Micro Focus Visual COBOL 5.0J
Micro Focus Enterprise Developer 5.0J

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

開発支援機能の強化

- COBOL 開発支援機能の強化
- Micro Focus Unit Testing Framework の強化
- アプリケーション実行環境の拡張

実行環境機能の強化

- 統合トレース機能の拡張
- ランタイムシステムの強化
- Enterprise Server Common Web Administration の導入
- スケールアウトアーキテクチャの導入

Enterprise Developer 固有機能の拡張

- JES サポートの拡張
- CICS サポートの拡張
- データベースファイルハンドラ機能の追加
- PL / I サポートの拡張

対応プラットフォームとデータベース

開発支援機能の強化

COBOL 開発支援機能の強化

Eclipse　Visual Studio

■ COBOL 言語の拡張

次の COBOL 構文が拡張されました。

- 条件付きコンパイル

追加指令	DEFINE 64BIT

```
>>DEFINE 64BIT AS PARAMETER
>>IF 64BIT IS DEFINED
   CALL "INIT64"
>>ELSE
   CALL "INIT32"
>>END-IF
```

ISO2002 に基づく条件付きコンパイル方法が利用可能になりました。

- JSON PARSE 文

JSON テキストを COBOL データ形式に変換できる構文が追加されました。

```
Identification division.
Program-id. jparse1.
Data division.
Working-storage section.
01 msg.
  03 snd usage comp-1.
  03 pid pic 9999 usage display.
  03 txt pic x(12).
Linkage section.
01 json-str pic x(53).
Procedure division using json-str.
display "Parsing....".
JSON PARSE json-str into msg with DETAIL
END-JSON.
if snd equal to 9 then
   display "PID is " pid
   display "Message text is '" txt "'"
end-if.
goback.
End program jparse1.
```

- SET 文

set [string] 構文のサポートが追加されました。

- 非同期および待機構文（.NET COBOL のみ）

async および await 構文を使用した非同期プログラミングのサポートが追加されました。

MICRO FOCUS

COBOL 開発支援機能の強化

Eclipse　Visual Studio

■ COBOL エディターの強化

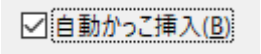
- END 文の自動補完

  IF, Evaluate 文など、各ステートメントの END 句を自動挿入するように強化されました。設定で無効にすることもできます。

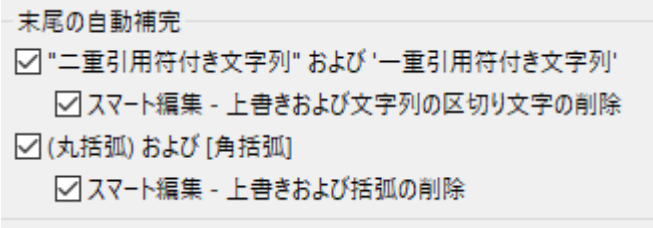
- 末尾の自動補完

  自動的に閉じ引用符を挿入するように強化されました。設定で無効にすることもできます。

【 Visual Studio 2019 】



【 Eclipse 】



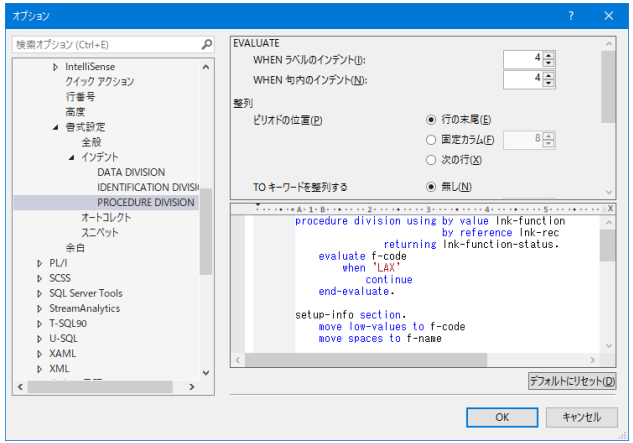
COBOL 開発支援機能の強化

Eclipse　Visual Studio

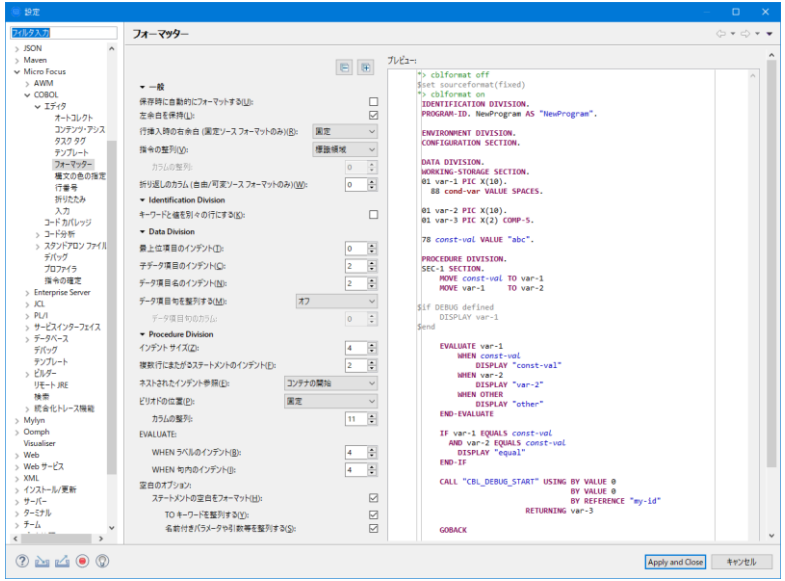
■ COBOL エディターフォーマッター機能の追加

COBOL プログラム全体や選択したコード、コードブロックに一貫した書式設定を適用できるように拡張されました。これによりコードの可読性が向上します。

【 Visual Studio 2019 】



【 Eclipse 】



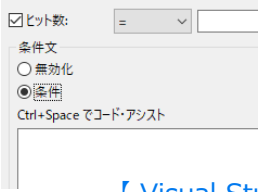
MICRO FOCUS

COBOL 開発支援機能の強化

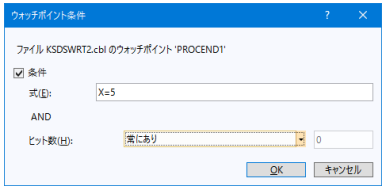
Eclipse　Visual Studio

■ デバッグ機能の強化

【 Eclipse 】



【 Visual Studio 2019 】



- 条件付きウォッチポイント

  条件を指定できるように拡張されました。大規模で複雑なプログラムをデバッグするときに役立ちます。

- コアダンプファイルの作成（ネイティブ COBOL のみ）

  CBL_CREATE_CORE を使用して、任意のタイミングでランタイムエラーを発生させることなく、プロセスのコアダンプファイル作成が可能になりました。コアダンプのデバッグにはソースファイルや idy ファイルが必要です。

```
01 flags pic x(4) comp-5 value 0.
01 process pic x(4) comp-5 value 18655.
01 filename pic x(100) value "myDump.%p" & x"00".
call "CBL_CREATE_CORE" using by value flags
                             by value process
                             by reference filename.
```


MICRO FOCUS

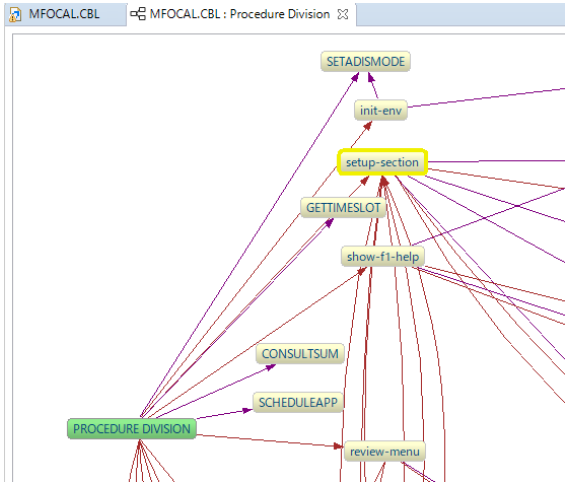
COBOL 開発支援機能の強化

Eclipse　Visual Studio

■ フロー分析を新たに追加

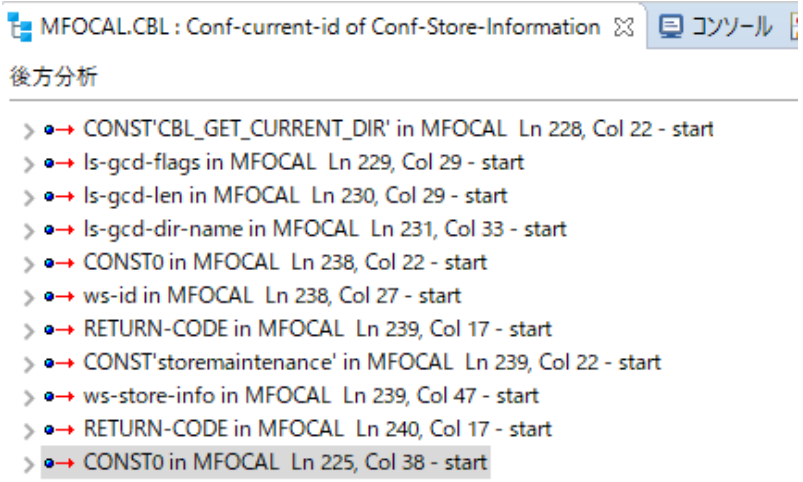
● プログラムフロー

選択したプログラムフロー図を作成し、プログラム全体像を俯瞰できます。



● データフロー

選択したデータ項目の影響範囲を一覧で確認できます。



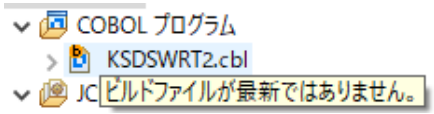
MICRO FOCUS

COBOL 開発支援機能の強化

Eclipse

■ UI 機能の強化

COBOL, PL/I エクスプローラでは b アイコンを使用してビルドされていないプログラムを示すように強化され、より視覚的に認知することが可能になりました。



■ DevHub ネットワークトラフィックの自動 SSH トンネリングの改善

SSH ポート転送を使用して DevHub ネットワークトラフィックをルーティングできるように DevHub 接続タイプが改善され、ファイアウォールを設定している UNIX ホスト上でリモート開発することがより簡単になりました。

MICRO FOCUS

COBOL 開発支援機能の強化

Eclipse

■ デバッグ機能の強化

- ライブレコードのパフォーマンス向上（Linux ターゲットマシンのみ）

  パフォーマンスを向上させるため、メモリ量を指定する環境変数 COB_LIVE_RECORD_SIZE が追加されました。

```
export COB_LIVE_RECORD_SIZE=mem-size
```

■ コンパイラ指令セットの追加

コンパイラ指令が XML を使用した .mfdirset ファイルに格納されるようになりました。既存のプロジェクトはインポートまたはロードされると .mfdirset ファイルを使用するようアップグレードされます。ソース管理ツールをご利用の場合は、必ず .mfdirset ファイルも格納してください。

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<x:dirset xmlns:x="http://microfocus.com/directivesset" schemaVersion="1.0">
<x:directive handler="project.cobolSettings" id="max-error" name="MAX-ERROR&quot;100&quot;"/>
<x:directive handler="project.cobolSettings" id="mainframecopyhandling" name="NOPANVALET NOLIBRARIAN"/>
<x:directive handler="project.cobolSettings" id="exitprogram" name="EXITPROGRAM&quot;ANSI&quot;"/>
<x:directive handler="project.cobolSettings" id="dialect" name="DIALECT&quot;ENTCOBOL&quot;"/>
<x:directive handler="project.cobolSettings" id="warning" name="WARNING&quot;1&quot;"/>
<x:directive handler="project.cobolSettings" id="charset" name="CHARSET&quot;ASCII&quot;"/>
<x:directive handler="project.cobolSettings" id="anim" name="anim"/>
<x:directive handler="project.cobolSettings" id="sourceformat" name="SOURCEFORMAT&quot;fixed&quot;"/>
<x:directive handler="project.cobolSettings" id="generate" name="generate"/>
<x:ecm enabled="false" handler="CICS" id="project.cicsSettings" postfix=")" prefix="CICSECM("/>
</x:dirset>
```

- .asm.mfdirset
- .bms.mfdirset
- .cbl.mfdirset
- .dbd.mfdirset
- .mfs.mfdirset
- .psb.mfdirset
- KSDSWRT2.cbl.mfdirset

MICRO FOCUS

COBOL 開発支援機能の強化

Eclipse

■ 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 4.6（Neon）

  サポートは終了しました。

MICRO FOCUS

COBOL 開発支援機能の強化

Visual Studio

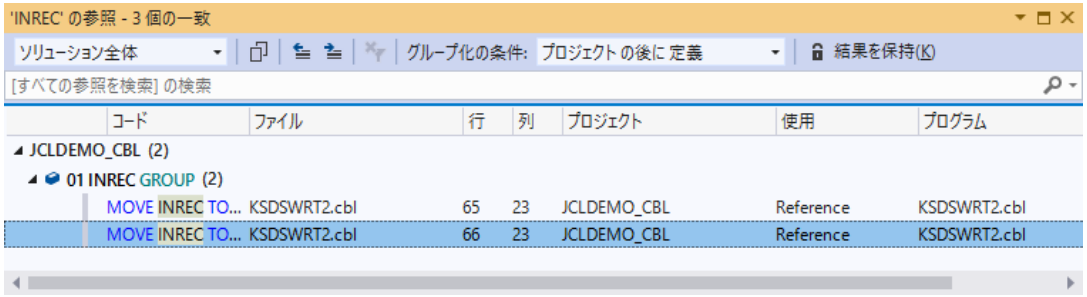
■ Visual Studio 最新版の 2019 をサポート

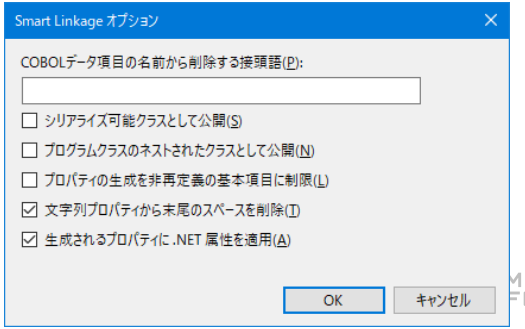
Visual Studio 2017 は引き続きサポートします。

Visual Studio 2013, 2015 のサポートは終了しました。

■ 開発支援機能の強化

- 参照の検索ウィンドウに、グループ化された結果出力とフィルター機能が追加されました。
- Smart Linkage オプションに指定項目が追加されました。（マネージのみ）





Micro Focus Unit Testing Framework の強化

Eclipse　Visual Studio

■ データドリブンテストの拡張

カンマ区切り値（CSV）フォーマットのデータソースを適用するデータドリブンテストのサポートが追加されました。

```
ENTRY "MFUM_MYTESTCASE".
   move "csv:MyCSVSource.csv" to MFU-MD-TESTDATA.
```

また、MFU-MD-DD-CSV-FILT-COND を使用することで、フィルタリング条件を使用することができ、どの行をテストするか制御できます。

```
move "Surname = Jones" to MFU-MD-DD-CSV-FILT-COND
```

テストケースの中でデータソースのデータを使用するには、作業場所節に列ヘッダー項目を定義します。

例）EarnedPoints 列の定義

```
01 mfu-dd-EarnedPoints is MFU-DD-VALUE external.
```


MICRO FOCUS

アプリケーション実行環境の拡張

Visual Studio

■ .NET Coreのサポート

EAP：早期導入機能

.NET Core は複数のオペレーティングシステムをサポートするオープンソース開発プラットフォームです。 COBOL アプリケーションを開発し、そのアプリケーションを .NET Core がサポートする任意のオペレーティングシステムにディプロイすることができます。

.NET Core アプリケーション用のプロジェクトテンプレートとして、クラスライブラリ（.NET Core）とコンソールアプリケーション（.NETCore）を用意しています。

MICRO FOCUS

実行環境機能の強化

統合トレース機能の拡張（Windows プラットフォームのみ）

■ Consolidated Trace Facility（CTF）

動的に構成することが可能になりました。構成ファイルとログを指定したインスタンスを起動後、新しいコマンドラインユーティリティ（cblctd）を使用すると、実行中プロセスのアプリケーションのトレースレベルを変更したり、トレースにコンポーネントを追加または削除することができます。

```
cblctd -Cf mf.rts:fsys -Ct mf.mffh.xfh:memory,catalog -p 6765
```

また、新しいコンポーネント mf.mfdbfh を使用すると、Micro Focus ネイティブデータベースファイルハンドラのアクティビティをトレースできます。

```
...
mftrace.comp.mf.mfdbfh#datastore=true
mftrace.comp.mf.mfdbfh#fh=true
...
```

ランタイムシステムの強化

（Linux Intel 64-bit, Solaris Intel 64-bit, Linux/390 プラットフォームのみ）

実行可能ファイルまたは共有オブジェクトにビルドされたネイティブ COBOL アプリケーションでランタイムシステムエラーが発生したときに、スタックダンプが生成可能になりました。（マネージ COBOL または INT / GNT フォーマットにビルドされたプログラムを除く）実行時チューナーへ stackdump_on_error および stackdump_filename の設定を行います。このダンプを解析することでプログラム内のエラーの正確な行を見つけることができます。

```
>>---set stackdump_on_error=--.-1-.--------->><
                              +-0-+
```

```
set stackdump_filename="/home/temp/stackdump/stack.%p@%t_%d"
```


MICRO FOCUS

スケールアウトアーキテクチャの導入

EAP：早期導入機能

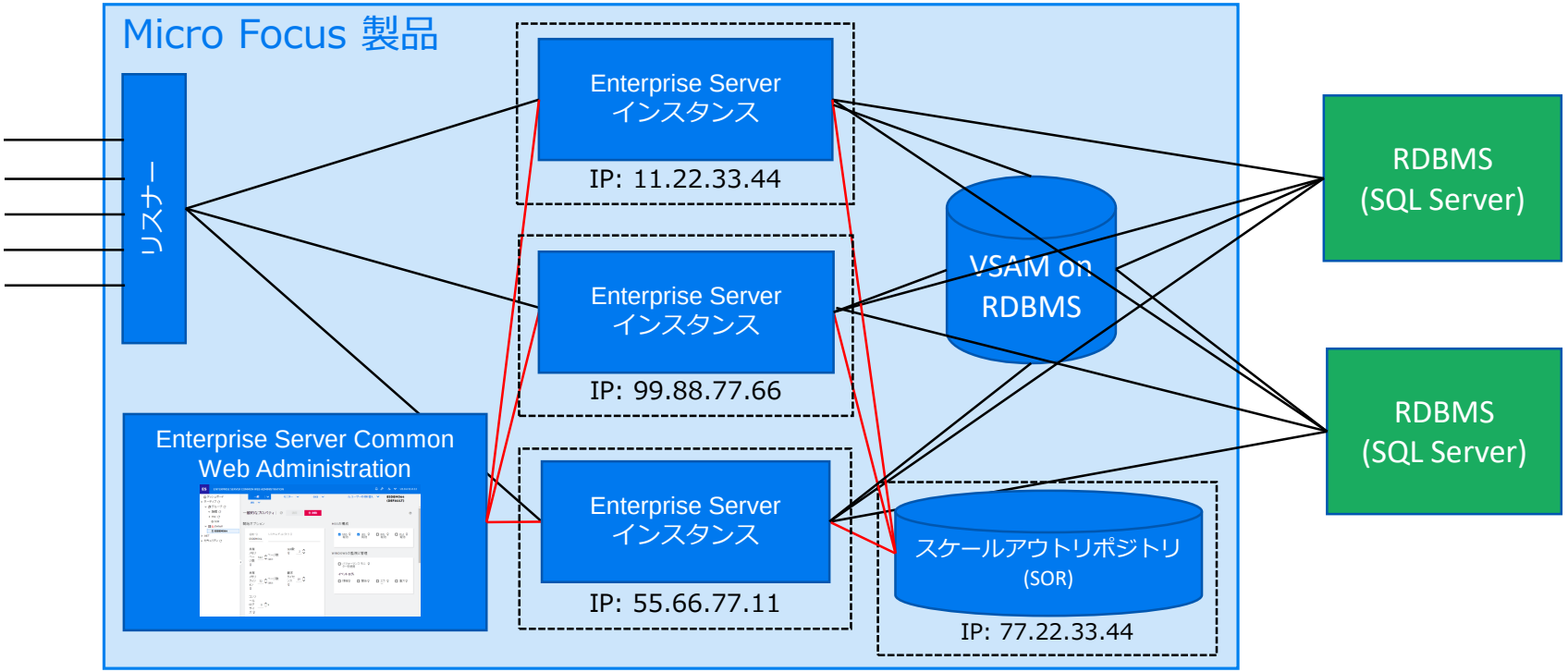
パフォーマンス・アベイラビリティ クラスター（PAC）を使用することで、インスタンスグループを単一のシステムイメージとして管理できるようになり、これを実現するためにスケールアウト リポジトリ（SOR）も提供されます。この機能拡張により以下のようなことが可能になります。

- ハードウェアコストの削減 … PAC を使用したスケールアウト
- 可用性の向上 … 異なるマシン上のインスタンスを運用可能
- 動的スケーリング … 随時 PAC へインスタンスを追加または削除が可能
- パフォーマンスの向上 … 単一マシンのリソースによる処理能力の制限なし（スケールアップ）
- 保守性の向上 … ESCWA（ Enterprise Server インスタンス集中管理機能）から単一のイメージとして管理可能

スケールアウトアーキテクチャの導入

EAP：早期導入機能

■ スケールアウトアーキテクチャ概念図の例



Enterprise Server Common Web Administration の導入（ESCWA）

EAP：早期導入機能

Enterprise Server インスタンスの集中管理やメンテナンス性向上のため、新たな Enterprise Server 用管理画面が導入されました。

スケールアウト機能管理のために論理グループを指定し、パフォーマンス・アベイラビリティ クラスター（PAC）とそれに関連するスケールアウト リポジトリ（SOR）を設定した運用が可能となります。

Enterprise Developer 固有機能の拡張

JES サポートの拡張

ハウスキーピングルールなどにより、様々なフォルダにアーカイブされたスプールファイルをマージして一括参照する機能が追加されました。

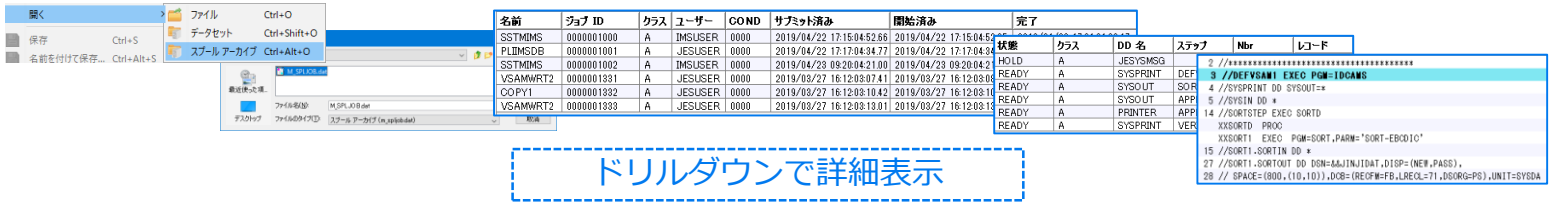
スプールファイルはコピーされ、結合されたアーカイブにマージされます。

マージプロセスを実行するには JCL をサブミットするか、コマンドラインユーティリティを実行します。

```
//MERGESPL JOB 'MERGE SPOOL ARCHIVES',CLASS=A,MSGCLASS=A
//*********************************************************************
//* Merge a spool archive into a merged archive
//*********************************************************************
//ARCHIVE  EXEC PGM=MERGESPL
//SYSPRINT DD SYSOUT=*
//SYSOUT   DD SYSOUT=*
//SYSERR   DD SYSOUT=*
//* SYSIN or CONFIGI dataset contains the program parameters
//SYSIN  DD *
* ---------------------------------------------------------
*   MERGED SPOOL HOUSEKEEPING CONFIGURATION FILE
* ---------------------------------------------------------
*   ARCHIVE LOCATION - the archive to read
* ---------------------------------------------------------
* ARCHIVE   LOC C:\ES_ERRORS\MERGED_ARCHIVES\SET_A\2018052611114975
* ---------------------------------------------------------
* TOP LEVEL DIRECTORY TO SCAN
  SCAN      LOC C:\spl-arc\
* ---------------------------------------------------------
*   MERGED ARCHIVE LOCATION - The location of the merged archive
* ---------------------------------------------------------
  MERGE LOC C:\spl-merge\Nov-18\
/*
//
```

データファイルツールを使用してマージ後の内容を参照します。

```
MergeArchives SCAN|ARCH arch-spool-file-directory/directories directory-for-merged-spool-files
```



ドリルダウンで詳細表示

CICS サポートの拡張

■ CRCN トランザクション

EAP：早期導入機能

定義されている XA 接続の状態を監視する CICS トランザクションが組み込まれました。接続が失われたことを検出した場合は定期的に再接続を試みます。 監視の頻度と開始は ES_XA_RECONNECT 環境変数（秒単位指定）によって制御されます。

```
CASX00041I The XA connection monitoring transaction CRCN started using a 00020 seconds interval.
CASX00034I CRCN started monitoring: XASQL Enabled 09:38:52
```

監視を除外したい XA 接続は ES_XA_EXCLUDE_MONITORING 環境変数へ指定します。名称は 8 文字までです。

```
ES_XA_EXCLUDE_MONITORING=RM1;RM2;RM3
ES_XA_EXCLUDE_MONITORING=MYRM01;MYRM003
```

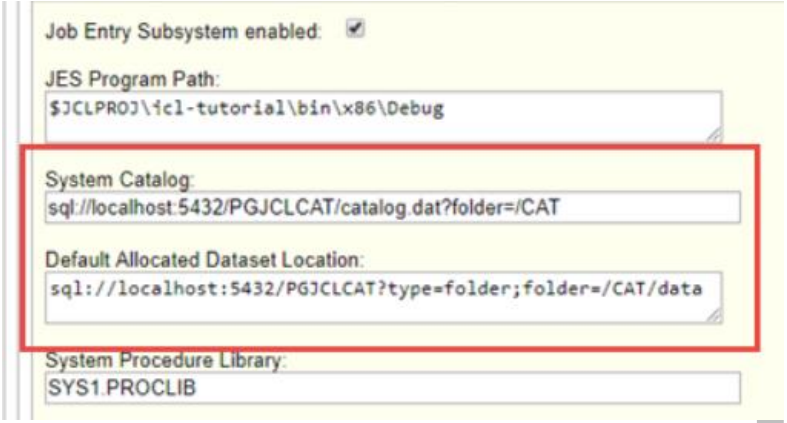

MICRO FOCUS

データベースファイルハンドラ機能の追加

EAP：早期導入機能

CICS およびバッチアプリケーションがデータベース内に格納されているデータファイルおよび特定のインフラストラクチャファイルと対話できるように、 Micro Focus データベースファイルハンドラ（MFDBFH）機能が追加されました。現時点では Microsoft SQL Server または PostgreSQL をサポートしており、ODBC を介してアクセスします。

ファイル処理のソースコードを変更する必要はありませんが、アプリケーションを実行する環境にいくつかの変更を加える必要があります。



PL / I サポートの拡張

Eclipse　Visual Studio

■ コンパイラの拡張

- GET DATA のフルサポート
- ビルトイン関数 ALLOCATE, INDEXR, MEMCONVERT, MEMINDEX, ONCONDCOND, PLIFREE, REM, REPLACEBY2 の追加
- INONLY, NONASSIGNABLE 属性のサポート
- *PROCESS OPT(TIME) のサポート
- DUMMY ALIGNED および DUMMY UNALIGNED 引数のサポート
- XML case asis および XML case Upper のサポート
- %INSCAN および %XINSCAN マクロプリプロセッサーのサポート
- データ構造体のメンバを初期化する VALUE の許可
- OPTIONS（REORDER）の許容（実行時は無視）

MICRO FOCUS

PL / I サポートの拡張

Eclipse　Visual Studio

■ EXEC プリプロセッサの強化

- PL/I パッケージに埋め込まれた SQL 文をサポート
- エラー最大数を設定する –maxmsg オプション指定をサポート
- シグナルプログラムに最大 4000 のインクルードファイルが指定可能

■ ランタイムシステムの強化

- PLIDUMP 呼出し後、オプションで Windows のミニダンプ もしくは Linux のコアダンプを生成することが可能
- TITLE オプションを使用したデータセットの動的割り当てをサポート
- サブルーチンが COBOL RTS エラーを発生させた場合、エラー状態を発生させることが可能

MICRO FOCUS

PL / I サポートの拡張

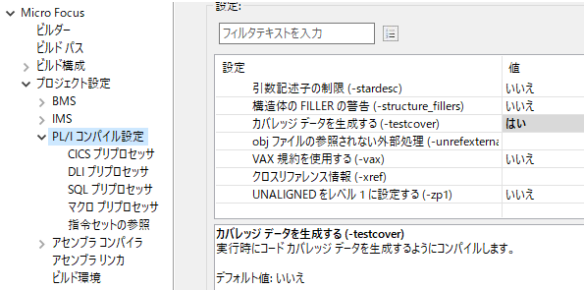
Eclipse

■ Eclipse デバッガの改善

- データ項目の宣言行にウォッチポイントを設定できるようになりました。
- リモートプロジェクト操作時にコンソールからの対話型入出力が可能になりました。
- 実行ポイント周辺の変数が [変数] ビューに表示されるようになりました。

■ コードカバレッジ

- コードカバレッジデータを生成することが可能になりました。



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 7, 8.1, 10 (*1)
  - Windows Server 2008 R2 SP1
  - Windows Server 2012 R2
  - Windows Server 2016, 2019

*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。

注意）Enterprise Developer は上記以前の Windows にインストールすることが可能ですが、テストはされていません。

MICRO FOCUS

対応プラットフォーム (2/3)

- **RHEL (x86-64)**
  - Red Hat Enterprise Linux 6.x, 7.x
  - Red Hat Enterprise Linux 8.x (Patch Update3 からサポート)
  - Oracle Linux 6.x, 7.x (Unbreakable Enterprise Kernel, Red Hat Compatible Kernel)
- **RHEL (390)**
  - Red Hat Enterprise Linux 6.x, 7.x
- **RHEL (POWER)**
  - Red Hat Enterprise Linux 7.1, 7.2
- **SLES (x86-64), SLES (390), SLES (POWER)**
  - SUSE Enterprise Linux Server 12, 15

対応プラットフォーム (3/3)

- **IBM AIX (POWER)**
  - AIX 7.1, 7.2
- **HP-UX (Itanium)**
  - HP-UX 11iv3 (11.31)
- **Solaris (SPARC)**
  - Solaris 11
- **Solaris (x86-64)**
  - Solaris 11

対応データベース

- **Oracle**
  - 12c R2, 18c
- **Microsoft SQL Server**
  - 2012, 2014, 2016, 2017
- **IBM DB2**
  - 10.5, 11.1
- **PostgreSQL 10.5**
- **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)**
  - Eclipse は Java11 では実行できません。

対応 Java Application Servers

- Tomcat 9.0.x
- JBOSS EAP 7.1.4
- Oracle WebLogic 12.2.1
- IBM WebSphere 9.0

対応 Middleware

- IBM's MQ series
  - 8.0

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。

www.microfocus.co.jp

© 2019 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。