル フォル...	
Plugin_Variables	2015/12/10 14:44	ファイル フォル...	
Plugin_VIManager	2015/12/10 14:44	ファイル フォル...	
Plugin_WindowController	2015/12/10 14:44	ファイル フォル...	
Release	2015/12/10 14:44	ファイル フォル...	
Android.mk	2014/12/25 20:14	Makefile	1 KB
Application.mk	2014/12/25 20:14	Makefile	1 KB
ChangeLog.txt	2014/12/25 20:16	TXT ファイル	925 KB
COPYRIGHT.txt	2014/12/25 20:14	TXT ファイル	3 KB
Makefile	2014/12/25 20:16	ファイル	15 KB
MMDAgent_vs2008.sln	2014/12/25 20:14	Microsoft Visual...	14 KB
MMDAgent_vs2010.sln	2014/12/25 20:14	Microsoft Visual...	14 KB
NEWS.txt	2014/12/25 20:14	TXT ファイル	4 KB
README.txt	2014/12/25 20:15	TXT ファイル	7 KB

▼ Procedure（Android Studio）







Copy specified files and directories

To directory: C:¥Users¥[redacted]¥AndroidStudioProjects¥MMDAgent¥app¥src¥main¥jni

Use Ctrl+スペース for path completion

☑ Open copy in editor

[Notes]

This includes files not used with the Android version, but they do not affect development, so we do not delete them.

Creating system and content directories on an Android terminal

For testing, an MMDAgent system directory and content directory must be created on the run-time environment Android terminal.
(When completing the application for distribution, the directory is used for storing data downloaded from the server to the application and other tasks.)

Create the system directory

Create a system directory to store the AppData directory on the Android terminal.

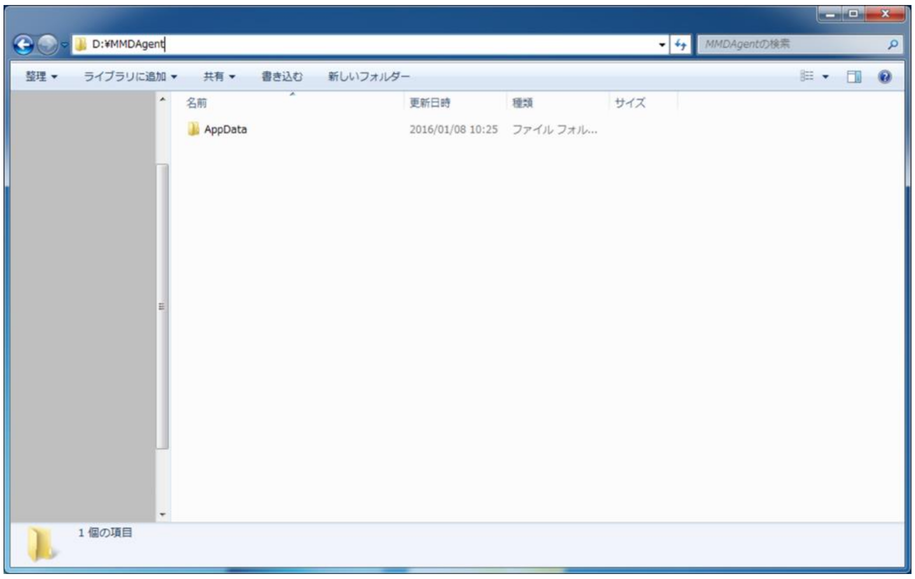
▼ Procedure

1. Create a transfer source folder
   Create a transfer source folder called MMDAgent in a suitable location on the PC.

2. Copy AppData to the transfer source folder.
   Copy the contents of the Release¥AppData folder in the downloaded source code to the transfer source folder.

3. Copy the source folder to the run-time environment terminal.
   Copy the transfer source folder created on the PC to the run-time environment terminal. Any path can be specified as the copy destination, but here we use the "/sdcard" folder in internal storage.
   * Depending on the terminal, /sdcard may not necessarily be internal storage.

▼Example of the transfer source folder



Creating the content directory

Create the content directory for storing the Android terminal sample script.

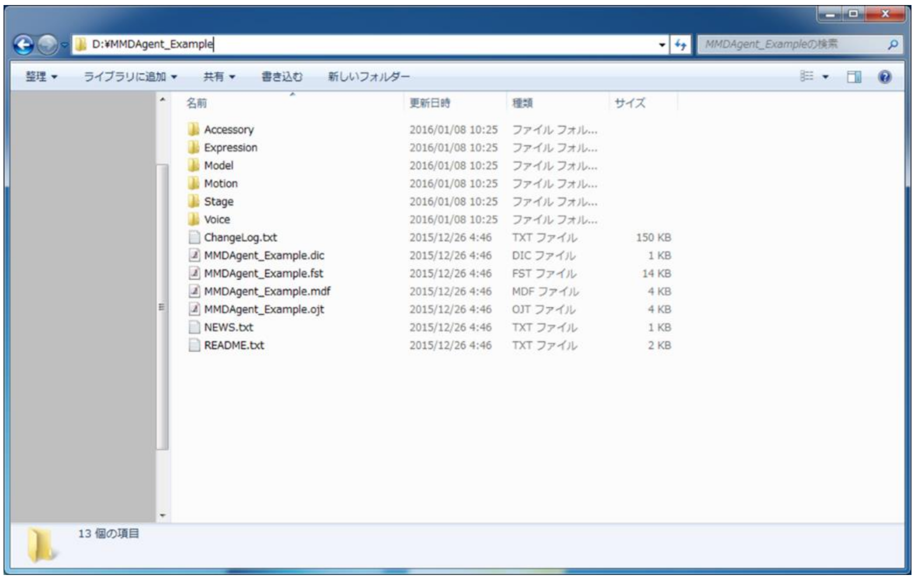
▼ Procedure

1. Create a transfer source folder
   Create a transfer source folder called MMDAgent_Example in a suitable location on the PC.

2. Copy Sample Script to the transfer source folder.
   Copy the contents of the downloaded Sample Script to the transfer source folder.

3. Copy the source folder to the run-time environment terminal.
   Copy the transfer source folder created on the PC to the run-time environment terminal. Any path can be specified as the copy destination, but here we use the "/sdcard" folder in internal storage.
   * Depending on the terminal, /sdcard may not necessarily be internal storage.

▼ Example of the transfer source folder



D:¥MMDAgent_Example

名前	更新日時	種類	サイズ
Accessory	2016/01/08 10:25	ファイル フォル...	
Expression	2016/01/08 10:25	ファイル フォル...	
Model	2016/01/08 10:25	ファイル フォル...	
Motion	2016/01/08 10:25	ファイル フォル...	
Stage	2016/01/08 10:25	ファイル フォル...	
Voice	2016/01/08 10:25	ファイル フォル...	
ChangeLog.txt	2015/12/26 4:46	TXT ファイル	150 KB
MMDAgent_Example.dic	2015/12/26 4:46	DIC ファイル	1 KB
MMDAgent_Example.fst	2015/12/26 4:46	FST ファイル	14 KB
MMDAgent_Example.mdf	2015/12/26 4:46	MDF ファイル	4 KB
MMDAgent_Example.ojt	2015/12/26 4:46	OJT ファイル	4 KB
NEWS.txt	2015/12/26 4:46	TXT ファイル	1 KB
README.txt	2015/12/26 4:46	TXT ファイル	2 KB

13 個の項目

Editing files

To build the Android version of MMDAgent, some files must be edited.
Edit the files indicated below, with reference to the lists of Edit items and examples of edited files.

Edit AndroidManifest.xml

▼ File path

MMDAgent¥app¥src¥main¥AndroidManifest.xml

* There are multiple files with the same name, so be sure to edit the correct one.

▼ Edit items

1. Change the package name (Line 1)
   Change the Package attribute to the value automatically entered when creating the project.

2. Add permissions to read and write to external storage (Lines 4 and 5)
   Add permissions to read and write to external storage.
   * MMDAgent 1.6 and greater require permissions to write.

3. Add permissions to use the microphone (Line 6)
   Add permissions to use the microphone for speech recognition.

4. Add an activity tag (Lines 15 to 21)
   Enter the names of shared libraries loaded by the NativeActivity in a meta-data tag, and enter the activity main settings and settings to register it in the Android launcher in an intent-filter tag.

▼File contents after editing

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
package="com.example.mmdagent">

    <!-- add permission -->
    <uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" />
    <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
    <uses-permission android:name="android.permission.RECORD_AUDIO" />

    <application
        android:allowBackup="true"
        android:label="@string/app_name"
        android:icon="@mipmap/ic_launcher"
        android:theme="@style/AppTheme">

        <!-- add NativeActivity -->
        <activity android:name="android.app.NativeActivity" android:label="@string/app_name">
            <meta-data android:name="android.app.lib_name" android:value="main" />
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>

    </application>
</manifest>
```

[Notes]
If including Java code when extending the Android version of MMDAgent, the android:hasCode attribute must be added to the application tag.

・android:hasCode attribute values

Value	Summary
true (default)	Includes Java code.
false	Does not include Java code.

Edit local.properties

▼ File path

MMDAgent¥local.properties

▼ Edit items

1. Add the Android NDK path (Line 11)
   Add the Android NDK install folder path.

▼File contents after editing

```
 1 ## This file is automatically generated by Android Studio.
 2 # Do not modify this file -- YOUR CHANGES WILL BE ERASED!
 3 #
 4 # This file should *NOT* be checked into Version Control Systems,
 5 # as it contains information specific to your local configuration.
 6 #
 7 # Location of the SDK. This is only used by Gradle.
 8 # For customization when using a Version Control System, please read the
 9 # header note.
10 sdk.dir=C¥:¥¥Users¥¥<user name>¥¥AppData¥¥Local¥¥Android¥¥sdk
11 ndk.dir=C¥:¥¥Users¥¥<user name>¥¥AppData¥¥Local¥¥Android¥¥ndk
```

* Modify <user name> to match your environment.

* Use “¥¥” as the folder separator.

Edit build.gradle

▼ File path

MMDAgent¥app¥build.gradle

＊ There are multiple files with the same name, so be sure to edit the correct one.

▼ Edit items

1. Change the package name (Line 9)
   Change the applicationId to the value automatically input when creating the project.

2. Disable Android Studio build (Line 16)
   To use the ndk-build command, disable Android Studio build.

3. Add a task to generate config.h (Lines 31 to 68)
   Add a task to generate the settings file for Julius (the speech recognition module).

4. Add tasks to perform the compile (Lines 70 to 79)
   Add tasks to run the ndk-build command.
   ＊ Modify ndkDir on line 72 to match your environment.

5. Add task dependency relationships (Lines 81 to 84, 103 to 104)
   Configure the dependency relationships between tasks in this file.

6. Add a task to delete config.h (Lines 86 to 91)
   Add a task to delete the configuration file for Julius (the speech recognition module).

7. Add a task to delete ndk-build related files (Lines 93 to 101)
   Add a task to delete files generated by the ndk-build command.
   ＊ Modify ndkDir in line 95 to match your environment.

▼ File contents after editing

```
apply plugin: 'com.android.application'
import org.apache.tools.ant.taskdefs.condition.Os

android {
    compileSdkVersion 21
    buildToolsVersion "21.1.2"

    defaultConfig {
        applicationId "com.example.mmdagent"
        minSdkVersion 15
        targetSdkVersion 21
        versionCode 1
        versionName "1.0"
    }

    sourceSets.main.jni.srcDirs = [] // avoid using NdkCompile task

    buildTypes {
        release {
            minifyEnabled false
            proguardFiles(getDefaultProguardFile('proguard-android.txt'),
'proguard-rules.pro');
        }
    }
}

dependencies {
    compile fileTree(dir: 'libs', include: ['*.jar'])
    compile 'com.android.support:appcompat-v7:21.0.3'
}

task makeConfigFile << {
    def configFile1 = file("src/main/jni/Library_Julius/include/julius/config.h");
    configFile1.createNewFile()
    configFile1.write('#define JULIUS_PRODUCTNAME ""' + System.getProperty("line.separator"))
    configFile1.append('#define JULIUS_VERSION "4.3"' + System.getProperty("line.separator"))
    configFile1.append('#define JULIUS_SETUP "fast"' + System.getProperty("line.separator"))
    configFile1.append('#define JULIUS_HOSTINFO ""' + System.getProperty("line.separator"))
    configFile1.append('#define RETSIGTYPE void' + System.getProperty("line.separator"))
    configFile1.append('#define STDC_HEADERS 1' + System.getProperty("line.separator"))
    configFile1.append('#define UNIGRAM_FACTORING 1' + System.getProperty("line.separator"))
    configFile1.append('#define LOWMEM2 1' + System.getProperty("line.separator"))
    configFile1.append('#define PASS1_IWCD 1' + System.getProperty("line.separator"))
    configFile1.append('#define SCAN_BEAM 1' + System.getProperty("line.separator"))
    configFile1.append('#define GPRUNE_DEFAULT_BEAM 1' + System.getProperty("line.separator"))
    configFile1.append('#define CONFIDENCE_MEASURE 1' + System.getProperty("line.separator"))
    configFile1.append('#define LM_FIX_DOUBLE_SCORING 1' +
System.getProperty("line.separator"))
    configFile1.append('#define GRAPHOUT_DYNAMIC 1' + System.getProperty("line.separator"))
    configFile1.append('#define GRAPHOUT_SEARCH 1' + System.getProperty("line.separator"))
    configFile1.append('#define HAVE_STRCASECMP 1' + System.getProperty("line.separator"))

    def configFile2 = file("src/main/jni/Library_Julius/include/sent/config.h");
    configFile2.createNewFile()
    configFile2.write('#define LIBSENT_VERSION "4.3"' + System.getProperty("line.separator"))
    configFile2.append('#define AUDIO_API_NAME ""' + System.getProperty("line.separator"))
    configFile2.append('#define AUDIO_API_DESC ""' + System.getProperty("line.separator"))
    configFile2.append('#define AUDIO_FORMAT_DESC ""' + System.getProperty("line.separator"))
    configFile2.append('#define GZIP_READING_DESC ""' + System.getProperty("line.separator"))
    configFile2.append('#define STDC_HEADERS 1' + System.getProperty("line.separator"))
    configFile2.append('#define USE_MIC 1' + System.getProperty("line.separator"))
    configFile2.append('#define USE_ADDLOG_ARRAY 1' + System.getProperty("line.separator"))
    configFile2.append('#define HAVE_SOCKLEN_T 1' + System.getProperty("line.separator"))
    configFile2.append('#define HAVE_UNISTD_H 1' + System.getProperty("line.separator"))
    configFile2.append('#define HAVE_ZLIB 1' + System.getProperty("line.separator"))
    configFile2.append('#define HAVE_STRCASECMP 1' + System.getProperty("line.separator"))
    configFile2.append('#define HAVE_SLEEP 1' + System.getProperty("line.separator"))
    configFile2.append('#define CLASS_NGRAM 1' + System.getProperty("line.separator"))
    configFile2.append('#define MFCC_SINCOS_TABLE 1' + System.getProperty("line.separator"))
}

```

```
task buildNative(type:Exec) {
    //def ndkDir = project.plugins.findPlugin('com.android.application').getNdkFolder()
    def ndkDir = "C:¥¥Users¥¥<ユーザー名>¥¥AppData¥¥Local¥¥Android¥¥ndk"
    def jOption = '-j'+Runtime.runtime.availableProcessors()
    if(Os.isFamily(Os.FAMILY_WINDOWS)){
        commandLine("$ndkDir/ndk-build.cmd", jOption, '-C', file('src/main').absolutePath, 
'NDK_APP_LIBS_OUT=jniLibs');
    }else{
        commandLine("$ndkDir/ndk-build", jOption, '-C', file('src/main').absolutePath, 
'NDK_APP_LIBS_OUT=jniLibs');
    }
}

buildNative.dependsOn 'makeConfigFile'
tasks.withType(JavaCompile) {
    compileTask -> compileTask.dependsOn 'buildNative'
}

task cleanConfigFile << {
    def configFile1 = file('src/main/jni/Library_Julius/include/julius/config.h');
    configFile1.delete();
    def configFile2 = file('src/main/jni/Library_Julius/include/sent/config.h');
    configFile2.delete();
}

task cleanNative(type:Exec){
    //def ndkDir = project.plugins.findPlugin('com.android.application').getNdkFolder()
    def ndkDir = "C:¥¥Users¥¥<ユーザー名>¥¥AppData¥¥Local¥¥Android¥¥ndk"
    if(Os.isFamily(Os.FAMILY_WINDOWS)){
        commandLine("$ndkDir/ndk-build.cmd", 'clean', '-C', file('src/main').absolutePath, 
"NDK_APP_LIBS_OUT=jnilibs");
    }else{
        commandLine("$ndkDir/ndk-build", 'clean', '-C', file('src/main').absolutePath, 
"NDK_APP_LIBS_OUT=jnilibs");
    }
}
```

```

cleanNative.dependsOn 'cleanNative'
clean.dependsOn 'cleanNative'

```

Edit Android.mk

▼ File path

MMDAgent¥app¥src¥main¥jni¥Library_MMDAgent¥Android.mk

* There are multiple files with the same name, so be sure to edit the correct one.

▼ Edit items

1. Change the path for DMMDAGENT_OVERWRITEEXEFILE (Line 38)
   Change it to that of the fst file in the Content directory. Note that this will be referenced as an exe file, so enter the filename extension as exe.

   ```
   ¥"/sdcard/MMDAgent_Example/MMDAgent_Example.exe¥"
   ```

   * If following the instructions in this manual exactly, enter the path above.

2. Change the path for DMMDAGENT_OVERWRITECONFIGFILE (Line 39)
   Change it to that of the mdf file in the Content directory.

   ```
   ¥"/sdcard/MMDAgent_Example/MMDAgent_Example.mdf¥"
   ```

   * If following the instructions in this manual exactly, enter the path above.

3. Change the path for DMMDAGENT_OVERWRITESYSDATADIR (Line 40)
   Change it to that of the AppData folder in the System directory.

   ```
   ¥"/sdcard/MMDAgent/AppData¥"
   ```

   * If following the instructions in this manual exactly, enter the path above.

4. Change the path for DMMDAGENT_OVERWRITEPLUGINDIR (Line 41)
   Change it to match the Package name of the created project.

   ```
   ¥"/data/data/<packageName>/lib¥"
   ```

   * Enter the value automatically input when creating the project in place of <packageName> above.

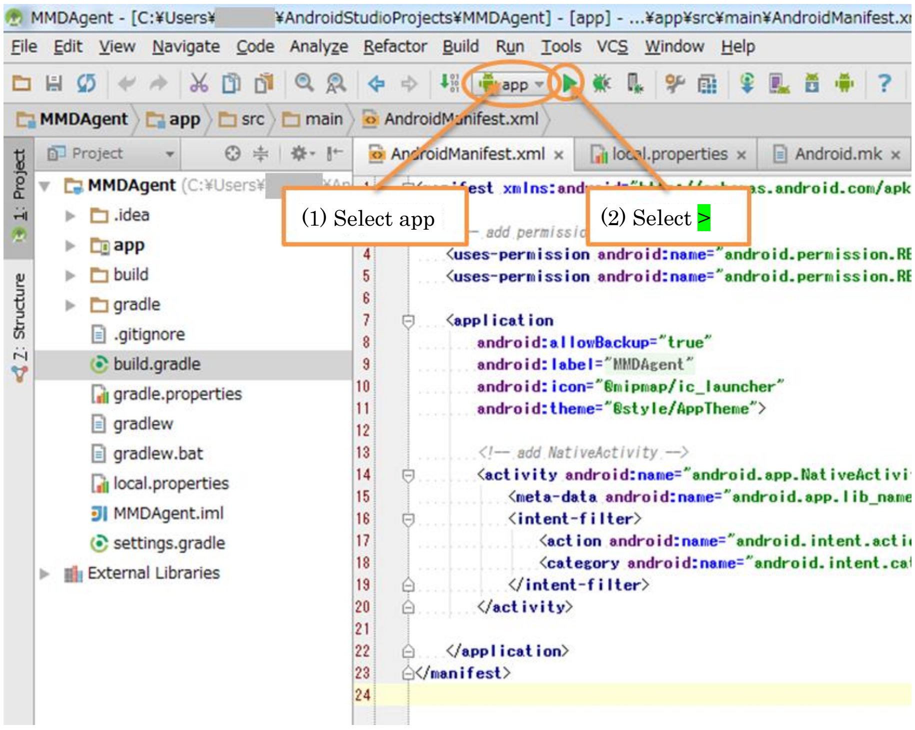
▼ File contents after editing

```
LOCAL_PATH := $(call my-dir)

include $(CLEAR_VARS)

LOCAL_MODULE     := MMDAgent
LOCAL_SRC_FILES  := src/lib/BoneController.cpp ¥
                    src/lib/LipSync.cpp ¥
                    src/lib/LogText.cpp ¥
                    src/lib/Message.cpp ¥
                    src/lib/MMDAgent.cpp ¥
                    src/lib/MMDAgent_utils.cpp ¥
                    src/lib/MotionStocker.cpp ¥
                    src/lib/Option.cpp ¥
                    src/lib/PMDObject.cpp ¥
                    src/lib/Plugin.cpp ¥
                    src/lib/Render.cpp ¥
                    src/lib/ScreenWindow.cpp ¥
                    src/lib/Stage.cpp ¥
                    src/lib/FreeTypeGL.cpp ¥
                    src/lib/TileTexture.cpp ¥
                    src/lib/Timer.cpp
LOCAL_C_INCLUDES := $(LOCAL_PATH)/include ¥
                    $(LOCAL_PATH)/../Library_JPEG/include ¥
                    $(LOCAL_PATH)/../Library_Bullet_Physics/include ¥
                    $(LOCAL_PATH)/../Library_GLee/include ¥
                    $(LOCAL_PATH)/../Library_libpng/include ¥
                    $(LOCAL_PATH)/../Library_zlib/include ¥
                    $(LOCAL_PATH)/../Library_MMDFiles/include ¥
                    $(LOCAL_PATH)/../Library_GLFW/include ¥
                    $(LOCAL_PATH)/../Library_FreeType/include ¥
                    $(LOCAL_PATH)/../Library_UTF8-CPP/include
LOCAL_CFLAGS     += -DMMDAGENT_DONTRENDERDEBUG ¥
                    -DMMDAGENT_DONTUSESHADOWMAP ¥
                    -DMMDAGENT_DONTPICKMODEL ¥
                    -DMMDAGENT_DONTUSEMOUSE ¥
                    -DMMDAGENT_DONTUSEWINDOW ¥
                    -DMMDAGENT ¥
                    -DMMDAGENT_OVERWRITEEXEFILE="¥"/sdcard/MMDAgent_Example/MMDAgent_Example.exe¥"" ¥
                    -DMMDAGENT_OVERWRITECONFIGFILE="¥"/sdcard/MMDAgent_Example/MMDAgent_Example.mdf¥"" ¥
                    -DMMDAGENT_OVERWRITESYSDATADIR="¥"/sdcard/MMDAgent/AppData¥"" ¥
                    -DMMDAGENT_OVERWRITEPLUGINDIR="¥"/data/data/com.example.mmdagent/lib¥""

include $(BUILD_STATIC_LIBRARY)

```

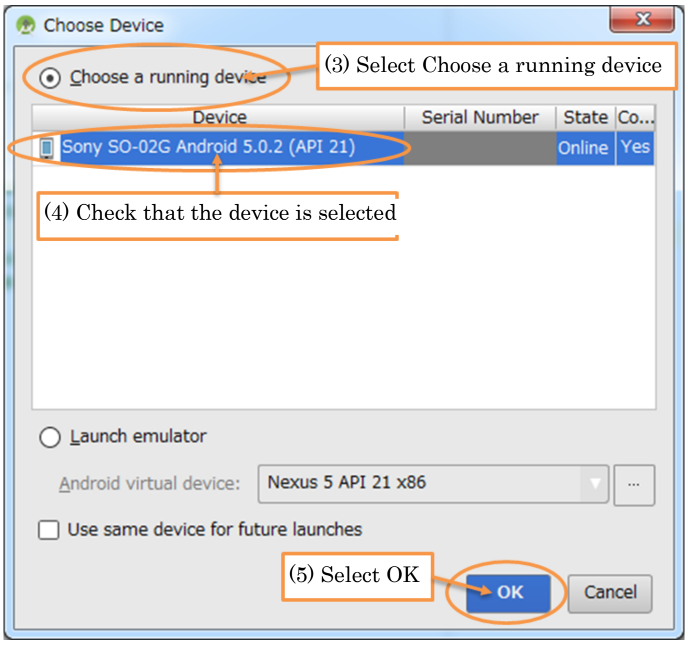
Building and running source code

Build the source code and run the app on an Android terminal.

▶ Procedure


MMDAgent - [C:¥Users¥[redacted]¥AndroidStudioProjects¥MMDAgent] - [app] - ...¥app¥src¥main¥AndroidManifest.xı
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
app
MMDAgent app src main AndroidManifest.xml
Project
AndroidManifest.xml × local.properties × Android.mk ×
MMDAgent (C:¥Users¥[redacted]
.idea
app
build
gradle
.gitignore
build.gradle
gradle.properties
gradlew
gradlew.bat
local.properties
MMDAgent.iml
settings.gradle
External Libraries
1: Project
Z: Structure
(1) Select app
(2) Select >

```
<manifest xmlns:android="http://schemas.android.com/apk

    <!-- add permissio
    <uses-permission android:name="android.permission.RE
    <uses-permission android:name="android.permission.RE

    <application
        android:allowBackup="true"
        android:label="MMDAgent"
        android:icon="@mipmap/ic_launcher"
        android:theme="@style/AppTheme">

        <!-- add NativeActivity -->
        <activity android:name="android.app.NativeActivi
            <meta-data android:name="android.app.lib_name
            <intent-filter>
                <action android:name="android.intent.acti
                <category android:name="android.intent.ca
            </intent-filter>
        </activity>

    </application>
</manifest>
```


Choose Device
(3) Select Choose a running device
(4) Check that the device is selected
(5) Select OK
OK
Cancel

◉ Choose a running device

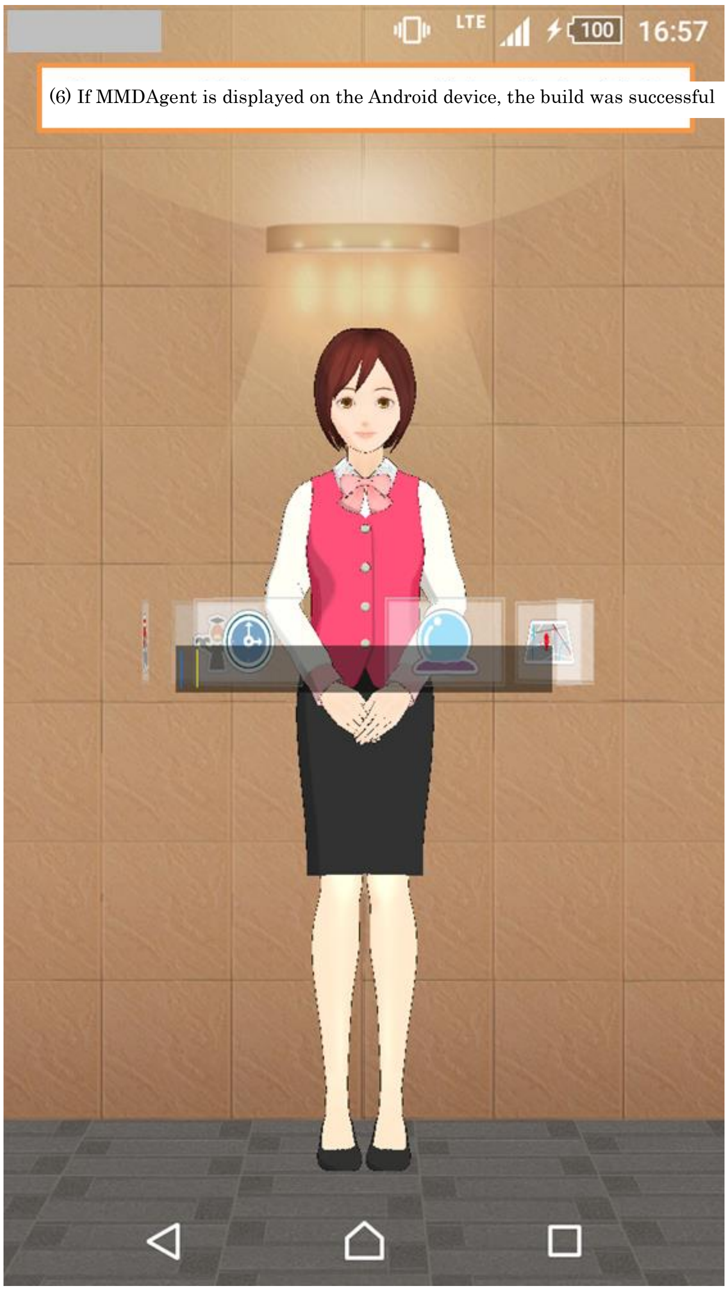
Device	Serial Number	State	Co...
Sony SO-02G Android 5.0.2 (API 21)		Online	Yes

○ Launch emulator

Android virtual device: Nexus 5 API 21 x86

☐ Use same device for future launches

* If no errors occur during the build, the window above will automatically appear.



(6) If MMDAgent is displayed on the Android device, the build was successful