

Micro Focus Enterprise Developer を使用した COBOL 開発業務のモダナイゼーション

IBM メインフレームは現在も世界のビジネストランザクションの主要な業務を担い、日々稼働し続けています。メインフレーム上の基幹システムは多くの企業にとってその業務の心臓部分を担うものであり、欠くことのできない経営資産となっています。

一方で、メインフレームアプリケーションの稼働には多大な運用コストがかかる現実があり、かつ周辺システムとのデータ連携に制約があるなどの理由から、多くの企業がクラウドを含むオープン環境への移行を検討または実施しています。

Micro Focus Enterprise Developer / Enterprise Server は、実績のある COBOL, PL/I アプリケーションを無駄にせず、再活用しながらオープン環境へのリホストを実現できる開発 / 実行環境製品です。COBOL コンパイラは IBM メインフレームの各種コンパイラバージョンに対して高い互換性を持っています。

さらに Enterprise Server 機能が備える JES, CICS, IMS ミドルウェアエミュレーション機能により、オープン環境移行後も JCL, EXEC CICS 構文、EXEC DLI, CBLTDLI 構文などを利用できるため、アプリケーションの品質を損なうことなく最小限のコストと期間でメインフレームからのオープン化を実現することができます。また、Enterprise Developer / Enterprise Server は Visual COBOL / COBOL Server の全機能を含む上位製品となり、リホスト後は最新技術を利用した更なるモダナイゼーションを目指すこともできます。

本文書では、Enterprise Developer を活用した COBOL アプリケーションの開発業務におけるモダナイゼーションについて解説します。

目次

1. 「リホスト」の定義.....	1
2. モダナイゼーション支援製品	1
3. 製品の機能分布	2
4. 製品の実行機能	2
5. Enterprise Developer を利用した開発業務	3
1) 統合開発環境 (IDE)	3
2) COBOL プロジェクト	4
3) COBOL エディタ機能	4
4) JCL エディタ機能.....	5
5) CICS 構文のサポート	5
6) IMS 構文のサポート	5
7) コンパイル機能.....	6
8) デバッグ機能.....	7
9) カバレッジ機能.....	7
10) Micro Focus Unit Testing Framework (MFUnit) 機能	8
11) コード分析機能.....	8
12) プログラムフローグラフ機能	9
13) データファイルツール.....	9
6. Enterprise Server を利用した実行.....	10
1) Enterprise Server Common Web Administration (ESCWA)	10
2) JCL 実行機能.....	10
3) CICS 実行機能	11
4) IMS 実行機能	11
5) Micro Focus Database File Handler (MFDBFH) 機能.....	12
6) スケールアウト パフォーマンス/可用性クラスター機能	12
7. メインフレーム コンパイラとの違い	13
1) COBOL 構文の差異.....	13
2) COBOL 実行時挙動の差異	13
8. Enterprise Developer チュートリアルと例題	14
9. おわりに.....	15
補足：稼働環境	15
1) OS	15
2) プロセッサ	15
3) システムの種類.....	15
4) Micro Focus 製品 バージョン	15
5) 製品マニュアル バージョン.....	15

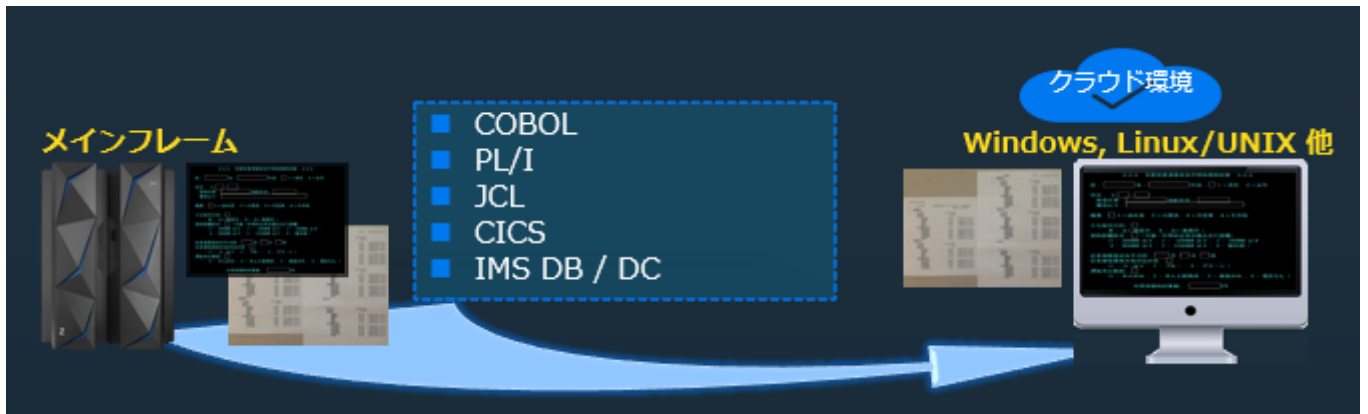
各機能の詳細に関しては、製品マニュアルページからご利用になるバージョンを選択後、内容をご確認ください。

<https://www.microfocus.co.jp/>

[サポート] > [COBOL・エンタープライズ製品のカスタマーケア：詳細をみる] > [製品マニュアル]

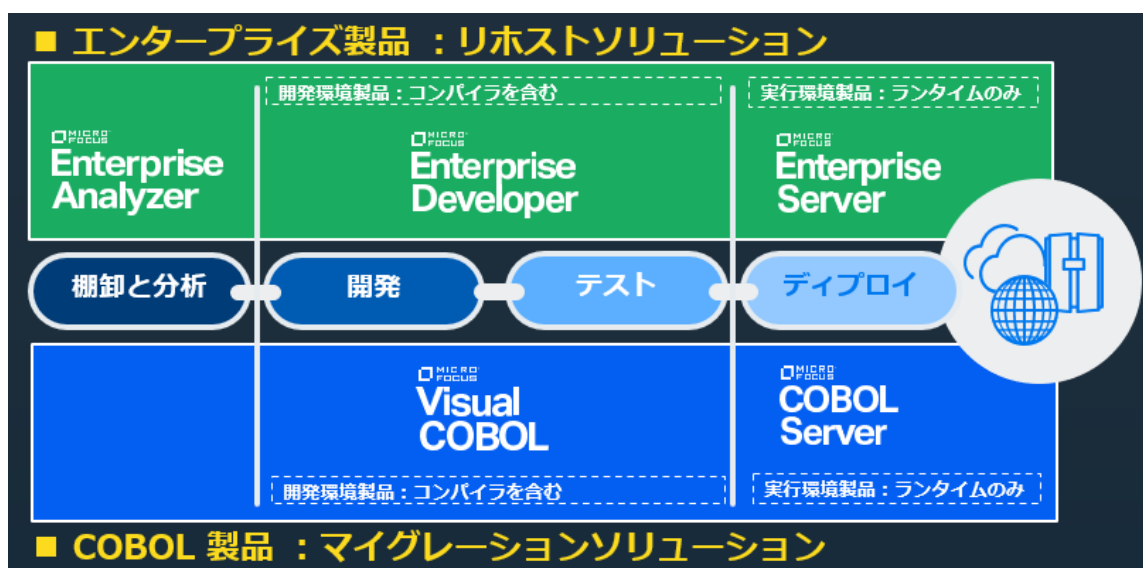
1. 「リホスト」の定義

「リホスト」という言葉の意味は様々な解釈がありますが、本文書では「IBM メインフレームの既存資産を有効活用するため、それらを出来るだけ修正せずにオープン環境へ移行してメインフレームの運用や開発にかかるコストの削減を目指す手法」とします。



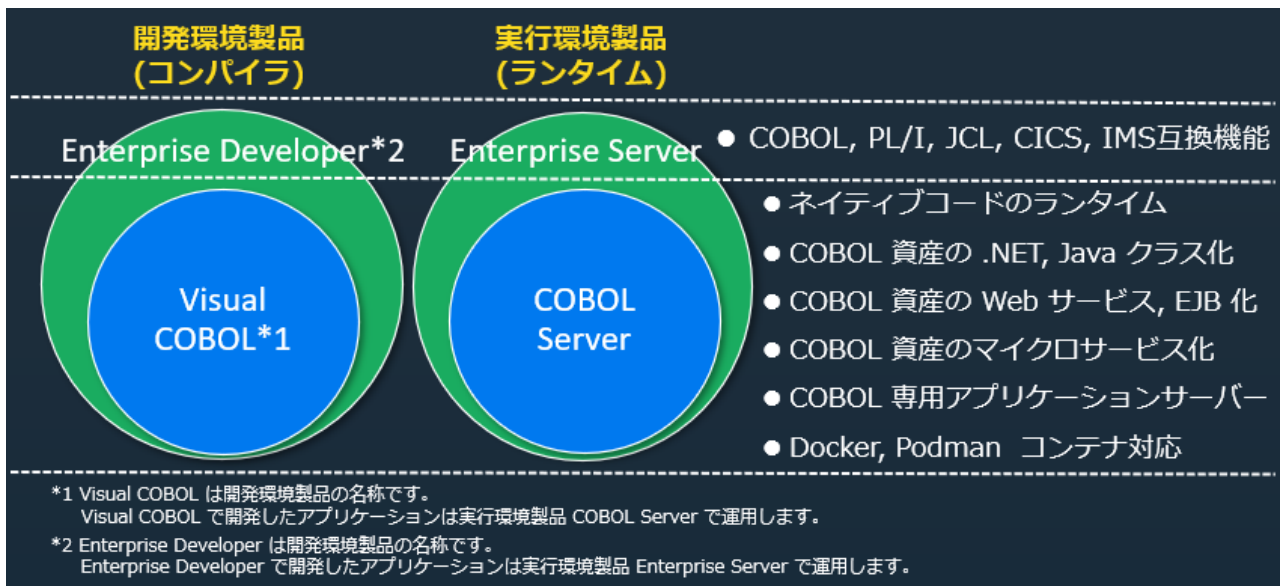
2. モダナイゼーション支援製品

マイクロフォーカスのモダナイゼーション支援製品は大きく2つに分類されています。IBM メインフレームを対象としたリホストを実現するエンタープライズ製品群には、静的解析ツールの Enterprise Analyzer、コンパイラを含み、開発や単体テストに使用する開発環境製品の Enterprise Developer、ランタイムのみを含み、本番環境で使用する実行環境製品の Enterprise Server が含まれます。IBM、国産メインフレーム、オープンレガシーを対象とした COBOL 製品群も同様に、コンパイラを含む開発環境製品の Visual COBOL とランタイムのみを含む実行環境製品の COBOL Server があります。



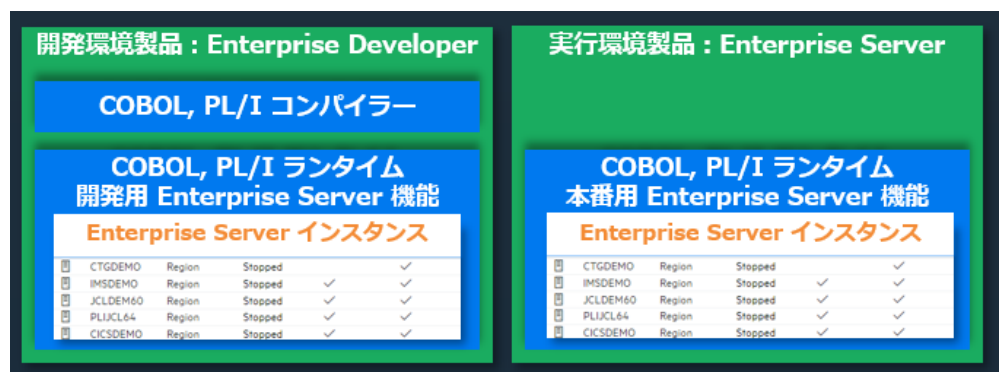
3. 製品の機能分布

エンタープライズ製品は COBOL 製品の上位製品となり、クラウド上でマイクロサービスを導入する際に欠かせないコンテナ型仮想化にも対応しています。また、マイクロフォーカスのコンパイラ技術により、COBOL ソースはそのままで、ネイティブコード、Java クラス化、.NET クラス化した実行可能ファイルを生成することができます。ネイティブコードは製品に含まれている COBOL 専用のアプリケーションサーバーを使用した REST 形式に加え、SOAP 形式の Web サービス、EJB 連携を実現することができます。これらの機能に加えて、PL/I 言語のサポートや JCL, CICS, IMS をエミュレートする機能を持ち IBM メインフレームからのリホストに最適な製品です。



4. 製品の実行機能

開発環境製品の Enterprise Developer は COBOL, PL/I コンパイラを含み、かつテスト実行が可能な開発用の Enterprise Server 機能を内蔵しています。一方、実行環境製品の Enterprise Server は COBOL, PL/I ランタイムのみを含む、本番環境に適した製品です。実行単位である Enterprise Server インスタンスは、例えば JCL を対象としたバッチ用、IMS, CICS などのオンライン用など、運用用途に合わせて複数設定することが可能です。



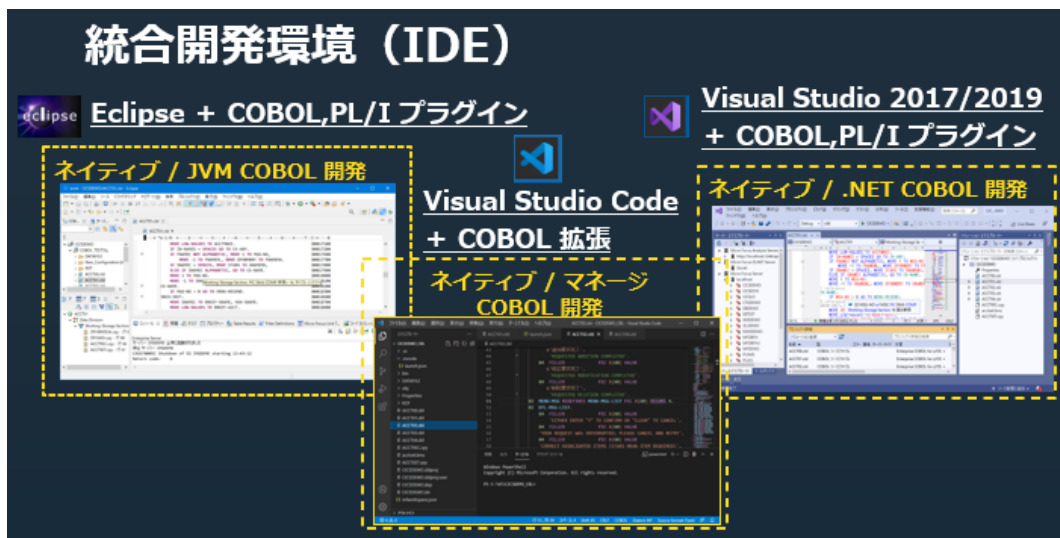
5. Enterprise Developer を利用した開発業務

COBOL アプリケーションをオープン環境へ移行後は、開発環境製品である Enterprise Developer を使用した開発業務を行います。Enterprise Developer が持つ様々な機能を利用することにより、作業の効率化を計れ、開発用の実行環境である Enterprise Server 機能を利用することにより、メインフレームを独り占めするイメージで単体テストを実施することができます。

この章では、Enterprise Developer の具体的な機能について説明します。PL/I の開発機能については、「PL/I 開発業務のモダナイゼーション」ホワイトペーパーをご参照ください。また、一般的なリホストにおける注意点や手順については「COBOL, PL/I アプリケーションのリホスト手順と注意点」ホワイトペーパーをご参照ください。

1) 統合開発環境 (IDE)

業界標準の IDE である Eclipse, Visual Studio, Visual Studio Code を利用して、コーディング、コンパイル、デバッグ、テストなどの開発業務を効率的に行うことができます。もちろん、テキストエディターでソースを編集後、コマンドを利用したコンパイルも可能ですので、開発者のスタイルに合わせた開発業務が可能になります。



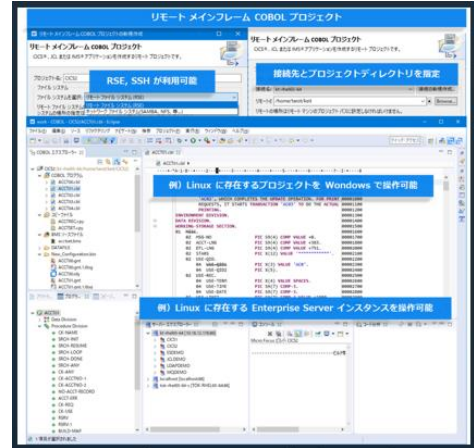
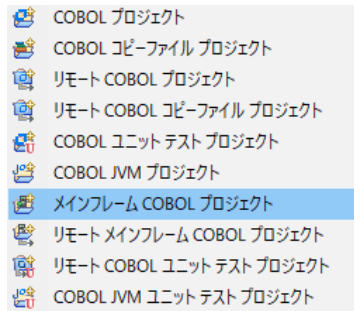
例えば、COBOL ソースはそのまま、Java クラス化した実行可能ファイルを生成する計画がある場合は JVM COBOL プロジェクトがある Eclipse を、.NET クラス化した実行可能ファイルを生成する必要がある場合は Visual Studio を、軽量で速い Visual Studio Code を利用してコーディングを行うなど、利用用途に合わせて IDE を選択することができます。

さらに、ステップ実行可能な各 IDE のデバッガを利用することにより、変数の値を可視化でき、開発業務のモダナイゼーションを実現することができます。

2) COBOL プロジェクト

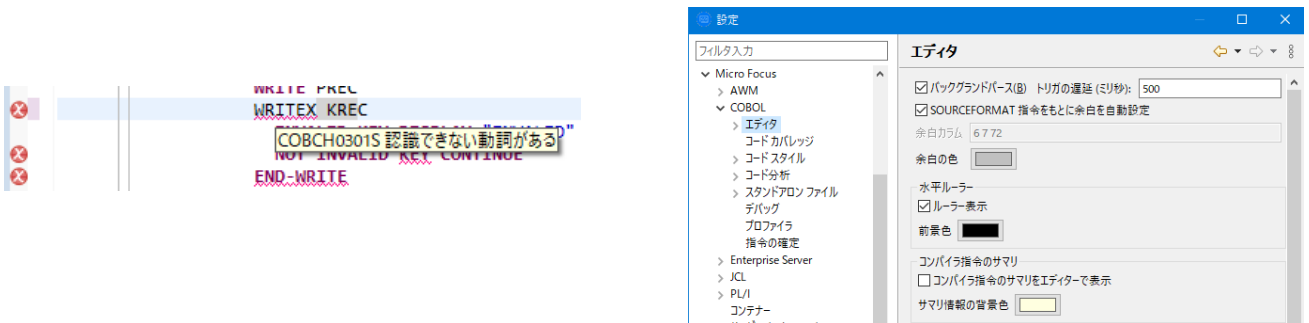
Eclipse, Visual Studio では COBOL 専用のプロジェクトを作成します。Windows 上の Eclipse ではリモート開発が可能なプロジェクトも含まれており、Linux マシンに配備したソースを Eclipse で編集し、コンパイルにより Linux マシンへ実行可能ファイルを生成することができます。

【 Eclipse リモート開発 】

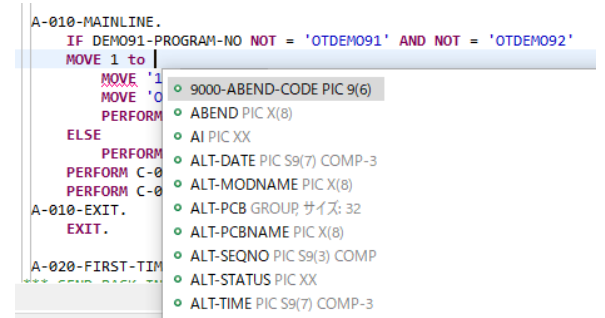
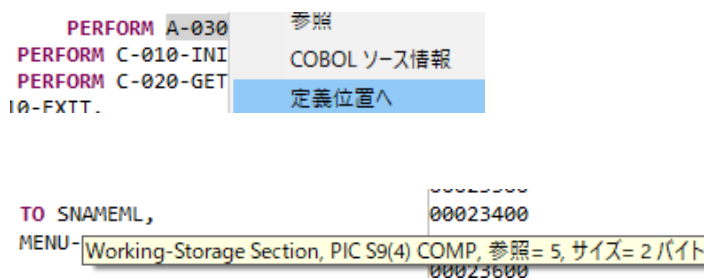


3) COBOL エディタ機能

COBOL プロジェクトを作成後は、ソースやコピー句などのファイルをこのプロジェクトにインポートし、COBOL 専用エディタを使用してコーディングを行います。入力時、リアルタイムに入力値エラーを検出できるバックグラウンドパース機能を利用することができます。



また、項目定義の位置へジャンプすることや、マウスオーバーによるデータ属性の確認、転送先候補の表示ができ、コーディングミス未然に防ぐことで、開発工数の削減に貢献できる機能です。



4) JCL エディタ機能

JCL 専用のエディタを利用してコーディングを行うことができます。COBOL ソースと同様に、リアルタイムに入力値エラーを検出できるバググランドパース機能を利用することができます。

```
//VSAMNRT3 JOB CLASS=A,MSGCLASS=A
//*****
//DEFVSAM1 EXECE PGM=IDCAM5
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
```

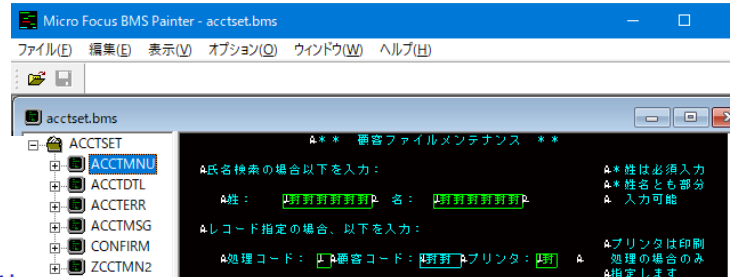
5) CICS 構文のサポート

EXEC CICS 構文や、一般的に使用される API, SPI¹ の大部分をサポートしています。BMS 定義もサポートしており、画面イメージでメンテナンス可能な BMS ペインターも利用することができます。

```
EXEC CICS SEND
  MAP('ACCTMNU')
  MAPSET('ACCTSET') FREEKB
  ERASE MAPONLY
END-EXEC
EXEC CICS RETURN TRANSID('AC01') END-EXEC
```

```
ACCTMNU DFHMDI SIZE=(24,80)
DFHMDF ATTRB=(ASKIP,NORM),
COLOR=TURQUOISE,
LENGTH=36,
POS=(1,20),
PS=8,
INITIAL='** 顧客ファイルメンテナンス **'
```

【 BMS ペインター機能 】



6) IMS 構文のサポート

CALL インターフェイスである CBLTDLI や EXEC DLI といった IMS プログラム構文をサポートしており、データのセグメント構成も保持できます。また、メインフレームで使用している IMS 資源定義マクロをそのまま利用することができるため、新しくマクロを作成する必要がありません。JCL で使用する MPP, BMP, DLI もサポートしています。

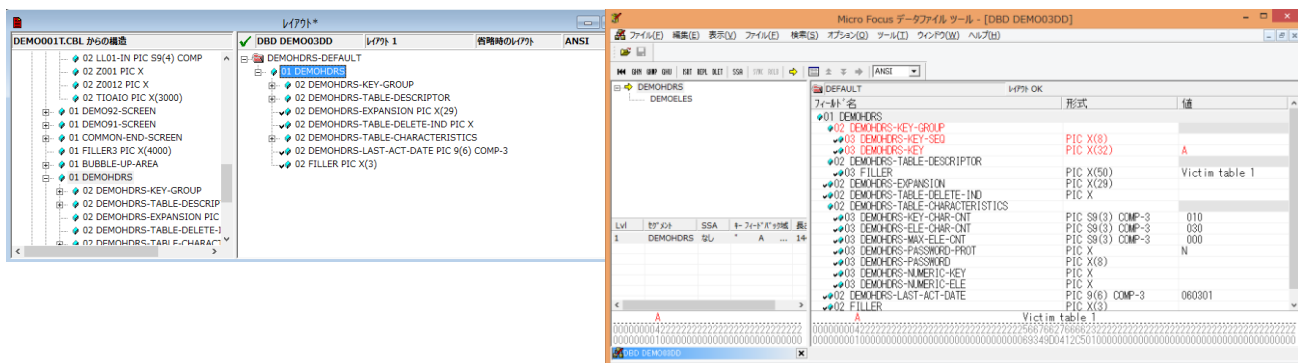
```
PROCEDURE DIVISION USING PCB-LT
                      ALT-PCB
                      DEMO-PCB.
ENTRY 'DLITCBL' USING PCB-LT
                      ALT-PCB
                      DEMO-PCB.

OTDEMO91 MSG TYPE=OUTPUT,SOR=(DEMO91,IGNORE),FILL=NULL,PAGE=YES,
           NXT=INDEMO91
           SEG
           MFLD FLD0000,LTH=0079
           MFLD FLD0001,LTH=0008
           MFLD FLD0002,LTH=0035
           MFLD LTERM,LTH=0008

//S01 EXEC PGM=DFSRR00,REGION=4M,
//      PARM='BMP,DEMO001B,DEMO001T,,,,,,,,,CDLI,,N,N'
```

¹ 製品マニュアル) [Micro Focus Enterprise Developer] > [リファレンス] > [メインフレーム リファレンス] > [CICS サポートコマンド]

また、IMS データのセグメントレイアウトに沿ってデータのメンテナンスを行うことができるデータファイルツールも付属されており、レコードレイアウトを利用してデータを管理することができます。



7) コンパイル機能

COBOL プロジェクトまたはソースファイルにプロパティとしてコンパイラ指令を指定します。Enterprise Developer は IBM メインフレームの COBOL コンパイラバージョンに沿った方言²を指定でき、これによりコンパイラ挙動を模倣するよう、複数のコンパイラ指令が暗黙的に指定され、高い互換性を保つことができます。

【 Eclipse の方言指定画面 】

▼ 一般	
文字セット	ASCII
言語の方言	Enterprise COBOL for z/OS
ソースフォーマット	Enterprise COBOL for z/OS
メインフレームのコピー処理	COBOL for OS/390
指令ファイルを生成する	COBOL for MVS
リストファイルを生成	COBOL/370 Release 1
デバッグ用にコンパイル	VS COBOL II Release 4
	VS COBOL II Release 3
	VS COBOL II Release 2
言語の方言	OS/VS COBOL
指定された言語の方言と適合するコンパイル時および実行時の	DOS/VS COBOL
デフォルト値: Micro Focus	Micro Focus

注意)

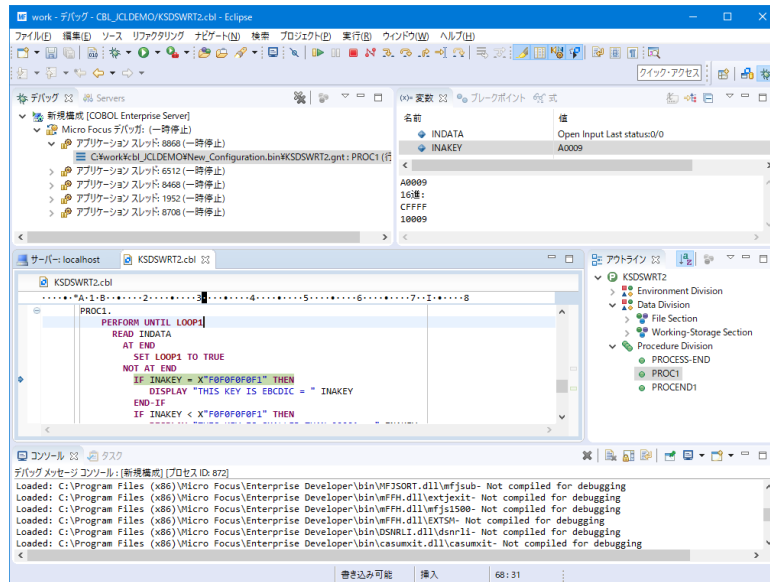
メインフレームで指定しているコンパイラ指令を精査し、デフォルトと異なる指令が存在する場合は同様の意味を持つコンパイラ指令を個別に指定します。その他の注意点については、「メインフレームコンパイラとの違い」の章で説明します。

² 製品マニュアル) [Micro Focus Enterprise Developer] > [リファレンス] > [コンパイラ指令] > [Compiler Directives - Alphabetical List] > [DIALECT]

8) デバッグ機能

COBOL ソースのデバッグは、各 IDE 内部のデバッガを利用して、簡単にプログラムのステップ実行ができ、変数の値を 16 進数で確認することもできます。また、通過ルートの動的な確認もでき、単体テストの品質向上に貢献できる機能です。

【 Eclipse のデバッグ画面 】



9) カバレッジ機能

デバッグ機能に加えてカバレッジ機能を利用すれば、通過ロジックは緑、未通過ロジックは赤と色分けされたカバレッジ結果が出力され、静的にも通過ルートが確認でき、より品質を高めることができます。

【 カバレッジ表示画面 】

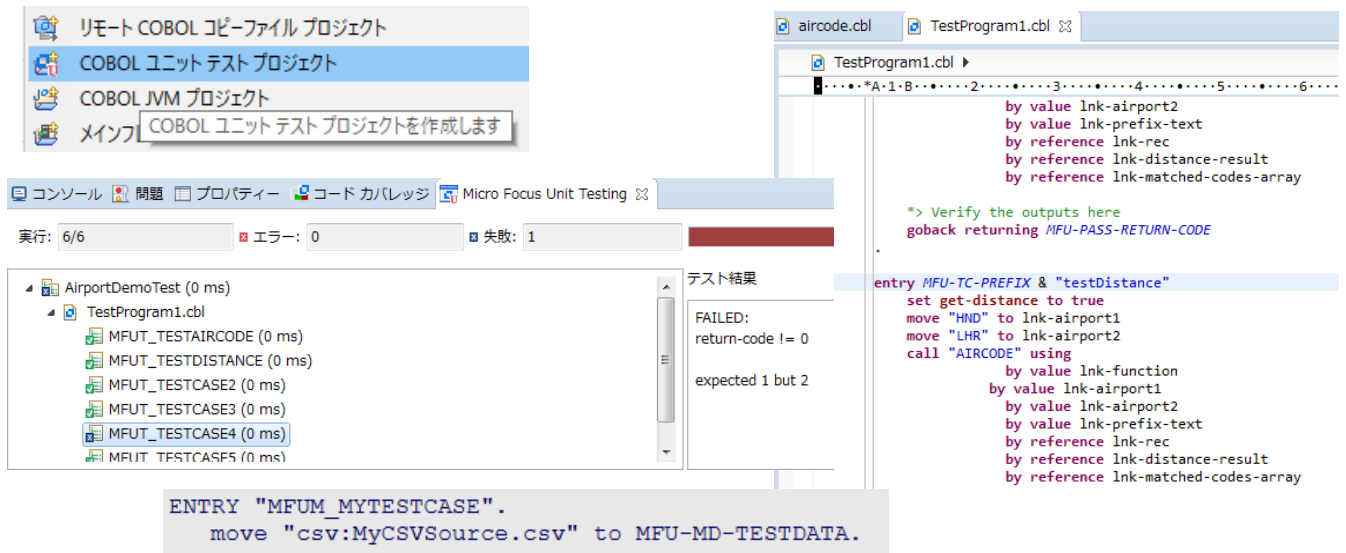
```

50      IF "00" NOT = FSTAT-I OR FSTAT-K OR FSTAT-P
51      THEN
52          CONTINUE;
53      ELSE
54          PERFORM PROC1 THRU PROCEND1;
55      END-IF.
56  PROCESS-END.
57  DISPLAY "*** OPEN ERROR ***".
58  STOP RUN.
59
60  PROC1.
61  PERFORM UNTIL LOOP1
62  READ INDATA
63  AT END
64  SET LOOP1 TO TRUE
65  NOT AT END
66  MOVE INREC TO KREC
67  MOVE INREC TO PREC
68  WRITE PREC
69  WRITE KREC
70  INVALID KEY DISPLAY "INVALID"
71  NOT INVALID KEY CONTINUE
72  END-WRITE
73  END-READ
74  END-PERFORM.
75  PROCEND1.
76  CLOSE INDATA.
77  CLOSE INDEXFILE.
78  CLOSE PRTPFILE.
                    
```

要素	カバレッジ	カバー済みブロック	未カバーブロック	ブロック全体
JCLDEMO	72.7 %	8	3	11
KSDSWRT2	72.7 %	8	3	11
PROC1	83.3 %	5	1	6
PROCEND1	100.0 %	1	0	1
PROCESS-END	0.0 %	0	1	1
Procedure Division	72.7 %	8	3	11

10) Micro Focus Unit Testing Framework (MFUnit) 機能

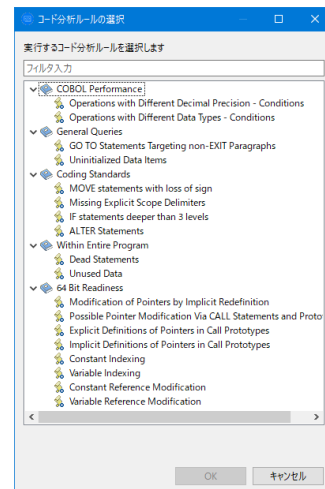
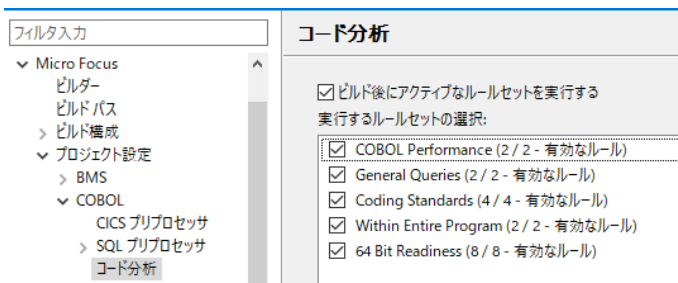
Enterprise Developer には MFUnit と称する COBOL 専用の単体テストフレームワークが含まれています。コマンドラインや IDE から利用できる xUnit スタイルのテストフレームワークで、ネイティブ COBOL、コマンドラインからはマネージ COBOL がサポートされ、単体テストの作成、コンパイル、実行、およびデバッグが可能です。この機能を利用することにより、テストの効率化が実現でき、CI ツールのサイクルに組み込むこともできます。一部、制限事項³がありますので、製品マニュアルをご確認ください。



11) コード分析機能

デフォルトで組み込まれている静的解析が利用できます。コンパイル後にコーディングチェックを目的とした静的解析を行うことも、任意のタイミングで静的解析を行うこともできます。また、CI ツールを利用する際は ソースチェックイン → コンパイル → 静的解析 → デプロイなどのサイクルに組み込むことができます。追加で独自の静的解析クエリを作成するには Micro Focus Enterprise Analyzer⁴ が必要になります。

【コンパイル後の解析指定】

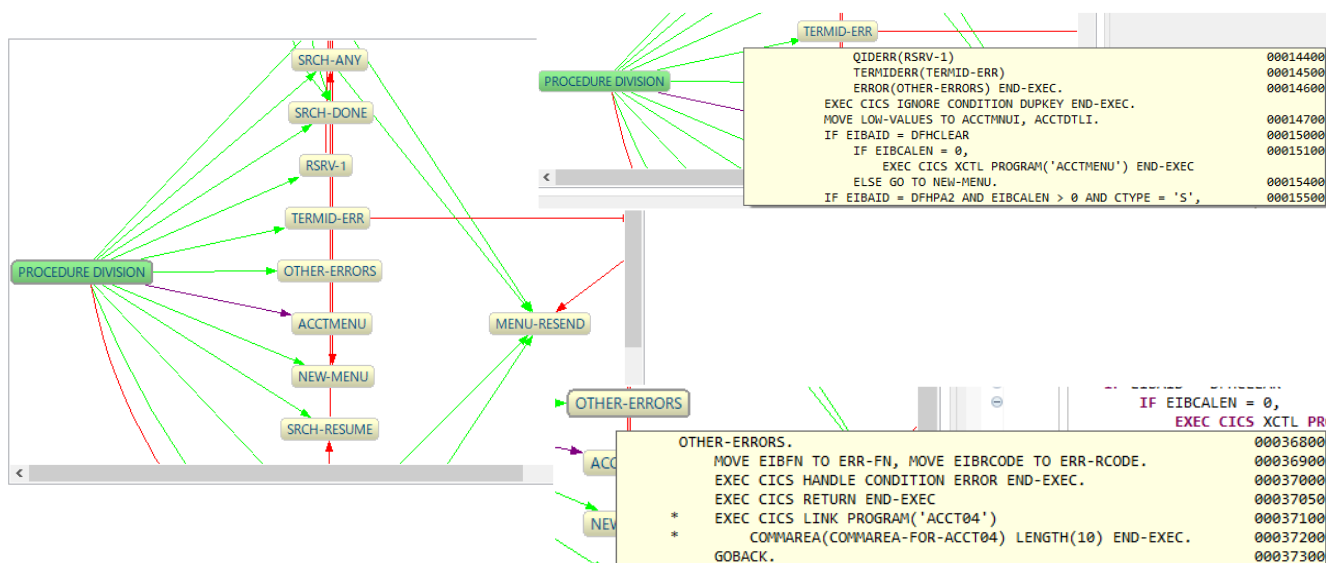


³ 製品マニュアル) [Micro Focus Enterprise Developer] > [プログラミング] > [Micro Focus Unit Testing Framework]

⁴ <https://www.microfocus.co.jp/mfproducts/enterprise/analyzer/>

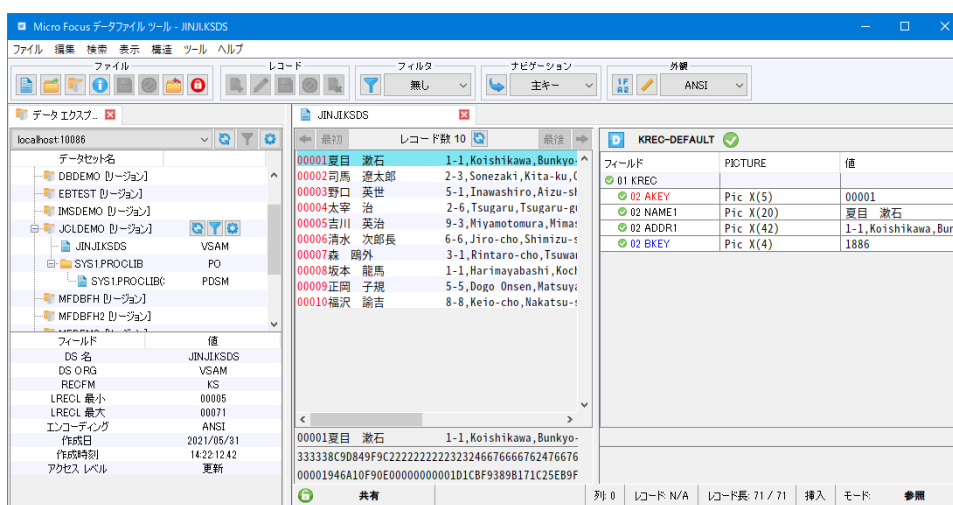
12) プログラムフローグラフ機能

プログラムフローの描画機能が利用できます。複雑なプログラムを可視化し、内容を把握するために役立ちます。



13) データファイルツール

EBCDIC 文字コード、ASCII 文字コードを持つデータをメンテナンスできるデータファイルツールを使用することにより、レコードレイアウトに沿ったメンテナンスや COMP-3 などの COBOL 特有のデータ型であっても数値として扱うことができます。また、JES 機能を持つ Enterprise Server インスタンスのカタログファイルと連携したデータメンテナンスも行うことが可能です。

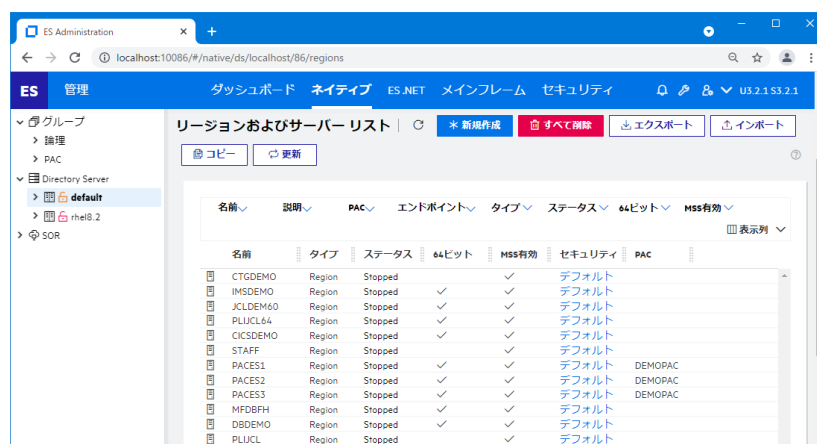


6. Enterprise Server を利用した実行

Enterprise Server には利便性を追求した機能やメインフレームとの互換性を保つための機能、運用と管理をサポートする機能など、様々な機能が備わっています。この章では実行環境の機能について説明します。

1) Enterprise Server Common Web Administration (ESCWA)

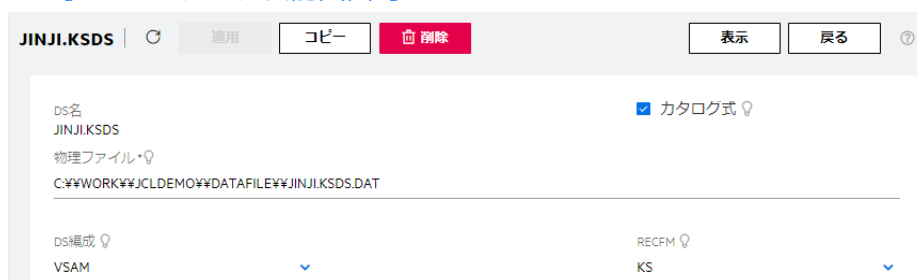
実行結果など、運用管理者が参照する Enterprise Server インスタンスの管理画面です。Web ベースの画面を利用することにより、物理的に異なり、かつ異なる OS を持つホストマシンに作成した Enterprise Server インスタンスを一括管理することができます。また、異なるホストマシン間で Enterprise Server インスタンスのコピーを行うことができ、利便性にも優れた機能です。



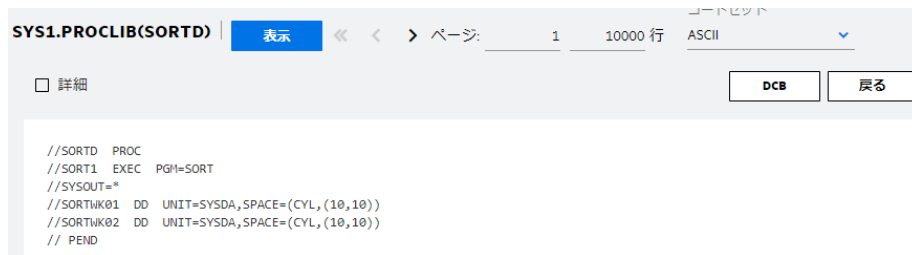
2) JCL 実行機能

実行単位である Enterprise Server インスタンスが持つ JCL エミュレーション機能により、メインフレームで使用していた JCL をオープン環境で実行できます。一般的に使用される IBM JCL ユーティリティの多くをサポートしており、スプール、カタログファイル、プロシージャ、世代管理ファイルもサポートしています。また、ACCEPT 文による入力要求に対する返答も可能です。

【 JES カタログ機能画面 】



【 JES プロシージャ表示画面 】



3) CICS 実行機能

Enterprise Server インスタンスが TP モニターの役割を持つため、OLTP ツールを別途用意する必要はありません。TN3270 エミュレータを Enterprise Server インスタンスが持つリスナーポートへ接続するだけで BMS ソースファイルに定義された画面が表示されます。

【 CICS リソース管理画面 】



4) IMS 実行機能

IMS オンライン処理においては CICS 機能と同様に Enterprise Server インスタンスが TP モニターの役割を持つため、OLTP ツールを別途用意する必要はありません。TN3270 エミュレータを Enterprise Server インスタンスが持つリスナーポートへ接続するだけで MFS ファイルに定義された画面が表示されます。また、IMS データベース制御コマンド⁵を利用することもできます。

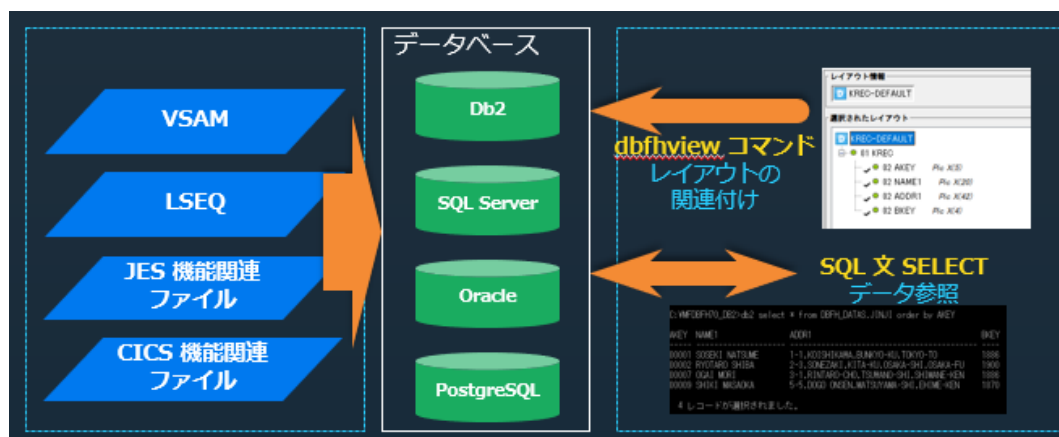
【 IMS コントロール機能画面 】



⁵ 製品マニュアル) [Micro Focus Enterprise Developer] > [プログラミング] > [メインフレーム プログラミング] > [IMS サポート] > [リファレンス] > [ユーザー インターフェイスのリファレンス] > [Enterprise Server] > [ESMAC] > [IMS 制御]

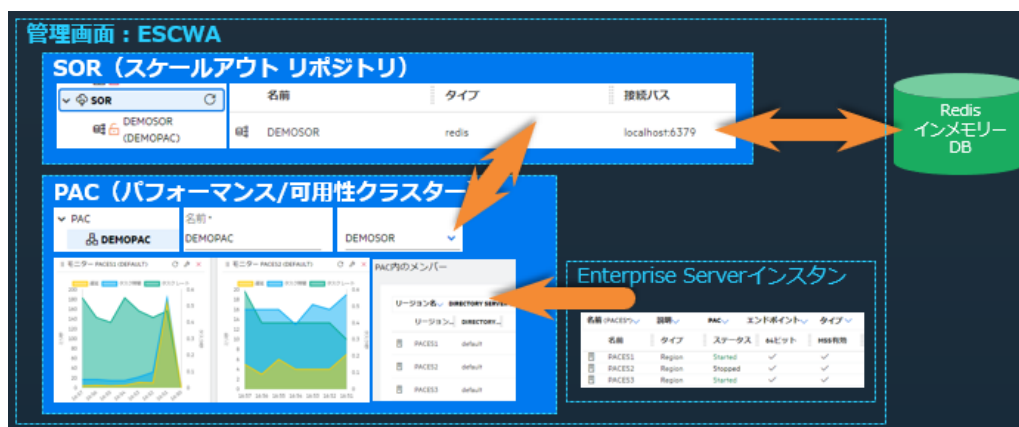
5) Micro Focus Database File Handler (MFDBFH) 機能

ソース記述の変更なく、JCL, CICS で使用するファイルをデータベースに格納できる機能です。JES 機能で使用するカタログファイルや CICS の FCT に登録されたファイルなどを、複数の Enterprise Server インスタンス間で共有することもできます。レコードレイアウトをデータファイルと関連付けることにより、SQL 文の SELECT ステートメントからレコードを参照することもできます。要件⁶については製品マニュアルをご確認ください。



6) スケールアウト パフォーマンス/可用性クラスター機能

複数の Enterprise Server インスタンスをグループ化し、負荷分散⁷を実現することができます。OS、製品バージョンなどの前提条件が一致すれば、異なるホストマシンに存在する Enterprise Server インスタンスをグループに含めることができ、物理的なリスクも回避した運用を実現することができます。また、パフォーマンスを考慮したインメモリーデータベースを利用可能です。



⁶ 製品マニュアル) [Micro Focus Enterprise Developer] > [ディプロイ] > [ディプロイ済みアプリケーションのメインフレーム サポート] > [MSS 構成および管理] > [Micro Focus ネイティブ データベース ファイル処理およびエンタープライズ サーバー リージョン データベース管理]

⁷ 製品マニュアル) [Micro Focus Enterprise Developer] > [ディプロイ] > [構成および管理] > [Enterprise Server の構成および管理] > [スケールアウト パフォーマンス/可用性クラスター]

最後に、メインフレームコンパイラとの違いに起因する注意点について、次の章で説明します。

7. メインフレーム コンパイラとの違い

メインフレームと Enterprise Developer ではコンパイラが異なるため、これに起因してコンパイルエラーが発生する可能性があります。次に示す代表的なエラーが発生するプログラムが存在し、それについて Enterprise Developer のサポートがない場合は独自のプリコンパイラを使用する、もしくはプログラムを修正するなどの対策が必要となります。

1) COBOL 構文の差異

- ① 準拠する COBOL 規格が異なる場合⁸

Enterprise Developer のサポート有 :

コンパイラ指令により EXAMINE などの旧構文を使用可能にします。

- ② 古い COBOL コンパイラ方言を使用している場合

- ③ COBOL 言語仕様としては不正であるがメインフレームコンパイラが許容している場合

- ④ 追加された予約語が利用者語と衝突している場合

Enterprise Developer のサポート有 :

REMOVE コンパイラ指令により予約語から排除することができます。また、別の予約語に読み替える指令も用意しています。

- ⑤ 特殊構文を使用している場合

Enterprise Developer のサポート有 :

コンパイラ指令により PANVALET ++INCLUDE 文などを使用可能にします。

2) COBOL 実行時挙動の差異

COBOL 標準言語仕様により結果不定と定義されている事柄について、メインフレーム実行時と Enterprise Server インスタンス実行時の挙動が異なる可能性がある代表的なものを列挙します。

- ① ON SIZE ERROR 句が指定されていない桁あふれの結果

Enterprise Developer のサポート有 :

CHECKDIV, HOSTARITHMETIC などのコンパイラ指令により不定の結果を回避できます。

⁸ 製品マニュアル) [Micro Focus Enterprise Developer] > [リファレンス] > [COBOL 言語リファレンス] > [サポートされる COBOL 言語および著作権]

- ② 数値データ項目に数値以外の値が含まれている場合
- ③ 初期値を設定していないデータを参照する場合

Enterprise Developer のサポート有：

INIT-BY-TYPE, DEFAULTBYTE コンパイラ指令により初期値を設定することが可能です。

- ④ 算術結果が異なる場合

Enterprise Developer のサポート有：

Enterprise Developer では IBM 社が公開している中間結果精度の仕様をシミュレートするコンパイラ指令を用意しており、これにより互換性を保っています。例えばコンパイラ指令 ARITHMETIC"OSVS" を指定した場合は、OS/VS の規則に則って切り捨てを行うなど、COBOL バージョンの指定に依存して中間結果仕様を適用しています。

ENTCOBOL

Enterprise COBOL for z/OS および OS/390 の規則に従って切り捨てを行います。

MF、ISO2002、ANSI

中間結果を切り捨てません。ARITHMETIC"MF"、ARITHMETIC"ISO2002"、または ARITHMETIC"ANSI" を指定した場合、切り捨てが行われないため、式の計算が最も正確になります。

OS390

COBOL for OS/390 の規則に従って切り捨てを行います。

OSVS

OS/VS COBOL の規則に従って切り捨てを行います。

VSC2

VS COBOL II および COBOL/370 の規則に従って切り捨てを行います。

8. Enterprise Developer チュートリアルと例題

Enterprise Developer の製品マニュアルには COBOL と PL/I ユーザーに向けた JCL, CICS, IMS などの様々なチュートリアル⁹を準備しています。これらのチュートリアルには IDE を使用したプロジェクトの作成方法からコンパイル、実行、デバッグまでを具体的に記述しており、例題もダウンロードできます。

メインフレーム チュートリアル

このセクションには、メインフレームまでの開発を対象としたチュートリアル (CICS、IMS、JCL、および Open PL/I アプリケーションの開発方法など) が含まれています。

[Micro Focus Enterprise Developer - CICS チュートリアル](#)

このチュートリアルのセットは、Micro Focus Enterprise Developer for Eclipse による CICS アプリケーションの開発および移行方法を案内します。

使用する例題はこちらからダウンロードしてください。

[Micro Focus Enterprise Developer - JCL チュートリアル](#)

このチュートリアルのセットは、Micro Focus Enterprise Developer for Eclipse による JCL バッチアプリケーションの開発および移行方法を案内します。

使用する例題はこちらからダウンロードしてください。

[Micro Focus Enterprise Developer - IMS チュートリアル](#)

このチュートリアルのセットは、Micro Focus Enterprise Developer for Eclipse による IMS アプリケーションの開発および移行方法を案内します。

使用する例題はこちらからダウンロードしてください。

[Micro Focus Enterprise Developer - DB アクセスチュートリアル](#)

このチュートリアルのセットは、Micro Focus Enterprise Developer for Eclipse による SQL アクセスの開発および移行方法を案内します。

使用する例題はこちらからダウンロードしてください。

[Micro Focus Enterprise Developer - PL/I JCL チュートリアル](#)

このチュートリアルでは Micro Focus Enterprise Developer for Eclipse による PL/I 言語の JCL アプリケーションの開発および移行方法を案内します。

使用する例題はこちらからダウンロードしてください。

[Micro Focus Enterprise Developer - PL/I プログラムチュートリアル](#)

このチュートリアルのセットは、Micro Focus Enterprise Developer for Eclipse による PL/I プログラムの開発方法を案内します。

使用する例題はこちらからダウンロードしてください。

[Micro Focus Enterprise Developer - CICS システム開発チュートリアル](#)

このチュートリアルでは Micro Focus Enterprise Developer for Eclipse を使用して CICS システム開発を実施する方法を案内します。

⁹ 製品マニュアル) [Micro Focus Enterprise Developer] > [ここからはじめよう] > [Getting Started] > [メインフレーム チュートリアル]

9. おわりに

メインフレーム環境では資源を開発者全員で共有するために、急を要するコンパイルは優先度の調整を行う、変数の値を確認するためにデバッグ文を入れたプログラムを再コンパイルする、などの開発スタイルが多く見受けられます。一方、オープン環境の Enterprise Developer は Eclipse, Visual Studio といった業界標準の IDE にアドオンする形で COBOL, PL/I アプリケーションの開発ができ、Visual Studio Code では COBOL 拡張機能をインストールすることで COBOL の開発環境をより軽く迅速に行うことができます。他開発言語と同様の開発スタイルを採用したステップ実行を利用したデバッグ、変数値の動的な確認、処理の流れの動的な把握を行うことができ、開発業務のモダナイゼーションにより効率的な開発環境を整えることができる製品です。

また、開発用実行環境の Enterprise Server インスタンスを開発者ごとに占有することができるため、コンパイルや実行時に他開発者との調整は必要なく、これによる開発工数の削減も見込めます。

Enterprise Developer の機能をご理解いただき、開発業務のモダナイゼーションをご検討いただければ幸いです。

補足 : 稼働環境

本書は下記環境を基に記述しています。

1) OS

Windows 10 Enterprise

2) プロセッサ

Intel® Core™ i7-3770 CPU @ 3.40HGz 3.39 GHz

3) システムの種類

64 ビットオペレーティング システム x64 ベース プロセッサ

4) Micro Focus 製品 バージョン

Micro Focus Enterprise Developer 7.0J for Windows

注意) Micro Focus Enterprise Server 7.0J for Windows と同等の開発用実行環境を含んでいます。

5) 製品マニュアル バージョン

Micro Focus Enterprise Developer 7.0 for Eclipse

Micro Focus Enterprise Developer 7.0 for Visual Studio 2019