

記事

[Toshihiko Minamoto](#) · 2021年7月20日 7m read

## Windows環境でnginxを構築する

皆さん、こんにちは。

InterSystems製品ではCache 2015.1からNGINXに対応しているのですが、インストールマニュアルを見るとC言語のソースファイルが提供されていて、「むむむ。。。コンパイルか。。。」と躊躇してしまいますね(ಠ\_ಠ)。  
そこで、実際にWindows環境でNGINXをビルドしてみましたのでレポートします。


### IRISのインストール

IRISまたはIRIS for Healthのキットを実行し、インストールします。

Nginxのアクセスに必要なUniversal Web gateway

Moduleをインストールするには、セットアップタイプでCustomを選択し、

Web Server Gatewayを明示的にインストールする必要がありますので、ご注意ください。

 InterSystems IRIS - Installation



セットアップ タイプ



ご利用方法に合わせて最適なセットアップ タイプを選択してください。

- Development**      Installs InterSystems IRIS server, InterSystems IRIS Studio, xDBC and all supported language bindings.
- Server**              Installs InterSystems IRIS server and Web Gateway.
- Client**                Installs InterSystems IRIS Studio and xDBC.
- Web Server**         Installs Web Gateway.
- Custom**              Specify the list of features you want installed.

< Back

Next >

Cancel



また、今回はIRIS for Healthをインストールしていますので、C:\InterSystems\IRISHealthディレクトリにIRISをインストールしています。

IRISだったり、他のディレクトリにインストールされている場合は適宜インストールディレクトリを読み替えてください。

## VisualStudioのインストール

Windows環境でコンパイルが必要ですのでVisualStudioをインストールします。バージョンはそんなに古くなければ大丈夫かと思います。ちなみに私はWindows 10の環境にインストールしていたVisual Studio 2017を使用しました。

## MSYSのインストール

MSYSはSourceForge(<https://sourceforge.net/projects/mingw/files/MSYS/>)からダウンロードします。以下の画面の「Download Latest Version」ボタンをクリックします。

Home / Browse / Development / Build Tools / MinGW - Minimalist GNU for Windows / Files

# MinGW - Minimalist GNU for Windows

A native Windows port of the GNU Compiler Collection (GCC)

Brought to you by: [cstrauss](#), [earnie](#), [gressett](#), [keithmarshall](#)

Summary **Files** Reviews Support News Wiki

**Download Latest Version**  
mingw-get-setup.exe (86.5 kB)

Get Updates

Home / MSYS

ダウンロードされた、mingw-get-setup.exeを実行します。

MinGW Installation Manager Setup Tool

mingw-get version 0.6.2-beta-20131004-1



Written by Keith Marshall

Copyright © 2009-2013, MinGW.org Project

<http://mingw.org>

This is free software; see the product documentation or source code, for copying and redistribution conditions. There is NO WARRANTY; not even an implied WARRANTY OF MERCHANTABILITY, nor of FITNESS FOR ANY PARTICULAR PURPOSE.

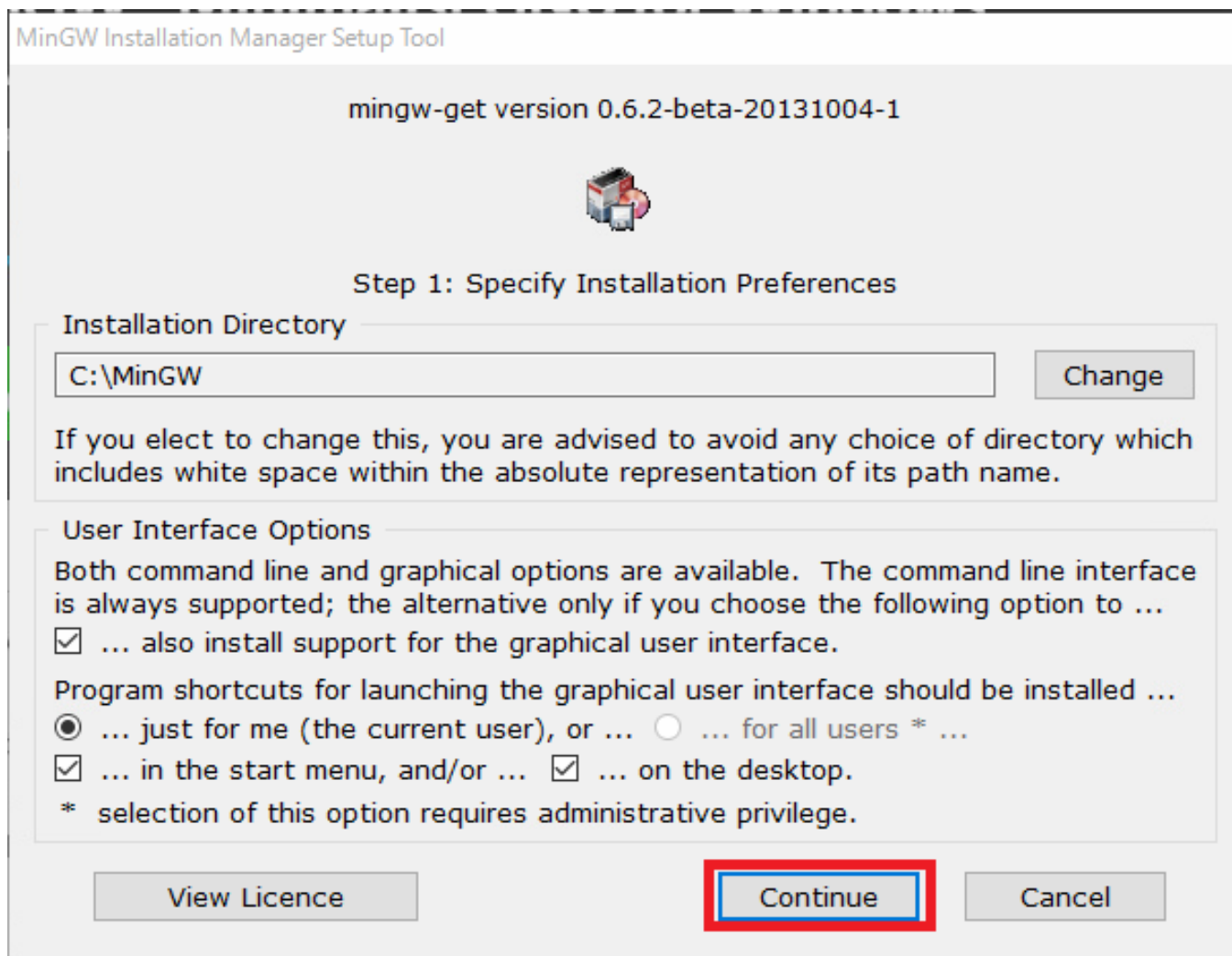
This tool will guide you through the first time setup of the MinGW Installation Manager software (mingw-get) on your computer; additionally, it will offer you the opportunity to install some other common components of the MinGW software distribution.

After first time setup has been completed, you should invoke the MinGW Installation Manager directly, (either the CLI mingw-get.exe variant, or its GUI counterpart, according to your preference), when you wish to add or to remove components, or to upgrade your MinGW software installation.

View Licence

Install


Cancel



以上でインストールが始まります。

## MinGW Installation Manager Setup Tool

mingw-get version 0.6.2-beta-20131004-1



**Step 2: Download and Set Up MinGW Installation Manager**

**Download Progress**

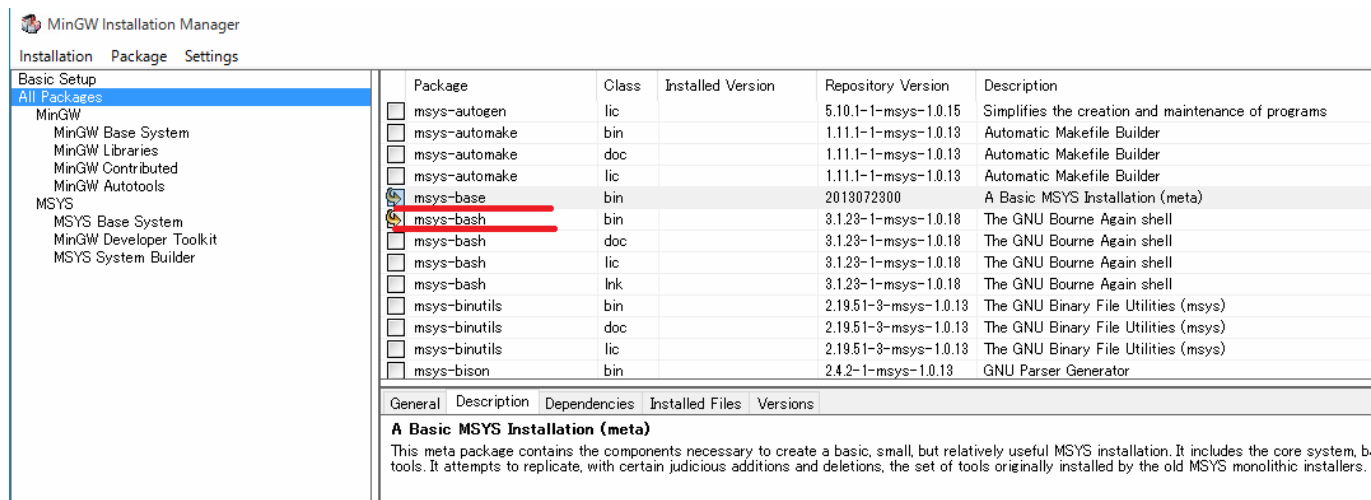
Catalogue update completed; please check 'Details' pane for errors.

Processed 113 of 113 items : 100 %

**Details**

```
mingw-get: *** INFO *** setup: unpacking mingw-get-setup-0.6.2-mingw32-beta-20131004-1-xml.tar.xz
mingw-get: *** INFO *** setup: updating installation database
mingw-get: *** INFO *** setup: register mingw-get-0.6.2-mingw32-beta-20131004-1-bin.tar.xz
mingw-get: *** INFO *** setup: register mingw-get-0.6.2-mingw32-beta-20131004-1-gui.tar.xz
mingw-get: *** INFO *** setup: register mingw-get-0.6.2-mingw32-beta-20131004-1-lic.tar.xz
mingw-get: *** INFO *** setup: installation database updated
```

インストールが終わると以下のInstallation Managerが表示されますので、Package欄のmsys-base,msys-bashを選択、「installation」メニューの「update catalogue」をクリックし、これらのパッケージをインストールします。



MinGW Installation Manager

Installation Package Settings

Basic Setup

All Packages

- MinGW
  - MinGW Base System
  - MinGW Libraries
  - MinGW Contributed
  - MinGW Autotools
- MSYS
  - MSYS Base System
  - MinGW Developer Toolkit
  - MSYS System Builder

Package	Class	Installed Version	Repository Version	Description
<input type="checkbox"/> msys-autogen	lic		5.10.1-1-msys-1.0.15	Simplifies the creation and maintenance of programs
<input type="checkbox"/> msys-automake	bin		1.11.1-1-msys-1.0.13	Automatic Makefile Builder
<input type="checkbox"/> msys-automake	doc		1.11.1-1-msys-1.0.13	Automatic Makefile Builder
<input type="checkbox"/> msys-automake	lic		1.11.1-1-msys-1.0.13	Automatic Makefile Builder
<input checked="" type="checkbox"/> msys-base	bin		2013072300	A Basic MSYS Installation (meta)
<input checked="" type="checkbox"/> msys-bash	bin		3.1.23-1-msys-1.0.18	The GNU Bourne Again shell
<input type="checkbox"/> msys-bash	doc		3.1.23-1-msys-1.0.18	The GNU Bourne Again shell
<input type="checkbox"/> msys-bash	lic		3.1.23-1-msys-1.0.18	The GNU Bourne Again shell
<input type="checkbox"/> msys-bash	lnk		3.1.23-1-msys-1.0.18	The GNU Bourne Again shell
<input type="checkbox"/> msys-binutils	bin		2.19.51-3-msys-1.0.13	The GNU Binary File Utilities (msys)
<input type="checkbox"/> msys-binutils	doc		2.19.51-3-msys-1.0.13	The GNU Binary File Utilities (msys)
<input type="checkbox"/> msys-binutils	lic		2.19.51-3-msys-1.0.13	The GNU Binary File Utilities (msys)
<input type="checkbox"/> msys-bison	bin		2.4.2-1-msys-1.0.13	GNU Parser Generator

General Description Dependencies Installed Files Versions

**A Basic MSYS Installation (meta)**

This meta package contains the components necessary to create a basic, small, but relatively useful MSYS installation. It includes the core system, b tools. It attempts to replicate, with certain judicious additions and deletions, the set of tools originally installed by the old MSYS monolithic installers.

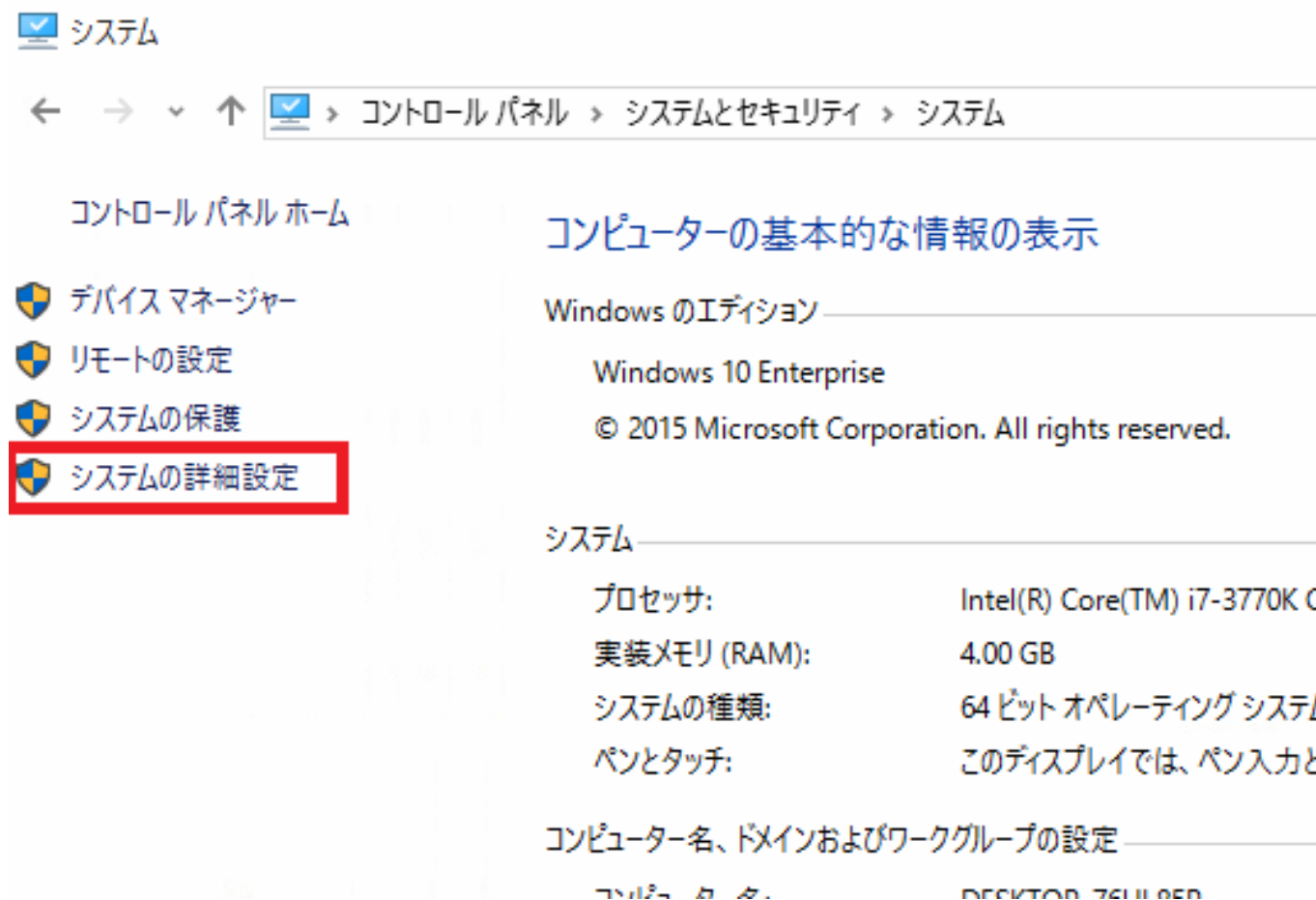
## PATH環境変数の更新

MinGWのコマンドが実行されるようにPATH環境変数を変更します。

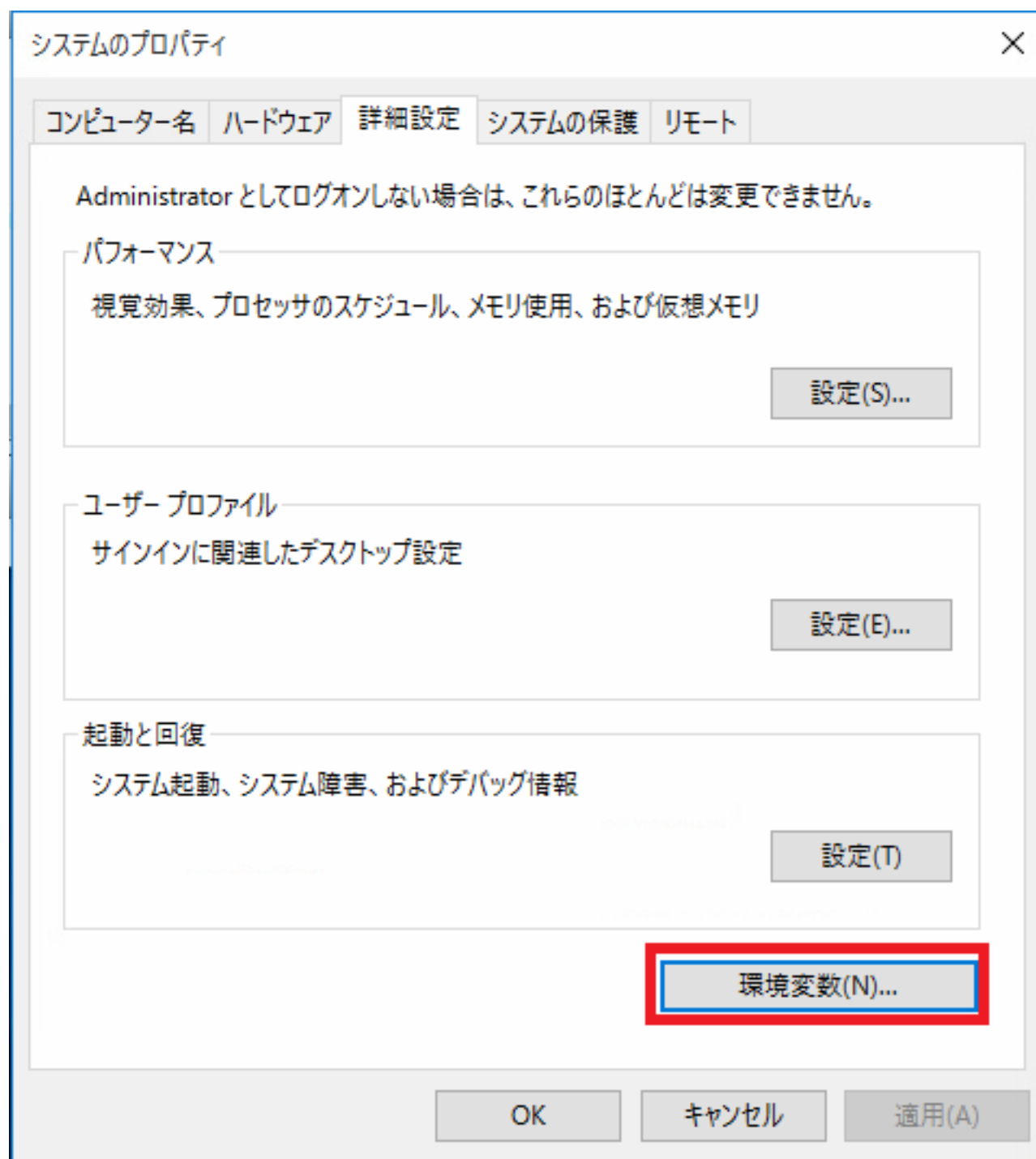
Windowsのスタートメニューの「すべてのアプリ」をクリックし、「Windowsシステムツール」にある「コントロールパネル」をクリックします。

「システムとセキュリティ」の「システム」メニューをクリックします。

以下の画面が表示されますので、「システムの詳細設定」をクリックします。



システムのプロパティが表示されますので、「環境変数...」をクリックします。すると環境変数一覧が表示されますので、システム環境変数のPathをクリックし、「編集」ボタンをクリックします。





以下の画面にて変数値の最後にMinGWのコマンド実行パス「;C:\MinGW\MSYS\1.0\bin\」を追加し、「OK」ボタンで確定します。

システム変数の編集

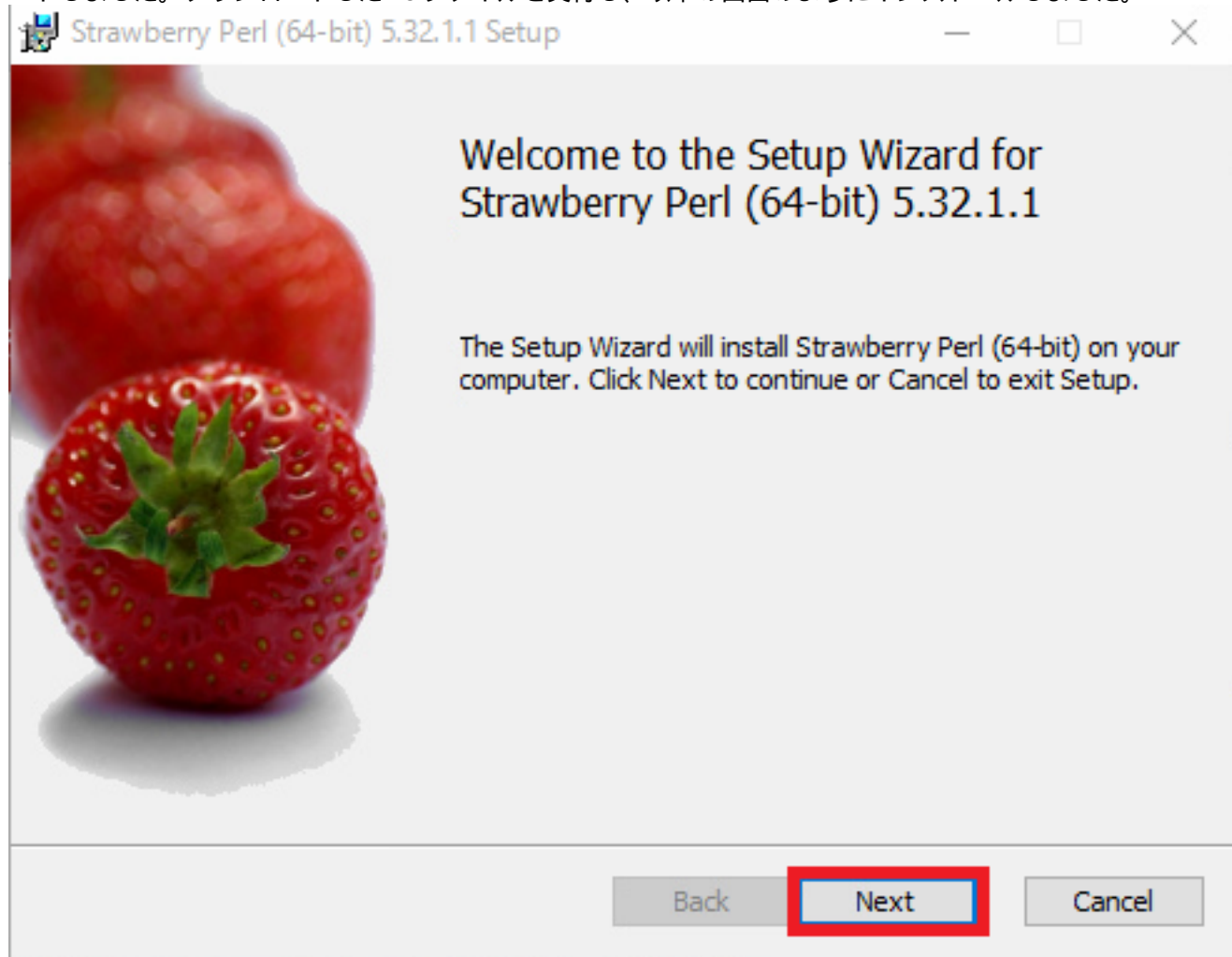
変数名(N): Path

変数値(V): Files (x86)¥Windows Kits¥10¥Windows Performance Toolkit¥;c:¥MinGW¥MSYS¥1.0¥bin¥

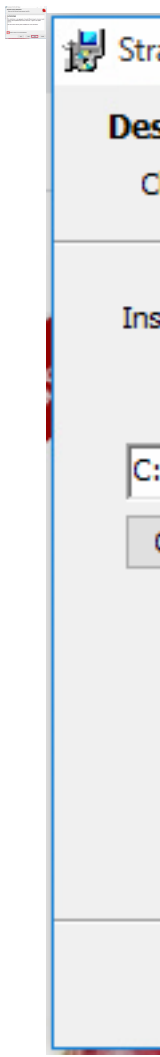
OK キャンセル

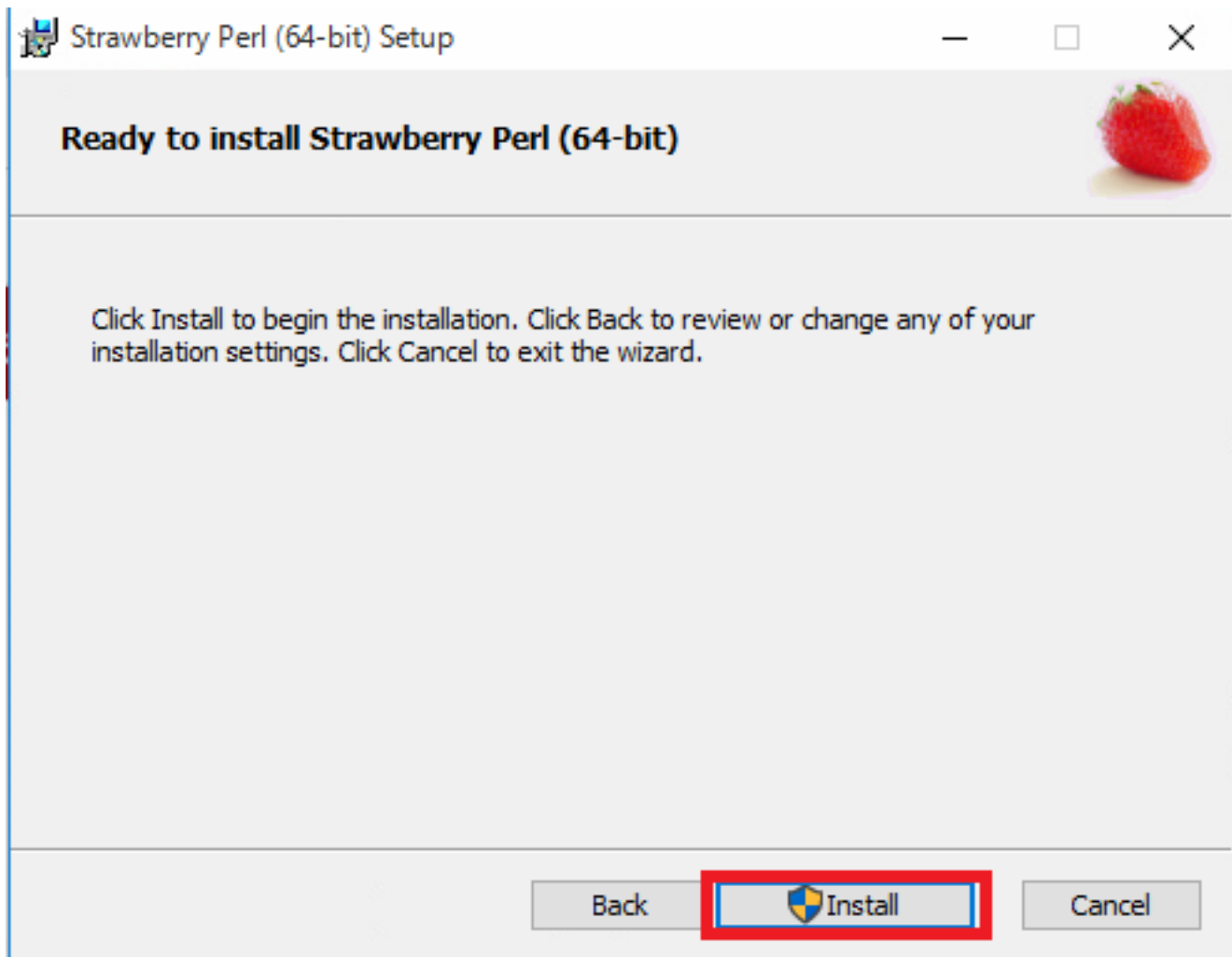
## Strawberry perlのインストール

SSL対応に必要なPerlとして、Strawberry Perl (<https://strawberryperl.com>)をダウンロードします。今回は最新版のStrawberry-perl-5.32.1.1-64bit.msiをダウンロードしました。ダウンロードしたmsiファイルを実行し、以下の画面のようにインストールしました。





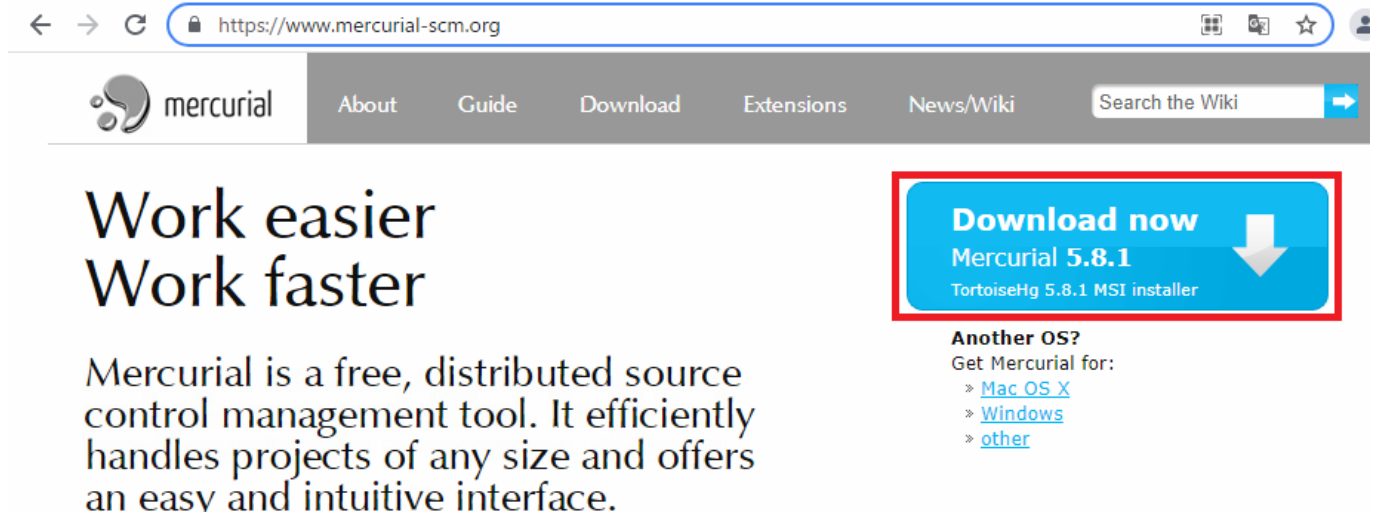




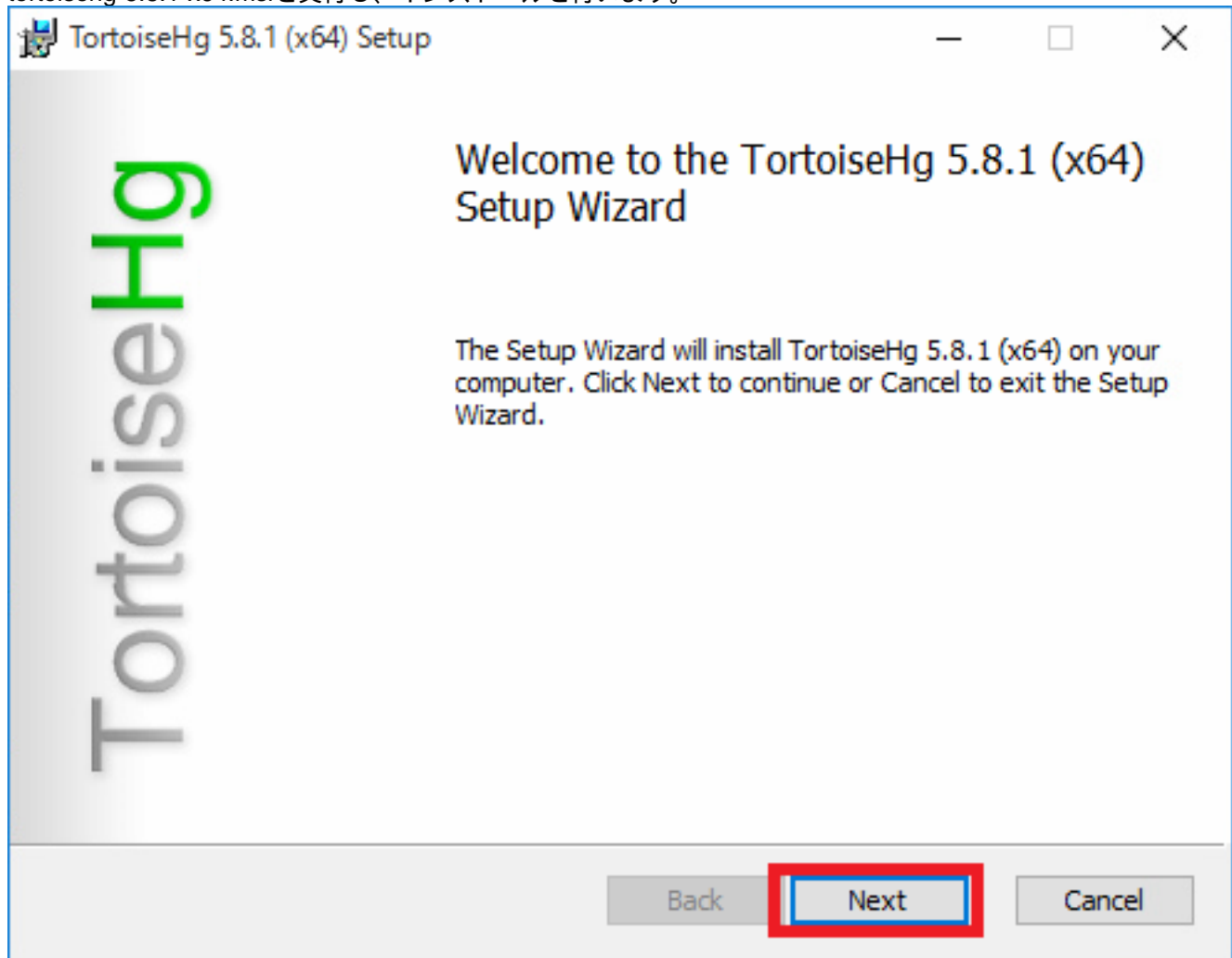


## Mercurialクライアントのインストール

Mercurial(TortoiseHg) 5.8.1をサイト(<https://www.mercurial-scm.org/>)からダウンロードします。  
以下の画面の赤枠部分をクリックします。



tortoisehg-5.8.1-x64.msiを実行し、インストールを行います。





TortoiseHg 5.8.1 (x64) Setup



### Custom Setup

Select the way you want features to be installed.



Click the icons in the tree below to change the way features will be installed.

- TortoiseHg
  - Shell Extension x64
  - Shell Extension x86
  - Translations
  - Documentation
  - KDiff3
  - SSH Utils
  - Miscellaneous

The complete package

This feature requires 0KB on your hard drive. It has 6 of 7 subfeatures selected. The subfeatures require 83MB on your hard drive.

Location: C:\Program Files\TortoiseHg\

Browse...

Reset

Disk Usage

Back

Next

Cancel



## ライブラリ(PCRE、zlib、OpneSSL)のダウンロード

PCREはダウンロードサイト( <https://ftp.pcre.org/pub/pcre/pcre-8.45.tar.gz> )からダウンロードします。

zlibは <http://zlib.net/zlib-1.2.11.tar.gz> からダウンロードします。

Open SSLは <https://www.openssl.org/source/openssl-1.1.1k.tar.gz> からダウンロードします。

これらのgzipファイルはドキュメントフォルダのnginxdevフォルダに移動しておきます。

## ビルド環境用フォルダの作成とソースの配置

ビルドに必要なソフトウェアはインストールできましたので、構築環境を作成していきたいと思います。

まず、ドキュメントフォルダ下にnginxdevフォルダを作成します。

Windowsスタートメニューから「すべてのアプリ」「Visual studio 2017 フォルダ」「VS 2017用 x64 Native Toolsコマンドプロンプト」メニューを選択し、コマンドプロンプトを起動します。

bashを起動します。

```
> c:\MinGW\MSYS\1.0\bin\bash
```

nginxdevフォルダに移動します。

```
$ cd $USERPROFILE/Documents/nginx_dev
```

nginxソースをダウンロード

Mercurialを使用してnginxのサイトからソースコードをダウンロードします。

```
$ hg clone http://hg.nginx.org/nginx
```

ライブラリ格納用のフォルダを作成します。

```
$ mkdir objs
$ mkdir objs/lib
$ cd objs/lib
```

ライブラリの解凍

```
$ tar -xzf ../../../../pcre-8.44.tar.gz
$ tar -xzf ../../../../zlib-1.2.11.tar.gz
$ tar -xzf ../../../../openssl-1.1.1k.tar.gz
```

CSPのソースファイルをコピー

libフォルダ配下にcspフォルダを作成し、IRISに含まれるWeb Gatewayコンポーネントのソースファイルをcspフォルダにコピーします。

```
$ mkdir csp
$ cp /c/InterSystems/IRISHealth/dev/csp/nginx/*.ch csp
```

cspフォルダにconfigファイルを作成

```
$ cd csp
$ cat > config
ngx_addon_name=ngx_http_csp_module_sa
HTTP_MODULES="$HTTP_MODULES ngx_http_csp_module_sa"
NGX_ADDON_SRCS="$NGX_ADDON_SRCS $ngx_addon_dir/nginx_http_csp_module_sa.c"
```

以上で準備は完了です。

## ソースのビルド

configureを実行し、ビルドに必要なファイルを生成します。

```
$ cd ../../../../
$ auto/configure \
  --with-cc=cl \
  --with-debug \
  --prefix= \
  --conf-path=conf/nginx.conf \
  --pid-path=logs/nginx.pid \
  --http-log-path=logs/access.log \
  --error-log-path=logs/error.log \
  --sbin-path=nginx.exe \
```

```
--http-client-body-temp-path=temp/client_body_temp \  
--http-proxy-temp-path=temp/proxy_temp \  
--http-fastcgi-temp-path=temp/fastcgi_temp \  
--http-scgi-temp-path=temp/scgi_temp \  
--http-uwsgi-temp-path=temp/uwsgi_temp \  
--with-cc-opt=-DFD_SETSIZE=1024 \  
--with-pcre=objs/lib/pcre-8.45 \  
--with-zlib=objs/lib/zlib-1.2.11 \  
--with-openssl=objs/lib/openssl-1.1.1k \  
--with-openssl-opt=no-asm \  
--with-http_ssl_module \  
--with-http_auth_request_module \  
--add-module=objs/lib/csp
```

nmakeでビルドを行います。

```
$ nmake -f objs/Makefile
```

これでobjsフォルダに nginx.exe が作成されます。

## nginxのインストール

nginxのインストールはnginxダウンロードサイト(<https://nginx.org/en/download.html>)からzipファイルをダウンロードします。zip ファイルを c:/に展開すると「nginx-x.x.x」というフォルダが作成されます。

c:/nginx-x.x.xフォルダにある nginx.exe を先ほど作成した nginx.exe で上書きします。

## Universal Web gateway moduleの設定

をを作成し、C:/InterSystems/IRISHealth/binフォルダにあるCSPx.dll、CSPxSYS.dllをコピーします。C:/InterSystems/IRISHealth/CSP/binフォルダにもCSPx.dllはありますが、これは32bitバージョンのdllですのでご注意ください。

また、C:/InterSystems/IRISHealth/CSP/binフォルダのCSP.iniをc:/nginx-x.x.x/cspフォルダにコピーします。

コピーが終わりますと、C:/nginx-x.x.x/confフォルダにある、nginx.confを編集します。以下のように http {} 設定ブロックの中に以下のようにCSPModulePathを追加します。

```
:  
#gzip on;  
  
CSPModulePath csp/;  
  
server {  
:  
}
```

また、server {} 設定ブロックには以下のセクションを追加します。

```
server {  
    listen      80;  
    server_name localhost;  
:  
}
```

```
location / {
    root    html;
    index  index.html index.htm;
}

# ??????????
location /csp {
    CSPFileTypes  csp cls zen cxw;
}
:
```

以上でnginx.confの編集は終了です。  
nginxを起動します。

```
> start c:\nginx-x.x.x\nginx
```

起動に成功すると、以下の画面が表示されますので、アクセスを許可したいネットワークを選択し、「アクセスを許可する」ボタンをクリックします。

The screenshot shows a Windows Security warning dialog titled "Windows セキュリティの重要な警告". The main message is "このアプリの機能のいくつかは Windows Defender ファイアウォールでブロックされています". Below this, it states "すべてのパブリック ネットワークとプライベート ネットワークで、Windows Defender ファイアウォールにより nginx の機能のいくつかがブロックされています".

	名前(N):	nginx
	発行元(P):	不明
	パス(H):	C:\nginx\nginx.exe

nginx にこれらのネットワーク上での通信を許可する:

- プライベート ネットワーク (ホーム ネットワークや社内ネットワークなど)(R)
- (パブリック ネットワーク (空港、喫茶店など) (非推奨))(U)  
(このようなネットワークは多くの場合、セキュリティが低いかセキュリティが設定されていません)

[アプリにファイアウォールの経由を許可することの危険性の詳細](#)

At the bottom, there are two buttons: "アクセスを許可する(A)" (highlighted with a red box) and "キャンセル".

以上で、nginxからCSPアクセスできるようになりました。  
手順が多いですが、ぜひチャレンジしてみてください。

また、ご意見、ご質問等ございましたら、お気軽にコメント欄にお寄せください。  
よろしくお願いいたします。

[#REST API](#) [#Web Gateway](#) [#InterSystems IRIS](#) [#InterSystems IRIS for Health](#)



<https://jp.community.intersystems.com/post/windows%E7%92%B0%E5%A2%83%E3%81%A7nginx%E3%82%92%E6%A7%8B%E7%AF%89%E3%81%99%E3%82%8B>