White Paper Micro Focus Visual COBOL

# Micro Focus Visual COBOL を 活用した開発作業の効率化

Visual COBOL は Micro Focus が提供する最新版の COBOL 統合開 発環境製品です。本製品は COBOL のコーディング・コンパイル機能やデバッ グ機能のみならず、さまざまなシーンでの使用を想定した COBOL 専用の開 発補助機能を豊富に搭載しています。これらはバッチ、シェル、CI ツール等 からの利用も想定して設計されており、有効活用することで変化に強い高速 な開発スタイルの実現も可能です。本書ではこれらの Visual COBOL に 装備された主な開発補助機能を概説いたします。



## White Paper Micro Focus Visual COBOL

# 目次

| 1. はじめに                                                                                                                                                    | 3               |
|------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------|
| 1.1 Micro Focus Visual COBOL とは                                                                                                                            | 3               |
| 1.2 Visual COBOL を活用した開発の自動化                                                                                                                               | 3               |
| <ol> <li>自動実行も可能、Visual Studio、Eclipse COBOLプロジェクトのビノ</li> <li>2.1 MSBuild による Visual Studio プロジェクトのビルド</li> <li>2.2 Ant による Eclipse プロジェクトのビルド</li> </ol> | <b>レド 3</b><br> |
| 3. Test Coverage - COBOL 専用のカバレッジ分析ツール                                                                                                                     | 12              |
| 3.1 Test Coverage の基本動作                                                                                                                                    | 12              |
| 3.2 IDE を通じた Test Coverage の操作                                                                                                                             | 13              |
| 4. コード分析 – COBOL 専用の静的コード解析ツール                                                                                                                             | 15              |
| 4.1 コード分析の基本動作                                                                                                                                             | 15              |
| 4.2 Enterprise Analyzer 上のクエリールールの活用                                                                                                                       | 17              |
| 4.3 MSBuild, Ant による実行                                                                                                                                     | 18              |
| 5. mfunit - COBOL 専用の xUnit 系単体テストフレームワーク                                                                                                                  | 19              |
| 5.1 mfunit の基本機能                                                                                                                                           | 19              |
| 5.2 JUnit 互換形式の出力フォーマットの活用                                                                                                                                 | 23              |
| 6. Profiler – COBOL 専用のパフォーマンス分析ツール                                                                                                                        | 25              |
| 6.1 コマンドによる処理                                                                                                                                              | 25              |
| 6.2 IDE 上での処理                                                                                                                                              | 26              |
| 7. おわりに                                                                                                                                                    |                 |
|                                                                                                                                                            |                 |

# 1 はじめに

### 1.1 Micro Focus Visual COBOL とは

Visual COBOL は長年 COBOL 業界をリードしてきた Micro Focus が提供する最新版の COBOL 統合開発環境製品です。 Windows, Linux, UNIX ターゲットのアプリケーションを最新テクノロジー環境にて様々なミドルウェア等と連携した COBOL アプリケー ション開発を可能とします。Visual COBOL に装備された優れた開発ツール群を活用することで開発期間の短縮やコスト削減も望めます。

Windows ターゲットであれば Visual Studio や Eclipse IDE 上で COBOL アプリケーションを開発できます。 Visual COBOL はこれ らの IDE 向けに様々な COBOL 専用の開発支援機能を提供しています。例えば、COBOL 開発用に用意されたエディターを利用すれば、 COBOL プログラムのコーディング・メンテナンスの生産性を著しく向上できます<sup>1</sup>。

Linux/UNIX をターゲットとする場合でも Windows をターゲットとする場合と同様に Eclipse IDE 上で Windows の開発と同じ感覚で 開発が可能です<sup>2</sup>。

これらの環境上で開発した COBOL モジュールは Native アプリケーション、.NET アプリケーション、JVM アプリケーション、EJB、 RESTful サービス、もしくは SOAP Web サービスとして運用が可能であり、目的に合ったアプリケーションを構築できます。

# 1.2 Visual COBOL を活用した開発の自動化

Visual COBOL は COBOL 開発における高機能なエディター、デバッガーを備えたコンパイラ製品に留まりません。Micro Focus が 約 40 年に渡って培った COBOL アプリケーション開発ノウハウを反映させた COBOL 向けの開発支援機能も豊富に搭載しています。こ のうち本書では、以下の機能について概説いたします<sup>3</sup>。

> ビルド

- > カバレッジ分析
- > 静的コード解析
- > 単体テスト
- > パフォーマンス分析

これらの機能は様々なシーンでの使用を想定して設計されており、IDE上での実行はもちろんのこと、Windows であればバッチや PowerShell、Linux/UNIX であればシェル経由で実行することも可能です。更には CI ツールによる実行を意識したインターフェースを 用意したものもあり、これらの作業プロセスを自動化し継続的インテグレーションに組み入れるといった使い方も可能です。

# 2 自動実行も可能、Visual Studio、Eclipse COBOL プロジェクトのビルド

Visual COBOL は Visual Studio IDE 並びに Eclipse IDE 上で COBOL アプリケーションの開発を可能とします。 C# や Java の開 発と同様に IDE 上で COBOL アプリケーションのビルド命令を発行することも可能ですが、 IDE の外でビルド命令を発行することも可能 です。IDEを使用しないビルドについては従来よりコマンドラインインターフェースを提供していますが、ここでは IDE 上で作成したプロジェ クトをビルドする方法を紹介いたします。各 IDE のプロジェクトとしてビルドするため、プロジェクトに設定したビルド命令、コンパイラ指令、 プリコンパイル指令等をプロジェクトでビルド対象とするリソースに対して反映させます。

# 2.1 MSBuild による Visual Studio プロジェクトのビルド

MSBuild は Microsoft 社が提供するソースコードのコンパイルやパッケージング等の自動化を支援するビルドツールです。本ツールを使え ば Visual Studio IDE がない環境<sup>4</sup> であっても Visual Studio のソリューション並びにプロジェクトを Visual Studio IDE 上で設定したプロ ジェクト設定の内容に基づきビルドが出来るようになります。この MSBuild を活用した自動実行をサポートするツールは Microsoft Team Foundation Server を始め Jenkins のプラグイン等でも提供されており、コマンドラインのみならず様々な Cl ツールより実行が可能です。 Visual COBOL for Visual Studio がインストールされた環境であれば、Visual COBOL の COBOL プロジェクトについても自動ビルドが 可能となります。ここでは Visual COBOL for Visual Studio 2015 で作成した COBOL のプロジェクトを Jenkins 経由でビルドするよ うすをご紹介します<sup>5</sup>。

\_\_\_\_\_

エディターを活用した実際の開発イメージについては下記ページ中で公開しています動画「進化する COBOL エディター ~ Visual Studio 編~」及び「進化する COBOL エディター ~ Eclipse 編~」でご覧いただけます。
 http://www.microfocus.co.jp/resources/video/visualcobol/

**<sup>2</sup>** Linux/UNIX をターゲットとしたアプリケーションを Windows 上で Eclipse IDE を使って開発するようすを下記ページ中の [Eclipse IDE を使って UNIX/Linux ターゲットの COBOL アプリケーションをリモート開発」にてご覧いただけます。 http://www.microfocus.co.jp/resources/video/visualcobol/

<sup>3</sup> 本書では Visual COBOL 2.3J Update 1 for Windows を用いて動作イメージを紹介しています。異なるバージョンや OS 向けの製品で記載内容を試すのであればそれぞれに該当 するマニュアルをご参照ください。

<sup>4</sup> 以前は、.NET Framework にバンドルされていたようですが、執筆時点の最新版については「Microsoft Build Tools 2015」にバンドルされているようです。 「Microsoft Build Tools 2015」はMicrosoft 社のホームページより無償でダウンロードできるようです。

<sup>5</sup> 本書では Jenkins 1.654 を用いたイメージを掲載しています。別のバージョンをお使いになる場合、画面やブラグインのバージョン等、掲載と異なるものがあれば適宜置き換えてご 覧ください。

#### ① Jenkins に MSBuild のプラグラインを追加

|                                  | 🗕 アップデートセンター [Jenkins] 🗙 📃                             |                             |                                                                                                                                              |                  |    |
|----------------------------------|--------------------------------------------------------|-----------------------------|----------------------------------------------------------------------------------------------------------------------------------------------|------------------|----|
|                                  | ← → C D localhost:8080/plu                             | iginManager/installe        | d                                                                                                                                            |                  |    |
|                                  | Jenkins > プラグインマネージャー                                  |                             |                                                                                                                                              |                  |    |
|                                  |                                                        | Ŷ                           | Maven 2/3 jobs via a special<br>project type.                                                                                                | <u>Z.1.1</u>     |    |
|                                  |                                                        |                             | MSBuild Plugin                                                                                                                               |                  |    |
|                                  |                                                        |                             | This plugin makes it possible<br>to build a Visual Studio project<br>(.proj) and solution files (.sln).                                      | <u>1.25</u>      |    |
|                                  |                                                        |                             | MSTest plugin                                                                                                                                |                  |    |
|                                  |                                                        | V                           | This plugin converts <u>MSTest</u><br>TRX test reports into JUnit<br>XML reports so it can be<br>integrated with Hudson's JUnit<br>features. | <u>0.19</u>      |    |
| ② 使用する<br>Jenkins の <sup>6</sup> | MSBuildを設定<br>管理画面にて [ システムの設定 ]                       | をクリック                       |                                                                                                                                              |                  |    |
|                                  | Jenkinsの管理                                             |                             |                                                                                                                                              |                  |    |
|                                  | ▲ Jenkinsの新しいバージョン(2.14)を空<br>システムの設定<br>システム全体の振る舞いやり | クンロードできます (変更)<br>パスを設定します。 | <sub>爾提)。</sub> 自動でアップグレード                                                                                                                   |                  |    |
| [MSBuild] ‡                      | 闌にて使用する MSBuild を指定。                                   | ここでは Visual S               | Studio 2015 にバンドルされる                                                                                                                         | る MSBuild を指定してい | ます |
|                                  | MSBuild                                                |                             |                                                                                                                                              |                  |    |
|                                  | インフトール溶みMSBのは                                          | III MOD III                 |                                                                                                                                              |                  |    |

| インストール済みMSBuild | MSBuild           |                                                     |
|-----------------|-------------------|-----------------------------------------------------|
|                 | 名前                | bundled with Visual Studio 2015                     |
|                 | MSBuild.exeファイルパス | C:\Program Files (x86)\MSBuild\14.0\Bin\MSBuild.exe |

③ ビルド対象のアプリケーションを含む Visual Studio プロジェクトを用意

ここでは、Native COBOL、.NET Managed COBOL、C#、VB.NET のプロジェクトを含むソリューションを用意しました:



#### ④ Jenkins にてビルド用のジョブを用意

[フリースタイル・プロジェクトのビルド]を選択の上、ジョブ名を命名



[OK] ボタンを押下

# ⑤ ジョブに MSBuild によるビルド命令を追加 [ビルド手順の追加] にて [MSBuild の実行]を選択

| 2 | Conditional step (single)             |
|---|---------------------------------------|
| 1 | Antの呼び出し                              |
| L | Conditional steps (multiple)          |
| L | Execute Silk Performer Tests          |
|   | MSBuildの実行                            |
| E | Mavenの呼び出し                            |
| ł | Run unit tests with MSTest            |
|   | SilkCentral Test Manager Execution    |
| L | Trigger/call builds on other projects |
| L | Windows PowerShell                    |
| L | Windowsバッチコマンドの実行                     |
| ł | シェルの実行                                |
|   | ビルド手順の追加 🔻                            |
|   |                                       |

[MSBuild バージョン] 欄では②で用意した MSBuild 設定を選択

| ビルF             |                                              |
|-----------------|----------------------------------------------|
| MSBuildの実行      |                                              |
| MSBuild バージョン   | (Default)                                    |
| MSBuild ビルドファイル | (Default)<br>bundled with Visual Studio 2015 |
| コマンバライン引数       |                                              |

[MSBuild ビルドファイル]欄に Visual Studioの COBOL プロジェクトフォルダ配下にあるプロジェクトファイル < プロジェクト名 >.cblproj を指定

| ビルド             |                                                        |  |
|-----------------|--------------------------------------------------------|--|
| MSBuildの実行      |                                                        |  |
| MSBuild バージョン   | bundled with Visual Studio 2015                        |  |
| MSBuild ビルドファイル | C:\work\wp\DevTools\Book_WPF\COBCLASS\COBCLASS.cblproj |  |
|                 |                                                        |  |

[保存]ボタンを押下

### ⑥ Visual Studio プロジェクトのビルドを実行

用意した Jenkins のプロジェクトにて [ビルド実行]をクリック



正常にビルドされると [ビルド履歴]欄にて青いアイコンが表示されます。(失敗した場合は、赤いアイコンが表示されます。)



ビルド番号をクリックの上、[コンソール出力]をクリックするとビルド時のコンソール出力を確認できます:



⑦ MSBuild のビルドターゲットにソリューションを指定し、COBOL や他の .NET 言語のプロジェクトをまとめてビルド

⑤で構成した Jenkins のビルド設定画面にて [MSBuild ビルドファイル ] 欄にプロジェクトファイルではなく .sln の拡張子を持つソリューショ ンファイルを指定:

| ビルド             |                                             |
|-----------------|---------------------------------------------|
| MSBuildの実行      |                                             |
| MSBuild バージョン   | bundled with Visual Studio 2015             |
| MSBuild ビルドファイル | C:\work\wp\DevTools\Book_WPF\ManagedSol.sIn |

#### 再度 Jenkins から⑥の要領でビルドを実行

ソリューションに設定されたプロジェクト間の依存関係に基づいたビルド順序に沿ってソリューション配下のプロジェクトがビルドされます。 ソリューションに設定されたビルド順序のイメージ:

| プロジェクトのも                                           | 衣存関係                 |       | ? | × |
|----------------------------------------------------|----------------------|-------|---|---|
| 依存関係                                               | ビルドの順序               |       |   |   |
| プロジェクト                                             | をビルドする順の             | 斧(R): |   |   |
| COBCLA<br>CsharpW<br>VBnetWi<br>LegacyA<br>CobolWi | SS<br>PF<br>PF<br>PF |       |   |   |

#### 2.2 Ant による Eclipse プロジェクトのビルド

Ant は Apache オープンソースプロジェクトが提供する Java ベースのビルドツールです。Eclipse IDE 上で作成された Visual COBOL のプロジェクトはこの Ant を使ってプロジェクトをビルドします。Ant のビルドファイル .cobolBuild はプロジェクトファイル .cobolProj より 自動生成されます。Java のプロジェクトのようにビルドファイルをプロジェクトからエクスポートして生成させる必要はありません。Visual COBOL は Eclipse 上で作成した COBOL プロジェクトをバッチやシェルを始めとする Ant を利用可能なツールから自動実行させることも 想定して設計されています。前項で紹介した CI ツール Jenkins にも Ant によるビルドインターフェースが用意されています。ここでは、 Visual COBOL for Eclipse のプロジェクトより自動生成された Ant のビルドファイルを Jenkins よりビルド実行させるイメージを紹介します。

① Visual COBOL for Eclipse のプロジェクトを用意



プロジェクトを作成すると自動で Ant ビルドファイル .cobolBuild が生成されます:



 Visual COBOL for Eclipse にバンドルされる Ant のバージョンを確認 Eclipse にて

[ウィンドウ]メニュー > [設定] で表示される画面にて [Ant] > [ランタイム]へとナビゲート

| 🔤 設定                                                                                                                                                                                   |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | 本書 新 聿 時 点<br>1.9.2. が バン                                                                                                                                                                 | での最新版 2.3 Opdate T には A<br>ドルされていることがわかります。                                                                                                                                                                                                                                                                                                                                | nt |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| <b>ጋ</b> ብルタ入力                                                                                                                                                                         | ランタイム                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |                                                                                                                                                                                           |                                                                                                                                                                                                                                                                                                                                                                            |    |
| <ul> <li>一般</li> <li>Ant</li> <li>ゴディター<br/>ランタイム</li> <li>BPEL</li> <li>BPMN2</li> <li>Data Management</li> <li>Drools</li> <li>Forge<br/>FreeMarker</li> <li>Fuse Tooling</li> </ul> | Ant ビレド・ファイルの実行時に使用される設定:           ◆ クラスパス(L)         ② タスク(K)         ④ タイプ(Y)         ③ プロパラ           ◆ Ant ホーム項目 (デフォルト)         ③ C×Users¥Public¥Micro Focus¥Visual COBOL¥           ● C×Users¥Public¥Micro Focus¥Visual COBOL¥ | F1(T)<br>ieclipse¥plugins¥org.apach<br>ieclipse¥plugins¥org.apach<br>ieclipse¥plugins¥org.apach<br>ieclipse¥plugins¥org.apach<br>ieclipse¥plugins¥org.apach<br>ieclipse¥plugins¥org.apach | e.ant_1.9.2.v201404171502¥iib¥ant-antirjar<br>e.ant_1.9.2.v201404171502¥iib¥ant-apache-bcel.jar<br>e.ant_1.9.2.v201404171502¥iib¥ant-apache-bsfjar<br>e.ant_1.9.2.v201404171502¥iib¥ant-apache-log4jjar<br>e.ant_1.9.2.v201404171502¥iib¥ant-apache-rorjar<br>e.ant_1.9.2.v201404171502¥iib¥ant-apache-regexp.jar<br>e.ant_1.9.2.v201404171502¥iib¥ant-apache-resolver.jar |    |

③ Jenkins に②で確認したバージョンの Ant を指定 [Jenkins の管理] をクリック

[システムの設定]をクリック

[Ant 追加] ボタンを押下:

|                        | description                            |
|------------------------|----------------------------------------|
| MSBuild                |                                        |
| インストール済みMSBuild<br>Ant |                                        |
| インストール済みAnt            | Ant追加                                  |
|                        | Jenkinsで利用する、このシステムにインストールされたAntの一覧です。 |

[バージョン]欄にて②で確認したバージョンを指定:

| Ant           |                               |
|---------------|-------------------------------|
| インストール 済み Ant | iiii Ant                      |
|               | 名前 Ant_192_in_line_with_VC231 |
|               |                               |
|               | Apachetからインストール               |
|               | バージョン 1.9.2 ▼                 |

④ Jenkins にてビルド用のジョブを用意 [新規ジョブ作成]をクリック

ジョブ名を命名の上、[フリースタイル・プロジェクトのビルド]を選択

ジョブ名 Build\_VS\_Projects\_by\_Ant フリースタイル・プロジェクトのビルド ちっとも汎用性の高いJenkinsの中核機能です。任意のSCMからソースコードをチェックアウトし、任意のビルドシステムでプロジェクトがビルドできます。往々にして、ソフトウェアのビルド以外にも様々な仕事の自動化に利用することができます。

# ⑤ ジョブに Ant によるビルド命令を追加

[ビルド手順の追加]にて [Ant の呼び出し]を選択

| - | Conditional step (single)             |
|---|---------------------------------------|
|   | Antの呼び出し                              |
| Г | Conditional steps (multiple)          |
| L | Execute Silk Performer Tests          |
| L | MSBuildの実行                            |
| L | Mavenの呼び出し                            |
| ł | Run unit tests with MSTest            |
|   | SilkCentral Test Manager Execution    |
| L | Trigger/call builds on other projects |
| L | Windows PowerShell                    |
| L | Windowsバッチコマンドの実行                     |
| ł | シェルの実行                                |
|   | ビルド手順の追加 🔻                            |

#### [使用する Ant] 欄にて③で用意した Ant を選択

| ビルド     |                                     |        |   |
|---------|-------------------------------------|--------|---|
| Antの呼び  | 出し                                  |        | ? |
| 使用するAnt | デフォルト                               | •      |   |
| ターゲット   | デフォルト<br>Ant_192_in_line_with_VC231 |        | ? |
|         |                                     | 高度な設定… |   |

#### [高度な設定]ボタンを押下

| ビルド     |                            |       |
|---------|----------------------------|-------|
| Antの呼び。 | 出し                         | 0     |
| 使用するAnt | Ant_192_in_line_with_VC231 | T     |
| ターゲット   |                            | ▼ 0   |
|         |                            | 高度な設定 |

[ターゲット] 欄には -lib <Visual COBOL のインストールフォルダ >\bin\mfAnt.jar を、[ビルドファイル] 欄には Eclipse プロジェ クトフォルダ配下にある Ant のビルドファイルを指定:

| ビルド     |                                                                      |  |
|---------|----------------------------------------------------------------------|--|
| Antの呼び出 | st.                                                                  |  |
| 使用するAnt | Ant_192_in_line_with_VC231                                           |  |
| ターゲット   | -lib "C:\Program Files (x86)\Micro Focus\Visual COBOL\bin\mfant.jar" |  |
| ビルドファイル | C:\work\wp\DevTools\ws\NativeCOBOL\.cobolBuild                       |  |

[保存]ボタンを押下

#### ⑥ Ant をインストール 用意した Jenkins のプロジェクトを一度実行させ、Ant をダウンロードさせます。

[ビルド実行]をクリック

Visual COBOL を使うための環境変数が指定されていないため、コンパイルできず、コンパイル処理の部分でエラーが返ります:

# **cobol.compile.cfg.New\_Configuration:** [cobol] Compiling CHECKDATE.cbl...

[cobol] Cobo L

[cobol

An unexpected error was encountered when executing the COBOL compile task Compilation complete with 2 errors, 0 warnings, 0 notices and an exit code of 12 Compiling CUISCRN.cbl... [cobol] [cobol]

cobo l

An unexpected error was encountered when executing the COBOL compile task Compilation complete with 2 errors, 0 warnings, 0 notices and an exit code of 12 Compiling UDT0036P.cbl... cobol

[cobol [cobol]

[cobol

[cobol] An unexpected error was encountered when executing the COBOL compile task [cobol] Compilation complete with 2 errors, 0 warnings, 0 notices and an exit code of 12

cobol.link.cfg.New\_Configuration:

#### ⑦ Visual COBOL の利用に必要な環境変数を設定するバッチファイルを用意 Visual COBOL コマンドプロンプトを開き、SET コマンドを実行

#### 出力される環境変数をメモ

#### 出力された環境を設定するバッチファイルを用意 バッチファイルの例:



⑧ ant.bat に Visual COBOL を利用するための環境変数を指定

ant のインストールフォルダ配下の bin フォルダを開く(例えば、ログインユーザが「tarot」、③で指定した Ant の名前が「Ant\_192\_ in\_line\_with\_VC231」であれば

C:\Users\tarot\.jenkins\tools\hudson.tasks.Ant\_AntInstallation\Ant\_192\_in\_line\_with\_VC231 となります。)

#### フォルダ中に格納されている ant.bat をテキストエディタで開く

ant.bat ファイル中の先頭行に⑦で用意したバッチファイルを call する命令を埋め込み:



正しく機能することを確認します。

出力例:

cobol.cfg.New\_Configuration:

**cobol.compile.cfg.New\_Configuration:** [cobol] Compiling CHECKDATE.cbl... cobo l Micro Focus COBOL Version 2.3.01142 Copyright (C) Micro Focus 1984-2016. All rights reserved. \* Checking complete with no errors - starting code generation \* Generating C:¥work¥wp¥DevTools¥ws¥NativeCOBOL¥New\_Configuration.bin¥CHECKDATE \* Data: 408 Code: 736 Literals: 132 Compilation complete with no errors. cobo I cobo I cobol cobo I cobo l cobol Compiling CUISCRN.cbl... cobo l cobo l Micro Focus CCBOL Version 2.3.01142 Copyright (C) Micro Focus 1984-2016. All rights reserved. \* Checking complete with no errors - starting code generation \* Generating C:¥work¥wp¥DevTools¥ws¥NativeCOBOL¥New\_Configuration.bin¥CUISCRN \* Data: 1844 Code: 1864 Literals: 1392 cobo I cobo l cobo I cobo l cobol Compilation complete with no errors. Compiling UDT0036P.cbl... cobol cobo I cobo I [cobol] Micro Focus COBOL [cobol] Version 2.3.01142 Copyright (C) Micro Focus 1984–2016. All rights reserved. [cobol] \* Checking complete with no errors - starting code generation [cobol] \* Generating C:¥work¥wp¥DevTools¥ws¥NativeCOBOL!Wew\_Configuration.bin¥UDT0036P [cobol] \* Data: 680 Code: 4216 [cobol] Compilation complete with no errors. Literals: 392

cobol.link.cfg.New\_Configuration:

# 3 Test Coverage - COBOL 専用のカバレッジ分析ツール

ホワイトボックステストの実施にあたり、テスト網羅性の計測はテスト品質を評価する上で重要な指標となり得ます。Micro Focus は Visual COBOL よりも前の世代の製品から COBOL 専用のカバレッジ分析ツール「Test Coverage」を提供しています。本ツールを 使えば COBOL ソース中の各文が各テストのポイントで通過した数を記録しレポート表示できます。

# 3.1 Test Coverage の基本動作

Test Coverage は TESTCOVER コンパイラ指令を指定してコンパイルされたモジュールに対してカバレッジを取得します。カバレッジ の分析動作については TESTCOVER 環境変数でポイントする構成ファイルにて構成します。取得されたカバレッジ情報は.tczの拡張子 を持つバイナリファイルに溜められます。このファイルに書き出されたデータは tcreport コマンドを使ってテキストもしくは HTML 形式の レポートに出力します。この一連の処理については特に特別な GUI ツール等を介していないため、バッチやシェル等で自動化も可能です。 つまり前章で紹介したような CI ツールから実行することも可能です。

以下に簡単なサンプルを用いた Test Coverage の実行例を記します:

### ① TESTCOVER コンパイラ指令を指定して対象のプログラムをコンパイル



#### Test Coverage の構成ファイルを準備



準備したファイルは環境変数 TESTCOVER でポイントします:

C:¥work¥wp¥DevTools¥TestCover>set TESTCOVER=C:¥work¥wp¥DevTools¥TestCover¥testcover.cfg C:¥work¥wp¥DevTools¥TestCover> ③ アプリケーションを実行 ④ カバレッジ情報を HTML 形式のレポートに成形 C:¥work¥wp¥DevTools¥TestCover>tcres.tcz browse html verbs Program ussteel.gnt processed normally Listing of reports produced is in: C:¥work¥wp¥DevTools¥TestCover} C:¥work¥wp¥DevTools¥TestCover} C:¥work¥wp¥DevTools¥TestCover}

html: HTML 形式のレポートを生成させるオプション verbs: COBOL 文のサマリをレポートに含めるオプション

#### 生成されたレポートのイメージ:



### 3.2 IDE を通じた Test Coverage の操作

Visual COBOL を使う場合、Visual Studio や Eclipse IDE 上でも COBOL アプリケーションの開発が可能です。これらの IDE 上 には COBOL 用に作りこまれたデバッガーが搭載されており、これらを使ってデバッグ実行して消化したテストのカバレッジを確認するこ とも可能です。前項で紹介した方法でカバレッジを取得することももちろん可能ですが、Visual COBOL ではこれらの IDE 上で前項で 紹介した設定内容を直感的な操作で構成し、カバレッジ結果を専用のビューで確認することも可能です<sup>6</sup>。以下は IDE 上における Test Coverage の操作イメージとなります<sup>7</sup>。

カバレッジの有効化/無効化の切り替え:



<sup>6</sup> IDE 上での操作機能は Visual COBOL 2.3 より実装されています。

<sup>7</sup> ここでは Visual Studio 2015 を使ったイメージを紹介してますが、 Visual Studio の他のバージョン並びに Eclipse でも同様の操作で構成し、同様のビューを使って結果を閲覧す ることが可能です。

#### カバレッジ動作の構成:



Test Coverage にカバレッジを取得させつつアプリケーションを実行:



#### IDE 内でカバレッジの結果を確認:

| 1240         IF WK-HH-IN IS GREAT           1247         ADD 24 TO WK-HH-OU           1248         IF WK-HF-IN IS GREAT           1249         ADD 100 TO WK-HF-OU           1249         ADD 100 TO WK-FF-OI           1250         SUBTRACT I FROM WK           1251         IF WK-SS-IN IS GREAT           1253         SUBTRACT I FROM WK           1253         SUBTRACT I FROM WK-SS-OU           1255         ADD 60 TO WK-SS-OU           1255         ADD 60 TO WK-MF-OU           1256         SUBTRACT I FROM WK | ER THAN WK-HH-OU<br>ER THAN WK-FF-OU<br>SS-OUT.<br>ER THAN WK-SS-OU<br>ER THAN WK-SS-OU<br>ER THAN WK-MM-OU<br>HH-OUT. | 「<br>」 通過<br>「<br>て表え                    | 行、未通<br>示します。 | 過行を色を分け         |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------|------------------------------------------|---------------|-----------------|
| 1257 COMPUTE TEST-SECONDS<br>1258 3600) + ((WK-MM-OU                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | ROUNDED = ((WK-I<br>T - WK-MM-IN) * I                                                                                  | HH-OUT - WK-HH-IN)<br>50) + (WK-SS-OUT - | *             | 従来のレポートの生成も     |
| 100 % 👻                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |                                                                                                                        |                                          | _             | 可能です。           |
| Micro Focus Code Coverage                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |                                                                                                                        |                                          |               |                 |
| Results_2016-07-20_13_29_37.tcz                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |                                                                                                                        | - 12                                     | られ苦           | x 🛱 🧲 🗕 🚽       |
| 階層                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | 未力パー (プロック)                                                                                                            | 未力/(- (% プロック)                           | 力パー済み (フ      |                 |
| 🔺 🚟 Results_2016-07-20_13_29_37.tcz                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | 12                                                                                                                     | 4.26 %                                   | 270           | 35.74 A         |
| 🔺 💷 IntGntProject1                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | 12                                                                                                                     | 4.26 %                                   | 270           | 95.74 %         |
| 🔺 🔩 プログラム: ussteel                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | 12                                                                                                                     | 4.26 %                                   | 270           | 95.74 %         |
| ☞ 箭: 000-DUMMY                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | 9                                                                                                                      | 3.36 %                                   | 259           | 96.64 %         |
| ☞ 節: 580-CLOCK-OUT                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | 3                                                                                                                      | 21.43 %                                  | 11            | 78.57 %         |
| ☞ 段落: 010-OPENING-PARAGRAPH                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | 4                                                                                                                      | 36.36 %                                  | 7             | 63.64 %         |
| ☞ 段落: 020-CLOCKING-ROUTINE                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | 0                                                                                                                      | 0%                                       | 3             | 100 %           |
| ☞ 段落: 030-CLOCKING-ROUTINE-END                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | 0                                                                                                                      | 0%                                       | 2             | 100 %           |
| ☞ 段落: 040-TIME-TEST01                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | 0                                                                                                                      | 0%                                       | 3             | 100 %           |
|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |                                                                                                                        | 0%                                       |               | 100 %           |
| ◎ 段落: 060-TIME-TEST01-PACKED                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | 0                                                                                                                      | 0%                                       | 3             | 100 %           |
| ◎ 段落: 070-TIME-TEST01-PACKED-END                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | 0                                                                                                                      | 0%                                       | 2             | 100 %           |
| ◎ 段落: 080-TIME-TEST01-BINARY                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | 0                                                                                                                      | 0%                                       | 3             | 100 %           |
| ◎ 段落: 090-TIME-TEST01-BINARY-END                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | 0                                                                                                                      | 0%                                       | 2             | 100 %           |
| <ul> <li>         の書かれることである         がた。100 TIME TECTOR<br/>Micro Focus Code Coverage パッケージ マネージャー コンソール         </li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                | 1<br>/ エラー一覧 出力 M                                                                                                      | হন থ<br>licro Focus Code Analys          | a<br>iis      | 段落毎のサマリを確認できます。 |

# 4 コード分析 - COBOL 専用の静的コード解析ツール

プログラムのコンパイルが通れば、言語の規約に準じたコーディングができていることをある程度は担保できます。しかし、言語規約に従っ たプログラムであってもプログラム中の問題が完全に排除されたことにはなりません。例えば、以下のような問題はこの場合でも混入し得ます:

> 潜在バグとなり得るようなコーディング (例:未初期化の変数の利用、動作が規定されない命令の利用等)

> 性能の観点上、非効率なコーディング (例:ゾーン十進変数を使った算術演算、リトルエンディアン環境におけるビッグエンディアン変数の利用等)

> ソースの可読性を低下させるようなコーディング (例:デッドコード、GO TO 文の多用 等)

多くの場合、各開発プロジェクトでコーディング規準を設けてこのようなコーディングが行われないよう徹底します。このような管理作 業はプログラムを実行せずに支援するツール「静的コード解析ツール」が Java や .NET 言語向けに多くのベンダーより提供されていま す。静的コード解析ツールはツールの性質上、一般的に対象の言語に合わせて作りこまれます。Micro Focus は長年積み上げてきた COBOL の知見を活かして COBOL 専用の静的コード解析機能を Visual COBOL に搭載しました。

### 4.1 コード分析の基本動作

Visual COBOL が提供する静的コード解析機能「コード分析」は、Visual Studio IDE 並びに Eclipse IDE 上にて操作が可能です。 Visual COBOL をインストール<sup>8</sup>しますと、エディターのコンテキストメニューからビルドインされたルールに基づいた分析を実行できます。

Eclipse IDE における分析処理の開始イメージ:

|   |                                                                                                                                                                                              | 表示                                                                                                       | Alt+シフト+W( <u>W</u> ) ▶ | L                                                                               |                                                                                                                                                                                                                                       |
|---|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------|-------------------------|---------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|   | EXIT PROGRAM                                                                                                                                                                                 | 切り取り(T)                                                                                                  | Ctrl+X                  |                                                                                 |                                                                                                                                                                                                                                       |
|   | STOP RUN                                                                                                                                                                                     | コピー(C)                                                                                                   | Ctrl+C                  |                                                                                 |                                                                                                                                                                                                                                       |
|   |                                                                                                                                                                                              | 貼り付け(P)                                                                                                  | Ctrl+V                  |                                                                                 |                                                                                                                                                                                                                                       |
| 0 | DO-READ-RECORD SECTION.<br>OPEN INDUT BOOKFILE<br>IF LS-FILE-STATUS <> "00<br>INITIALIZE LNK-B-DET<br>MOVE ALL '*' TO LNK-<br>MOVE LS-FILE-STATUS<br>EXIT SECTION<br>END-IF<br>EVALUATE TRUE | クイック・フィックス(<br>右ヘシフト(5)         左ヘシフト(H)         ソース         二         番号を付け直す         二         番号を削除する | Q) Ctrl+1               | 50<br>50<br>50<br>50<br>50<br>50<br>50<br>50<br>50<br>50                        | Operations with Different Decimal Precision - Conditions<br>ALTER Statements<br>Find PERFORM THRU Usage<br>GO TO Statements Targeting non-EXIT Paragraphs<br>Uninitialized Data Items<br>GO TO statement outside of PERFORMed section |
|   | WHEN LNK-B-STOCKNO <> S<br>MOVE LNK-B-STOCKNO T<br>READ BOOKFILE<br>WHEN LNK-B-TITLE <> SPA<br>MOVE LNK-B-TITLE TO                                                                           | プロファイル(P)<br>デバッグ(D)<br>実行(R)<br>Validate                                                                | ><br>><br>>             | 48<br>48<br>48<br>48                                                            | Look for MOVE statements with loss of sign<br>Explicit Scope Terminators<br>Dead Statements<br>Unused Data                                                                                                                            |
|   | READ BOOKFILE KEY IS<br>WHEN LNK-B-AUTHOR <> SP<br>MOVE LNK-B-AUTHOR TO<br>READ BOOKFILE KEY IS                                                                                              | 置換対象(L)<br>COBOL コピー表示で開<br>コンテンツ・アシスト                                                                   | ・<br>Iく F4<br>Ctrl+スペース | ()<br>()<br>()<br>()<br>()<br>()<br>()<br>()<br>()<br>()<br>()<br>()<br>()<br>( | COBOL Performance<br>General Queries<br>Coding Standards                                                                                                                                                                              |
|   | WHEN OTHER<br>*>NO KEY SPECIFIE                                                                                                                                                              | <ul> <li>レコード レイアウト コード分析</li> </ul>                                                                     | 7アイルを作成<br>▶            |                                                                                 | Within Entire Program<br>その他                                                                                                                                                                                                          |
|   | MOVE 23 TO LS-FILE                                                                                                                                                                           | Team                                                                                                     | •                       |                                                                                 |                                                                                                                                                                                                                                       |

#### Dead Statements の実行例:

| •                                            | END-EVALUATE<br>EXIT PROGRAM<br>STOP RUN<br>DO-READ-RECORD SECTION.<br>OPEN INPUT BOOKFILE<br>IF LS-FILE-STATUS <> "00"<br>INITIALIZE LNK-B-DETAILS | 分析対象の<br>PROGRAM<br>STOP RUN<br>なると検出さ | プログラムはサブプログラムとなるため、EXIT<br>にてプログラムは抜けます。そのため、次行の<br>N についてはフロー上未到達 (Dead Statement) に<br>hました。 |
|----------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------|------------------------------------------------------------------------------------------------|
| •                                            | MOVE ALL '*' TO LNK-B-TEXT-DETAILS<br>MOVE LS-FILE-STATUS TO LNK-FILE-STATUS<br>EXIT SECTION                                                        |                                        |                                                                                                |
| בילא ביל | レ 🔝 問題 🧔 タスク 🔝 プロパティー 🎴 コード カバレッ                                                                                                                    | ジ 🛍 コード分析 🛙                            |                                                                                                |
|                                              | Statements - Within Entire Program (1個の一致)                                                                                                          |                                        | _                                                                                              |
| ⊿ 🖻 BC                                       | DOK.cbl - ¥NativeBook (1 個の一致)<br>STOP RUN                                                                                                          |                                        |                                                                                                |

<sup>8</sup> コード分析機能は Visual COBOL 2.3 より実装された機能です。

分析対象は各ソース単位のみならず、Visual Studio のプロジェクトや Eclipse のプロジェクトへと対象を拡大することも可能です。

| Eclipse of | りプロジェクト | 単位による分 | 析処理イメージ | : |
|------------|---------|--------|---------|---|
|------------|---------|--------|---------|---|



Visual COBOL が提供するコード分析機能はビルド実行後に自動で処理させる構成オプションを用意しています。これにより、コンパイ ルチェック並びに静的コード解析が済んだソースが常に得られるよう環境を整えることが可能です。

Eclipse プロジェクトにおけるプロジェクト設定イメージ:

| 🔤 プロパティ: NativeBook |                                          |
|---------------------|------------------------------------------|
| フィルタ入力              | コード分析                                    |
| ▶ リソース              |                                          |
| Micro Focus         | New Configuration [使用中]                  |
| ビルドパス               | • • •                                    |
| ▲ ビルド構成             |                                          |
| COBOL               | ▼構成の固有な設定を可能にする(C)                       |
| SQL プリプロセ           |                                          |
| コード分析               |                                          |
| 追加のプリプロ             | ▼ ビルド後にアクティブなルールセットを実行する                 |
| イベント                |                                          |
| ビルド環境               |                                          |
| ▶ リンク               | COBOL Performance (3 / 3 - fly)/a/v=/v)  |
| ▷ プロジェクト設定          | General Queries (3 / 3 - 有効なルール)         |
| ▷ 実行時構成             | Coding Standards (2 / 2 - 有効なルール)        |
| Project Facets      | 📄 Within Entire Program (2 / 2 - 有効なルール) |
| Server              | CodingRuleConformation (3 / 4 - 有効なルール)  |
| Task Repository     |                                          |

## 4.2 Enterprise Analyzer 上のクエリールールの活用

Visual COBOL にビルドインされたコード分析の解析ルールは一般的な開発プロジェクトで検討され得る基本的なルールが取り揃えら れています。しかし、実際の開発現場で用意されるコーディング規準はこのルールだけに留まらないはずです。このような要件をも満た すための製品として Micro Focus は Enterprise Analyzer<sup>9</sup>という製品を開発・販売しています。この Enterprise Analyzer には Visual COBOL のコード分析機能よりも広い観点で解析ルールが取り揃えられています。

Enterprise Analyzer にビルドインされたクエリールールのイメージ:



Enterprise Analyzer にデフォルトで用意されていない観点で解析したい場合は、ルールをカスタマイズ作成することも可能です。

カスタマイズ作成画面イメージ:

| Find <u>Variable</u><br>is not Init                               |                                                                                              | $\sim$ |   |
|-------------------------------------------------------------------|----------------------------------------------------------------------------------------------|--------|---|
| and<br>such that <u>Var</u><br><u>and</u><br>which <u>has rek</u> | <u>Usage</u> = "OUT"<br><u>ited</u> Declaration (as Declaration) which <u>has no related</u> |        |   |
| Variable (a<br><u>or</u>                                          | ss Instances)<br>such that <u>VarUsage</u> = "IN"                                            |        |   |
|                                                                   | such that <u>VarUsage</u> = "INOUT"<br>条件                                                    |        |   |
| 適用先(A): 関係                                                        |                                                                                              |        | • |
| -関係タイプ( <u>B</u> )<br>Conditions                                  | エンティティー・タイプ(5)                                                                               |        |   |
| Declaration<br>File Name For                                      | Conditional Expression (with all descendants)<br>Constant<br>Data Dest                       |        | ^ |
| Port<br>ネスト済み<br>含む<br>ユ                                          | Data Port Target<br>Data Section<br>DD Name                                                  |        | ≡ |
| 親                                                                 | De France<br>Declaration (with all descendants)<br>Division<br>End of paragraph              |        |   |
|                                                                   | Expression                                                                                   |        | > |
| □ 関係を持っていない((                                                     | 0                                                                                            |        |   |
|                                                                   | <b>a</b> //                                                                                  | ير ملا |   |

••••••

 9 Enterprise Analyzerは COBOL ソースのコード解析以外にも様々な機能が搭載されています。本製品の詳細については下記ページをご覧ください。同ページではデータシートやデモ 動画にて製品概要を紹介しています。
 http://www.microfocus.co.jp/products/modernization/enterprise-analyzer/ カスタマイズ作成されたルールを含め Enterprise Analyzer 上のルールは Visual COBOL と共有することが可能です。即ち、例えば下記のようなシナリオの実現も可能です:

① 開発プロジェクトリーダーが Enterprise Analyzer を所有し、コーディング規準に基づいたルールを用意

- ② Enterprise Analyzer で用意されたルールをエクスポート
- ③ エクスポートしたルールを Visual COBOL のライセンスを持った開発メンバへ配布
- ④ 開発メンバは Enterprise Analyzer のルールを自身の開発環境に取り込みコーディング規準に従いコーディング

以下に Enterprise Analyzer 上でカスタマイズ作成したルールを Visual COBOL に取り込む手順を記します。

カスタマイズ作成したルールをエクスポート:



エクスポートされたルールを Visual COBOL for Eclipse で取り込み:

| 名前                                | メニュー表示 | 場所                                                  |                                    |
|-----------------------------------|--------|-----------------------------------------------------|------------------------------------|
| 🕨 📝 🧇 COBOL Performance           | (よい)   | ビルトイン                                               |                                    |
| V Coding Standards                | (はい)   | ビルトイン                                               |                                    |
| Image: CodingRuleConformation     |        | C:¥work¥EA_Rules¥CodingRuleConformation.General.xml |                                    |
| 👿 🐁 Computation_with_ZonedDecimal |        | C:¥work¥EA_Rules¥                                   | CodingRuleConformation.General.xml |
| 👿 🐁 Value_Clause_Used_In_FD       |        | C:¥work¥EA_Rules¥                                   | CodingRuleConformation.General.xml |
| 🕨 📝 🍫 General Queries             | (はい)   | ビルトイン                                               |                                    |
| 🕨 📝 🧇 Within Entire Program       | はい     | ビルトイン                                               |                                    |
|                                   |        |                                                     |                                    |
|                                   |        |                                                     |                                    |
| AB.                               |        |                                                     | インポートされたルール                        |
| -m.                               |        |                                                     |                                    |
|                                   |        | I                                                   |                                    |
|                                   |        |                                                     |                                    |
|                                   |        |                                                     |                                    |

#### 4.3 MSBuild, Ant による実行

Visual COBOL のコード分析機能は各プログラムソースに対して分析用の一時ファイルをディスク上に作成の上、分析処理します。その ため、分析対象のソースのボリュームによっては大幅な処理時間を要する可能性も否めません。このような状況下でビルドの度に分析処 理を流すことは非効率であり、生産性の低下を招く恐れがあります。Visual COBOL は先の章で紹介したビルドツール MSBuild 及び Ant 経由でコード分析処理を発行するインターフェースを備えています。即ち、CI ツールによるコード分析処理の発行も可能なため、ビ ルドが済んだ時点でコード分析処理を自動でキックさせる仕組みを用意することも可能です。

以下に Jenkins 経由でコード分析を処理させる設定イメージを紹介します。

Ant を使って Eclipse COBOL プロジェクトをコード分析させる設定例<sup>10</sup>:

| Antの呼び  | 出し                                            | analyze: コード分析処理を発行させます。<br>-DanalysisEailOnError: 何か検出された場合ビルドをエラーとします。 |
|---------|-----------------------------------------------|--------------------------------------------------------------------------|
| 使用するAnt | Ant_192_in_line_with_VC231                    |                                                                          |
| ターゲット   | -lib "C:\Program Files (x86)\Micro Focus\Visu | ıal COBOL\bin\mfant.jar" analyze -DanalysisFailOnErro≔true               |

<sup>10</sup> Ant 経由で命令を発行する場合、Eclipse プロジェクトに設定されたルールに基づいて分析をします。MSBuild を使う場合は、Visual Studio に設定されたルールもしくは MSBuild にオプションとして渡すルールを実行するか選択可能です。

#### ルールに基づいたクエリで問題が検出された場合の出力例:

perform.analysis.cfg.New\_Configuration: [analysis] Micro Focus Enterprise Analyzer Command - Line Tool, Version 3.6.1 [analysis] Copyright (C) Micro Focus 2015, All Rights Reserved. [analysis] Analysis started.\_\_\_ [analysis] Cityorkight (0) micro rocus 2013, Arr regits Reserved. [analysis] Analyzing CHECKDATE ... [analysis] Analyzing CUISCRN ... [analysis] Analyzing UDI3GEP ... [analysis] CityorkiyopyDevToolsiywsiNativeCOBOLiUDT0036P.cbl(223,28,223,42): error MFCA0001: CodingRuleConformation: Computation with ZonedDecimal: WK-DEPRECIATION [analysis] CityorkiyopyDevToolsiywsiNativeCOBOLiUDT0036P.cbl(228,28,226,42): error MFCA0001: CodingRuleConformation: Computation with ZonedDecimal: WK-DEPRECIATION [analysis] CityorkiyopyDevToolsiywsiNativeCOBOLiUDT0036P.cbl(227,37,227,51): error MFCA0001: CodingRuleConformation: Computation with ZonedDecimal: WK-DEPRECIATION [analysis] CityorkiyopyDevToolsiywsiNativeCOBOLiUDT0036P.cbl(221,37,227,51): error MFCA0001: CodingRuleConformation: Computation with ZonedDecimal: WK-DEPRECIATION [analysis] CityorkiyopyDevToolsiywsiNativeCOBOLiUDT0036P.cbl(221,32,231,42): error MFCA0001: CodingRuleConformation: Computation with ZonedDecimal: WK-DEPRECIATION [analysis] CityorkiyopyDevToolsiywsiNativeCOBOLiUDT0036P.cbl(232,37,232,51): error MFCA0001: CodingRuleConformation: Computation\_with\_ZonedDecimal: WK-DEPRECIATION [analysis] CityorkiyopiDevToolsiywsiNativeCOBOLiUDT0036P.cbl(232,37,232,51): error MFCA0001: CodingRuleConformation: Computation\_with\_ZonedDecimal: WK-DEPRECIATION [analysis] Analysis Complete. [analysis] Analysis Complete. [analysis] [analysis] [analysis] Done: 00:00:05 [analysis] Finished: FAILURE [delete] Deleting directory C:¥work¥wp¥DevTools¥ws¥NativeCOBOL¥.analysis¥data BUILD FAILED C:¥work¥wp¥DevTools¥ws¥NativeCOBOL¥.cobolBuild:242: C:¥work¥wp¥DevTools¥ws¥NativeCOBOL¥.cobolBuild:249: The following error occurred while executing this line: C:¥work¥wp¥DevTools¥ws¥NativeCOBOL¥.cobolBuild:145: Analysis failure Total time: 9 seconds Build step 'Antの呼び出し' marked build as failure Finished: FAILURE

ルールに基づいたクエリで問題が検出されなかった場合の出力例:

perform.analysis.cfg.New\_Configuration: [analysis] Micro Focus Enterprise Analyzer Command - Line Tool, Version 3.6.1 [analysis] Copyright (C) Micro Focus 2015, All Rights Reserved. [analysis] Analysis started. [analysis] Analysis stattet. [analysis] Analyzing CHECKDATE ... [analysis] Analyzing CUISCRN ... [analysis] Analyzing UDT0036P ... [analysis] Analysis Complete. [analysis] Finalysis Complete. [analysis] [analysis] Done: 00:00:06 [analysis] Finished: SUCCESS [delete] Deleting directory C:¥work¥wp¥DevTools¥ws¥NativeCOBOL¥.analysis¥data BUILD SUCCESSFUL Total time: 9 seconds Finished: SUCCESS

上の結果と Jenkins のビルド結果が紐付きます:

| 🚕 ビル        | レド履歴             | <u>推移</u> |
|-------------|------------------|-----------|
| find        |                  | Х         |
| ) <u>#4</u> | 2016/07/20 18:16 |           |
| 🥥 <u>#3</u> | 2016/07/20 18:14 |           |

# 5 mfunit - COBOL 専用の xUnit 系単体テストフレームワーク

mfunit は Visual COBOL に搭載された xUnit 系の単体テストフレームワークです<sup>11</sup>。 xUnit はオブジェクト指向型の単体テストフレー ムワーク SUnit に起源を持つ JUnit や RUnit 等の単体テストフレームワークの総称です。 mfunit は xUnit の設計アーキテクチャーや 仕組みは取り入れつつも COBOL 開発者にとって扱いやすい手続き型の COBOL を対象とした単体テストフレームワークという設計思想 の下、開発されました。

本章ではこの mfunit の基本動作について紹介いたします。

#### 5.1 mfunit の基本機能

mfunitでは以下のような手順でテストを進めます。

<sup>11</sup> 本機能は本書執筆時点では Technology Preview 扱いとなっており、サポート対象外の機能です。お試しになる際は、マニュアルをご参照いただき最新のステータスをご確認の上ご 自身の判断でご利用ください。尚、本機能は Visual COBOL 2.3 より実装された機能です。

#### ① Test Fixture の作成

Test Fixture(テストを実行、成功させるために必要な状態や前提条件の集合)を COBOL でコーディングして用意します。COBOL プログラム中に複数の Test Fixture を含めることも可能です。

#### ② Test Fixture をコンパイル

Windows 版であればダイナミックリンクライブラリ (.dll) に、Linux/UNIX 版であれば呼び出し可能な共有オブジェクト (.so) にコンパイ ルします。

#### ③ テストを実行

Windows 版であれば mfurun、Linux/UNIX 版であれば cobmfurun32/ cobmfurun64 が mfunit の Test Runner として用意されています<sup>12</sup>。

### ④ 結果を確認

結果は標準出力に出力されます。加えて、テキストファイルや JUnit 互換のある XML 形式で出力することも可能です。

ここでは簡便化のために用意した電卓の機能を処理するサブモジュールを mfunit を使って Windows 環境上で幾つか単体テストするようすを紹介します。

| OT LNK-PARAM.<br>O3 LNK-OPERATOR PIC 9.<br>O3 LNK-TERM1 PIC S9(4) COMP-5.<br>O3 LNK-TERM2 PIC S9(6).<br>PROCEDURE DIVISION USING LNK-PARAM.<br>MAIN-PARA.<br>EVALUATE LNK-OPERATOR<br>WHEN 1<br>PERFORM ADDITION-PARA<br>WHEN 2<br>PERFORM SUBTRACTION-PARA<br>WHEN 3<br>PERFORM MULTIPLICATION-PARA<br>WHEN 4<br>PERFORM DIVISION-PARA<br>END-EVALUATE.<br>GOBACK.<br>ADDITION-PARA.<br>COMPUTE LNK-RES = LNK-TERM1 + LNK-TERM2<br>EXIT.<br>SUBTRACTION-PARA.<br>COMPUTE LNK-RES = LNK-TERM1 - LNK-TERM2<br>EXIT.<br>MULTIPLICATION-PARA.<br>COMPUTE LNK-RES = LNK-TERM1 * LNK-TERM2<br>EXIT.<br>MULTIPLICATION-PARA.<br>COMPUTE LNK-RES = LNK-TERM1 * LNK-TERM2<br>EXIT.<br>DIVISION-PARA.<br>COMPUTE LNK-RES = LNK-TERM1 * LNK-TERM2<br>EXIT.                                                                                                   | DATA DIVISION.<br>LINKAGE SECTION. |                    |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------|--------------------|
| 03 LNK-OPERATOR PIC 9.<br>03 LNK-TERM1 PIC S9(4) COMP-5.<br>03 LNK-TERM2 PIC S9(4) COMP-5.<br>03 LNK-RES PIC S9(6).<br>PROCEDURE DIVISION USING LNK-PARAM.<br>MAIN-PARA.<br>EVALUATE LNK-OPERATOR<br>WHEN 1<br>PERFORM ADDITION-PARA<br>WHEN 2<br>PERFORM SUBTRACTION-PARA<br>WHEN 3<br>PERFORM MULTIPLICATION-PARA<br>WHEN 4<br>PERFORM MULTIPLICATION-PARA<br>WHEN 4<br>PERFORM DIVISION-PARA<br>END-EVALUATE.<br>GOBACK.<br>ADDITION-PARA.<br>COMPUTE LNK-RES = LNK-TERM1 + LNK-TERM2<br>EXIT.<br>SUBTRACTION-PARA.<br>COMPUTE LNK-RES = LNK-TERM1 - LNK-TERM2<br>EXIT.<br>MULTIPLICATION-PARA.<br>COMPUTE LNK-RES = LNK-TERM1 * LNK-TERM2<br>EXIT.<br>DIVISION-PARA.<br>COMPUTE LNK-RES = LNK-TERM1 * LNK-TERM2<br>EXIT.<br>DIVISION-PARA.                                                                                                     |                                    |                    |
| <ul> <li>D3 LNK-TERM1 PIC S9(4) COMP-5.</li> <li>D3 LNK-TERM2 PIC S9(4) COMP-5.</li> <li>D3 LNK-RES PIC S9(6).</li> <li>PROCEDURE DIVISION USING LNK-PARAM.</li> <li>MAIN-PARA.</li> <li>EVALUATE LNK-OPERATOR</li> <li>WHEN 1</li> <li>PERFORM ADDITION-PARA</li> <li>WHEN 2</li> <li>PERFORM SUBTRACTION-PARA</li> <li>WHEN 3</li> <li>PERFORM MULTIPLICATION-PARA</li> <li>WHEN 4</li> <li>PERFORM DIVISION-PARA</li> <li>END-EVALUATE.</li> <li>GOBACK.</li> <li>ADDITION-PARA.</li> <li>COMPUTE LNK-RES = LNK-TERM1 + LNK-TERM2</li> <li>EXIT.</li> <li>SUBTRACTION-PARA.</li> <li>COMPUTE LNK-RES = LNK-TERM1 - LNK-TERM2</li> <li>EXIT.</li> <li>MULTIPLICATION-PARA.</li> <li>COMPUTE LNK-RES = LNK-TERM1 * LNK-TERM2</li> <li>EXIT.</li> <li>DIVISION-PARA.</li> <li>COMPUTE LNK-RES = LNK-TERM1 * LNK-TERM2</li> <li>EXIT.</li> </ul>    | 03 LNK-OPERATOR                    |                    |
| <ul> <li>DIS LINK-TERMIZ</li> <li>DIS LINK-RES</li> <li>PIC S9(4) COMP-S.</li> <li>O3 LNK-RES</li> <li>PIC S9(6).</li> <li>PROCEDURE DIVISION USING LNK-PARAM.</li> <li>MAIN-PARA.</li> <li>EVALUATE LNK-OPERATOR</li> <li>WHEN 1</li> <li>PERFORM ADDITION-PARA</li> <li>WHEN 2</li> <li>PERFORM SUBTRACTION-PARA</li> <li>WHEN 3</li> <li>PERFORM MULTIPLICATION-PARA</li> <li>WHEN 4</li> <li>PERFORM DIVISION-PARA</li> <li>END-EVALUATE.</li> <li>GOBACK.</li> <li>ADDITION-PARA.</li> <li>COMPUTE LNK-RES = LNK-TERM1 + LNK-TERM2</li> <li>EXIT.</li> <li>SUBTRACTION-PARA.</li> <li>COMPUTE LNK-RES = LNK-TERM1 - LNK-TERM2</li> <li>EXIT.</li> <li>MULTIPLICATION-PARA.</li> <li>COMPUTE LNK-RES = LNK-TERM1 * LNK-TERM2</li> <li>EXIT.</li> <li>DIVISION-PARA.</li> <li>COMPUTE LNK-RES = LNK-TERM1 * LNK-TERM2</li> <li>EXIT.</li> </ul> |                                    | PIC 59(4) COMP-5.  |
| <ul> <li>US LINK-RES PIC S9(6).</li> <li>PROCEDURE DIVISION USING LNK-PARAM.</li> <li>MAIN-PARA.</li> <li>EVALUATE LNK-OPERATOR</li> <li>WHEN 1</li> <li>PERFORM ADDITION-PARA</li> <li>WHEN 2</li> <li>PERFORM SUBTRACTION-PARA</li> <li>WHEN 3</li> <li>PERFORM MULTIPLICATION-PARA</li> <li>WHEN 4</li> <li>PERFORM DIVISION-PARA</li> <li>END-EVALUATE.</li> <li>GOBACK.</li> <li>ADDITION-PARA.</li> <li>COMPUTE LNK-RES = LNK-TERM1 + LNK-TERM2 EXIT.</li> <li>SUBTRACTION-PARA.</li> <li>COMPUTE LNK-RES = LNK-TERM1 - LNK-TERM2 EXIT.</li> <li>MULTIPLICATION-PARA.</li> <li>COMPUTE LNK-RES = LNK-TERM1 * LNK-TERM2 EXIT.</li> <li>DIVISION-PARA.</li> <li>COMPUTE LNK-RES = LNK-TERM1 * LNK-TERM2 EXIT.</li> </ul>                                                                                                                       |                                    | PIC 59(4) COMP-5.  |
| PROCEDURE DIVISION USING ENR-PARAM.<br>MAIN-PARA.<br>EVALUATE LNK-OPERATOR<br>WHEN 1<br>PERFORM ADDITION-PARA<br>WHEN 2<br>PERFORM SUBTRACTION-PARA<br>WHEN 3<br>PERFORM MULTIPLICATION-PARA<br>WHEN 4<br>PERFORM DIVISION-PARA<br>END-EVALUATE.<br>GOBACK.<br>ADDITION-PARA.<br>COMPUTE LNK-RES = LNK-TERM1 + LNK-TERM2<br>EXIT.<br>SUBTRACTION-PARA.<br>COMPUTE LNK-RES = LNK-TERM1 - LNK-TERM2<br>EXIT.<br>MULTIPLICATION-PARA.<br>COMPUTE LNK-RES = LNK-TERM1 * LNK-TERM2<br>EXIT.<br>DIVISION-PARA.<br>COMPUTE LNK-RES = LNK-TERM1 * LNK-TERM2<br>EXIT.                                                                                                                                                                                                                                                                                       |                                    | PIC S9(6).         |
| MAIN-PARA.<br>EVALUATE LNK-OPERATOR<br>WHEN 1<br>PERFORM ADDITION-PARA<br>WHEN 2<br>PERFORM SUBTRACTION-PARA<br>WHEN 3<br>PERFORM MULTIPLICATION-PARA<br>WHEN 4<br>PERFORM DIVISION-PARA<br>END-EVALUATE.<br>GOBACK.<br>ADDITION-PARA.<br>COMPUTE LNK-RES = LNK-TERM1 + LNK-TERM2<br>EXIT.<br>SUBTRACTION-PARA.<br>COMPUTE LNK-RES = LNK-TERM1 - LNK-TERM2<br>EXIT.<br>MULTIPLICATION-PARA.<br>COMPUTE LNK-RES = LNK-TERM1 * LNK-TERM2<br>EXIT.<br>DIVISION-PARA.<br>COMPUTE LNK-RES = LNK-TERM1 * LNK-TERM2<br>EXIT.                                                                                                                                                                                                                                                                                                                              | PROCEDURE DIVISION USING           | LNK-PARAM.         |
| <ul> <li>EVALUATE LINK-OPERATOR</li> <li>WHEN 1</li> <li>PERFORM ADDITION-PARA</li> <li>WHEN 2</li> <li>PERFORM SUBTRACTION-PARA</li> <li>WHEN 3</li> <li>PERFORM MULTIPLICATION-PARA</li> <li>WHEN 4</li> <li>PERFORM DIVISION-PARA</li> <li>END-EVALUATE.</li> <li>GOBACK.</li> <li>ADDITION-PARA.</li> <li>COMPUTE LNK-RES = LNK-TERM1 + LNK-TERM2<br/>EXIT.</li> <li>SUBTRACTION-PARA.</li> <li>COMPUTE LNK-RES = LNK-TERM1 - LNK-TERM2<br/>EXIT.</li> <li>MULTIPLICATION-PARA.</li> <li>COMPUTE LNK-RES = LNK-TERM1 * LNK-TERM2<br/>EXIT.</li> <li>DIVISION-PARA.</li> <li>COMPUTE LNK-RES = LNK-TERM1 * LNK-TERM2<br/>EXIT.</li> </ul>                                                                                                                                                                                                       |                                    | 2                  |
| <ul> <li>WHEN 1</li> <li>PERFORM ADDITION-PARA</li> <li>WHEN 2</li> <li>PERFORM SUBTRACTION-PARA</li> <li>WHEN 3</li> <li>PERFORM MULTIPLICATION-PARA</li> <li>WHEN 4</li> <li>PERFORM DIVISION-PARA</li> <li>END-EVALUATE.</li> <li>GOBACK.</li> <li>ADDITION-PARA.</li> <li>COMPUTE LNK-RES = LNK-TERM1 + LNK-TERM2<br/>EXIT.</li> <li>SUBTRACTION-PARA.</li> <li>COMPUTE LNK-RES = LNK-TERM1 - LNK-TERM2<br/>EXIT.</li> <li>MULTIPLICATION-PARA.</li> <li>COMPUTE LNK-RES = LNK-TERM1 * LNK-TERM2<br/>EXIT.</li> <li>DIVISION-PARA.</li> <li>COMPUTE LNK-RES = LNK-TERM1 * LNK-TERM2<br/>EXIT.</li> </ul>                                                                                                                                                                                                                                       | EVALUATE LINK-OPERATOR             | 1                  |
| <ul> <li>PERFORM ADDITION-PARA</li> <li>WHEN 2</li> <li>PERFORM SUBTRACTION-PARA</li> <li>WHEN 3</li> <li>PERFORM MULTIPLICATION-PARA</li> <li>WHEN 4</li> <li>PERFORM DIVISION-PARA</li> <li>END-EVALUATE.</li> <li>GOBACK.</li> <li>ADDITION-PARA.</li> <li>COMPUTE LNK-RES = LNK-TERM1 + LNK-TERM2 EXIT.</li> <li>SUBTRACTION-PARA.</li> <li>COMPUTE LNK-RES = LNK-TERM1 - LNK-TERM2 EXIT.</li> <li>MULTIPLICATION-PARA.</li> <li>COMPUTE LNK-RES = LNK-TERM1 * LNK-TERM2 EXIT.</li> <li>DIVISION-PARA.</li> <li>COMPUTE LNK-RES = LNK-TERM1 * LNK-TERM2 EXIT.</li> </ul>                                                                                                                                                                                                                                                                       | WHEN 1                             |                    |
| <ul> <li>WHEN 2</li> <li>PERFORM SUBTRACTION-PARA</li> <li>WHEN 3</li> <li>PERFORM MULTIPLICATION-PARA</li> <li>WHEN 4</li> <li>PERFORM DIVISION-PARA</li> <li>END-EVALUATE.</li> <li>GOBACK.</li> <li>ADDITION-PARA.</li> <li>COMPUTE LNK-RES = LNK-TERM1 + LNK-TERM2</li> <li>EXIT.</li> <li>SUBTRACTION-PARA.</li> <li>COMPUTE LNK-RES = LNK-TERM1 - LNK-TERM2</li> <li>EXIT.</li> <li>MULTIPLICATION-PARA.</li> <li>COMPUTE LNK-RES = LNK-TERM1 * LNK-TERM2</li> <li>EXIT.</li> <li>DIVISION-PARA.</li> <li>COMPUTE LNK-RES = LNK-TERM1 * LNK-TERM2</li> <li>EXIT.</li> </ul>                                                                                                                                                                                                                                                                  | PERFORM ADDITION-PAF               | łA                 |
| <ul> <li>PERFORM SUBTRACTION-PARA</li> <li>WHEN 3</li> <li>PERFORM MULTIPLICATION-PARA</li> <li>WHEN 4</li> <li>PERFORM DIVISION-PARA</li> <li>END-EVALUATE.</li> <li>GOBACK.</li> <li>ADDITION-PARA.</li> <li>COMPUTE LNK-RES = LNK-TERM1 + LNK-TERM2</li> <li>EXIT.</li> <li>SUBTRACTION-PARA.</li> <li>COMPUTE LNK-RES = LNK-TERM1 - LNK-TERM2</li> <li>EXIT.</li> <li>MULTIPLICATION-PARA.</li> <li>COMPUTE LNK-RES = LNK-TERM1 * LNK-TERM2</li> <li>EXIT.</li> <li>DIVISION-PARA.</li> <li>COMPUTE LNK-RES = LNK-TERM1 * LNK-TERM2</li> <li>EXIT.</li> </ul>                                                                                                                                                                                                                                                                                  | WHEN 2                             |                    |
| <ul> <li>WHEN 3</li> <li>PERFORM MULTIPLICATION-PARA</li> <li>WHEN 4</li> <li>PERFORM DIVISION-PARA</li> <li>END-EVALUATE.</li> <li>GOBACK.</li> <li>ADDITION-PARA.</li> <li>COMPUTE LNK-RES = LNK-TERM1 + LNK-TERM2<br/>EXIT.</li> <li>SUBTRACTION-PARA.</li> <li>COMPUTE LNK-RES = LNK-TERM1 - LNK-TERM2<br/>EXIT.</li> <li>MULTIPLICATION-PARA.</li> <li>COMPUTE LNK-RES = LNK-TERM1 * LNK-TERM2<br/>EXIT.</li> <li>DIVISION-PARA.</li> <li>COMPUTE LNK-RES = LNK-TERM1 * LNK-TERM2<br/>EXIT.</li> </ul>                                                                                                                                                                                                                                                                                                                                        |                                    | N-PARA             |
| <ul> <li>PERFORM MULTIPLICATION-PARA</li> <li>WHEN 4</li> <li>PERFORM DIVISION-PARA</li> <li>END-EVALUATE.</li> <li>GOBACK.</li> <li>ADDITION-PARA.</li> <li>COMPUTE LNK-RES = LNK-TERM1 + LNK-TERM2</li> <li>EXIT.</li> <li>SUBTRACTION-PARA.</li> <li>COMPUTE LNK-RES = LNK-TERM1 - LNK-TERM2</li> <li>EXIT.</li> <li>MULTIPLICATION-PARA.</li> <li>COMPUTE LNK-RES = LNK-TERM1 * LNK-TERM2</li> <li>EXIT.</li> <li>DIVISION-PARA.</li> <li>COMPUTE LNK PES = LNK TERM1 / LNK TERM2</li> </ul>                                                                                                                                                                                                                                                                                                                                                   | WHEN 3                             |                    |
| WHEN 4<br>PERFORM DIVISION-PARA<br>END-EVALUATE.<br>GOBACK.<br>ADDITION-PARA.<br>COMPUTE LNK-RES = LNK-TERM1 + LNK-TERM2<br>EXIT.<br>SUBTRACTION-PARA.<br>COMPUTE LNK-RES = LNK-TERM1 - LNK-TERM2<br>EXIT.<br>MULTIPLICATION-PARA.<br>COMPUTE LNK-RES = LNK-TERM1 * LNK-TERM2<br>EXIT.<br>DIVISION-PARA.<br>COMPUTE LNK PES = LNK TERM1 / LNK TERM2                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |                                    | UN-PARA            |
| END-EVALUATE.<br>GOBACK.<br>ADDITION-PARA.<br>COMPUTE LNK-RES = LNK-TERM1 + LNK-TERM2<br>EXIT.<br>SUBTRACTION-PARA.<br>COMPUTE LNK-RES = LNK-TERM1 - LNK-TERM2<br>EXIT.<br>MULTIPLICATION-PARA.<br>COMPUTE LNK-RES = LNK-TERM1 * LNK-TERM2<br>EXIT.<br>DIVISION-PARA.<br>COMPUTE LNK PES = LNK TERM1 / LNK TERM2                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |                                    |                    |
| GOBACK.<br>ADDITION-PARA.<br>COMPUTE LNK-RES = LNK-TERM1 + LNK-TERM2<br>EXIT.<br>SUBTRACTION-PARA.<br>COMPUTE LNK-RES = LNK-TERM1 - LNK-TERM2<br>EXIT.<br>MULTIPLICATION-PARA.<br>COMPUTE LNK-RES = LNK-TERM1 * LNK-TERM2<br>EXIT.<br>DIVISION-PARA.<br>COMPUTE LNK RES = LNK TERM1 / LNK TERM2                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |                                    | А                  |
| ADDITION-PARA.<br>COMPUTE LNK-RES = LNK-TERM1 + LNK-TERM2<br>EXIT.<br>SUBTRACTION-PARA.<br>COMPUTE LNK-RES = LNK-TERM1 - LNK-TERM2<br>EXIT.<br>MULTIPLICATION-PARA.<br>COMPUTE LNK-RES = LNK-TERM1 * LNK-TERM2<br>EXIT.<br>DIVISION-PARA.<br>COMPUTE LNK RES = LNK TERM1 / LNK TERM2                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | END-EVALUATE.                      |                    |
| ADDITION-PARA.<br>COMPUTE LNK-RES = LNK-TERM1 + LNK-TERM2<br>EXIT.<br>SUBTRACTION-PARA.<br>COMPUTE LNK-RES = LNK-TERM1 - LNK-TERM2<br>EXIT.<br>MULTIPLICATION-PARA.<br>COMPUTE LNK-RES = LNK-TERM1 * LNK-TERM2<br>EXIT.<br>DIVISION-PARA.<br>COMPUTE LNK RES = LNK TERM1 / LNK TERM2                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |                                    |                    |
| EXIT.<br>SUBTRACTION-PARA.<br>COMPUTE LNK-RES = LNK-TERM1 - LNK-TERM2<br>EXIT.<br>MULTIPLICATION-PARA.<br>COMPUTE LNK-RES = LNK-TERM1 * LNK-TERM2<br>EXIT.<br>DIVISION-PARA.<br>COMPUTE LNK RES = LNK TERM1 / LNK TERM2                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |                                    |                    |
| SUBTRACTION-PARA.<br>COMPUTE LNK-RES = LNK-TERM1 - LNK-TERM2<br>EXIT.<br>MULTIPLICATION-PARA.<br>COMPUTE LNK-RES = LNK-TERM1 * LNK-TERM2<br>EXIT.<br>DIVISION-PARA.<br>COMPUTE LNK RES = LNK TERM1 / LNK TERM2                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | EXIT                               |                    |
| COMPUTE LNK-RES = LNK-TERM1 - LNK-TERM2<br>EXIT.<br>MULTIPLICATION-PARA.<br>COMPUTE LNK-RES = LNK-TERM1 * LNK-TERM2<br>EXIT.<br>DIVISION-PARA.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | SUBTRACTION-PARA                   |                    |
| EXIT.<br>MULTIPLICATION-PARA.<br>COMPUTE LNK-RES = LNK-TERM1 * LNK-TERM2<br>EXIT.<br>DIVISION-PARA.<br>COMPUTE LNK RES = LNK TERM1 / LNK TERM2                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | COMPUTE I NK-BES = I NK-           | TEBM1 - I NK-TEBM2 |
| MULTIPLICATION-PARA.<br>COMPUTE LNK-RES = LNK-TERM1 * LNK-TERM2<br>EXIT.<br>DIVISION-PARA.<br>COMPUTE LNK RES = LNK TERM1 / LNK TERM2                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | EXIT.                              |                    |
| COMPUTE LNK-RES = LNK-TERM1 * LNK-TERM2<br>EXIT.<br>DIVISION-PARA.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | MULTIPLICATION-PARA.               |                    |
| EXIT.<br>DIVISION-PARA.<br>COMPLITE LNK DES – LNK TEDM1 / LNK TEDM2                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | COMPUTE LNK-RES = LNK-             | TERM1 * LNK-TERM2  |
| DIVISION-PARA.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | EXIT.                              |                    |
| COMPLITE I NK DES - I NK TEDM1 / I NK TEDM2                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | DIVISION-PARA.                     |                    |
| COMPOTE ENR-LS - ENR-TERMIT / ENR-TERMZ                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | COMPUTE LNK-RES = LNK-             | TERM1 / LNK-TERM2  |
| EXIT.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | EXIT.                              |                    |

**<sup>12</sup>**.NET 用の mfurunil 等 native 以外のアプリケーションがターゲットの場合はそれぞれを対象とした Test Runner が用意されています。詳細についてはマニュアルをご参照ください。

#### ① Test Fixture を用意

DATA DIVISION. WORKING-STORAGE SECTION. COPY "MFUNIT.CPY". 01 WS-FAIL-MSG. PIC X(20) VALUE "EXPECTED VALUE WAS: ". 03 FILLER 03 WK-EXPECTED-VALUE PIC X(06). 03 FILLER PIC X(23) VALUE ". BUT WHAT WE GOT WAS: ". 03 WK-ACTUAL-VALUE PIC X(06). 01 WK-PARAM. 03 WK-OPERATOR PIC 9 VALUE 1. PIC S9(4) COMP-5 VALUE 10. 03 WK-TERM1 03 WK-TERM2 PIC S9(4) COMP-5 VALUE 2. 03 WK-RES PIC S9(6) VALUE 0. 01 EXPECTED-VAL1 PIC S9(6) VALUE 12. 01 EXPECTED-VAL2 PIC S9(6) VALUE 7. PROCEDURE DIVISION. ENTRY MFU-TC-PREFIX & "PTN1". CALL "COBCALC" USING WK-PARAM. IF WK-RES NOT = EXPECTED-VAL1 THEN MOVE EXPECTED-VAL1 TO WK-EXPECTED-VALUE MOVE WK-RES TO WK-ACTUAL-VALUE CALL "MFU\_ASSERT\_FAIL" USING BY REFERENCE WS-FAIL-MSG BY VALUE 65 END-CALL END-IF. GOBACK. ENTRY MFU-TC-PREFIX & "PTN2". MOVE 2 TO WK-OPERATOR. CALL "COBCALC" USING WK-PARAM. IF WK-RES NOT = EXPECTED-VAL2 THEN MOVE EXPECTED-VAL2 TO WK-EXPECTED-VALUE MOVE WK-RES TO WK-ACTUAL-VALUE CALL "MFU\_ASSERT\_FAIL" USING BY REFERENCE WS-FAIL-MSG **BY VALUE 65** END-CALL END-IF. GOBACK.

ここでは、2つのテストケースを用意しました。テストケースは「MFUT\_」から始まるエントリポイントで定義します<sup>13</sup>。1番目のテスト ケースは「10 + 2」をテスト対象のモジュールに処理させています。このテストケースでは mfunit 用の Assert ライブラリ「MFU\_ ASSERT\_FAIL」にこの処理の結果は「12」になるはずだと期待値を指定しています。2番目のテストケースでは「10 - 2」を処理させ、 敢えてテストが失敗するよう期待値には「7」を指定しています。

② ダイナミックリンクライブラリに Test Fixture をコンパイル



#### ④ 結果を確認

#### テキストファイル形式で出力された結果を確認します:

| 🗐 CALCUT-report.txt - メモ帳                                                                                |                                                                                                                         |                                                |
|----------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------|------------------------------------------------|
| ファイル(F) 編集(E) 君式(O) 表示(V                                                                                 | /) ヘルプ(H)                                                                                                               |                                                |
| Micro Focus COBOL - mfuru<br>Unit Testing Framework fo                                                   | n Utility<br>r Windows∕Native/32                                                                                        |                                                |
| Options : verbose=false,p<br>Fixture filename : CALCUT                                                   | rocess=separate,junit=fals<br>.dll                                                                                      | se,printfile=true                              |
| Test case<br>Status<br>Time taken in ms<br>Return Code                                                   | : MFUT_PTN1<br>: Completed / Passed<br>: 000000000<br>: 000000000                                                       |                                                |
| Test case<br>Status<br>Time taken in ms<br>Return Code<br>Fail message count<br>EXPECTED VALUE WAS: 0000 | : MFUT_PTN2<br>: CompTeted / Failed<br>: 000000000<br>: 000000000<br>: 00000000<br>: 00000001<br>: BUT WHAT WE GOT WAS: | 000008 2                                       |
| Test Run Summary<br>Overall Result<br>Tests run<br>Tests passed<br>Tests failed<br>Total execution time  | : Failed<br>: 2<br>: 1<br>: 1<br>: 0                                                                                    | 予め想定したようにテストケース1は成功していますが、<br>テストケース2は失敗しています。 |

13 本例ではテストケース用のブリフィックスを Visual COBOL が提供する mfunit 用のコピーファイル MFUNIT.CPY 中の定数を通じて利用しています。

.....

## 5.2 JUnit 互換形式の出力フォーマットの活用

mfunit は前項で紹介したようにテキスト形式で結果を出力することも可能ですが、JUnit 互換の XML 形式で結果を出力することも可能 です。更に mfunit は先の章でも紹介した CI ツール Jenkins での使用を想定した結果を出力するオプションも用意しています。加えて Micro Focus Silk Central のようなテスト管理ツールとも連携させればより効率的にチーム内でテストを管理できます。 本項ではこれらのオプションを活用し Jenkins 経由で mfunit を実行し、結果を Jenkins 上で確認するようすを紹介します。

#### ① JUnit 用のプラグインを Jenkins に追加

追加後のイメージ:

|       | This allows other plugins to use jQuery in UI.                                                                              |             |
|-------|-----------------------------------------------------------------------------------------------------------------------------|-------------|
| ۲     | <u>JUnit Attachments Plugin</u><br>This plugin can archive certain files (attachments) together<br>with your JUnit results. | <u>1.3</u>  |
| 4     | <u>JUnit Plugin</u><br>Allows JUnit-format test results to be published.                                                    | <u>1.13</u> |
|       | JUnit Realtime Test Reporter Plugin                                                                                         | <u>0.2</u>  |
| all a | LDAP Plugin                                                                                                                 | <u>1.11</u> |

 mfunit を実行する Jenkins のプロジェクトを作成 フリースタイルのプロジェクトを用意:

| ジョブ名             | Perform_Sample_mfunit                                                                                                                              |
|------------------|----------------------------------------------------------------------------------------------------------------------------------------------------|
| • <del>رار</del> | ー<br>ースタイル・ブロジェクトのビルド<br>もっとも汎用性の高いJenkinsの中核機能です。任意のSCMからソースコードをチェックアウトし、任意のビルドシステムでブロジ<br>ェクトがビルドできます。往々にして、ソフトウェアのビルド以外にも様々な仕事の自動化に利用することができます。 |

③ mfunit の Test Fixture が格納されたフォルダをプロジェクトディレクトリに指定 [プロジェクトの高度なオプション]欄にて [高度な設定] ボタンを押下:

| ブロジェクトの高度なオブション |   |        |
|-----------------|---|--------|
|                 | × | 高度な設定… |
| ソースコード管理        |   |        |

[カスタムワークスペースを使用] にチェックを入れ、[ディレクトリ] 欄に Test Fixture が格納されたフォルダを指定:

| 1 | カスタムワークスページ | スを使用                  |
|---|-------------|-----------------------|
|   | デルクトリ       | C-twork/wn\DouToole\m |

| 140217     | C:\work\wp\DevTools\mfunit |
|------------|----------------------------|
| 表示用プロジェクト名 |                            |

#### ④ mfunit のテスト実行コマンドを追加

[ビルド手順の追加]にて [Windows バッチコマンドの実行]を選択:

|    | Windows PowerShell |    |
|----|--------------------|----|
| (  | Windowsバッチコマンドの    | 実行 |
| ł  | シェルの実行             |    |
| -1 | ビルド手順の追加 🔻         |    |
| Ľ  | ルド後の処理             |    |

#### mfunit のテストを実行するコマンドを追加:

|||| Windowsバッチコマンドの実行

| יאעקב | call C:¥work¥Jenkins¥cob32set.bat<br>mfurun -report:junit -jenkins-ci -outdir:C:¥work¥wp¥DevTools¥mfunit¥mfunit_res CALCUT.dll |
|-------|--------------------------------------------------------------------------------------------------------------------------------|
|       | $\mathbf{\Lambda}$                                                                                                             |
|       | -report:junit junit => 互換形式の結果を出力させます。                                                                                         |
|       | (テキスト形式の結果生成を抑制したい場合は、-report:noprintfile も追加で指定します。                                                                           |
|       | -jenkins-ci => 結果を Jenkins 互換形式にします。                                                                                           |
|       | -outdir => 結果の出力フォルダを指定します。                                                                                                    |

#### ⑤ テスト結果を JUnit 用のプラグラインを使って集計するよう構成 [ビルド後の処理の追加]にて [JUnit テスト結果の集計]を選択:

| A | 中期的1-安行                                       | _                           |
|---|-----------------------------------------------|-----------------------------|
|   | Console output (build log) parsing            |                             |
| C | Emma力バレッジレポートを記録                              |                             |
| 1 | JUnitテスト結果の集計                                 |                             |
| I | Javadocの保存                                    |                             |
| l | Publish HTML reports                          | pat                         |
| l | Publish MSTest test result report             | ;i -outdir:C:¥work¥wp¥DevTo |
| l | ファイル指紋を記録してファイルの利用状況を追跡                       |                             |
| l | 下流プロジェクトのテスト結果を集約                             |                             |
| l | 他のプロジェクトのビルド                                  |                             |
| l | 成果物を保存                                        |                             |
| l | Build other projects (manual step)            |                             |
| ł | E-mail通知                                      |                             |
| ł | Trigger parameterized build on other projects |                             |
| 1 | ビルド後の処理の追加 🔻                                  |                             |

mfunit\_res\\*.xml

[テスト結果 XML] 欄に③で指定したディレクトリから mfurun の outdir で指定したディレクトリまでの相対パス及び「\*.xml」を指定

| ビルド後の処理         |                  |    |
|-----------------|------------------|----|
| ∭ JUnitテスト結果の集計 |                  | _  |
| テスト結果XML        | mfunit_res\* vml | 74 |

| 沐1+ 小スノで1+ | Γ | 保存 | 13 | ボタン | を押 |  |
|------------|---|----|----|-----|----|--|
|------------|---|----|----|-----|----|--|

#### ⑥ ビルドを実行

Jenkins のプロジェクトの処理実行後、履歴欄に表示される該当の番号をクリックすると「テスト結果」へのリンクが表示されます:



クリックするとテスト結果を専用のビューで確認できます:

| テスト結果                                                                                                                                         |                   |           |           |      |                                      |                                  |  |  |
|-----------------------------------------------------------------------------------------------------------------------------------------------|-------------------|-----------|-----------|------|--------------------------------------|----------------------------------|--|--|
| 1個の失敗 (±D)                                                                                                                                    |                   |           |           |      |                                      |                                  |  |  |
|                                                                                                                                               |                   |           |           |      | 2個のテ:<br>所要時<br>1200<br>1200<br>1200 | スト (±0)<br>間 <u>5 ms</u><br>月を記入 |  |  |
| 失敗したすべてのテスト                                                                                                                                   |                   |           |           |      |                                      |                                  |  |  |
| テスト名                                                                                                                                          |                   |           |           |      | テスト<br>所要時<br>間                      | 時期                               |  |  |
| <ul> <li><u>■ MEUT_PTN2.MEUT_PTN2</u></li> <li>ニ エラー詳細</li> <li>EXPECTED VALUE WAS: 000007. BUT WHAT WE GOT WAS: 000008 2</li> <li></li></ul> |                   |           |           |      |                                      |                                  |  |  |
| すべてのテスト                                                                                                                                       |                   |           |           |      |                                      |                                  |  |  |
| パッケージ                                                                                                                                         | テスト所 失敗<br>要時間 失敗 | (差分) スキッフ | タ (差分) パス | (差分) | 合計                                   | (差分)                             |  |  |
| (root)                                                                                                                                        | 5 ms              | 1         | 0         | 1    | 2                                    |                                  |  |  |

「テスト履歴」をクリックするとこれまでのテストの履歴を確認できます。

下図は、テストケースを修正し一回実行した後に表示したイメージです:



# 6 Profiler - COBOL 専用のパフォーマンス分析ツール

Profiler は Visual COBOL が提供する COBOL 専用のパフォーマンス分析ツールです。COBOL の節や段落単位で処理に要した時間、 節や段落へのエントリ回数、エントリ毎の平均処理時間等の統計情報を取得します。プログラムのパフォーマンスが許容基準を超える場合 に、ボトルネック検出するのに非常に有用なツールです。

### 6.1 コマンドによる処理

PROFILE コンパイラ指令を指定してコンパイルしたモジュールは Profiler による分析の対象となります。Profiler は分析対象のモジュー ルが実行されると計測情報をバイナリファイルに書き溜めます。計測情報をテキスト形式のレポートに成形するには profiler コマンドを利用 します。Test Coverage と同様に本ツールについてもバッチやシェル等による自動化も可能です。 以下に簡単なサンプルを用いた Test Coverage の実行例を記します:

① PROFILE コンパイラ指令を指定してプログラムをコンパイル



② アプリケーションを実行

| ③ 生成され7     | たバイナリフ                                                                                                 | レからレポ-                                                                         | ートを生成                                                                                            | < プログラム名 >.ipf: Profiler による計測結果を記録              |                                                                                                                                                                                                                                                                                        |                                                                                                                                                            |   |        |  |  |
|-------------|--------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------|--------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------|---|--------|--|--|
|             | <b>GN</b> Visual COB                                                                                   | ועקב וס                                                                        | <sup>ะ</sup> プロンプト (32-bi                                                                        | it)                                              |                                                                                                                                                                                                                                                                                        | したファイル +list: レボートをコンソールへ書き出させる<br>オプション                                                                                                                   |   |        |  |  |
| )<br>x<br>x | C:¥work¥wp¥<br>* Micro Foc<br>* Copyright<br><u>* usstee</u> l                                         | DevToo<br>us COB<br>(C) M                                                      | ls¥Profile<br>OL Profile<br>licro Focus                                                          | r≻profiler uss<br>r<br>1984-2016. Al             | teel.ipf +list 🧲<br>  rights reserved.                                                                                                                                                                                                                                                 |                                                                                                                                                            |   |        |  |  |
|             | Total time:<br>% time                                                                                  | 1<br>time                                                                      | 08684 mill<br>entries                                                                            | iseconds.<br>micros/entry                        | Module called once.<br>paragraph                                                                                                                                                                                                                                                       |                                                                                                                                                            |   |        |  |  |
|             | $\begin{array}{c} 22.07\\ 19.51\\ 13.67\\ 13.51\\ 7.46\\ 6.60\\ 6.59\\ 6.50\\ 0.39\\ 0.29 \end{array}$ | 23988<br>21206<br>14860<br>14680<br>8103<br>7169<br>7159<br>7065<br>419<br>316 | 1<br>25000001<br>25000001<br>26000001<br>25000000<br>25000000<br>25500001<br>500000<br>500000    | 23988000<br>1<br>1<br>0<br>0<br>0<br>0<br>1<br>1 | Unnamed Paragraph 00<br>300-TIME-TEST12 0F 0<br>260-TIME-TEST10 0F 0<br>280-TIME-TEST11 0F 0<br>490-DATA-ADDRESS-RT<br>530-DATA-ADDRESS-RT<br>500-DATA-ADDRESS-RT<br>500-DATA-ADDRESS-RT<br>220-TIME-TEST08 0F 0<br>240-TIME-TEST09 0F 0                                               | 080<br>000-DUMMY<br>000-DUMMY<br>0F 000-DUMMY<br>-INDEX OF 000-DUMM<br>-INDEX-EXIT OF 000<br>-BINARY OF 000-DUM<br>000-DUMMY<br>000-DUMMY                  | 4 | 計測結果の例 |  |  |
|             | 0.27<br>0.23<br>0.21<br>0.20<br>0.19<br>0.19<br>0.16<br>0.16<br>0.16<br>0.16<br>0.15                   | 297<br>250<br>231<br>219<br>218<br>205<br>203<br>173<br>172<br>169<br>159      | 500000<br>500001<br>500000<br>500001<br>500000<br>500000<br>499999<br>500001<br>500001<br>500000 | 1<br>0<br>0<br>0<br>0<br>0<br>0<br>0<br>0<br>0   | 180-TIME-TEST06 OF 0<br>040-TIME-TEST01 OF 0<br>200-TIME-TEST07 OF 0<br>060-TIME-TEST01-PAC<br>400-TIME-TEST01-PAC<br>400-TIME-TEST05 OF 0<br>320-TIME-TEST05 OF 0<br>520-DATA-ADDRESS-RT<br>080-TIME-TEST01-BIN<br>380-TIME-TEST01-BIN<br>380-TIME-TEST06 OF 0<br>020-CLOCKING-ROUTIN | 000-DUMMY<br>000-DUMMY<br>KED OF 000-DUMMY<br>000-DUMMY<br>000-DUMMY<br>000-DUMMY<br>-INDEX-EXIT OF 000<br>ARY OF 000-DUMMY<br>000-DUMMY<br>E OF 000-DUMMY |   |        |  |  |

# 6.2 IDE 上での処理

Profiler は Visual Studio 並びに Eclipse IDE 上でも利用が可能です<sup>14</sup>。この場合、前項で紹介したようなコンパイラ指令指定や profiler コマンドによるレポート生成作業は不要です。 以下に本機能の動作イメージを記します。

Visual Studio 2015 における設定イメージ:

| 出力              |                  |
|-----------------|------------------|
| 出力パス:           | .¥bin¥x86¥Debug¥ |
| 🗌 指令ファイルの生成     | □ リストファイルを生成     |
| 🔲 コード カバレッジを有効に | する               |

Profiler の挙動も IDE で構成可能です:

E.

.....

| オプション                                                                                             |   |                                                               |
|---------------------------------------------------------------------------------------------------|---|---------------------------------------------------------------|
| 検索オプション (Ctrl+E)                                                                                  | ٩ | レポートの生成                                                       |
| Web Service Clients<br>▶ コードカパレッジ<br>サービスインターフェイス<br>データファイル ツール                                  | ^ | ✓ プロファイラ実行の終了時に結果を表示(S) ✓ 起動時にプロファイラ情報がない場合に警告する(W) レポートオブ/コン |
| ノーファイト<br>▲ プロファイラ<br>レポートの生成<br>ト 指令                                                             |   | □ 完全な性総プロファイルを表示(E)<br>□ 段落別にアルファベット順で性総プロファイルを表示( <u>A</u> ) |
| <ul> <li>▷ NuGet パッケージ マネージャー</li> <li>▷ SQL Server ツール</li> <li>▷ Web パフォーマンス テスト ツール</li> </ul> |   | 結果ディレクトリ: Profiling                                           |

<sup>13</sup> IDE 上における操作機能は Visual COBOL 2.3 より実装されています。

Profiler による計測をさせつつアプリケーションを実行:

| ual Studio          |                |                    |                   |            |        |          |       |  |
|---------------------|----------------|--------------------|-------------------|------------|--------|----------|-------|--|
| V) プロジェクト(P) ビルド(B) | Ŧ/(            | シグ(D)              | チーム(M)            | ツール(T)     | テスト(S) | Silk4NET | 分析(N) |  |
| 📲 🄊 - 🤆 - 🛛 Debug - |                | ウインド               | <sup>(</sup> ウ(W) |            |        |          | +     |  |
|                     |                | グラフィ               | ックス(C)            |            |        |          | + I   |  |
|                     | ~              | 診断り                | ールの表示(D           | )          |        | Ctrl+Al  | t+F2  |  |
| 構成(C): アクラ          | ►              | デパッグの開始(S)         |                   |            |        | F5       |       |  |
|                     | ▶              | <u>デバック</u>        | ブなしで開始(H          | l)         |        | Ctrl+F5  |       |  |
| ヨロヤウ                | 2              | デパッグなしで診断ツールを開始(R) |                   |            |        | Alt+F2   |       |  |
| 最大エラー               | é <sup>#</sup> | プロセス               | スにアタッチ(P)         |            |        | Ctrl+Al  | t+P   |  |
|                     |                | その他                | のデパッグター           | ゲット(H)     |        |          | •     |  |
| 出力                  |                | Micro              | Focus Code        | Coverage 7 | 開始     |          |       |  |
| 出力パス:               |                | Micro              | Focus Profil      | erで開始      |        |          |       |  |

アプリケーションの処理が完了すると IDE 内にレポートが出力されます:

| Profiler - Microsoft Visual Studio |                                              |              |            |            |              |        |                      |                          |                         |             |    |
|------------------------------------|----------------------------------------------|--------------|------------|------------|--------------|--------|----------------------|--------------------------|-------------------------|-------------|----|
| ファイ                                | ſJI (I                                       | F) 編集(E)     | 表示(V)      | プロジェクト(P)  | ビルド(B) デ/    | (ッグ(D) | チーム(M)               | ツール(T)                   | テスト(S)                  | Silk4NET 3  | 分4 |
|                                    | θ,                                           | • 🖸 🕆 📬      | 1 HA 🖓     | 9-9-       | Debug 👻 🛪    | (86    | •                    | ▶ 開始 ◄                   | 🏓 🚽 🖁 🛃                 | 2 🔵 🚽 🕅 📃   |    |
| 4-1                                | Re                                           | port.prf 👳 🗙 | IntGntProj | ject1      |              |        |                      |                          |                         |             |    |
| Г<br>Н                             | * Micro Focus COBOL Profiler 20-Jul-16 19:09 |              |            |            |              |        |                      |                          |                         |             |    |
| 77                                 |                                              | ∗ ussteel    |            |            |              |        |                      |                          |                         |             |    |
| H                                  |                                              | Total time   | e: 1       | 10285 mill | iseconds     | Module | called               | once.                    |                         |             |    |
| J.                                 |                                              | >% time      | time       | entries    | micros/entry | paragr | apn<br>              |                          |                         |             |    |
| Ţ.                                 |                                              | 22.20        | 24486      | 1          | 24486000     | Unname | d Paragr             | aph 0080                 |                         |             |    |
| 9                                  |                                              | 19.57        | 21578      | 25000001   | 1            | 300-TI | ME-TEST1             | 2 OF 000-I               | DUMMY                   |             |    |
| l ⊱                                |                                              | 14.17        | 15622      | 25000001   | 1            | 260-TI | ME-TEST1             | 0 OF 000-I               | DUMMY                   |             |    |
| 퀑                                  |                                              | 12.94        | 14268      | 25000001   | 1            | 280-TI | ME-TEST1             | 1 OF 000-I               | DUMMY                   |             |    |
| 2                                  |                                              | 7.02         | 7743       | 25500001   | 0            | 500-DA | TA-ADDRE             | SS-RT-BIN                | ARY OF OO               | O-DUMMY     |    |
| L.                                 |                                              | 6.92         | 7637       | 26000001   | U            | 490-DA | TA-ADDRE             | SS-RI OF I               | UUU-DUMMY               | NUMBU       |    |
|                                    |                                              | 6.72         | 7412       | 25000000   | U            | 530-DA | TA-AUURE             | SS-RI-INDI               | EX UF UUU<br>EV EVIT () | E OOO DUWWY |    |
|                                    |                                              | 0.00         | 1220       | 500000     | 1            | 220-TI | NE_TESTO             | 8 OF 000-1               | EAFEAIL C<br>Dummy      | F UUU-DUMMI |    |
|                                    |                                              | 0.42         | 267        | 500000     | 1            | 180-TT | ME-TESTO             | 6 OF 000-1               |                         |             |    |
|                                    |                                              | 0.24         | 267        | 500000     | i            | 400-TI | ME-TEST1             | 7 OF 000-1               | DUMMY                   |             |    |
|                                    |                                              | 0.24         | 265        | 500000     | i            | 240-TI | ME-TESTO             | 9 OF 000-1               | DUMMY                   |             |    |
|                                    |                                              | 0.23         | 252        | 500001     | 1            | 040-TI | ME-TESTO             | 1 OF 000-I               | DUMMY                   |             |    |
|                                    |                                              | 0.21         | 234        | 500001     | 0            | 120-TI | ME-TESTO             | 3 OF 000-I               | DUMMY                   |             |    |
|                                    |                                              | 0.21         | 233        | 500001     | 0            | 060-TI | ME-TESTO             | 1-PACKED                 | OF 000-DU               | MMY         |    |
|                                    |                                              | 0.18         | 203        | 500001     | 0            | 080-TI | ME-TESTO             | 1-BINARY                 | OF 000-DU               | MMY         |    |
|                                    |                                              | 0.18         | 200        | 500000     | 0            | 200-TI | ME-TESTO             | 7 OF 000-1               | DUMMY<br>Door Dury      |             |    |
|                                    |                                              | 0.17         | 190        | 5000001    | U            | 020-CL | UCKING-N             | DUTINE OF                | UUU-DUMM                | Y           |    |
|                                    |                                              | 0.16         | 172        | 500000     | 0            | 100-TT | ME-TESTI<br>ME-TESTO | p OF 000-1<br>2 OF 000-1 | DUMMY                   |             |    |
|                                    |                                              | 0.16         | 172        | 500001     | 0            | 320-TI | ME-TESTU             | 3 OF 000-1               | DUMMI                   |             |    |
|                                    |                                              | 0.16         | 172        | 499999     | ň            | 520-DA | TA-ADDRE             | SS-RT-IND                | EX-EXIT C               | E OOO-DUMMY |    |
| _                                  |                                              | 0.14         | 156        | 500001     | ŏ            | 340-TI | ME-TEST1             | 4 OF 000-1               | DUMMY                   |             |    |
|                                    |                                              | 0.13         | 141        | 500000     | Ō            | 550-RE | SET-INDX             | OF 000-D                 | UMMY                    |             |    |
| _                                  |                                              | 0.13         | 140        | 1          | 140000       | 420-TI | ME-TEST1             | 8 OF 000-I               | DUMMY                   |             |    |
|                                    |                                              | 0.11         | 126        | 500001     | 0            | 140-TI | ME-TESTO             | 4 OF 000-1               | DUMMY                   |             |    |
|                                    |                                              | 0.10         | 109        | 499999     | 0            | 510-DA | TA-ADDRE             | SS-RT-IND                | EX OF 000               | -DUMMY      |    |

# 7 おわりに

近年、ビジネス環境を取り巻く変化のスピードは高速化の一途を辿っています。このように先の見通しが立てにくい環境にあって、開 発期間中における不測の要件変更に対しても柔軟に追従することが求められる時代に突入しているとも捉えられます。実際、30年以 上も前からこのような不確実性に強く、迅速に反応できる開発モデルが様々な企業や団体によって考案されてはいましたが、近年一層 注目されるようになっています。ただし、このようなアジャイル開発モデルは従来モバイルや IoT 等を活用して人との関係を構築す る SoE(System of Engagement)には向いているが、システムの安定性やデータ保全性が強く求められる SoR(System of Record)には向かないと評価されてきたようです。COBOL で開発される基幹システムは SoR に分類されることが多く、このような モデルが採用されることはあまりありませんでした。しかし、近年ではこのような開発モデルをエンタープライズ目的でも利用できるよう 拡張したエンタープライズアジャイルといったモデルも考案されるようになり、採用企業も増えてきているようです。

本書で紹介した機能を活用した継続的インテグレーションはこのようなエンタープライズアジャイルを実現する第一歩となり得ます。本書 ではそれぞれのツールを単体で処理する部分を紹介しましたが、開発方針に則りそれぞれの処理をパイプラインでつなぎ継続的インテグ レーションのシナリオに基づいた自動処理を組み上げることが可能です。構成管理ツールと組み合わせソースを安全に管理しつつチェッ クインなどのタイミングで上述の処理をキックさせるなど、より保全性の高い仕組みを築き上げることも可能です。更には要件管理ツー ルやデプロイツール等も組み合わせ、デリバリー、運用部分が一体になって協調する仕組みへと拡げていくことも可能です。COBOL アプリケーションの開発ツールとしての利用はもちろんのこと、このような環境変化に強い開発モデル構築への切り口としても是非 Micro Focus Visual COBOL をお役立てください。

> 記載の会社名、製品名は各社の商標または登録商標です。 本ホワイトペーパーは2016 年 7 月に作成したものです。 MFWP05-1811-00MC | © 2018 Micro Focus. All rights reserved.

