



# Micro Focus Visual COBOL 6.0J Micro Focus Enterprise Developer 6.0J

## 新機能・強化機能概要

マイクロフォーカス合同会社

# 主な新機能、強化機能一覧

## ■ 開発環境機能の強化

- .NET Core 3.1 を正式サポート
- パッケージ構成ユーティリティの追加
- COBOL 言語の拡張
- パフォーマンスの向上
- COBOL コンパイラ指令の追加 / オプションの追加
- コンソールサイズの拡張 / コード抽出機能の拡張
- データベースアクセスの強化
- データファイルツールの拡張
- Interface Mapping Toolkit (IMTK) 機能の強化
- Micro Focus Unit Testing Framework の強化
- デバッグ機能の強化
- Eclipse IDE の強化
- Visual Studio IDE の強化
- Eclipse / Visual Studio サポートバージョン

# 主な新機能、強化機能一覧

## ■ 実行環境機能の強化

- インスタンス定義のインポート/エクスポート機能拡張
- ファイルハンドリング機能の強化
- Enterprise Server Common Web Administration (ESCWA) の強化
- スケールアウト パフォーマンス/可用性クラスタの強化：早期導入機能

## ■ Enterprise Developer/Server 固有機能の拡張

- データベース ファイル ハンドラ (MFDBFH) 機能の強化
- IMS サポートの拡張
- JCL サポートの拡張
- PL/I サポートの拡張
- CICS サポートの拡張：テクニカルプレビュー機能

## ■ 対応プラットフォームとデータベース



# 開発環境機能の強化

# 開発環境機能の強化

## ■ .NET Core 3.1 を正式サポート

長期サポート (LTS) の .NET Core 3.1 プラットフォーム上で、  
COBOL 資産を活用することができるようになりました。

注意)

Visual Studio 2017 は .NET Core 3.1 をサポートしていないため、  
Visual Studio 2019 Version 16.4.4 以上が必要です。

.NET Core ADO プロバイダを利用することで、次のデータベースと  
連携するアプリケーションを開発できます。

- DB2
- PostgreSQL
- MySQL
- SQL Server
- Oracle

目次へ

## ■ パッケージ構成ユーティリティの追加

Windows では COBOL 機能との依存関係があることから、製品セット  
アップツールにより Microsoft Build Tool と Windows SDK パッケージ  
が製品と同時にインストールされるようになりました。

構成ユーティリティを使用して、COBOL環境で使用されているパッケージのバージョンを表示したり、インストールされたパッケージの他バージョンを使用するように環境を設定することができます。

ユーティリティ例) Windows 10 SDK のバージョン 10.0.17134.0 を使用するように COBOL 環境を更新

```
cblms -USDK:C:\WinSDK\10?10.0.17134.0
```

## ■ COBOL 言語の拡張 (1/3)

### ● IBM Enterprise COBOL 6.2 互換性の強化

- 新しい組み込み関数がサポートされました。

BIT-OF, BIT-TO-CHAR, BYTE-LENGTH, HEX-OF, HEX-TO-CHAR, NUMVAL-F, TEST-NUMVAL, TEST-NUMVAL-C, TEST-NUMVAL-F, TRIM

```
01 WK-HEX-ITEM PIC X(10).  
01 WK-CHAR-ITEM PIC X(5) VALUE 'ABCDE'.  
  
PROCEDURE DIVISION.  
  
    MOVE FUNCTION HEX-OF(WK-CHAR-ITEM) TO WK-HEX-ITEM.
```

WK-CHAR-ITEM	ABCDE
WK-HEX-ITEM	4142434445

ABCDE  
16@:  
44444  
12345

- 次の組み込み関数が National データをサポートするようになりました。  
REVERSE, ULENGTH, UPOS, USUBSTR, USUPPLEMENTARY, UVALID, UWIDTH

## ■ COBOL 言語の拡張 (2/3)

### ● IBM Enterprise COBOL 6.2 互換性の強化

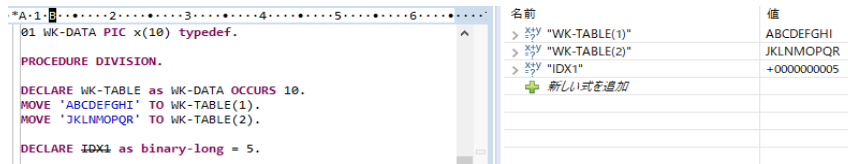
ALLOCATE ステートメントに LOC 句が追加されました。メモリが 16 MB 境界の上位または下位に割り当てられるかを定義することができます。

### ● CONSTANT 修飾子の追加

定数項目を作成する機能がネイティブ COBOL へ追加されました。コードの読みやすさと使いやすさが向上し、プログラミングエラーの検出に役立ちます。

### ● ローカル変数宣言の強化：ネイティブ COBOL のみ

マネージ COBOL では可能な  
PROCEDURE DIVISION 内  
インラインのデータ項目宣言が可能になりました。



```
*A-1-0-.....2-.....3-.....4-.....5-.....6-.....
01 WK-DATA PIC x(10) typedef.

PROCEDURE DIVISION.

DECLARE WK-TABLE as WK-DATA OCCURS 10.
MOVE 'ABCDEFGHI' TO WK-TABLE(1).
MOVE 'JKLMNOPQR' TO WK-TABLE(2).

DECLARE IDX1 as binary-long = 5.
```

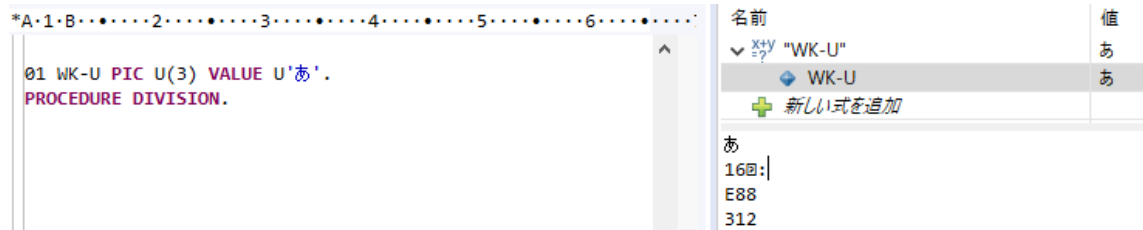
名前	値
> %Y "WK-TABLE(1)"	ABCDEFGHI
> %Y "WK-TABLE(2)"	JKLNMPQR
> %Y "IDX1"	+000000005
+ 新しい式を追加	



## ■ COBOL 言語の拡張 (3/3)

### ● UTF-8 文字データ型の拡張 : ネイティブ COBOL のみ

IBM Enterprise COBOL バージョン 6.3 から導入された PIC U データ型がサポートされました。



### ● パフォーマンスの向上

最適化レベル opt(4) を指定して生成されるネイティブ COBOL のパフォーマンス、および PERFORM 範囲構文をインラインで配置しているコードのパフォーマンスの向上が確認されています。とくに 32 ビット Intel x86 プラットフォームでパフォーマンスが向上しています。

## ■ COBOL コンパイラ指令の追加

### ● DISPLAY-AT

色属性を指定しない DISPLAY AT および ACCEPT AT ステートメントのデフォルトの前景色と背景色が指定可能になりました。

### ● DPC-IN-DATA

DECIMAL-POINT IS COMMA 句を XML GENERATE および JSON GENERATE ステートメント出力時に適用するか制御可能になりました。

### ● ILCONDITIONPARAM : JVM COBOL, .NET COBOL のみ

メソッドパラメータとしての条件式のサポートを有効にします。デフォルトはオンです。古い COBOL 構文を使用してカスタム属性に名前付きパラメータを指定するアプリケーションは指定できません。

## ■ COBOL コンパイラ指令オプションの追加

### ● CHECKREFMOD : JVM COBOL, .NET COBOL のみ

参照変更の境界チェックをオンにする指令に NOZEROLENGTH オプションが追加されました。コンパイル時に長さがゼロであることが判明している場合は、エラーとなります。

### ● PROTOTYPE

COMP, BINARY, COMP-4, COMP-5 および COMP-X データ項目の呼び出しプロトタイプのチェック方法を指定する指令に、プロトタイプが必要かどうかを指定するオプションおよびプロトタイプの不一致に対して発行されるエラーメッセージの重大度が指定可能になりました。

## ■ コンソールサイズの拡張：ネイティブ COBOL のみ

Screen Section で使用するコンソールサイズが最大 255 行、255 列に拡張されました。実行時チューナー (COBCONFIG) へ screen\_lines, screen\_cols を指定することにより実現します。

Example

```
UNIX:  
COBCONFIG=/home/mydir/cobconfig  
export COBCONFIG
```

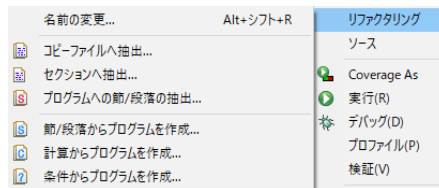
```
Windows:  
SET COBCONFIG=/home/mydir/cobconfig
```

## ■ コード抽出機能の拡張

### ● プログラムコードの抽出

COBOL プログラムのロジックを抽出して新しいプログラムやコピー句に移動するためのコマンドが、エディターのコンテキストメニューから利用可能になりました。Eclipse ではリファクタリングメニューからも利用することができます。

Eclipse)



目次へ

## ■ データベースアクセスの強化 (1/2)

### ● HCO for DB2 LUW の強化

複数行のフェッチ (MRF) および挿入 (MRI) ステートメントと配列の更新と削除ステートメントが追加されました。DB2 LUW バージョン 11.1 Mod 4 フィックスパック以降でサポートされます。

### ● XA スイッチモジュール拡張 : Windows のみ

Vault 指定で機密情報を保存できるようになりました。製品にはカスタマイズ可能なプログラム ESXAEXTCFG.CBL が含まれおり、これを使用して xa-open 文字列をシークレットとして保存することができます。

```
procedure division using by reference xaCustomizationInfo.
    PERFORM SETUP-VAULT
    PERFORM OPEN-VAULT

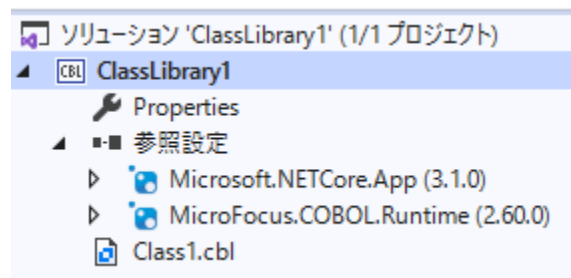
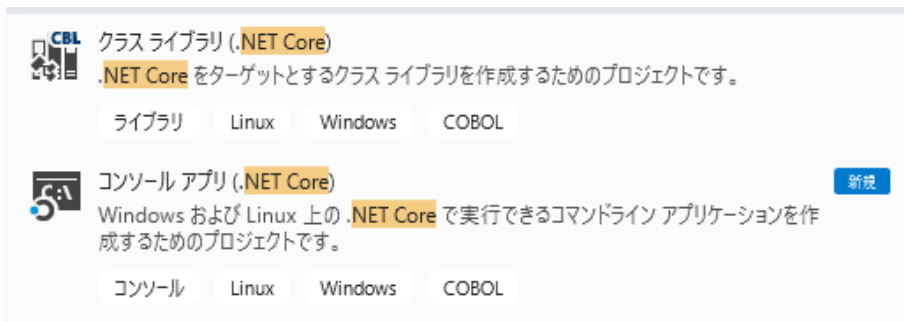
01 xaCustomizationInfo.
03   returnCode          pic 9(4) comp-5.
03   reasonCode         pic 9(4) comp-5.
03   xaOpenString       pic x(256).
03   password           pic x(32).
03   entriesUsed        pic 9(4) comp-5.
03   nameValuePairs occurs 64.
05     vName            pic x(16).  *> space terminated*
05     vValue          pic x(256).  *> null terminated*
```

# 開発環境機能の強化

## ■ データベースアクセスの強化 (2/2)

### ● OpenESQL の強化

.NET Core 3.1 と OpenESQL 指令である DBMAN=ADO を使用した「コンソールアプリケーション」および「クラスライブラリ」に COBOL .NET Core プロジェクトテンプレートが追加されました。

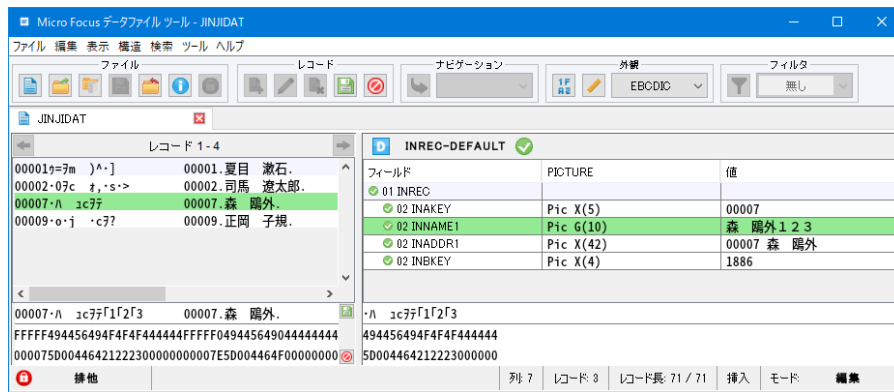
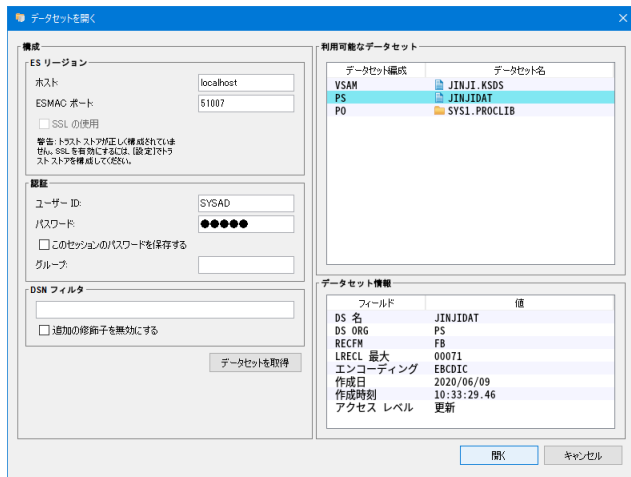




## ■ データファイルツールの拡張 (2/3)

### ● Enterprise Server インスタンスとの連携

JES 機能のカタログファイルと接続し、レコードフォーマットを適用して DBCS データを含むデータメンテナンスを実施することができます。





## ■ データファイルツールの拡張 (3/3)

### ● ファイル操作

- DBCS データを含むファイルを、排他または共有モードでオープンすることができます。
- デバッグ実行ファイルから DBCS 項目を含むレコードレイアウトを作成することができます。

The screenshot displays the Micro Focus Data File Tool interface. On the left, a table shows record data with a red box highlighting the '排他' (Exclusive) mode button. The main window shows a file tree with '01 INREC' selected, and a right-hand pane showing the selected record layout 'INREC-DEFAULT' with its sub-elements.

レコード	フィールド	値
00001	夏目	激石
00002	司馬	遼太郎
00007	森	隼外
00009	正岡	子規

利用可能なレイアウト

- FD INDATA
  - 01 INREC
    - 02 INAKEY Pic X(5)
    - 02 INNAME1 Pic G(10)
    - 02 INADDR1 Pic X(42)
    - 02 INBKEY Pic X(4)
- FD PRTFILE
  - 01 PREC
- Working-Storage Section

選択されたレイアウト

- INREC-DEFAULT
  - 01 INREC
    - 02 INAKEY Pic X(5)
    - 02 INNAME1 Pic G(10)
    - 02 INADDR1 Pic X(42)
    - 02 INBKEY Pic X(4)

## ■ Interface Mapping Toolkit (IMTK) 機能の強化 (1/2)

### ● RESTful サービスのフィルタリング機能

- URL パスやクエリパラメータを利用して、サービス応答結果の絞り込みが行えるようになりました。

下記の例では、fuelType が gasoline の結果を戻します。

URL パスの例)

<http://localhost:9003/temp/path/CarTracker/1.0/cars?doors=2&fuelType=gasoline>

- サービスの応答結果から取得項目を制限する機能が追加されました。

下記の例では、応答結果には stocknumber, price 以外の項目は戻されません。

URL パスの例)

[http://localhost:9003/temp/path/CarTracker/1.0/cars?\\$fields=stocknumber,price](http://localhost:9003/temp/path/CarTracker/1.0/cars?$fields=stocknumber,price)

## ■ Interface Mapping Toolkit (IMTK) 機能の強化 (2/2)

### ● Open API の対応

Open API 3.0, Swagger 2.0 仕様に対応し、定義ファイルからテストクライアントの生成や、サービスディプロイ時にサービス定義ファイルが出力されるようになりました。

### ● REST API Discovery の対応

ディプロイされたサービス API 情報を参照する API Discovery に対応しました。

URL パスの例)

<http://localhost:9003/tempopath/CarTracker/1.0>

```
[
  {
    "rel" : "GetCars",
    "href" : "/cars"
  },
  {
    "rel" : "AddCar",
    "href" : "/cars"
  },
  {
    "rel" : "AddCarFeature",
    "href" : "/cars/{stockNumber}/features"
  }
]
```

目次へ

## ■ Micro Focus Unit Testing Framework の強化

### ● データドリブンテストの拡張

- テストデータ (csv) 形式にテストデータ毎の説明文を外部ファイル指定で定義、参照できるようになりました。

```
call MFU-GET-FILE using
    by reference ws-filename
    by reference ws-pointer
    by reference ws-size
end-call
```

InitialStatus	EarnedPoints	FinalStatus	Description
Bronze	100	Bronze	Test Case 1
Platinum	14000	Diamond	@lorem.txt

- データドリブンテスト用のテスト前処理 (MFUDS\_testcasename)、後処理 (MFUDE\_testcasename) が追加されました。

```
entry "MFUDS_" & "TESTFLYER_FILE".
.
entry "MFUDE_" & "TESTFLYER_FILE".
.
```

### ● 単体テスト対象モジュールの追加

実行モジュールを利用して単体テストを実行できるようになりました。

目次へ

# 開発環境機能の強化

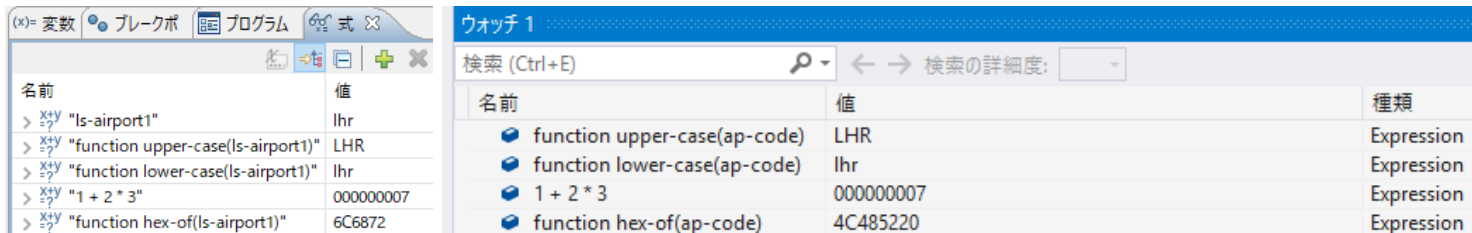
## ■ デバッグ機能の強化 (1/3)

### ● 計算や組み込み関数をサポート

Eclipse

Visual Studio

Eclipse では 式ビュー、Visual Studio ではウォッチウィンドウにて、計算や変数を参照する組み込み関数の呼び出しをサポートしました。



名前	値	種類
> $x+y$ "ls-airport1"	lhr	
> $x+y$ "function upper-case(ls-airport1)"	LHR	Expression
> $x+y$ "function lower-case(ls-airport1)"	lhr	Expression
> $x+y$ "1 + 2 * 3"	000000007	Expression
> $x+y$ "function hex-of(ls-airport1)"	6C6872	Expression

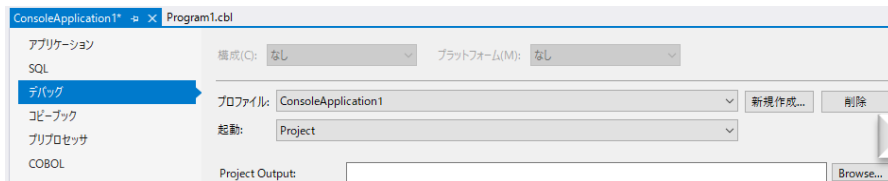
  

名前	値	種類
function upper-case(ap-code)	LHR	Expression
function lower-case(ap-code)	lhr	Expression
1 + 2 * 3	000000007	Expression
function hex-of(ap-code)	4C485220	Expression

### ● デバッグプロファイル機能の追加

Visual Studio

複数のデバッグ設定をプロファイルとして保存し、プロファイルを利用したデバッグ実行が可能になりました。



目次へ

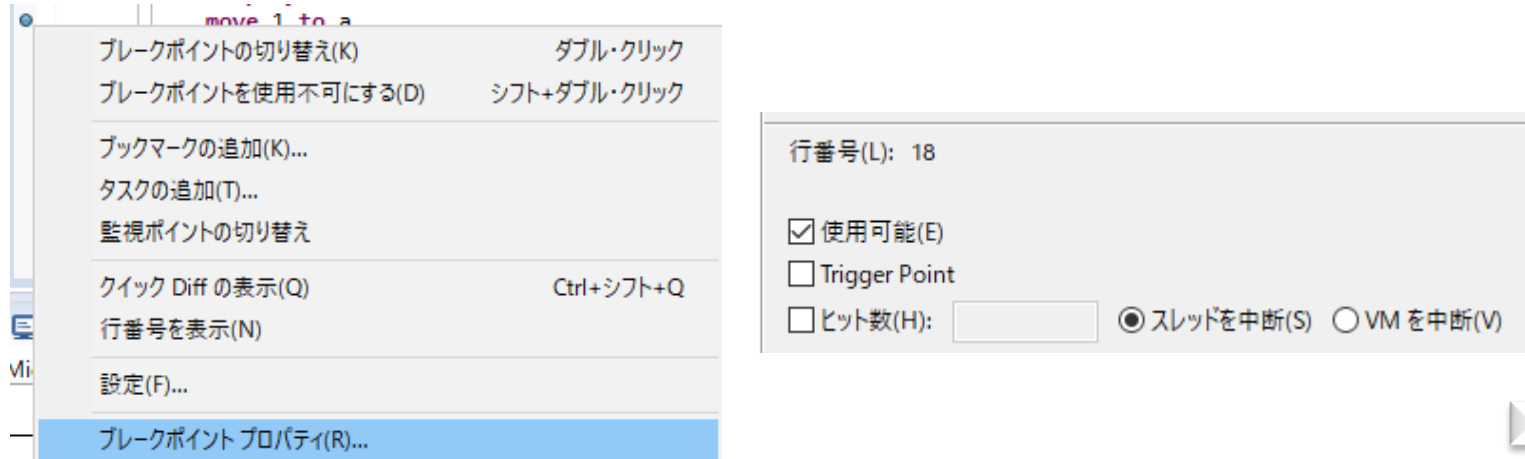
# 開発環境機能の強化

## ■ デバッグ機能の強化 (2/3)

### ● コンテキストメニューの拡張

Eclipse

JVM COBOL プロジェクトでブレークポイント プロパティ（ブレークポイントの設定）画面への移動がコンテキストメニューから行えるようになりました。ヒット数による一時停止条件指定や、トリガーポイント設定が行えます。



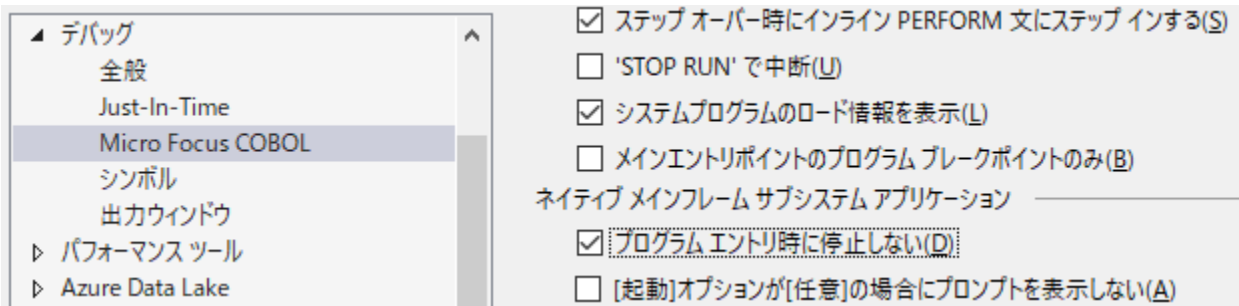
# 開発環境機能の強化

## ■ デバッグ機能の強化 (3/3)

Visual Studio

### ● デバッグオプションの追加

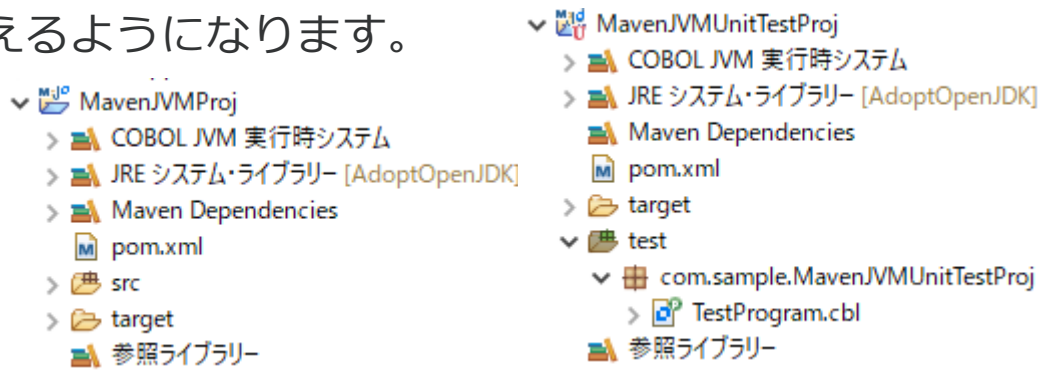
ネイティブ COBOL プログラムのエントリ時点でデバッグの一時停止を抑止するオプションが追加されました。



### ■ Eclipse IDE の強化 (1/3)

#### ● Maven プロジェクト形式のサポート

JVM COBOL / JVM COBOL ユニットテストプロジェクトが Maven プロジェクト形式をサポートしました。Maven プロジェクトを利用することで、プロジェクト管理を容易に行えるようになります。



既存プロジェクトから Maven プロジェクトへの変換も可能です。

比較対象(A)	>	Create module-info.java
構成	>	Convert to Maven Project



### ■ Eclipse IDE の強化 (2/3)

#### ● COBOL ビルド設定サマリー表示機能の強化

ビルド設定表示時に上位レベルから継承した設定値と、個別設定した設定値を区別できるようになりました。

COBOL コンパイル設定: <括弧内の継承された設定>

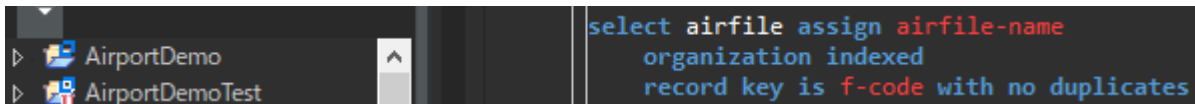
```
<CHARSET"ASCII" DIALECT"MF" SOURCEFORMAT"fixed" anim EXITPROGRAM"ANSI"  
WARNING"1" MAX-ERROR"100"> LIST()
```

#### ● 参照先コピーブックのパス表示

COPY 句にカーソルを合わせることで、参照しているコピーブックへのパスが表示されるようになりました。

#### ● Eclipse IDE の ダークテーマに対応

外観にダークテーマを選択して COBOL 開発が行えます。



```
select airfile assign airfile-name  
organization indexed  
record key is f-code with no duplicates
```

### ■ Eclipse IDE の強化 (3/3)

#### ● COBOL エディター上の折りたたみ機能の強化

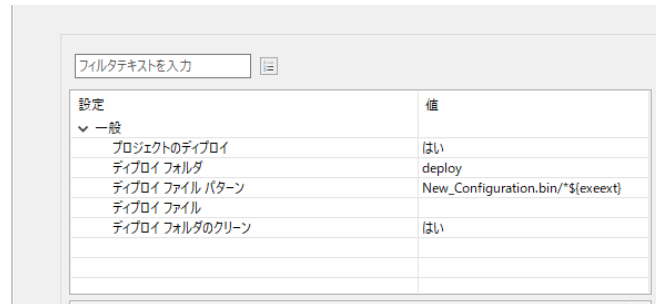
IF, EVALUATE, PERFORM 句単位でコードの折りたたみが可能になりました。

```
evaluate true
  when get-matches
    perform get-code-matches
  when get-distance
    perform distance-between-airports
  when get-details
    perform lookup-one-airport
  when open-file
    perform open-airfile
  when close-file
    perform close-airfile
  when display-record
    perform display-airport
end-evaluate
```

#### ● ビルド後のディプロイ設定を追加

ビルドにより生成されたモジュールを指定フォルダへコピーする設定が追加されました。アーカイブファイルを作成するフォルダなどへコピーが自動で行えます。

- Micro Focus
  - ビルダー
  - ビルドパス
  - ビルド構成
    - COBOL
    - イベント
    - ディプロイ
    - ビルド環境
    - リンク
  - プロジェクト設定
    - 指令の確認
    - 実行時構成
  - WikiText
  - タスク・タグ
  - タスク・リポジトリ



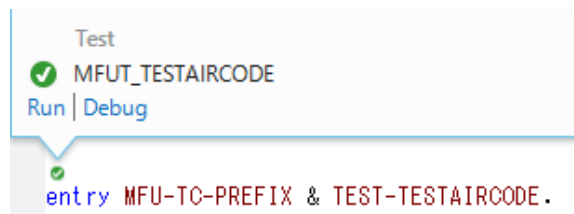
設定	値
▼ 一般	
プロジェクトのディプロイ	はい
ディプロイフォルダ	deploy
ディプロイファイル パターン	New_Configuration.bin/"\${exeext}
ディプロイファイル	
ディプロイフォルダのクリーン	はい

## ■ Visual Studio IDE の強化 (1/2)

### ● コードレンズに対応

COBOL エディター上でクラス、段落、節の参照数の表示に加え、単体テストではテスト実行指示と結果が表示されるようになりました。

```
1 reference  
calculate-airport-distance section.  
    move a1-latitude to file-angle  
    perform calculate-airport-distance
```



### ● END 動詞のツールチップに開始文を表示

END-IF, END-PERFORM, END-OF-STATEMENTのツールチップで開始文が表示されるようになりました。

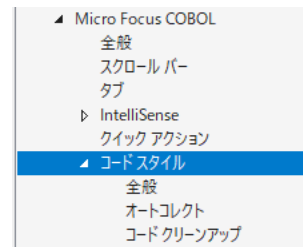
```
if flag1 = 1  
    move f-rec to a1-rec  
    move lnk-airport2 to airport  
    perform find-airport  
    if flag2 = 1  
        move f-rec to a2-rec  
        perform calculate-airport-distance  
        move distance to distance-km  
        move distance-m to distance-miles  
    end-if  
end-if
```

if flag2 = 1

## ■ Visual Studio IDE の強化 (2/2)

### ● コードスタイル設定を整理

コードスタイル設定を容易に行えるよう、新たなカテゴリに集約しました。



### ● 電球アイコンの追加

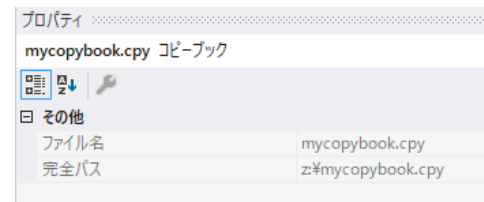
スコープの終了文 (END-IFなど) を挿入したり、セクションを新しい



プログラムに抽出するクイックアクションが使用できるようになりました。

### ● コピーブックプロパティに完全パスを表示

コピーブックファイルのプロパティウィンドウに、ファイル名に加え、完全パスが表示されるようになりました。



## ■ Eclipse 4.8 (Photon) 64ビットがデフォルトインストール

### ● Eclipse 4.8 (Photon)

製品に同梱されています。

### ● Eclipse 4.7 (Oxygen)

サポートされています。

- Eclipse を実行するには 64 ビットの Java 8 が必要です。
- Adopt OpenJDK 8 (u202) と Hotspot がインストールされます。
- Java 11 で Eclipse は実行できません。

## ■ Visual Studio 2017, 2019 をサポート

.NET Core 3.1 を使用する場合は Visual Studio 2019 Version 16.4.4 以上が必要です。



# 実行環境機能の強化

# 実行環境機能の強化

## ■ インスタンス定義のインポート/エクスポート機能拡張

### ● casesxml ユーティリティの追加

エンタープライズサーバーインスタンスの定義を xml ファイルにインポートおよびエクスポートできるようになりました。 インスタンス領域の構造化定義、CICSリソース定義ファイル、およびカタログ情報を含む xml ファイルを扱うことができます。

```
c:\YES>casesxml export -sa -rJCLDEMO -wc:yes
Processing -x option...
(c) Copyright 1991 - 2020 Micro Focus or one of its affiliates.
Micro Focus Directory Server daemon: Version 1.26.46
Exporting of resource entries in progress...
Export of resource entries complete.
Exporting of resource entries in progress...
Export of resource entries complete.
Catalog XML utility ended normally
```

名前

- JCLDEMO.xml
- JCLDEMO\_catlg.xml
- JCLDEMO\_export.xml
- JCLDEMO\_sit.xml
- JCLDEMO\_stul.xml

### 【カタログ情報】

```
<r-CAT-COMPOSITE-REC><R-CAT-REC><R-DSN><R-DSNAME>JINJI.KSDS</R-DSNAME><R-MEMBER> </R-MEMBER></R-DSN>
N>&lt;CatalogFolder&gt;%JINJI.KSDS.DAT</R-PC-DSN><R-CATALOGED>1</R-CATALOGED><R-VOLUME> </R-VOLUME><
KS</R-RECFM><R-DSORG>VSAM</R-DSORG><R-CHARSET>A</R-CHARSET><R-EXPDT>0</R-EXPDT><hex.R-SPACE-UNIT>00C
ex.R-SPACE-UNIT><R-CREATE-DATE>20200521</R-CREATE-DATE><R-CHG-DATE>20200521</R-CHG-DATE><R-CHG-TIME>
</R-CHG-TIME><R-COMPILE-DATE>0</R-COMPILE-DATE><R-COMPILE-TIME>00000000</R-COMPILE-TIME><R-DEFAULT-E
```

目次へ

# 実行環境機能の強化

## ■ ファイルハンドリング機能の強化

### ● mfsecretsadmin ユーティリティの追加

Fileshare 機能のパスワードファイルを Vault 指定で保存できるようになり、機密性の高いユーザー資格情報が暗号化されるようになりました。通常の方法でパスワードファイルを作成し、mfsecretsadmin ユーティリティを使用してアップロードします。

example:

```
mfsecretsadmin write microfocus/fh/pass.dat -file pass.dat -vault aesvault
```

Vault 指定で保存されているファイルを使用するには、/ uv オプションを使用して Fileshare サーバーを起動します。

目次へ

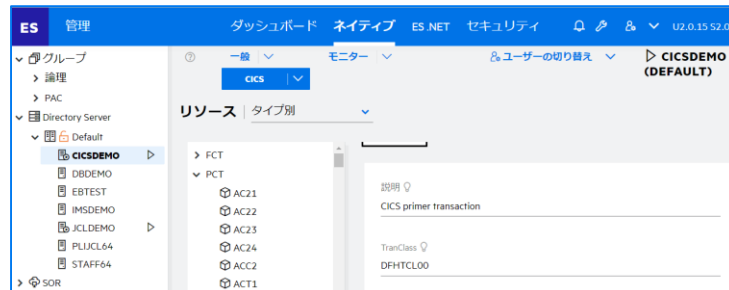


# 実行環境機能の強化

## ■ Enterprise Server Common Web Administration (ESCWA)(1/2)

### ● 新たな Enterprise Server 管理画面のリリース

既存管理画面と同等に各インスタンスに IMS、PL/I、MQ、XA設定やセキュリティの設定が利用可能になりました。

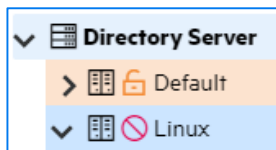
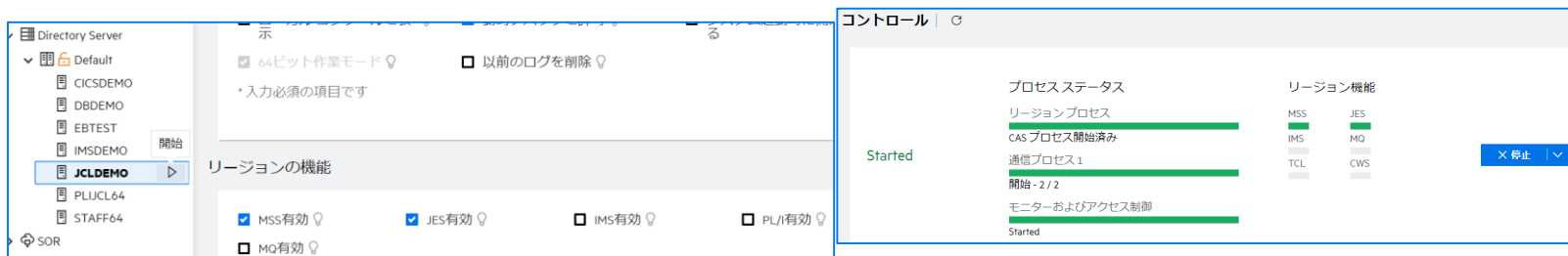


# 実行環境機能の強化

## ■ Enterprise Server Common Web Administration (ESCWA)(2/2)

### ● 操作性の向上

Directory server の色分けや、ナビゲーションツリーからインスタンスを開始、停止することができるなど、操作性が向上しました。



### 【最低限必要なブラウザバージョン】

- Microsoft Edge (version 80)
- Firefox (version 68)
- Google Chrome (version 80)
- Opera (version 67)
- Internet Explorer (IE) はサポートされていません。

### 【プラットフォーム】

- HP-UX はサポートされていません。

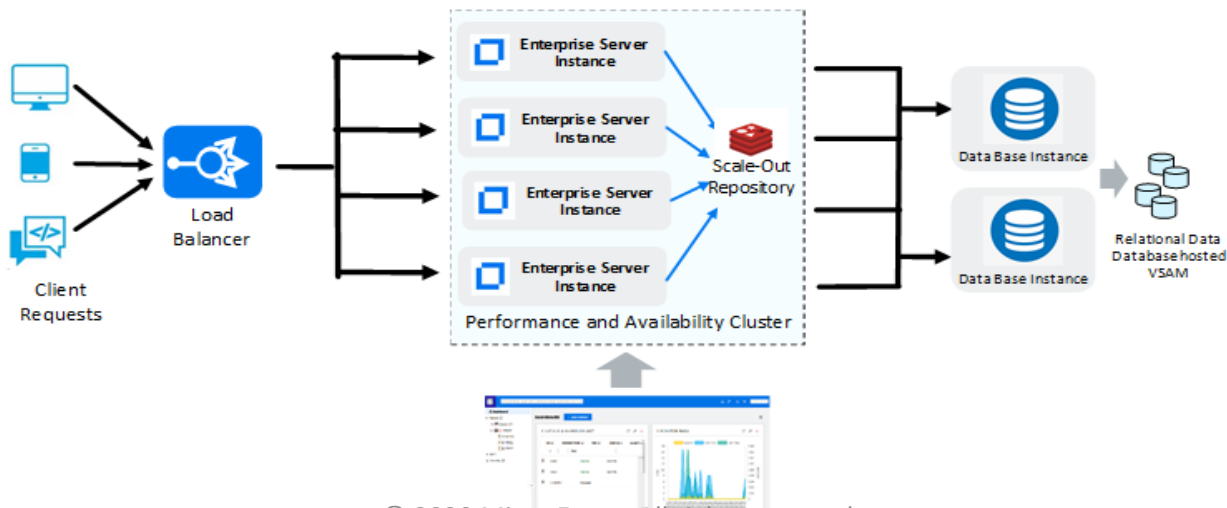
目次へ

# 実行環境機能の強化

## ■ スケールアウト パフォーマンス/可用性クラスターの強化 (1/2) (Performance and Availability Clusters (PAC))

EAP : 早期導入機能

PAC は複数のエンタープライズサーバーインスタンスが連携して動作し、スケールリングと可用性を向上させるように設計されており、そのアーキテクチャにより、障害 (single points of failure (SPOF)) を最小限に抑えることができる機能です。



# 実行環境機能の強化

## ■ スケールアウト パフォーマンス/可用性クラスターの強化 (2/2) (Performance and Availability Clusters (PAC))

EAP : 早期導入機能

### ● 強化と改善

#### ● Redis のサポート

SOR として利用することができます。  
また、MFREDIS 構成ファイルにより、  
ネットワーク再接続の指定が可能になりました。

開発環境製品に同梱されている Windows  
バージョンの Redis は、開発用のテスト  
使用を目的としています。

#### ● Kubernetes のサポート : Linux のみ

Kubernetes クラスターで実行されている場合、クラスター内で MFDS をホストしている Pod を自動的に検出し、ESCWA ユーザーインターフェースに表示  
するよう設定可能になりました。

目次へ



# Enterprise Developer/Server 固有機能の拡張

## ■ データベース ファイル ハンドラ (MFDBFH) 機能の強化 (1/2)

CICS および JCL アプリケーションがファイルをデータベースに格納できる MFDBFH 機能で、既存の Microsoft SQL Server, PostgreSQL に加えて DB2 と Microsoft Azure データベースが利用できるようになりました。

### 【 Linux/UNIX 】

Microsoft SQL Server 2017 の最小要件でもある、Red Hat Linux 7.3 または 7.4 x86-64 と SUSE v12 SP2 x86-64 でサポートされます。

Database	ODBC driver
Microsoft SQL Server 2017 (including Express and Azure SQL Database editions)	ODBC 13.1 for SQL Server  <b>Note:</b> SQL Server 2017 installs versions 13 and 17 (x64 only)
PostgreSQL 10.x	psqlodbc 11
Db2 LUW 10.5	n/a

Component	Version
Driver manager (MSSQL, Azure, and PostgreSQL installations only)	unixODBC 2.3.2

### 【 Windows 】

Database	ODBC driver
Microsoft SQL Server 2008 (including Express and Azure SQL Database editions)	ODBC 13.1 for SQL Server
PostgreSQL 10.x (incl. Amazon Aurora Postgres)	psqlodbc 11
Db2 LUW 10.5	n/a

## ■ データベース ファイル ハンドラ (MFDBFH) 機能の強化 (2/2)

- 順編成ファイル、行順編成ファイル、OUTPUT オープンされた ESDS/KSDS/RRDSファイルの I-O を最適化します。
- ODBC DSN を使用する代わりに、接続文字列を使用してデータベース接続を確立できます。
- スクリプトファイルを使用してデータベースを作成できます。
- dbfhconfig コマンドを使用してデータベース構成ファイルを管理できます。

【 commands.txt の内容例 】

```
-add -file:my.cfg -server:MYSRV -provider:ss -comment:"SQL Server server"  
-add -file:my.cfg -server:MYSRV -dsn:SS.MASTER -type:database -name:master  
-add -file:my.cfg -server:MYSRV -dsn:SS.FILES -type:datastore -name:FILES  
-add -file:my.cfg -server:MYSRV -dsn:SS.CAS.ESDEMO -type:region -name:ESDEMO
```

【 実行コマンド 】

dbfhconfig @commands.txt

JES タブ設定値の例)

The screenshot shows a configuration window for JES (Job Entry Subsystem). It contains several text input fields. The 'System Catalog' field is highlighted with a red box and contains the value 'sql://localhost:5432/PJCLCAT/catalog.dat?folder=/CAT'. The 'Default Allocated Dataset Location' field contains 'sql://localhost:5432/PJCLCAT?type=folder;folder=/CAT/data'. Other fields include 'Job Entry Subsystem enabled' (checked), 'JES Program Path' (set to '\$JCLPROJ\icl-tutorial\bin\x86\Debug'), and 'System Procedure Library' (set to 'SYS1 PROCLIB').

## ■ IMS サポートの拡張 (1/3)

### ● メッセージキューへのアクセスサポート (BMP)

バッチメッセージ処理プログラム (BMP) は、既存のバッチタイプ処理およびデータアクセス機能に加えて、入出力を目的とした IMS メッセージキューへのアクセスを、JCL から実行することが可能になりました。

```
//S01 EXEC PGM=DFSRRRC00,REGION=&RGN,  
// PARM=(BMP,&MBR,&PSB,&IN,&OUT,  
// &OPT&SPIE&TEST&DIRCA,&PRLD,  
// &STIMER,&CKPTID,&PARDLI,&CPUTIME,  
// &NBA,&OBA,&IMSID,&AGN,&SSM,
```

### ● コマンドの強化

/START TRAN ALL, /DISPLAY Q TRAN, /DISPLAY STATUS TRAN  
コマンドが追加されました。



## ■ IMS サポートの拡張 (2/3)

### ● IMS データベース制御コマンドの強化

アクティブなトランザクションへの SUSPEND コマンドをサポートします。新しい作業単位 (UOW) の要求をブロックし、アクティブな UOW を完了させて一時停止します。これにより、ログのアーカイブなど、管理タスクのサポートが可能になります。

```
///S01 EXEC PGM=DFSRRC00,REGION=4M,
///REPORT1 PARM='BMP,DEM0001B,DEM0001T,.....,CDLI..N.N'
///IMSSERR DD SYSOUT=*
///PRINTDD DD SYSOUT=*
///SYSOUT DD SYSOUT=*
///SYSPRINT DD SYSOUT=*
///IMSTRACE DD SYSOUT=*
//SYSOMP DD *
SSA*
PCB*
LOADAREA*
/*
```

### ● IMS BTSトレースの改善

アクティブな SSTM およびバッチジョブからも表示できるようになりました。これにより、トレース出力を表示するためにジョブステップを完了させる必要や MPR を停止する必要がなくなりました。

```
Sample BTS output
This sample BTS output is from the DEM0001B program, and shows the DL/I call function, SSAs, PCB, and I/O area:
CBLTOLI: .....1-----2-----3-----4-----5
CBLTOLI: CALL :GN
CBLTOLI: 4422
CBLTOLI: 7E00
CBLTOLI: SSAS : NONE
CBLTOLI: PCB=003:DEM0030001 AP ....DEMOHDRS...(.....TEST
CBLTOLI: 444433443322452200004444440500002000000000000545322
CBLTOLI: 450F034401001000000450F423000800200000000453400
```

目次へ

## ■ IMS サポートの拡張 (3/3)

### ● ユーザー DB ハンドラー出口の拡張

全機能データベースで使用可能だった出口が GSAM データベースでも使用できるようになりました。これによりデータベースの DB カタログ情報をプログラムで制御できます。製品には出口の作成を支援するためにテンプレートファイル USERDB.CBL と説明テキストファイル USERDB.TXT が含まれています。

```
01 IMSDB-SSA14-GRP. PIC X(4000).
01 IMSDB-SSA15-GRP. PIC X(4000).
PROCEDURE DIVISION USING IMSDB-PARMS
                          IMSDB-DLI-FUNC
                          IMSDB-DLI-PCB
                          IMSDB-DLI-IO-AREA
                          IMSDB-SSA01-GRP
                          IMSDB-SSA02-GRP
                          IMSDB-SSA03-GRP
                          IMSDB-SSA04-GRP
                          IMSDB-SSA05-GRP
                          IMSDB-SSA06-GRP
                          IMSDB-SSA07-GRP
                          IMSDB-SSA08-GRP
                          IMSDB-SSA09-GRP
                          IMSDB-SSA10-GRP
                          IMSDB-SSA11-GRP
                          IMSDB-SSA12-GRP
                          IMSDB-SSA13-GRP
                          IMSDB-SSA14-GRP
                          IMSDB-SSA15-GRP.
MOVE ZERO TO IMSDB-RC.
IF IMSDB-VERS = ZERO
OR IMSDB-VERS > 5
  DISPLAY 'WARNING - UNRECOGNIZED MFIMS USERDB VERSION'.
END-IF.
IF IMSDB-FUNC = 'DLI'
  PERFORM 1000-DLI THRU 1000-EXIT.
ELSE
  IF IMSDB-FUNC = 'GETTYPE'
    PERFORM 2000-GETTYPE THRU 2000-EXIT.
```

```
-----
IMSDB-FUNC = 'DLI'
A 'DLI' function indicates that this program must process a DLI
database call for a confirmed user database.
VERS      : Contains the API version level 'agreed to' during the
           GETVER call.
RC        : Return code - is initialized to zero. It should not be
           changed when using API levels 1 and 2. For VERS values
           at least 3, setting the return code to 8 or larger causes
           MFIMS to issue a pop-up ABEND error code 5001 which also
           terminates the application and performs rollback.
PSBNAME   : The name of the PSB for the program issuing the DLI call.
IO-LTH    : This can be set to indicate the length of the I/O area
           for the MFIMS DLI call trace facility. It has no other
           purpose.
DLI-PARMS : This value is the number of parameters which follow this
           parameter block. A value of '1' (not possible) would
           indicate only a DLI function followed. A value of '4'
           would mean a function, PCB, I/O area, and 1 SSA followed.
FSA-LTH   : This indicates the length of the FSA when a 'FLD' call
           is made to a Fast Path MSDB database. The FSA is the
           I/O area.
```

## ■ JCL サポートの拡張

### ● プリンター出口の拡張

2つの新しいエントリにより拡張情報を取得できるようになりました。  
(ws)-extended-output には拡張出力データのバージョンが（現在は 1 に設定）設定され、(ws)-extended-data にはポインターが設定されます。  
このポインターが null の場合、拡張データがないことを意味します。製品には出口サンプルプログラムの sampprnx.cbl が含まれています。

```
*> Extended output indicator.␣  
*> If extended output is included then the following␣  
*> pointer will point to an memory block containing a␣  
*> 'outpext.cpy' block.␣  
7 ()-extended-output          pic x comp-x.␣  
7 ()-extended-data pointer.␣  
  
$if P64 set␣  
7 pic x(122).␣  
  
$else␣  
7 pic x(126).␣  
  
$end␣
```

```
*> Extended Output data items.␣  
*> Version numbers␣  
78 extended-output-version1 value 1.␣  
  
*> Current version␣  
78 extended-output-version value 1.␣  
78 afpstats-report value 'Y'.␣  
78 afpstats-none value 'N'.␣  
78 ds-name-len value 54.␣  
78 mail-address-len value 60.␣  
78 userpath-len value 260.␣  
  
5 outp-afpstats          pic x.␣  
5 outp-afpparams        pic x(ds-name-len).␣  
5 outp-replyto          pic x(mail-address-len).␣  
5 outp-mailfrom         pic x(mail-address-len).␣  
5 outp-mailto           pic x(mail-address-len) occurs 32.␣  
5 outp-mailcc           pic x(mail-address-len) occurs 32.␣  
5 outp-mailbcc          pic x(mail-address-len) occurs 32.␣  
5 outp-userpath         pic x(userpath-len) occurs 8.␣
```

## ■ PL/I サポートの拡張 (1/2)

### ● サポート項目の追加

- ENTRY 宣言の GENERIC 属性サポート
- GET ステートメントの COPY オプションサポート
- ENTRY LIMITED 変数への ENTRY 割り当てサポート
- INONLY 属性サポート
- WHEN または OTHERWISE のない SELECT 構文のサポート
- 中間演算の FIXED DECIMAL 精度を指定できる  
新しいコンパイルオプションのサポート
- 要素名に \* を含む構造体に  
LIKE を使用する機能のサポート

`-fdmaxp n,[m]`  
Specifies the maximum FIXED DECIMAL precision *n*. Recommended values are:

15	The z/OS default.
18	Open PL/I default for Enterprise Developer release 2.1 and earlier.
31	Open PL/I default for Enterprise Developer release 2.2 and later. Fixed decimal precision 31 is the maximum allowed.
15,15	Corresponds to 15.
15,31	

目次へ

# Enterprise Developer/Server 固有機能の拡張

## ■ PL/I サポートの拡張 (2/2)

### ● 組み込み関数の拡張

Eclipse

Visual Studio

- FIXEDDEC および FIXEDBIN 機能の追加
- 構造体への TRIM 関数の使用をサポート

### ● IO 機能の拡張

Eclipse

Visual Studio

- FILEREAD, FILEWRITE, FILETELL, FILESEEK ステートメントを介した PL/I バイトストリームの IO サポート

### ● デバッグ機能の拡張

Eclipse

- Solaris (SPARC) 11.4 以降を使用したデバッグのサポート
- PLIDUMP : STATIC 変数の情報を生成し、HEAP データストレージを表示

目次へ

# Enterprise Developer/Server 固有機能の拡張

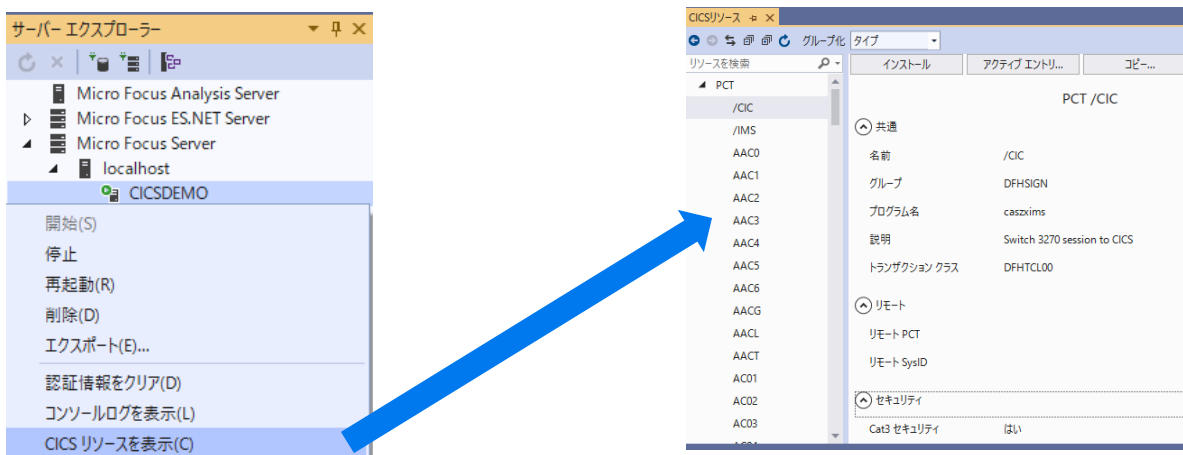
テクニカルプレビュー機能

Visual Studio

## ■ CICS サポートの拡張

### ● CICS PCT リソースの表示

Visual Studio 内から CICS PCT リソースを表示できるようになりました。  
サーバーエクスプローラーのエンタープライズサーバーから、コンテキストメニューの [CICSリソースの表示] を選択して表示することができます。





対応プラットフォームとデータベース

# 対応プラットフォーム (1/3)

## ■ Microsoft Windows (x86-64)

- Windows 8.1, 10 (\*1)
- Windows Server 2012 R2
- Windows Server 2016, 2019
- Windows Server 2012 (\*2)

\*1)

32 ビット Windows の場合は一部のコンポーネントがインストールされません。  
詳細については製品マニュアルの「既知の問題と制限事項」をご参照ください。  
なお、64 ビット Windows の場合は 32 / 64 ビットのアプリケーションが開発  
できます。

\*2)

COBOL Server, COBOL Server for SOA のみ対応しています。

目次へ



## 対応プラットフォーム (2/3)

### ■ RHEL (x86-64)

- Red Hat Enterprise Linux 6.x, 7.x, 8.x

8 以降では OCI コンテナの Podman をサポートしています。

- Oracle Linux 6.x, 7.x, 8.x

(Unbreakable Enterprise Kernel, Red Hat Compatible Kernel)

8.x は 6.0 Patch Update 2 以上でサポートされます。

### ■ RHEL (390)

- Red Hat Enterprise Linux 6.x, 7.x

### ■ CentOS (x86-64)

- CentOS 7.x, 8.x

目次へ

## 対応プラットフォーム (3/3)

### ■ SLES (x86-64), SLES (390)

- SUSE Enterprise Linux Server 12, 15

### ■ IBM AIX (POWER)

- AIX 7.1, 7.2

### ■ HP-UX (Itanium)

- HP-UX 11iv3 (11.31)

### ■ Solaris (SPARC)

- Solaris 11

### ■ Solaris (x86-64) : Enterprise Developer/Server は除く

- Solaris 11

目次へ

## 対応データベース

### ■ Oracle

- 18c, 19c

### ■ Microsoft SQL Server

- 2014, 2016, 2017, 2019

### ■ IBM DB2

- 10.5, 11.1, 11.5

### ■ PostgreSQL 10.x (x は 5 以降), 11.x

### ■ MySQL 5.7

## 対応 Java バージョン

### ■ Oracle Java 8 / Adopt OpenJDK 8 (32 or 64-bit)

### ■ Oracle Java 11 (64-bit) / Adopt OpenJDK 11 (64-bit)

Java11 で Eclipse は実行できません。

目次へ

## 対応 Java Application Servers

- Tomcat 9.0.x
- JBOSS EAP 7.1.x (JCA利用時: x は 4 以降)  
7.2.x
- Oracle WebLogic 12.2.1
- IBM WebSphere 9.0  
WebSphere Liberty 19.0.0.x  
(JVM COBOL利用時: x は 12 以降)

## 対応 Middleware

- IBM's MQ series
  - 8.0, 9.0, 9.1

記載した対応情報は代表例となります。  
個別対応の例もございますので、対応状況については別途お問合せください。

目次へ



[www.microfocus.co.jp](http://www.microfocus.co.jp)

© 2020 Micro Focus

本スライドに記載の会社名・製品名は、各社の商標または登録商標です。  
本スライドに記載の内容をマイクロフォーカスの許可無く転載することを禁じます。