- 0. 前提条件
- 当手順は、Windowsへopensource COBOL環境を構築する手順である。 手順確認時のOSと時期は以下の通りである。
- OS
- Windows 8.1 (64bit)
- ・時期 2010
- 2018/09
- 1. 構築手順
- 1. 1 Visual Stdio のインストール 以下のHPからVisual Studio Community 2017 をダウンロードする。 https://www.microsoft.com/ja-jp/dev/campaign/free-edition.aspx
- 1. 2 C++のコンパイラのインストール
- (1) Visual Studioを立ち上げ、C++のコンパイラとC++のヘッダをダウンロードする。



# (2) cl. exeの存在を確認する。



□ピー 貼り付け ■ピート貼り付け 記 ショートカットの影	移動先 コピー先 削除 名前の 変更	) 新しい プロパティ フォルダー	<ul> <li>浸編集 部 選択</li> <li>№ 履歴 部 選択</li> </ul>	解除 の切り替え
クリップボード	整理	新規 目	1 道	訳
🛞 🄄 🔻 🕇 📕 « VC 🕨 T	ools → MSVC → 14.15.26726 → bin	> Hostx64 > x64	~ Ċ	x64の検索 🔎
🍌 添付ファイル 🔷	名前	更新日時	種類	サイズ ^
	1041	2018/09/15 16:32	ファイル フォルダー	
	🚳 atlprov.dll	2018/09/15 16:19	アプリケーション拡張	214 KB
	💷 bscmake.exe	2018/09/15 16:19	アプリケーション	104 KB
え ホールガループ	🚳 c1.dll	2018/09/15 16:19	アプリケーション拡張	2,388 KB
	🚳 c1xx.dll	2018/09/15 16:19	アプリケーション拡張	8,027 KB
	🚳 c2.dll	2018/09/15 16:19	アプリケーション拡張	7,097 KB
PC	Cl.exe	2018/09/15 16:19	アプリケーション	419 KB
ダウンロード	Ŷ_] cl.exe.config	2018/09/15 16:19	XML 構成ファイル	1 KB
▶ デスクトップ	ScppCoreCheck.dll	2018/09/15 16:19	アプリケーション拡張	600 KB
	💷 cvtres.exe	2018/09/15 16:19	アプリケーション	54 KB
🍺 ピクチャ	land d3dcompiler_47.dll	2018/09/15 16:19	アプリケーション拡張	4,364 KB
🍺 ビデオ		2018/09/15 16:19	アノリケーション払張	1,349 KB
] ミュージック	adithin and	2018/09/15 16:19	アプリケーション	29 KB
L TI31289800A (C:)	editoin.exe	2018/09/15 16:19	アプリケーション	29 KB
🚥 CD ドライブ (D:) Spei 🗸 🗸	💌 Espitengine.dii	2018/09/15 10:19	アプリグーンヨン北方長	2,380 KB V
48 個の項目 1 個の項目を選択	418 KB			:==

(3) ヘッダファイル「stdio.h」のパスを確認しておく



1. 3 MPIRのインストール

下記HPよりMPIR 3.0.0 source zip をダウンロードし、7zipで解凍する。 http://mpir.org/

- 1. 4 Windows8.1 SDKのインストール
- (1) mpir.slnをVisual Studio 2017 で開く。



		🖻 mpir.sln	2017/02/17 13:18	SLN ファイル	6 KB	
	E ビクチャ	mpir toctc cln     通柄・SLN ファイル	2017/02/13 19:19	SLN ファイル	128 KB	
		回 n サイズ: 5.59 KB	2017/02/01 0:36	SLN ファイル	5 KB	
		🚳 n 更新日時: 2017/02/17 13:18	2017/02/23 1:18	Windows バッチ フ	2 KB	
-		run-speed.py	2017/02/22 1:22	PY ファイル	11 KB	
	- CD F517 (D:) Spee	version_info.py	2017/02/01 0:36	PY ファイル	1 KB	¥
	~	<			>	
	22 個の項目 1 個の項目を選	択 5.59 KB				

(2) 下記手順でWindows 8.1 SDKをインストールする。

手順1:VisualStudio 2017 のメニューから『ファイル』→『新規作成』→『プロジェクト』を選択する

手順2:『新しいプロジェクト』のダイアログが開くので、その左側の『Visual Studio インストーラーを開く』をクリック

手順3:最初は『ワークロード』タブが選択されているので、『個別のコンポーネント』を選択する。

手順4:表示されるリストの中から『Windows 8.1 SDK』にチェックを付けて右下の『変更』ボタンをクリックする。

※この時、Visual Studio 2017 は終了しておく

1. 5 lib\_mpir\_gcのビルド

(1) ビルドパスの設定で、1.1(3) で確認したヘッダファイル「stdio.h」のパスを追加する。



	インクルード ディレクトリ VC++ プロジェクトのビルド中、インクルード ファイルを検索するときに使用されるパスです。環境変数 INCLUDE と一致します。
	OK キャンセル 適用(A)

(2)64bitを選択し、releaseでlib\_mpir\_gcをビルドし、mpir.libを生成する。 「Release」「x64」「lib\_mpir\_gc」を選択。

刘 mpir - Microsoft Visual Studio	<ul><li>マ P</li><li>クイック起動 (Ctrl+Q)</li><li>P</li></ul>
ファイル(E) 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(I) テスト(S) 分析(N) ウィンドウ(W) ヘルプ(H)	
💿 - 💿   🎦 - 🖕 💾 💾 🏸 - 🤍 - Release - x64 トローカル Windows デバッガー -   🎜 😑	
े इ	
	ソリューション エクスプローラー の検索 (Ctrl+:)
	MJ ソリューション 'mpir' (11 プロジェクト)
	▶ the dll_mpir_core2
	Implify the swell avx
	Mail Miler Market Ma
	Market
	▶ the lib_mpir_core2
	▶ N lib mpir ac
	Ib_mpir_haswell_avx
	▶ 强 lib_mpir_p3
	Iib_mpir_skylake_avx
	ソリューション エクスプローラー チーム エクスプローラー
	プロパティ
	lib_mpir_gc プロジェクトのプロパティ
	□ その他
	(名前) lib_mpir_gc
	プロジェクトの放在関係
	ルート名前空間 mpir
	(名前)
	プロジェクト名を指定します。
□ この項目はプレビューをサポートしていません	<b>↑</b> ע–ג
刘 mpir - Microsoft Visual Studio	マ P クイック起動 (Ctrl+Q)
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)	
○ - ○ 裕 - 二 単 単 ウ - ペ - I 感 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 エクスプローラー
「 」 シリューションのクリーン(C)	
シリューションの完全なプログラム データベース ファイルを構築する	
クリューションでコード分析を実行(Y) Alt+F11	991-937 1927日-9- の検系 (Uti+:)
ib_mpir_gc のビルド(U)	ig」 タリューション mpir (11 ノロシェクト) ト 国家 dll mpir core2
lib_mpir_gc のリビルド(E)	▶ The dil_mpir_gc
lib_mpir_gc のクリーン(N)	Maintoine Mai
lib_mpir_gc でコード分析を実行(A)	Image: Second
プロジェクトのみ(J) ト	P
バッチ ビルド(T)	▶ The information is the information of the inf
構成マネージャー(0)	▶ 💽 lib_mpir_gc
	Ibb_mpir_haswell_avx
	▷ 1 lib_mpir_p3



下記の表示が出ればビルド成功。 ======= ビルド∶1 正常終了、0 失敗、0 更新不要、0 スキップ =========

mpir-3.0.0¥build.vc15¥lib\_mpir\_gc¥x64¥Release に下記ファイルが出力されているので、退避する。 gmp.h mpir.lib

# (3) mpir.dllを生成するため、プロパティページの構成の種類を変更し、リビルドする。 ※この手順を行うとmpir.libが生成されないので、mpir.libは先に退避する

# 構成の種類をダイナミックライブラリに変更する。

	lib_mpir_gc プロ/	パティページ ? ×
構成( <u>C</u> ): アクティブ(Release)	✓ プラットフォーム(P): アクティブ(x64)	✓ 構成マネージャー( <u>○</u> )
▲ 構成プロパティ	▲ プロジェクトの既定値	
全般	構成の種類	ダイナミック ライブラリ (.dll)
デバッグ	MFC の使用	標準 Windows ライブラリを使用する
VC++ ディレクトリ	文字セット	
▷ C/C++	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
▶ ライブラリアン	.NET Framework 対象バージョン	
▷ XML ドキュメント ジェネレータ	プログラム全体の最適化	プログラム全体の最適化なし
▶ フラウサー 情報	Windows ストア アプリのサポート	いいえ
	▲ 全般	
	ターゲット プラットフォーム	Windows
	Windows SDK バージョン	8.1
	出力ディレクトリ	\$(Platform)¥\$(Configuration)¥
	中間ディレクトリ	\$(Platform)¥\$(Configuration)¥
	ターゲット名	mpir
	ターゲットの拡張子	.lib
	クリーン時に削除する拡張子	*.cdf;*.cache;*.obj;*.obj.enc;*.ilk;*.ipdb;*.iobj;*.resources;*.tlb;*
	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 2017 (v141)
	マネージド インクリメンタル ビルドを有効にする	いいえ
	構成の種類	
	この増加が生成する出力の種類を指定します。	
< >>		
		OK キャンセル 適用( <u>A</u> )

# 「~ucrt¥x64」の下にあるlibucrt.libを見つけ、ライブラリディレクトリにパスを追加する。

	libucrt.libのプロパティ	×	梢	食素ツール	libucrt - 検索場所	f: Progra	m Files (>	(86) -	. 🗆	×
全般 セキュ!	Jティ 詳細	7	ŧ	検索			-			^ ?
	libucrt.lib	部	「」 新日	<sup>1</sup> 分類 ▼ <sup>1</sup> サイズ ▼ <sup>1</sup> その他のプロパティ ▼	<ul> <li>● 最近の検索内容</li> <li>● 詳細オプション、</li> <li>● 検索条件を保存</li> </ul>	ファイルの 場所を開く	) 検索結果 を閉じる			
ファイルの種類:	: オブジェクト ファイル ライブラリ (.lib)			絞り込み	オプション					
プログラム:	不明なアプリケーション	を更(C)	Progra	ram Files (x86)				✓ 🖒 libu	crt	×
				C:¥Program Files (x	86)¥Windows Kits¥1	0¥I ib¥10 0	実利口の サイブ・1	त. 2010/04/20 4.3 53 0 MB	•	^
場所:	36)¥Windows Kits¥10¥Lib¥10.0.17134.	0¥ucrt¥x64		C.+riogram nics (x		0+210+10.0.		+		
サイズ:	62.1 MB (65,220,492 バイト) C:¥Pi	rogram Files (x86)	)¥Win	dows Kits¥10¥Lib¥10	.0.17134.0¥ucrt¥x64	∔ o≠Lib¥10.0.	更新日間 サイズ:(	守: 2018/04/20 4:34 51.6 MB	Ŧ	
ディスク上 のサイズ:	62.1 MB (65,220,608 バイト)			<mark>libucrt</mark> d.lib C:¥Program Files (x8	86)¥Windows Kits¥1	0¥Lib¥10.0.	更新日8 サイズ: (	寺: 2018/04/20 2:10 54.0 MB	C	
作成日時:	2018年4月19日、11:10:28			libucrt.lib			更新日日	寺: 2018/04/20 2:0	9	
更新日時:	2018年4月19日、11:10:28			C:¥Program Files (x	86)¥Windows Kits¥1	0¥Lib¥10.0.	サイズ: (	51.5 MB		
アクセス日時:	2018年9月17日、17:25:15	i i		<mark>libucrt</mark> d.lib C:¥Program Files (x8	86)¥Windows Kits¥1	0¥Lib¥10.0.	更新日8 サイズ: (	寺: 2018/04/19 22: 50.8 MB	37	





リビルドを行う。



mpir-3.0.0¥build.vc15¥lib\_mpir\_gc¥x64¥Release に下記ファイルが出力されているので、退避する。 mpir.dll

1. 6 PDcursesのインストール・ビルド

(1) HPから pdcurs36.zip をダウンロードし、7zipで解凍する。 https://pdcurses.sourceforge.io/

(2) VS 2017 用 x64 Native Tools コマンド プロンプトを開き、コンパイルする。 VS 2017 用 x64 Native Tools コマンド プロンプトを開く



pdcurs36¥wincon へcdコマンドで移動し下記コマンドを入力。 set PLATFORM=X64 nmake -f Makefile.vc WIDE=Y DLL=Y

コンパイルに成功したら、 pdcurs36 の下記ファイルを退避する。 curses.h curspriv.h

また、 pdcurs36¥wincon の下記ファイルを退避する。 pdcurses.dll pdcurses. lib

1. 7 opensource COBOLのインストール・ビルド (1) HPよりopensource-cobol-1.5.1J.zip をダウンロードし、7zipで解凍する。 http://www.osscons.jp/osscobol/download/

(2) 退避したファイルの配置

(1)で解凍したフォルダのwin32ディレクトリに移動し、includeフォルダ、libフォルダ、share¥configフォルダを作成する。

その後、includeフォルダに退避した.hファイルを、libフォルダに退避した.libファイルや.dllファイルを配置する。

(3) defaults.hの編集

win32ディレクトリに存在するdefaults.hを編集する。 この情報は生成されるexeファイルに書き込まれ、exeファイル実行時の環境設定となる(重要)。

COB_CFLAGS	includeフォルダを-Iで指定
COB_LIBS	ibフォルダ¥ ibcob. ibを指定
COB_CONFIG_DIR	configフォルダを指定

C:¥Program Files (x86)¥opensource-cobol-1.5.1J¥win32¥defaults.h - sakura 2.2.0.1	
ファイル(E) 編集(E) 変換(C) 検索(S) ツール(I) 設定(Q) ウィンドウ(W) ヘルプ(H)	
- C 🔊 ▼ 🗖 🚰 🗠 ∾   ≪  ↔   🗞 🂝 💩 🐝 🌦 🔩 📢 🎆 🖏 🗱 🗎	
Q	
5行 28 桁 LF 3A SJIS REC 挿入	:

(4) opencobol.slnをVisual Studioで開く
 その際、ソリューション操作の再ターゲットのポップアップが出てきたら実施する

ソリューション操作の再ターゲット
プロジェクトの再ターゲット
次のプロジェクトは、以前のバージョンの Visual C++ プラットフォーム ツールセットを使用しています。これらの プロジェクトは、最新の Microsoft ツールセットをターゲットとするようにアップグレードできます。また、お使いのマ シンにインストールされているものからターゲットの Windows SDK バージョンを選択することもできます。
Windows SDK バージョン: 10.0.17134.0 、
プラットフォーム ツールセット: v141 へのアップグレード v
✓¥libcob¥libcob.vcxproj
✓¥vbisam¥vbisam.vcxproj
✓¥cobc¥cobc.vcxproj
✓¥cobcrun¥cobcrun.vcxproj
<ul> <li>OK ≠ャンセル</li> </ul>

(5)一部ヘッダファイルのルートディレクトリへの配置 gmp.h、curses.h、curspriv.hをOpenCobolのルートディレクトリに配置。

\_ 🗆 🗙 🍌 l ⊋ 🚺 👳 l opensource-cobol-1.5.1J ^ ? ファイル ホーム 共有 表示 🖺 開く 🔹 🔡 すべて選択 **₽** þ 👗 切り取り ▲ アロパティ → 福集 3:3 選択解除 **\_**  $\mathbf{\Lambda}$ 🐜 パスのコピー コピー 貼り付け 移動先 コピー先 削除 名前の 新しい 📲 選択の切り替え 🕢 履歴 🖻 ショートカットの貼り付け 変更 フォルダー

クリップボード	整理	新規目		沢
( → ↑ ) ≪ TI3128	89800A (C:) → Program Files (x86) → c	pensource-cobol-1.5.1J	~ Č	opensou 🔎
🍶 添付ファイル 🔷	名前	更新日時	種類	サイズ ^
山白	COPYING.DOC	2016/12/09 10:26	Microsoft Word	20 KB
●●● 白青黒	COPYING.LIB	2016/12/09 10:26	オブジェクト ファイル	26 KB
無 (1) (1) (1) (1) (1) (1) (1) (1) (1) (1)	📋 cpucheck.c	2016/12/09 10:26	C ファイル	2 KB
	depcomp	2016/12/09 10:26	ファイル	16 KB
📢 ホームグループ	🖺 gmp.h	2018/09/17 17:05	Η ファイル	96 KB
	INSTALL	2016/12/09 10:26	ファイル	10 KB
	📄 install-sh	2016/12/09 10:26	ファイル	10 KB
PC	🖺 libcob.h	2016/12/09 10:26	H ファイル	2 KB
	📄 ltmain.sh	2016/12/09 10:26	SH ファイル	195 KB
	Makefile.am	2016/12/09 10:26	AM ファイル	3 KB
	Makefile.in	2016/12/09 10:26	IN ファイル	25 KB
	i missing	2016/12/09 10:26	ファイル	11 KB
	mkinstalldirs	2016/12/09 10:26	ファイル	4 KB
	NEWS	2016/12/09 10:26	ファイル	23 KB
CD K= (T (D)) Creat	README	2016/12/09 10:26	ファイル	8 KB 🗸
	<			>
42 個の項目 1 個の項目を選	択 95.1 KB			!=== <b>E</b>

# (6)プロパティで(2)で作成したlibフォルダへのパスを通しておく

ライブラリ ディレ	クトリ	?	x
	*	<b>X V</b>	Ť
C:¥Program Files (x86)¥opensource-cobol- %LIB%	1.5.1J¥win32¥lib		Ŷ
<		>	
評価された値:			
C:¥Program Files (x86)¥opensource-cobol-	1.5.1J¥win32¥lib		Û
<		2	
継承の値:			
\$(VC_LibraryPath_x64) \$(WindowsSDK_LibraryPath_x64)			^
\$(WINDOWSSER_LIBRALYPach_XO4)			~
□親またはプロジェクトの既定値から継承(1)		マクロ( <u>M</u> )>	>
	ОК	キャンセル	,

# (7)プロパティでkernel32.libへのパスを通す kernel32.libが入っているディレクトリを探す。ディレクトリは「~um¥x64」である必要がある。

🔐   ⊋ 🕕 =	хб4	1		- 🗆 🗙
ファイル ホーム 共有 装	表示			^ <b>(</b> )
「「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     ショートカットの     クリップボード			<ul> <li>□ 編集</li> <li>□ 編集</li> <li>□ 編集</li> <li>□ 2 μ</li>     &lt;</ul>	催択 3除 切り替え R
€ 🦻 ▾ ↑ 📙 « Windo	ws Kits → 10 → Lib → 10.0.17134.0	→ um → x64	v ¢	x64の検索 🔎
ふ付ファイル	名前	更新日時	種類	サイズ ^
🎍 白	🖩 kerbcli.lib	2018/04/20 4:22	オブジェクト ファイル	100 KB
🔰 白青黒	🖩 kernel32.Lib	2018/04/20 3:37	オブジェクト ファイル	291 KB
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	🏢 kernel32 種類: オブジェクト ファイル	ライブラリ 2018/04/20 4:19	オブジェクト ファイル	5,014 KB
	III keycredn サイズ: 290 KB 亜新日時: 2018/04/20	2018/04/19 11:10	オブジェクト ファイル	3 KB
🜏 ホームグループ	III KSProxy.LTD	2018/04/20 3:37	オブジェクト ファイル	3 KB
	🔠 ksuser.lib	2018/04/20 4:22	オブジェクト ファイル	340 KB
	👪 ktmw32.lib	2018/04/19 11:01	オブジェクト ファイル	12 KB
PC	🔠 LoadPerf.Lib	2018/04/20 3:45	オブジェクト ファイル	5 KB
🍌 ダウンロード	locationapi.lib	2018/04/20 4:22	オブジェクト ファイル	198 KB
▶ デスクトップ	Lz32.Lib	2018/04/19 11:14	オブジェクト ファイル	4 KB
1/2×2=7	magnification.lib	2018/04/20 3:38	オブジェクト ファイル	7 KB
	MAPI32.Lib	2018/04/20 3:47	オブジェクト ファイル	28 KB
	mbnapi.tlb	2018/04/19 10:42	TLB ファイル	66 KB
<u> い</u> ミュージック	🏙 mbnapi_uuid.lib	2018/04/19 11:02	オブジェクト ファイル	19 KB
TI31289800A (C:)	🛄 mciole32.lib	2018/04/20 3:48	オブジェクト ファイル	4 KB 🗸
CD Not (D:) Spee	<			>
465 個の項目 1 個の項目を選	択 290 KB			!== <b></b>

上記ディレクトリへのパスを通す。

ライブラリ ディレクトリ ?	×
🎽 🗙 🗸	1
Program Files (x86)¥Windows Kits¥10¥Lib¥10.0.17134.0¥um¥x64	<b>^</b>
<	>
評価された値:	
C:¥Program Files (x86)¥opensource-cobol-1.5.1J¥win32¥lib	\$
<	>
継承の値:	
\$(VC_LibraryPath_x64) \$(WindowsSDK_LibraryPath_x64)	^
	~
親またはプロジェクトの既定値から継承(I) マクロ(M)	>>
OK キャンセ	ιL

(8)プロパティでmsvcrt.lib、ucrt.libへのパスを通す (7)同様の手順でmsvcrt.libへのパスを追加する。ディレクトリは「~onecore¥x64」である必要がある。

msvcrt.libのプロパティ	×
------------------	---

全般 セキュリティ 詳細						
	msvcrt.lib					
ファイルの種類:	オブジェクト ファイル ライブラリ (.lib)					
プログラム:	不明なアプリケーション 変更( <u>C</u> )					
場所:	y¥VC¥Tools¥MSVC¥14.15.26726¥lib¥onecore¥x64					
サイズ:	3.71 MB (3,898,764 /ርイト)					
ディスク上 のサイズ:	3.71 MB (3,899,392 バイト)					
作成日時:	2018年9月15日、16:31:51					
更新日時:	2018年9月15日、16:19:06					
アクセス日時:	2018年9月15日、16:31:51					
属性: □読み取り専用( <u>R</u> ) □ 隠しファイル( <u>H</u> ) 詳細設定( <u>D</u> )						
	OK キャンセル 適用(A)					

(7) 同様の手順でucrt. libへのパスを追加する。ディレクトリは「~ucrt¥x64」である必要がある。

alla	ucrt.libのプロパティ					
全般 セキュリティ 詳細						
	ucrt.lib					
ファイルの種類:	オブジェクト ファイル ライブラリ (.lib)					
プログラム:	不明なアプリケーション 変更( <u>C</u> )					
場所:	36)¥Windows Kits¥10¥Lib¥10.0.17134.0¥ <mark>ucrt¥x64</mark>					
サイズ:	279 KB (285,764 /ናイト)					
ディスク上 のサイズ:	280 KB (286,720 /(イト)					
作成日時:	2018年4月19日、11:02:04					
更新日時:	2018年4月19日、11:02:04					
アクセス日時:	2018年9月17日、16:03:33					
属性: □読み取り専用( <u>R</u> ) □隠しファイル( <u>H</u> ) 詳細設定( <u>D</u> )						
	OK キャンセル 適用(A)	]				

(9)ビルドの実施releaseのx64とし、libcobを選択し、ソリューションのビルド(リビルド)を選択



下記の表示が出ればビルド成功。 

### 1.8 生成ファイルの配置

任意のディレク	トリにosscobolフォルダを作成し、	下記ファイルを配置する。

フォルダ	ファイル	取得元
osscobol¥hin	cobc. exe	opensource-cobol-1.5.1J¥win32¥BIN¥x64¥Release
	cobcrun. exe	opensource-cobol-1.5.1J¥win32¥BIN¥x64¥Release
osscobol¥include	libcob.h	opensource-cobol-1.5.1J
	curses.h	opensource-cobol-1.5.1J
	curspriv.h	opensource-cobol-1.5.1J
	gmp.h	opensource-cobol-1.5.1J
	*. h	Windows Kits¥(中略)¥ucrt
	*. h	Microsoft Visual Studio¥(中略)¥include
	byteswap.h	opensource-cobol-1.5.1J¥libcob
aaaaba Wina Juda VLi baab	call.h	opensource-cobol-1.5.1J¥libcob
	coblocal.h	opensource-cobol-1.5.1J¥libcob
	codegen. h	opensource-cobol-1.5.1J¥libcob
	common.h	opensource-cobol-1.5.1J¥libcob
	exception.def	opensource-cobol-1.5.1J¥libcob
	fileio.h	opensource-cobol-1.5.1J¥libcob
03300001+11101000+110000	intrinsic.h	opensource-cobol-1.5.1J¥libcob
	move.h	opensource-cobol-1.5.1J¥libcob
	numeric.h	opensource-cobol-1.5.1J¥libcob
	screenio.h	opensource-cobol-1.5.1J¥libcob
	strings.h	opensource-cobol-1.5.1J¥libcob
	system.def	opensource-cobol-1.5.1J¥libcob
	termio.h	opensource-cobol-1.5.1J¥libcob
	vbisam.lib	opensource-cobol-1.5.1J¥win32¥BIN¥x64¥Release
	vbisam.dll	opensource-cobol-1.5.1J¥win32¥BIN¥x64¥Release
	libcob.lib	opensource-cobol-1.5.1J¥win32¥BIN¥x64¥Release
	libcob.dll	opensource-cobol-1.5.1J¥win32¥BIN¥x64¥Release
	mpir.dll	opensource-cobol-1.5.1J¥win32¥lib
osscobol¥lib	mpir.lib	opensource-cobol-1.5.1J¥win32¥lib
	pdcurses.dll	opensource-cobol-1.5.1J¥win32¥lib
	pdcurses.lib	opensource-cobol-1.5.1J¥win32¥lib
	*.lib	Microsoft Visual Studio¥(中略)¥lib¥x64
	*.lib	Windows Kits¥10¥(中略)¥um¥x64
	*.lib	Windows Kits¥10¥(中略)¥ucrt¥x64
	boundary-limit.conf	opensource-cobol-1.5.1J¥config
	bs2000. conf	opensource-cobol-1.5.1J¥config
	ChangeLog	opensource-cobol-1.5.1J¥config
	cobol85.conf	opensource-cobol-1.5.1J¥config
	cobol2002.conf	opensource-cobol-1.5.1J¥config
	default.conf	opensource-cobol-1.5.1J¥config
osscobol¥sbare¥config	default-en. conf	opensource-cobol-1.5.1J¥config
	default-jp.conf	opensource-cobol-1.5.1J¥config
	ibm.conf	opensource-cobol-1.5.1J¥config
	jp-compat.conf	opensource-cobol-1.5.1J¥config
	Makefile.am	opensource-cobol-1.5.1J¥config
	Makefile.in	opensource-cobol-1.5.1J¥config
	mf.conf	opensource-cobol-1.5.1J¥config
	mvs.conf	opensource-cobol-1.5.1J¥config

※上記取得元と名前が被るファイルは上書きしない

※上記取得元と名前が被るファイルは上書きしない

# 1. 9 Windowsのユーザ環境変数の追加

- Windowsのユーザ環境変数に下記のパスをを追加設定する。
  - …osscobol¥bin
  - ····Microsoft Visual Studio¥ (中略) ¥bin¥Hostx64¥x64
  - …Windows Kits¥10¥bin¥(中略)¥x64

 10 パスが通っていることの確認 コマンドプロンプトで下記コマンドを入力し、 cobc --version 下記のようにバージョンが表示されれば構築完了 ※cobc. exeにパスを通す

※cl.exeにパスを通す

※mt.exeにパスを通す

c:¥>cobc --version opensource COBOL 1.5.1J OSS Consortium's patched version of OpenCOBOL1.1(Feb.06 2009) ----

cobc (opensource COBOL) 1.5.1.0 Copyright (C) 2001-2009 Keisuke Nishida / Roger While Built Sep 22 2018 14:12:02 Packaged Feb 06 2009 10:30:55 CET

c:¥>

C:4.

2. 稼働確認

2.1 テストプログラム作成

フォルダ「osscobol¥src¥01\_helloworld」にソースコードを格納する。



コマンドプロンプトでフォルダ「osscobol¥lib」へ移動し、下記コマンドを実行。 cobc -x ...¥src¥01\_helloworld¥COBHELLO.cob

下記のようにエラーなく返ってくればコンパイル成功。 C:¥osscobol¥lib>cobc -x ..¥src¥01\_helloworld¥C0BHELLO.cob cobAB37.c

C:¥osscobol¥lib>

# コンパイルに成功すると、exeファイルが生成される。

👪 l 💽 👪 👳 l	アプリケーション ツール		01_	helloworld		-	×
ファイル ホーム 共有 昇	表示    管理						^ <b>(</b> )
□ピー 貼り付け		★ ① 前の 変更	● 新しい項目・ 予ショートカット・ 新しい フォルダー	プロパティ プロパティ ● 履歴	<ul> <li>■ すべて選択</li> <li>● 選択解除</li> <li>● 選択の切り替え</li> </ul>		
クリップボード	整	理	新規	開く	選択		
🔄 💮 🔻 🕆 퉬 🕨 PC 🕨	TI31289800A (C:) → oss	cobol → src → 0	1_helloworld			∨ C 01_he	<b>م</b> oll
▶ 無 ^	名前		更新日時	種類	サイズ		
3 + 1 5 a f	COBHELLO.cob		2018/09/22 13:50	COB ファイル	1 KB		
₩ <u>₩</u>	COBHELLO.exe		2018/09/22 15:17	アプリケーション	12 KB		
● PC ● ダウンロード ● デスクトップ ● ドキュメント ● ビクチャ ● ビデオ ● ビデオ ● ミュージック ■ TI31289800A (C:) ■ CD ドライブ (D:) Spee							
2 個の項目 1 個の項目を選択	₹ 11.5 КВ						:==

- 2. 3 ロードモジュールの実行 コマンドプロンプトでフォルダ「osscobol¥lib」へ移動し、下記コマンドを実行。 ..¥src¥01\_helloworld¥COBHELLO.exe

下記のように「HELLO WORLD!」が表示されればOK

<b>C31</b>	שלערב אעדב	-	×	
C:¥osscobol¥lib>¥src¥01_hellowo	rld¥COBHELLO.exe		^	
HELLO WORLD!				
C:¥osscobol¥lib>			~	

- 3.参考文献
- ・ 「ある又前へ
   ・ Install\_win ・ opensourcecobol-opensource-cobol Wiki ・ GitHub <u>https://github.com/opensourcecobol/opensource-cobol/wiki/Install win</u>
   ・ Windows上にopensource COBOLの環境を構築してみた 技術とか戦略とか <u>https://akira2kun.hatenablog.com/entry/2018/09/22/164247</u>