

opensource COBOL の Windows 対応について

東京システムハウス株式会社
発行：2015年3月31日

目次

| | | |
|-------|------------------------------------|----|
| 1 | はじめに..... | 1 |
| 1.1 | opensource COBOL とは..... | 1 |
| 1.2 | opensource COBOL の Windows 対応..... | 1 |
| 1.3 | opensource COBOL 動作イメージ..... | 2 |
| 2 | パッケージ概要..... | 3 |
| 2.1 | ディレクトリ構成..... | 3 |
| 3 | 動作環境..... | 4 |
| 4 | インストール..... | 4 |
| 4.1 | ダウンロードサイト..... | 4 |
| 4.2 | インストール環境について..... | 5 |
| 4.3 | パッケージの解凍..... | 5 |
| 4.4 | opensource COBOL v1.4.0J のビルド..... | 6 |
| 5 | コンパイル・実行..... | 10 |
| 5.1 | COBOL ソースファイルのコンパイル..... | 10 |
| 5.1.1 | 準備..... | 10 |
| 5.1.2 | コンパイルの実行..... | 11 |
| 5.2 | COBOL の実行..... | 12 |
| 6 | 性能比較..... | 13 |
| 6.1 | 性能比較環境..... | 13 |
| 6.2 | 演算パフォーマンスの性能比較と結果..... | 13 |
| 7 | 周辺機能..... | 13 |
| 8 | 既知の制約事項と問題点..... | 13 |
| 9 | まとめ..... | 14 |
| 10 | 付録..... | 14 |
| 10.1 | cobc コンパイラのコンパイラオプション例..... | 14 |
| 10.2 | 設定が必要な環境変数例..... | 14 |
| 10.3 | トラブルシューティング..... | 15 |

1 はじめに

1.1 opensource COBOL とは

opensource COBOL とは、日本特有のビジネス環境に即したカスタマイズやメンテナンスを加えた、日本発のオープンソース COBOL コンパイラである。開発の主体は OSS コンソーシアムのオープン COBOL ソリューション部会であり、2012 年に OpenCOBOL 1.1 Pre-release よりフォークされた。

※OpenCOBOL は、日本医師会総合政策研究機構 ORCA プロジェクトで開発が始められ、その原作者は西田圭介氏である

1.2 opensource COBOL の Windows 対応

opensource COBOL はソフトウェアライセンスなどの初期費用が掛からない事や、ソースコードの確認が可能 (=安心) である事、自社の仕様に容易に合わせる事が可能である事などの利点がある。しかし、今後、日本企業の COBOL システムを移行先として活用するには、次の点の課題の改善が必要となっている。

- Linux での利用に限定される。Windows での開発・動作環境が無いため、使用するためには Linux の環境とスキル習得が必要。
- COBOL 標準の CUI 画面の日本語環境でのサポートが不十分である。日本語表示・入力機能と、罫線の表示機能が正式に対応されていない。

今回リリースされた opensource COBOL v1.4.0J は、Visual Studio 2013 による COBOL コンパイラとランタイムのビルドと、CL コンパイラによる COBOL プログラムのコンパイルが正式に対応された。パッケージには、Visual Studio 2013 のソリューションファイルとプロジェクトファイルが同梱されており、Windows ユーザが COBOL プログラムを開発・実行できる様になっている。

更に、COBOL 標準の CUI 画面での日本語表示・入力機能と、罫線の表示機能が正式に対応された事により、汎用機・オフコンの画面を Windows 上で再現する事が可能になった。

ファイルシステムでは、OSS の ISAM File Handler である VBISAM が利用できる。従来は利用者が自分で入手をしてビルドをする必要があったが、opensource COBOL v1.4.0J では opensource COBOL と併せて配布している。

opensource COBOL v1.4.0J は、Visual Studio 2013 で構築されるため、その対応 OS に準じて、Windows のクライアント OS / サーバ OS の両環境に対応している。Windows クライアントのアプリケーションとして配布したり、Windows サーバのアプリケーションとして配置して RemoteApp でシンクライアントのシステムを実現したりすることができる。

1.3 opensource COBOL 動作イメージ

opensource COBOL の動作イメージ図を以下に記載する。v1.4.0J より、Linux だけではなく、Windows での利用が可能となった。

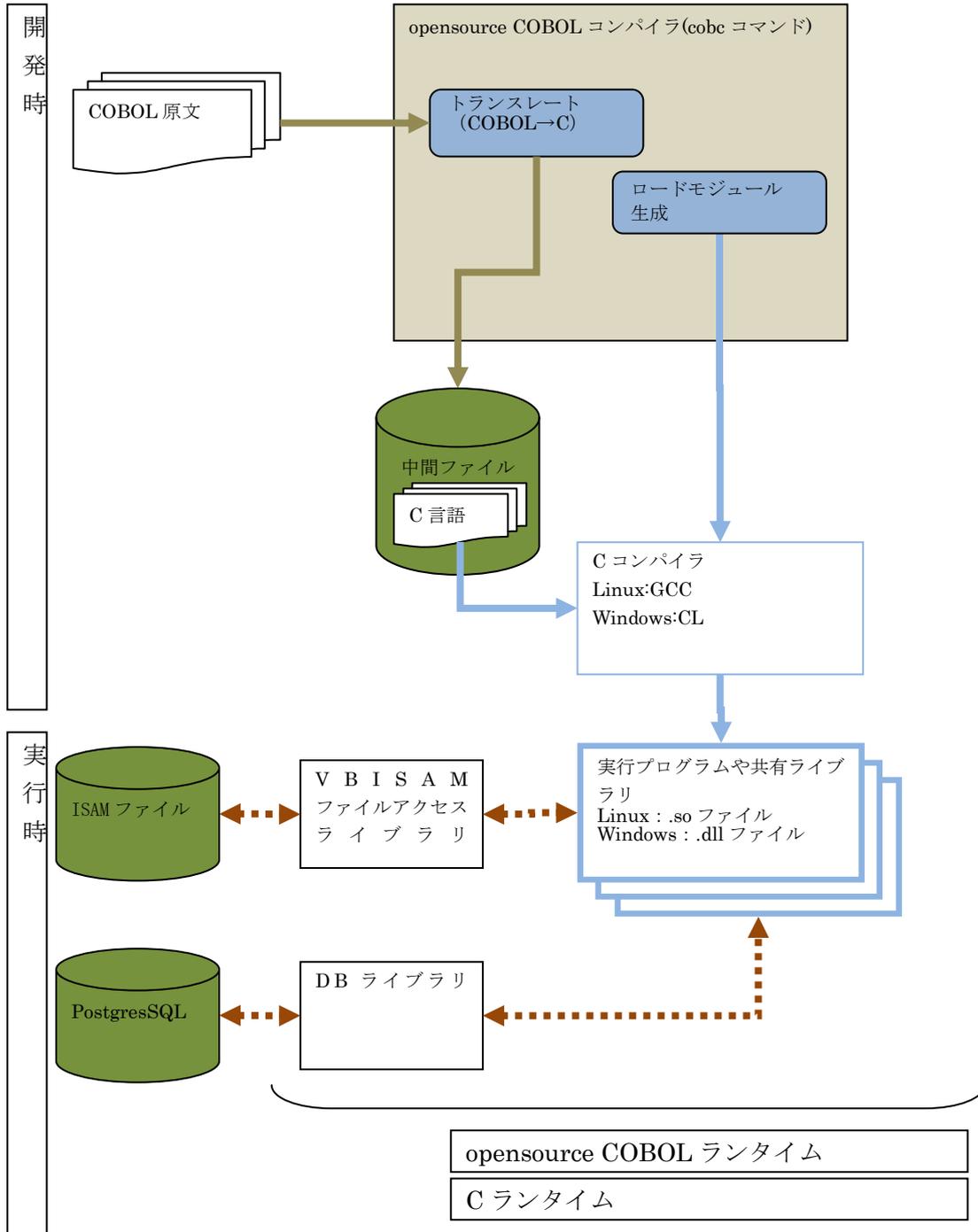


図 1.3-1 opensource COBOL の動作イメージ図

2 パッケージ概要

2.1 ディレクトリ構成

opensource COBOL v1.4.0J のディレクトリ構成を以下の表に示す。

| opensource COBOL-1.4.0J | | |
|-------------------------|------------------------------|------|
| bin | | 変更なし |
| cobc | | |
| config | | |
| copy | | |
| lib | | |
| libcob | | |
| m4 | | |
| po | | |
| tests | | |
| texi | | |
| vbisam | ISAM ファイルアクセスライブラリ | 追加 |
| win32 | Visual Studio ソリューションファイルを格納 | |
| cobc | cobc コンパイラのプロジェクトファイルを格納 | |
| cobcrun | cobcrun ランタイムのプロジェクトファイルを格納 | |
| libcob | libcob のプロジェクトファイルを格納 | |
| vbisam | VBISAM のプロジェクトファイルを格納 | |

表 2.1-1 opensource COBOL v1.4.0J ディレクトリ構成表

Windows 対応のため、新たに win32 ディレクトリを設け、そこに Visual Studio のソリューションファイルやプロジェクトファイルを配置している。また、外部ライブラリとして pdcurses や mpir を利用している。

vbisam ディレクトリには VBISAM のパッケージが格納されている。利用者が別途入手することなく、ビルドをして利用することができるようになった。

3 動作環境

opensource COBOL v1.4.0J は、以下の 3 種類の環境で開発と動作の確認を行った。

- 環境 1 (Windows7 クライアント スタンドアローン)
Microsoft Windows7 Professional Service Pack 1
Microsoft Visual Studio 2010 Express
- 環境 2 (Windows Server 2012 R2)
Microsoft Windows Server 2012 R2 Standard
Microsoft Visual Studio Professional 2013
Microsoft Windows7 Professional Service Pack 1
- 環境 3 (Linux)
CentOS release 6.5(Final)
gcc version 4.4.7 20120313 (Red Hat 4.4.7-4)

Windows クライアント OS とサーバーOS に加え、Linux においても同一パッケージでビルドから COBOL の実行までの動作が行えることを確認している。

4 インストール

4.1 ダウンロードサイト

opensource COBOL は、OSS コンソーシアム様のサイトでオープンソースソフトウェアとして公開している。

- OSS コンソーシアム 
<http://www.osscons.jp/>
- ダウンロードページ (名称 : opensource COBOL v1.4.0J)
http://www.osscons.jp/osscobol/download/v1_4j/
ファイル名 : opensourceCOBOL-1.4.0J.zip

4.2 インストール環境について

Windows でインストールを行う環境を、以下の表に示す。

| | |
|--------------|---------------------------------|
| OS | Windows Server 2012 R2 |
| 開発環境 | Visual Studio 2013 professional |
| インストール先 | C:\¥oscobol |
| 対象プラットフォーム | Win32 |
| 対象コンフィグレーション | Debug |

表 4.2-1 opensource COBOL インストール環境

4.3 パッケージの解凍

前述【4.1 ダウンロードサイト】よりダウンロードした配布パッケージを、インストール先ディレクトリに配置し、解凍を行う。

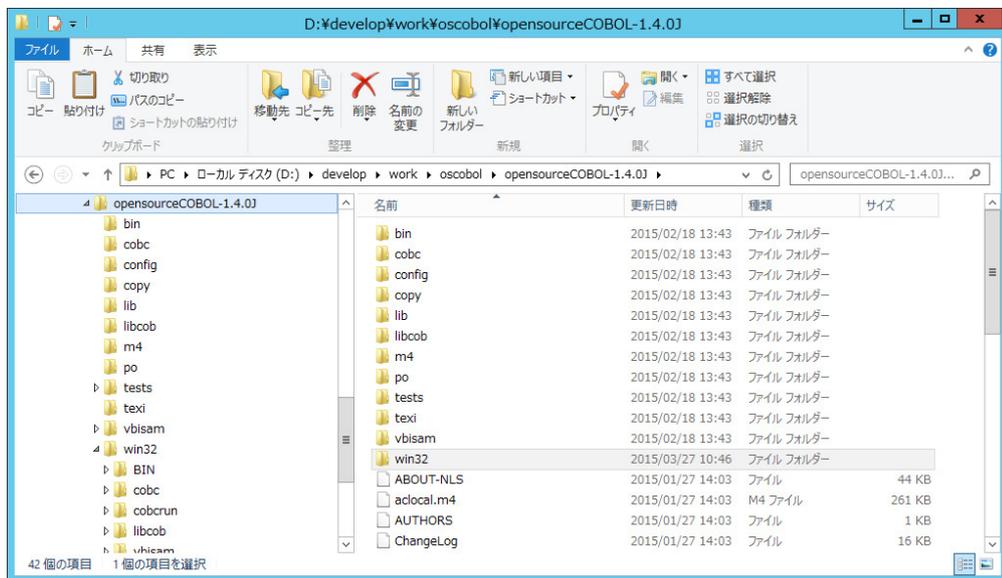


図 4.3-1 opensource COBOL v1.4.0J パッケージ解凍

4.4 opensource COBOL v1.4.0J のビルド

手順 1. 外部ライブラリの設定

opensource COBOL で利用する外部ライブラリ、MPIR と PDCurses を取得し、任意のディレクトリに設置する。その後、以下の環境変数を設定する。

- ・ LIB : mpir や pdcurses のライブラリを格納するディレクトリ
- ・ INCLUDE : mpir や pdcurses のヘッダファイルを格納するディレクトリ

手順 2. Visual Studio 2013 の起動

ダウンロードした配布パッケージに同梱されている Visual Studio 2013 ソリューションファイル opencobol.sln をダブルクリックし、Visual Studio 2013 を起動する。

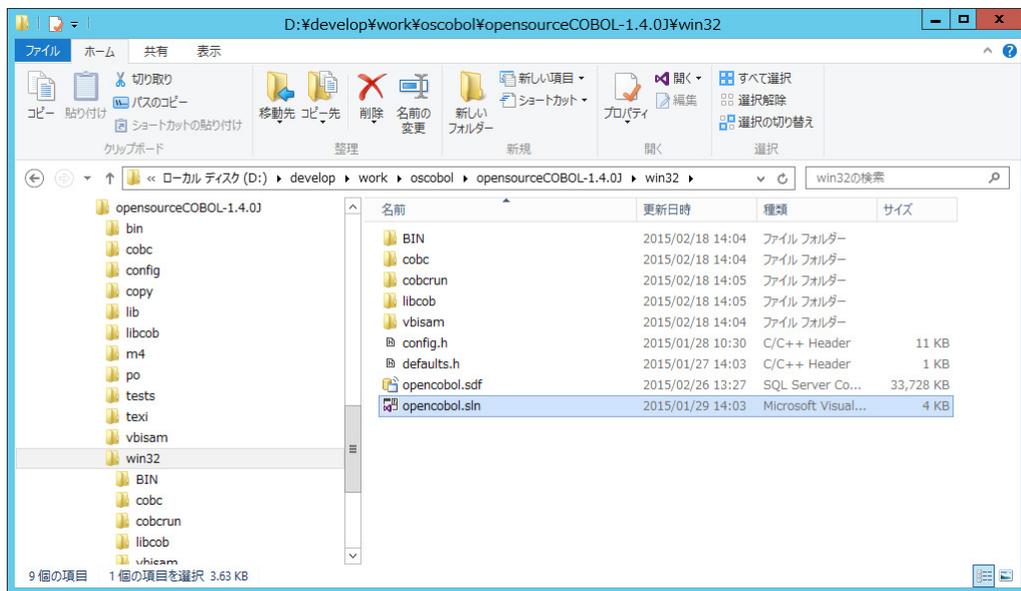


図 4.4-1 配布パッケージの解凍

手順 4. ビルド結果の確認

出力ウィンドウに表示された結果が、以下の様に正常終了していることを確認する。生成されたコンパイラやランタイムの実行ファイルは、win32¥BIN ディレクトリに格納される。

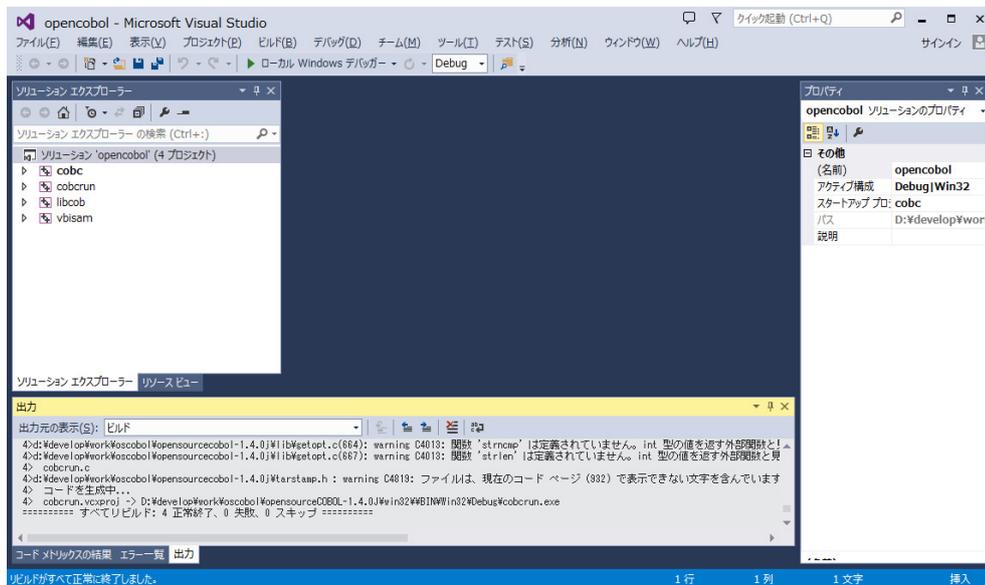


図 4.4-3 ビルド結果の確認

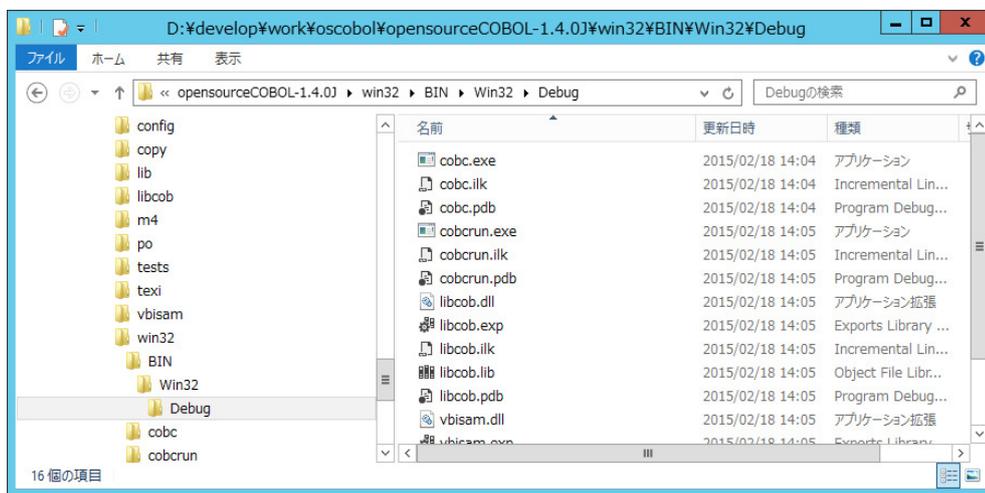


図 4.4-4 ビルドによる生成物の確認

その他、構成・プラットフォームの変更

ビルド構成やプラットフォームを変更する事が可能。ソリューションエクスプローラーのソリューションを右クリックし、【プロパティ】を選択する

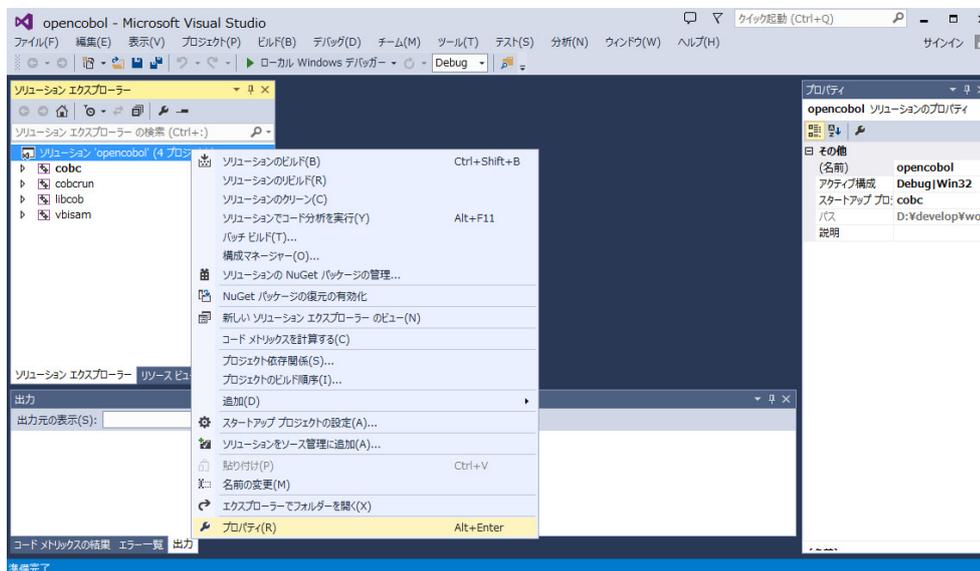


図 4.4-5 構成・プラットフォームの変更

構成プロパティから構成を選択し、表示された構成マネージャーから構成とプラットフォームの変更を行う。

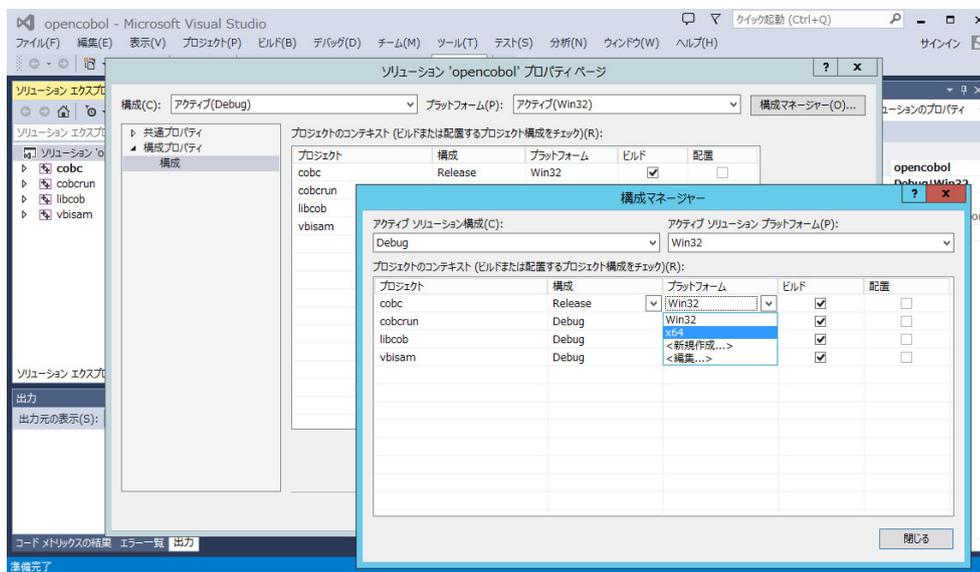


図 4.4-6 構成・プラットフォームの変更画面

5 コンパイル・実行

5.1 COBOL ソースファイルのコンパイル

5.1.1 準備

- CL コンパイラの指定

opensource COBOL で使用する Microsoft の CL コンパイラは、CPU ごとに使用する CL コンパイラが異なっている。Microsoft が提供しているバッチファイル VCVARSALL.BAT にパラメータを指定して実行し、決定する。詳細は下記 MSDN サイトを参照。

<https://msdn.microsoft.com/ja-jp/library/f2ccy3wt.aspx>

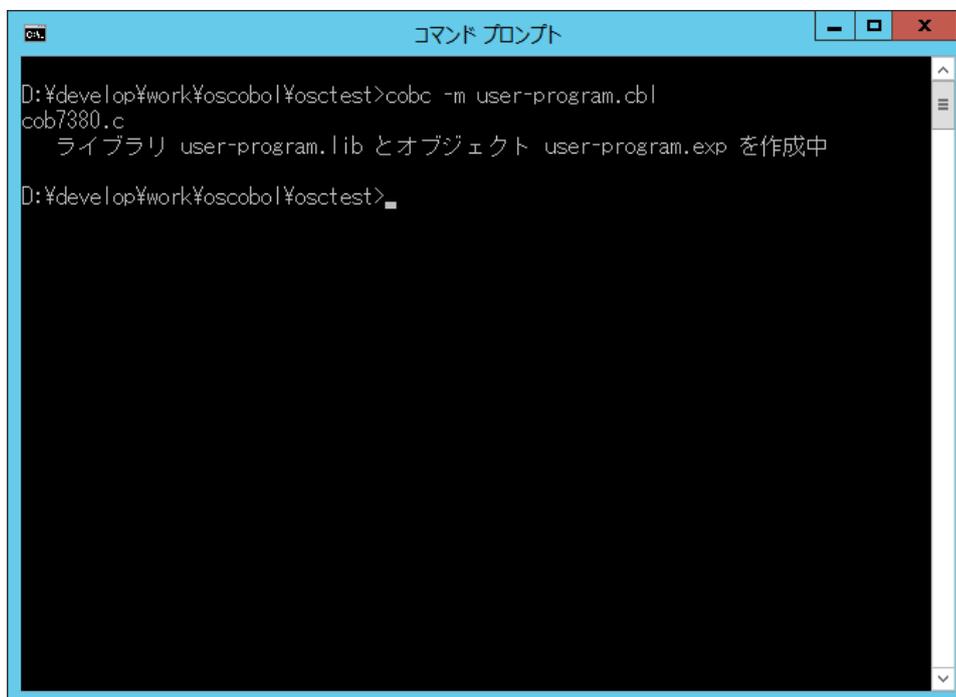
- 環境変数の設定

コンパイラやランタイムの実行には、後述【10.2 設定が必要な環境変数例】に記載されている設定が必要である。

| | |
|------------------|---|
| PATH | C:\oscobol\opensourceCOBOLV1.4.0J\win32\BIN; %LIB%;%INCLUDE%;%PATH% |
| COB_CONFIG_DIR | C:\oscobol\opensourceCOBOLV1.4.0J\config |
| COBCPY | C:\oscobol\Develop\copy |
| COB_LIBRARY_PATH | C:\oscobol\Develop\userdll |

5.1.2 コンパイルの実行

コンパイルは `cobc` コマンドを実行して行う。以下の画面は、「user-program.cbl」をコンパイルし、DLL ファイルを生成した例である。



```
コマンド プロンプト

D:\develop\work\oscobol\osctest>cobc -m user-program.cbl
cob7380.c
   ライブラリ user-program.lib とオブジェクト user-program.exp を作成中

D:\develop\work\oscobol\osctest>
```

図 5.1-1 cobc コマンド実行画面

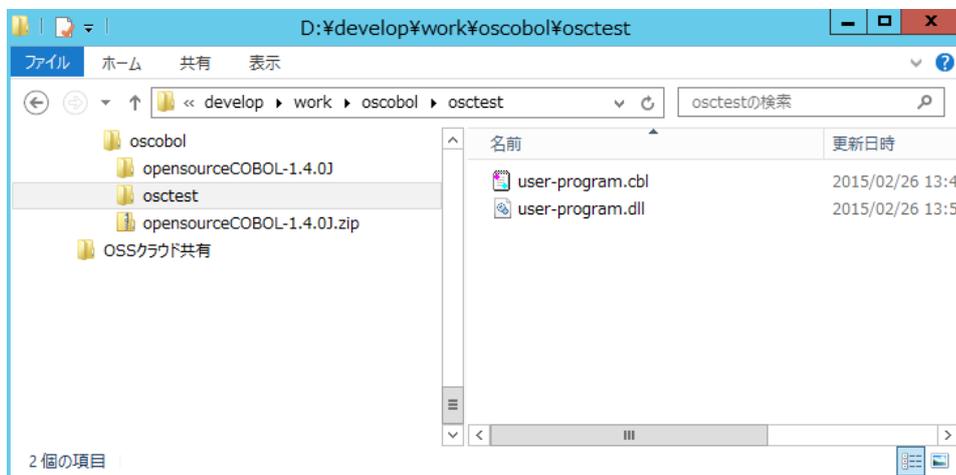
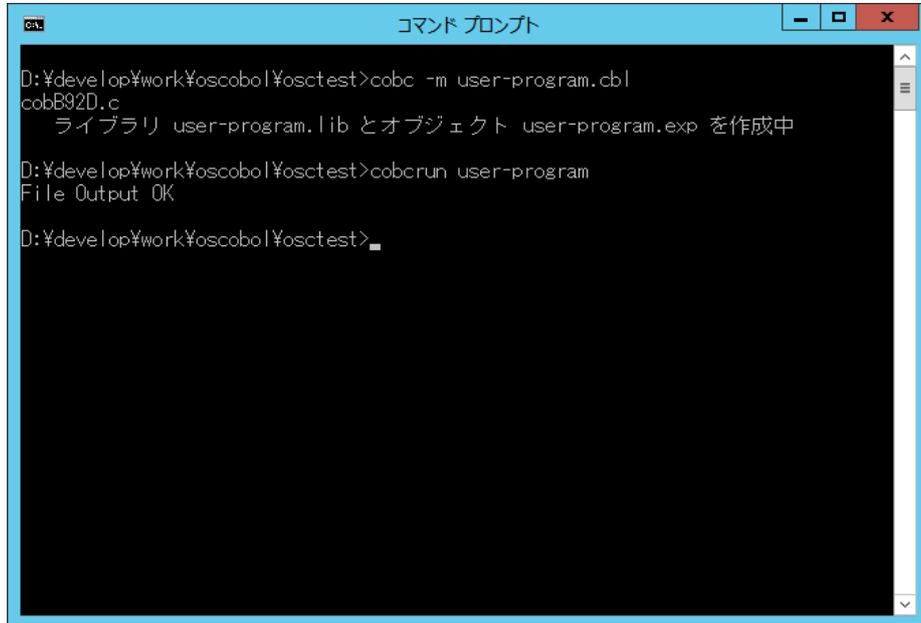


図 5.1-2 生成された DLL ファイルの確認

5.2 COBOL の実行

作成した DLL ファイルは、cobcrun コマンドを用いて実行する。以下の画面は、先ほどコンパイルを行った DLL ファイルを実行する例である。



```
コマンド プロンプト
D:\develop\work\oscobol\osctest>cobc -m user-program.cbl
cobB92D.c
ライブラリ user-program.lib とオブジェクト user-program.exp を作成中

D:\develop\work\oscobol\osctest>cobcrun user-program
File Output OK

D:\develop\work\oscobol\osctest>
```

図 5.2-1 cobcrun 実行画面

同様の手順で CUI プログラムを実行することができる。コマンドプロンプト上に CUI が表示され、従来は文字化けが起きていた日本語入力に対応したほか、罫線表示も対応できていることが確認できた。



```
コマンド プロンプト - cobcrun KEITEST

Half-size Characters
日本語文字
日本語+ANK
罫線も表示でき、日本語文字の入力もかろう

1:可能 2:可能ファイル 3:可能性 4:可能性が 5:可能性を 1/5
```

図 5.2-2 CUI プログラム実行画面

6 性能比較

opensource COBOL v1.4.0J の Windows 版と Linux 版との性能比較を実施した。実施と比較の結果は以下に記載の通り。

6.1 性能比較環境

- Windows
CPU : Intel(R) Xeon(R) CPU X5670 2.93GHz
メモリ : 16GB
- Linux
CPU : Intel(R) Xeon(R) CPU X5670 2.93GHz
メモリ : 16GB

6.2 演算パフォーマンスの性能比較と結果

演算にかかる時間を比較する。テストは 3 回行い、その平均時間を比較する。

- テストプログラム仕様

加算、減算、乗算、除算を、COMPUTE 文にて各々 1,000,000 回ループで実行を行い、かかった時間を計測する。100,000 ループ実施するごとに、コンソールにメッセージを表示する。

テストの実行結果は、以下の表の通り。

| | 1 回目 | 2 回目 | 3 回目 | 平均 |
|---------|--------|--------|--------|--------|
| Windows | 5.36 秒 | 5.30 秒 | 5.28 秒 | 5.31 秒 |
| Linux | 3.55 秒 | 3.55 秒 | 3.54 秒 | 3.55 秒 |

表 6.2-1 演算パフォーマンス結果の比較

演算にかかる時間を比較したところ、Windows 版は Linux 版よりも時間がかかる事が明らかになった。しかし、この時間の差は実用上特に問題が無い程度の差であると考えている。

7 周辺機能

オフコンなどの汎用機のマイグレーションでは、COBOL 以外の部分 (JCL、ソート、ファイル作成、帳票ツールや運用監視ツールなどとの結合、等) も必要となってくる。それらを実現するためのプログラム群が AJTOOL と呼ばれるプログラム群である。その AJTOOL も、Windows 版の opensource COBOL で使用する事が可能である。

8 既知の制約事項と問題点

- Windows 版は cobc コンパイラオプション “-L” が未実装となっている。Windows 版では、代替として PATH 環境変数に設定を行うことでライブラリディレクトリを参照することができる。

- Windows 版は、ファイルアクセスのロック機能が未実装。

9 まとめ

opensource COBOL v1.4.0J のリリースにより、Windows に標準対応し、CUI 機能の強化が行われたことを確認することができた。これにより、汎用機、オフコン上のシステムを opensourceCOBOL を用いて Windows サーバへの移行が可能となり、COBOL システムのオープン化に効果的であることが確認できた。

opensourceCOBOL は OSS 故のクラウドとの親和性により、今後の COBOL システムのオープン化やクラウド化において有効な選択肢となりえる。

10 付録

10.1 cobc コンパイラのコンパイラオプション例

| | |
|-------|-------------------------------|
| オプション | --help |
| 役割 | cobc コンパイラのコンパイラオプションを画面に表示する |

| | |
|-------|--|
| オプション | -x |
| 役割 | exe ファイルを生成する。exe ファイルは単体での実行が可能だが、他ユーザアプリケーションからの CALL 呼び出しに対応できない。 ※-m オプションとの同時指定は不可 |

| | |
|-------|---|
| オプション | -m |
| 役割 | dll ファイルを生成する。単体での実行はできず、cobcrun.exe より実行する。exe ファイルと違い、他ユーザアプリケーションからの CALL 呼び出しが可能。 ※-x オプションとの同時指定は不可 |

10.2 設定が必要な環境変数例

| | |
|-------|--|
| 環境変数名 | PATH |
| 役割 | コンパイラおよびランタイムの検索先パス |
| 設定例 | SET PATH=C:\oscobol\opensourceCOBOLV1.4.0J\win32\BIN;%PATH% |

| | |
|-------|--|
| 環境変数名 | COB_CONFIG_DIR |
| 役割 | コンパイラのコンフィグファイル格納パス |
| 設定例 | SET COB_CONFIG_DIR=C:\oscobol\opensourceCOBOLV1.4.0J\config |

| | |
|-------|------------------------------------|
| 環境変数名 | COBCPY |
| 役割 | COPY ファイルの格納パス |
| 設定例 | SET COBCPY=C:\oscobol\Develop\copy |

| | |
|-------|--|
| 環境変数名 | COB_LIBRARY_PATH |
| 役割 | ユーザ作成 DLL の格納パス |
| 設定例 | SET COB_LIBRARY_PATH=C:\¥oscobol¥Develop¥userdll |

10.3 トラブルシュート

| | |
|----|--|
| 現象 | cobc を実行した際に、以下のエラーが発生する。 'cobc' は、内部コマンドまたは外部コマンド、 操作可能なプログラムまたはバッチ ファイルとして認識されてい ません。 |
| 原因 | 環境変数 PATH に、cobc.exe を格納しているパスが定義されていな い |
| 対応 | 環境変数 PATH に、cobc.exe を格納しているパスを定義する |

| | |
|------|--|
| 現象 | cobcrun を実行した際に、以下のエラーが発生する。 libcob: Cannot find module 'user-program.dll' |
| 原因 1 | 環境変数 COB_LIBRARY_PATH に、作成した DLL ファイルが格 納されているディレクトリが設定されていない |
| 対応 1 | 環境変数 COB_LIBRARY_PATH に、作成した DLL ファイルが格 納されているディレクトリを設定する |
| 原因 2 | cobcrun に指定する DLL ファイルの拡張子は不要。 |
| 対応 2 | cobcrun に指定する DLL ファイル名から拡張子を外す。 例) cobcrun user-program.dll→cobcrun user-program |

| | |
|----|--|
| 現象 | cobcrun を実行した際に、以下のエラーが発生する。 libcob: LoadLibrary/GetProcAddress error 127 |
| 原因 | ソースファイル名と PROGRAM-ID で指定したファイル名が異な る。 |
| 対応 | ソースファイル名と PROGRAM-ID を同じにする。 |