

Enterprise Developer チュートリアル

メインフレーム COBOL 開発 : Linux/UNIX 環境での実行

1. 目的

Windows 開発環境でテストやデバッグが終了したアプリケーションは本番環境へ配備して運用されることとなりますが、この本番環境ではランタイムのみを持つ実行環境製品の Enterprise Server を使用することとなります。

本チュートリアルでは、メインフレームアプリケーションを Linux/UNIX 本番環境に配備する下記方法を体験することを目的とします。このチュートリアルでは Amazon Linux release 2 を使用していますが、すべての操作はその他の UNIX もほぼ共通です。また、データベースへ依存したプリコンパイルも行いますので、事前に正常に接続できることを確認しておいてください。

2. チュートリアル手順の概要

1. Linux/UNIX ターゲットのマイグレーション手順
2. Enterprise Developer のインストール
3. COBOL 環境変数の設定
4. XA スイッチモジュールの作成
5. Enterprise Server Common Web Administration (ESCWA) の起動
6. Micro Focus™ Directory Server の起動
7. 文字エンコーディングの設定
8. JES / CICS インスタンスの作成
9. COBOL プログラムのコンパイル
10. JCL のサブミット
11. CICS トランザクションの実行
12. Enterprise Server インスタンスの停止
13. デバッグについて

2.1 Linux/UNIX ターゲットのマイグレーション手順

マイグレーション手順について説明します。

1) ローカルマシンでの動作確認

Linux/UNIX ターゲットのマイグレーションでは、対象アプリケーションをまずは Windows 開発環境の IDE を利用してテストやデバッグを行い、実行環境プラットフォームにて再コンパイル後に実行ファイルを配置することを強く推奨しています。このため Linux/UNIX の Enterprise Developer 開発環境製品には Windows で稼働する Enterprise Developer for Eclipse ライセンスがセットで標準提供されています。Enterprise Developer とその実行環境製品である Enterprise Server はマイグレーションに際して COBOL, JES, CICS の基本部分を提供しますが、これだけでマイグレーションが実現できることはありません。たとえば、メインフレームで使用していたアセンブラルーチン、特殊ユーティリティ、簡易言語を書き換えた場合は、既存 COBOL ソースとの結合テストを Windows 開発環境の IDE にて事前にデバッグを行い動作確認します。ここまでできれば確認済み COBOL ソースをターゲットの Linux/UNIX 環境にコピーして一括コンパイルすることは容易です。コンパイルはターゲットマシン上のコマンドライン、または Windows の Eclipse からリモートで実行します。

2) Enterprise Server インスタンスの作成

開発環境でテスト用に作成した Enterprise Server インスタンスと同じものを Linux/UNIX 上にも作成します。開発環境の管理画面から対象インスタンスをエクスポートして、ターゲットマシンへインポートすることも可能です。ただし、ディレクトリの指定やホスト名または IP アドレスはターゲットマシンに合わせます。

3) 実行ファイルの配備

コンパイルした実行モジュールを、Enterprise Server インスタンスに指定したプログラムディレクトリへ配備します。

2.2 Enterprise Developer のインストール

1) SJIS ロケールの設定

Windows 開発環境でテストしたリソースを Linux/UNIX 上で使用するため、次のような手順で環境を SJIS ロケールに設定しておきます。

① カレント環境のロケール確認

次のコマンドを実行します。

コマンド) locale

```
# locale
LANG=ja_JP.UTF-8
LC_CTYPE=ja_JP.UTF-8
LC_NUMERIC=ja_JP.UTF-8
LC_TIME=ja_JP.UTF-8
LC_COLLATE=ja_JP.UTF-8
LC_MONETARY=ja_JP.UTF-8
LC_MESSAGES=ja_JP.UTF-8
LC_PAPER=ja_JP.UTF-8
LC_NAME=ja_JP.UTF-8
LC_ADDRESS=ja_JP.UTF-8
LC_TELEPHONE=ja_JP.UTF-8
LC_MEASUREMENT=ja_JP.UTF-8
LC_IDENTIFICATION=ja_JP.UTF-8
```

② 使用可能な日本語ロケール確認

次のコマンドを実行します。

コマンド) `locale -a | grep JP`

```
# locale -a | grep JP
ja_JP
ja_JP.eucjp
ja_JP.sjis
ja_JP.ujis
ja_JP.utf8
```

③ 使用可能な日本語ロケール確認

上記では SJIS ロケール `ja_JP.sjis` が使用できるように設定されていることがわかりますが、Linux では SJIS ロケールが設定されていない場合があります。その場合は `root` ユーザーで次のコマンドを実行し、ロケールを追加します。

コマンド例) `localedef -f SHIFT_JIS -i ja_JP ja_JP.sjis`

このコマンド実行時に以下のような警告メッセージが表示されても無視して構いません。

[警告メッセージ内容]

- キャラクタマップ `SHIFT_JIS' は ASCII 互換ではありません, ロケールは ISO C に従っていません。
- character map `SHIFT_JIS' is not ASCII compatible, locale not ISO C compliant

Windows 標準日本語コードを指定することでこの警告メッセージを回避することも可能ですが、MQ を使用する際にはこの CCSID をサポートしていない場合もありますので、確認後に適用してください。

コマンド例) `localedef -f WINDOWS-31J -i ja_JP ja_JP.sjis`

④ SJIS ロケールの設定

環境変数 `LANG` に SJIS ロケールを設定します。

コマンド) `export LANG=ja_JP.sjis`

2) Enterprise Developer のインストール

製品マニュアル [ここからはじめよう] > [製品情報] > [Enterprise Developer のインストール] > [Micro Focus™ Enterprise Developer for Linux and Unix の Readme] の [ダウンロードとインストールの手順 - Enterprise Developer Unix コンポーネント] 項目を参照してください。

3) ライセンス認証手順

弊社から発行された認証コードを用意します。製品をインストールした環境で稼働しているライセンス管理システムから、通常は直接インターネット経由で認証手続きを行います(オンライン認証)。もしそれが不可能な場合は、インターネットが使用可能な他の環境から弊社の認証サイトを使用し、認証コードやマシン ID 等の必要情報を入力することでライセンス文字列を入手できますので、それをライセンス管理システムにインストールすることも可能です(マニュアル認証)。

詳細は製品マニュアル [ここからはじめよう] > [製品情報] > [Enterprise Developer のインストール] > [Micro Focus™ Enterprise Developer for Linux and Unix の Readme] > [ライセンス] の [COBOL/エンタープライズ製品ライセンス 適用管理ガイド] を選択してご確認ください。

4) ファイアウォールの解除

このチュートリアルでは下記ポートへのアクセスを許可しておいてください。

Enterprise Server Common Web Administration へのアクセス:10086 / tcp

Enterprise Server Administration へのアクセス:86 / tcp

TN3270 エミュレータへのアクセス:9004 / tcp

なお、適宜、使用ポートを許可する必要があります。もしクローズした環境であれば、無効にすると便利です。

2.3 COBOL 環境変数の設定

インストールした製品を COBOL 実行環境に設定するため環境変数を設定します。製品ディレクトリの bin ディレクトリに cobsetenv が用意されていますので、これを実行します。

コマンド例). /opt/mf/ED10GA/bin/cobsetenv

実行すると環境変数 COBDIR にインストールした製品パス設定されます。

```
COBDIR set to /opt/mf/ED10GA
```

2.4 XA スイッチモジュールの作成

Windows 環境と同様に、ターゲットマシンで実行するプログラム内では XA スイッチモジュール経由でデータベースと接続することになります。使用するデータベース製品に合わせた XA スイッチモジュールを root ユーザーで作成します。

1) XA リソースのコピー

ビルドを行うため、インストールディレクトリ配下の \$COBDIR/src/enterpriseserver/xa をディレクトリごと書き込み権限があるパスへコピーします。

コピー元パス例:\$COBDIR/src/enterpriseserver/xa

コピー先パス例:/home/tarot/xa

2) XA スイッチモジュールのビルド準備

生成する環境の設定を行います。

① COBOL 作業モードの設定

接続するデータベースのビット数に合わせた数値を指定します。XA スイッチモジュールはこの設定値に沿って生成されます。cobmode コマンドまたは環境変数 COBMODE を使用して設定します。

64 ビット設定例)export COBMODE=64

② 環境変数 LD_LIBRARY_PATH の設定: Linux などの場合

DB 関連の必要なパスと、生成する XA スイッチモジュールを配置するパスを指定します。

Oracle 例)

```
export LD_LIBRARY_PATH=$ORACLE_HOME/lib:$ORACLE_HOME/precomp/lib:/home/tarot/xa:$LD_LIBRARY_PATH
```



注意

LD_LIBRARY_PATH 64 のような 64 bit 向けに環境変数を用意している OS もあり、データベースによってはこれを使用する場合があります。

③ 環境変数 LIBPATH の設定: AIX の場合

DB 関連の必要なパスと、生成する XA スイッチモジュールを配置するパスを指定します。

Db2 例)

```
export LIBPATH=$COBDIR/lib:/usr/lib:/lib:$HOME/sql/lib:/home/tarot/xa
```

④ 環境変数 DB2INSTANCE の設定

Db2 使用時にインスタンス名を指定します。

```
export DB2INSTANCE=db2inst1
```

3) XA スイッチモジュールのビルド実行

① 書き込み権限のあるコピー先パスへ移動します。

コマンド例) `cd /home/tarot/xa`

② Oracle を使用する場合は下記コマンドを実行し、XA スイッチモジュールを生成します。

コマンド). `build ora`

ビット数ごとに静的と動的登録用の 2 ファイルが生成されます。

ESORAXA.so	32-bit	static
ESORAXA64.so	64-bit	static
ESORAXA_D.so	32-bit	dynamic
ESORAXA64_D.so	64-bit	dynamic

③ Db2 を使用する場合は下記コマンドを実行し、XA スイッチモジュールを生成します。

コマンド). `build db2`

ビット数ごとに静的と動的登録用の 2 ファイルが生成されます。

ESDB2XA_S.so	32-bit	static
ESDB2XA64_S.so	64-bit	static
ESDB2XA.so	32-bit	dynamic
ESDB2XA64.so	64-bit	dynamic

XA スイッチモジュールのビルドに関する詳細は製品マニュアルを参照してください。

2.5 Enterprise Server Common Web Administration (ESCWA) の起動

後述の mfdss を一括管理できる ESCWA を、必要であれば起動します。例えば Windows マシンで起動した ESCWA から Linux マシンに存在する mfdss に接続して管理する場合は、Linux マシンの ESCWA を起動する必要はありません。

ESCWA は、デフォルトではループバックアドレスを使用した接続となり、Web ブラウザから「<http://<Linux IP アドレス>:10086>」を使用してアクセスできます。ループバックモードをオフにすることで、リモートマシンの Web ブラウザから ESCWA にアクセスすることができます。開始コマンドは root ユーザーで実行します。ループバックモードオフの開始コマンド例)

```
cd $COBDIR/bin
```

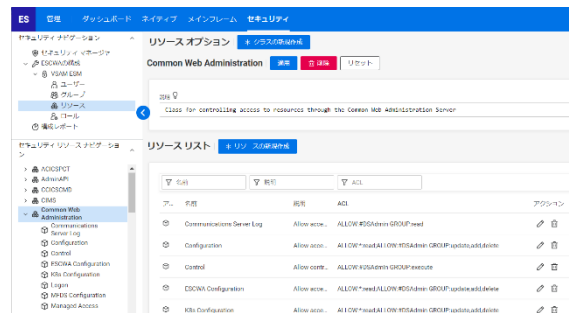
```
nohup escwa --BasicConfig.MfRequestedEndpoint="tcp*:10086" --write=true < /dev/null > escwa.out 2>&1 &
```

上記 & を付加すると、バックグラウンドで ESCWA のプロセスが起動されます。

起動した ESCWA では製品が提供する VSAM 外部セキュリティマネージャー(ESM)モジュールによるセキュリティがデフォルトで有効になっており、すべての処理において実行ユーザーの認証が行われます。

コマンドによりこの設定を無効にすることもできます。

ユーザー権限によっては実行を拒否されることがあり、この場合は ESCWA にログオン後、ESCWA に設定されているセキュリティ設定を確認、または関連するリソースをメンテナンスしてください。



詳細は製品マニュアルをご参照ください。

- ① デフォルトユーザーと初期パスワードを取得します。

Enterprise Developer コマンドプロンプトから次のコマンドを実行して、デフォルトユーザーと初期パスワードを取得します。

```
mfsecretsadmin read microfocus/temp/admin
```

```
$mfsecretsadmin read microfocus/temp/admin  
{ "mfUser": "SYSAD", "mfPassword": "oyuJIR1Z" }
```

上記例の場合、SYSAD がデフォルトユーザー、oyuJIR1Z が初期パスワードです。この情報をメモ帳などに記録してください。

- ② セキュリティを無効にする場合

Enterprise Developer コマンドプロンプトから次のコマンドを実行します。

```
DisableESDefaultSecurity.sh
```

```
$DisableESDefaultSecurity.sh  
Micro Focus DisableESDefaultSecurity utility: Version 1.0.0
```

この場合は、以降のユーザー認証にかかわる手順は省いてください。

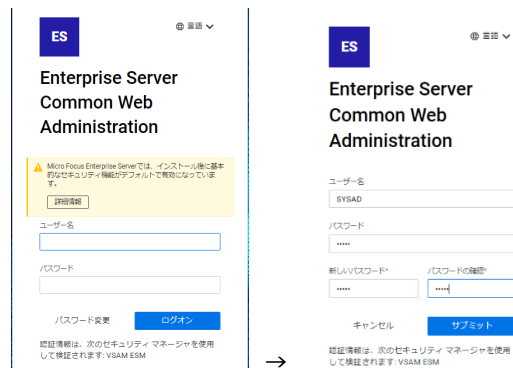
Windows マシンの Web ブラウザから Linux マシンで開始した ESCWA を参照すると、ユーザー認証を求められます。

セキュリティ確保のために初期パスワードを変更してログオンします。

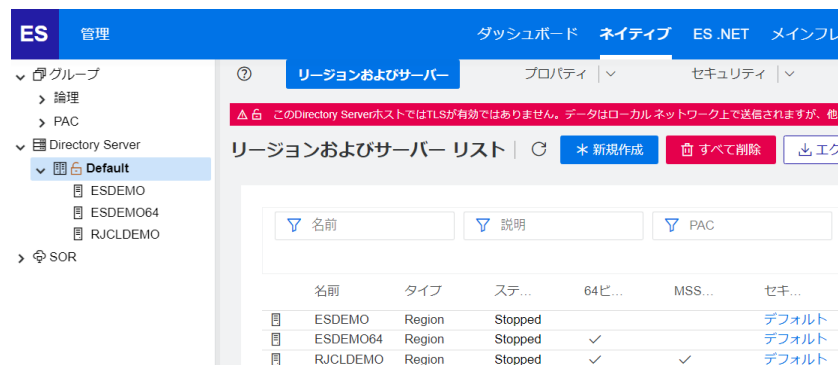
【パスワード変更】をクリックし、前項で確認したデフォルトユーザーと初期パスワード、新しいパスワードを入力して【サブミット】をクリックします。

変更後のパスワードはご自身の責任で管理してください。

パスワード変更後、画面がタイムアウトした場合は新しいパスワードを使用してログオンしてください。



例) `http://alinux2:10086/#/native/ds`



名前	タイプ	ステ...	64ビ...	MSS...	セキ...
ESDEMO	Region	Stopped			デフォルト
ESDEMO64	Region	Stopped	✓		デフォルト
RJCLDEMO	Region	Stopped	✓	✓	デフォルト

ESCWA を停止する場合は下記のコマンドを root ユーザーで実行します。

停止コマンド例)

```
cd $COBDIR/bin
```

```
escwa -p
```

2.6 Micro Focus™ Directory Server (mfds) の起動

root ユーザーで `mfd`s (Micro Focus™ Directory Server) コマンドを実行します。使用する環境によって、明示的に 32 bit 環境用に `mfd`s32 コマンド、64 bit 環境用に `mfd`s64 コマンドを実行することもできます。

コマンド例)mfd

上記 & を付加すると、前項の環境変数を基にバックグラウンドで `mfd`s のプロセスがインストール時の -ESadminID 指定による管理者ユーザー ID で起動されます。

2.7 文字エンコーディングの設定

Enterprise Server インスタンスを運用、管理する ESCWA では、スプールやデータ内容などに含まれる日本語を正しく表示させるために、事前に文字セットを所定のフォルダへ展開します。Windows マシンの ESCWA を使用する場合には、この設定を Linux マシンにする必要はありません。前項で Linux マシンの ESCWA を起動し、これを使用する場合に設定が必要となります。製品マニュアルの「リファレンス > コードセットの変換 > CCSID 変換テーブルのインストール > CCSID 変換テーブルをインストールするには」を参照しながら進めてください。

- 1) CCSID 変換テーブルをインストールします。

製品マニュアルにリンクされている下記の IBM CCSID 変換テーブルを、Web ブラウザから任意のフォルダへダウンロードします。アドレスは変更される可能性がありますので、製品マニュアルにてご確認ください。
<http://www.microfocus.com/docs/links.asp?vc=cdctables>

- 2) 製品インストールフォルダ配下の etc フォルダに ccsid ディレクトリがない場合はこれを作成します。

例)/opt/mf/ED10GA/etc/ccsid

- 3) ダウンロードファイルに含まれている Package2.zip を展開します。

- 4) 展開した Package2 フォルダに含まれる IBM-932.zip を展開します。

- 5) 展開した IBM-932 フォルダを切り取り、ccsid ディレクトリ配下へ貼り付け、14 ファイルが含まれていることを確認します。

```
03A434B0.MJ-R-A2 03A434B0.MJ-R-D 03A434B0.TPMAP110 03A434B0.TPMAP12A 03A434B0.UPMAP12A 34B003A4.RPMAP120 34B003A4.UM-E-A21
03A434B0.MJ-R-A3 03A434B0.PACKAGE 03A434B0.TPMAP11A 03A434B0.UPMAP120 03A434B0.UPMAP13A 34B003A4.RPMAP12A 34B003A4.UM-E-D12
```

詳細については、製品マニュアルの「ディプロイ > 構成および管理 > Enterprise Server の構成および管理 > Enterprise Server Common Web Administration > [Native] > [Directory Servers] > リージョンとサーバー > リージョン > エンタープライズ サーバー リージョンの文字エンコーディングのサポート」をご参照ください。

- 6) インスタンスの [構成情報] に次の環境変数を追加します。

MFACCCGI_CHARSET=Shift_JIS

追加設定

構成情報 ⓘ

[ES-Environment]
MFACCCGI_CHARSET=Shift_JIS

2.8 JES / CICS インスタンスの作成

- 1) インスタンスのディレクトリ構成決定

Windows 環境でのディレクトリ構成に倣い、各ファイルの配置を決定します。

- 2) リソース定義ファイルの移行

CICS 環境(IMS 機能でも使用)に必要なリソース定義は dfhdrdat(dfhdrdat.idx 付き形式も可能)という固有の索引ファイルで管理されています。Windows 環境でテスト済みのファイルをそのままバイナリ転送し、上記で構成した適切なディレクトリに配置します。ファイル名は小文字です。

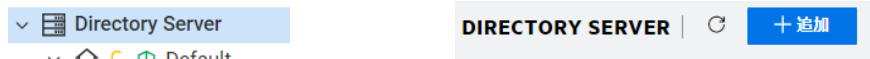
3) インスタンスの作成

ESCWA からインスタンスを作成します。パスの形式は Windows 環境とは異なり、/(スラッシュ)を使用します。¥(またはバックスラッシュ)は使用できません。複数パスを指定する場合は：(半角コロン)で区切ります。

- ① Web ブラウザで ESCWA を表示し、画面上部の【ネイティブ】をクリックします。



- ② Linux マシンの mfdcs である【Default】が表示されます。存在しない場合は右側ペインの【追加】をクリックして接続します。
存在する場合の表示)



追加する場合は、【名前】に任意の接続名称を、【ホスト】にはターゲットマシンのホスト名または IP アドレスを、【ポート】には MFDS のデフォルトポートである 86 を指定して【保存】をクリックします。

Directory Server

名前*

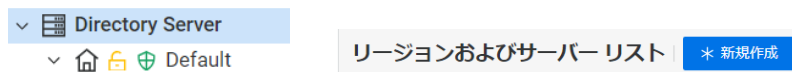
ホスト*

ポート*

説明

*入力必須の項目です [保存](#) [戻る](#)

- ③ 左側ペインの【Default】をクリックし、右側ペインの画面中央にある【新規作成】ボタンをクリックして Enterprise Server インスタンスをターゲットマシンに作成します。



- ④ 表示された画面の【名前】に MSSDEMO を入力します。64 ビットの実行可能ファイルを生成したため【64 ビット作業モード】にチェックし、TN3270 リスナーのポートには 9004 を入力して【保存】ボタンをクリックします。

リージョンの新規作成

名前 *
MSSDEMO

説明

▲ Directory Serverとリージョンの作業モードが一致しません。プラットフォームによっては、起動時に不具合が発生する可能性があります。

☒ 64ビット作業モード
☒ MSS有効
☒ TN3270リスナーの作成

TN3270リスナーポート 9004

* 入力必須の項目です

保存 戻る



重要

コンパイル環境で指定した稼働ビット数 =
Enterprise Server インスタンス動作モード = XA
リソースビット数 = データベースクライアント対応ビット数である必要があります。

- ⑤ 作成した MSSDEMO インスタンスが一覧に表示されますので、[編集] アイコンをクリックします。

名前	タイプ	ステータス	64ビット	MSS有効	セキュリティ	PAC	
MSSDEMO	Region	Stopped	✓	✓	デフォルト		

編集

- ⑥ 表示された画面の [リージョンの機能] > [JES 有効] にチェックし、[追加設定] > [構成情報] へ CICS マップの日本語英数カナ表示に必要な環境変数を指定して [適用] ボタンをクリックします。

設定値)
[ES-Environment]
MFICODESET=9122

追加設定

構成情報 ?

[ES-Environment]
MFICODESET=9122

リージョンの機能

☒ MSS有効 ? ☒ JES有効 ?
☐ IMS有効 ? ☐ MQ有効 ?



重要

インスタンス内の指定値は全て半角英数文字で入力してください。
2 バイト文字を使用すると正常に稼働しない場合があります。

- ⑦ 画面上部の [CICS] プルダウンメニューから [構成] を選択し、表示される画面の各項目を設定します。値を入力後、[適用] ボタンをクリックします。

項目名	説明
システム初期化テーブル(SIT)	CICS インスタンス設定の詳細が提供されるシステム初期化テーブルを指定します。ここでは例題に含まれている DBCS を指定します。
リソース定義ファイルパス	CICS リソース定義ファイルのパスを指定します。ここでは例題で用意されているリソース定義ファイルのパスを指定します。
トランザクションパス	実行される CICS プログラムの探索パスを指定します。
ファイルパス	データセットのデフォルトパスを指定します。ここでは例題で用意されている VSAM ファイルの置かれているパスを指定します。

マップパス	コンパイル済み BMS マップセットのパスを指定します。ここでは .MOD ファイルが配置されているパスを指定します。
-------	---

cicsの構成 |  **適用**

システム初期化テーブル (SIT) 
DBCS

リソース定義ファイルパス
/home/tarot/mf/MSSDEMO

トランザクションパス 
/home/tarot/mf/MSSDEMO

ファイルパス 
/home/tarot/mf/MSSDEMO/DATAFILE

マップパス 
/home/tarot/mf/MSSDEMO

☐ EZASOCKET サポート 

 **重要**
入力値は全て半角英数字で指定してください。
これらのフィールドでは改行を入れないように注意してください。

- ⑧ 画面上部の [JES] プルダウンメニューから [構成] を選択し、表示される画面の各項目を設定します。
値を入力後、[適用] ボタンをクリックします。

項目名	説明
JES プログラム パス	COBOL アプリケーションの実行ファイルが存在するパスを指定します。
システムカタログ	カタログファイルを出力するパスと、そのファイル名称を指定します。
データセットの省略時ロケーション	ジョブ実行時に生成されるスプールデータやカタログされるデータセットのデフォルトパスを指定します。
システムプロシージャライブラリ	プロシージャライブラリの名前を指定します。 ここでは SYS1.PROCLIB を入力します。

設定例)

JESの構成 |  **適用**

JES プログラムパス 
/home/tarot/mf/MSSDEMO

システムカタログ 
/home/tarot/mf/MSSDEMO/DATAFILE/catalog.dat

データセットの省略時ロケーション 
/home/tarot/mf/MSSDEMO/DATAFILE

システムプロシージャライブラリ 
SYS1.PROCLIB

Fileshare 構成ロケーション 

- ⑨ JES 画面の下部 [イニシエータ] 項目の [新規作成] ボタンをクリックします。

イニシエータ |  *** 新規作成**

- ⑩ 下記画面のように入力して【保存】ボタンをクリックします。この指定により MSSDEMO インスタンスが開始時にイニシエータが稼働し、ジョブクラス A,B,C のジョブが実行可能になります。

JESイニシエータ

名前 *
INITABC

クラス
ABC

説明
A,B,C クラスのイニシエーター

☐ 入力必須の項目です

保存

戻る

- ⑪ セキュリティ観点から、Web リスナーのデフォルトステータスは【Disabled】になっています。安全を確認したうえで、【一般】プルダウンメニューから【リスナー】を選択し、表示された Web リスナーのステータスを【Stopped】へ変更後、【適用】ボタンをクリックします。

TLS設定

ステータス
Disabled

ステータスの設定
Stopped

実際のアドレス
tcp:0.0.0.0:0

また、各リスナーの【tcp】の値がリモートマシンのホスト名または IP アドレスと一致させます。

リスナープロパティ

適用

削除

☐ 入力必須の項目です

名前
Web Services and JZEE

このエンドポイントはネットワーク経由でアクセス可能で、TLSが有効ではありません。

プロトコル
tcp

ホスト名またはIPアドレス
ALinux2

- ⑫ 前項で作成した XA スイッチモジュールを定義します。画面上部の【一般】プルダウンメニューから【XA リソース】を選択し、【新規作成】ボタンをクリックします。

XA リソース

🔍

* 新規作成

- ⑬ 必要項目を入力後【保存】ボタンをクリックします。

項目名	説明
ID	プログラムや JCL の IKJEFT ユーティリティに渡す ID を指定します。 ここでは XADB を指定します。 ID * XADB
名前	XA リソース名として任意の名前を指定します。 Oracle の場合は Oracle.XA 固定です。 名前 * ORACLE_XA
モジュール	前項で作成した XA スイッチモジュールのパスとファイル名を指定します。 【Oracle 使用時の例】 動的登録 /home/tarot/xa/ESORAXA64_D.so を入力します。 【Db2 使用時の例】

	動的登録 /home/tarot/xa/ESDB2XA64.so を入力します。
OPEN 文字列	対象データベースのオープン文字列を指定します。 【Oracle 使用時の例】 Oracle_XA+SesTm=100+SqlNet=tok-par+Acc=P/scott/tiger を入力します。 OPEN 文字列 ⓘ <u>Oracle_XA+SesTm=100+SqlNet=tok-par+Acc=P/scott/tiger</u> 【Db2 使用時の例】 DB=SAMPLE,uid=db2inst1,pwd=ibmdb2,AXLIB=casaxlib を入力します。 静的登録の場合は末尾に SREG=T を指定します。デフォルトは動的です。 OPEN 文字列 ⓘ <u>DB=SAMPLE,uid=db2inst1,pwd=ibmdb2,AXLIB=casaxlib</u>
有効	有効、無効切り替えチェックを指定します。ここではオンを指定します。

4) システムプロシージャライブラリ(例:SYS1.PROCLIB)の作成

JES 機能を使用する際のシステムプロシージャライブラリ指定も Windows 環境と同様です。カタログファイルに登録するパスヘディレクトリ(名前の例:SYS1.PROCLIB)を作成しておきます。また、インスタンス開始後に PO としてカタログすることも忘れないでください。

5) データセットの配備

Windows 環境でテスト済みのデータファイルをそのままバイナリ転送し、インスタンスへ指定した適切なディレクトリに配置します。

Linux/UNIX マシンではケースセンシティブであることに注意してください。

重要

バージョン 7.0 では、パフォーマンス向上の観点から JES 関連ファイルである SPLJOB.DAT のフォーマットが改善されています。そのため、旧バージョンのファイルを 7.0 以降で利用する場合は mfsplcnv コマンドを使用して新フォーマットにコンバートする必要があります。コンバートを実行すると、古いフォーマットのファイルは SPLJOB.bak として保存されます。

対象ファイルの特定には MFSYSCAT 環境変数を利用して、カタログファイルを指定します。

例)

```
export MFSYSCAT=/home/tarot/mf/MSSDEMO/DATAFILE/catalog.dat
mfsplcnv -2
```

詳しくは製品マニュアルをご参照ください。

2.9 COBOL プログラムのコンパイル

弊社の COBOL コンパイラは IBM 社歴代の COBOL に準拠し、また、代表的な RDBMS との連携をサポートするなど、多くのコンパイラ指令を用意しています。Linux/UNIX 環境では設定した COBOL の作業モードに沿ってコンパイルが実行されます。

ここでは Windows 環境の Eclipse で使用する CICS, JCL チュートリアルの例題を使用します。コマンド例ではルートユーザーでコンパイルを実施していますが、一般ユーザー行うことも可能です。



注意

COPY メンバファイルの拡張子は、コンパイル指令 COPYEXT(CPY,cpy) で大文字、小文字を指定しておくことでケースセンシティブに対応できます。

1) バッチプログラムのコンパイル

コマンド例)cob -u <ソースファイル名> -C"DIALECT(ENTCOBOL) CHARSET(ASCII)"

```
#cob -u KSDSWRT2.cbl -C"DIALECT(ENTCOBOL) CHARSET(ASCII)"
```



注意

コンパイルの方言指定で "DIALECT(ENTCOBOL)" を使用する場合は、EBCDIC 文字モードがデフォルトです。ASCII 文字コード使用時は CHARSET(ASCII) を指定します。

実行ファイルの .gnt、デバッグ用ファイルの .idy、中間コードの .int が生成されます。

```
KSDSWRT2.gnt KSDSWRT2.idy KSDSWRT2.int
```

2) CICS プログラムのコンパイル

コマンド例)cob -u <ソースファイル名> -C"DIALECT(MF) OSVS CICSECM()"

ACCT00.cbl ~ ACCT04.cbl までをコンパイルします。

```
#cob -u ACCT00.cbl -C"DIALECT(MF) OSVS CICSECM()"
Micro Focus CICS Option Preprocessor (CICS) - Version 7.0.00
```

3) BMS ファイルのコンパイル

コマンド例)mfbmscl acctset.bms

.MOD ファイルが生成されます。コンパイルオプションについては製品マニュアルをご参照ください。

4) SQL 文をもつプログラムのコンパイル

DB2 コマンド例)

cob -u TBLCRTE.cbl

-C"DB2(DB==SAMPLE,VALIDATE==RUN,PASS==db2inst1.ibmdb2)"

```
#cob -u TBLCRTE.cbl -C"DB2(DB==SAMPLE,VALIDATE==RUN,PASS==db2inst1.ibmdb2)"
```

2.10 JCL のサブミット

1) インスタンスの起動

コマンド例)casstart /r<インスタンス名>

セキュリティが有効な場合は、権限があるユーザーの /u ユーザー名 /p パスワード を追加指定して起動します。

```
#casstart /rMSSDEMO
..
CASC00167I ES Daemon successfully auto-started 08:25:15
CASC0005I /var/mfcobol/es/MSSDEMO/console.log 08:25:15
CASC00050I ES "MSSDEMO" initiation is starting 08:25:15
```

[開始] 状態になっても、各サービスが正常に起動されたかをコンソールログで確認してください。



注意

環境変数の COBOL 作業モードとインスタンスの動作モードが異なる場合は正常に開始されません。両モードのビット数を合わせる必要があります。

2) JCL のサブミット

コマンド例)cassub /r<インスタンス名> /j<JCL ファイル名>

セキュリティが有効な場合は、権限があるユーザーの /u ユーザー名 /p パスワード を追加指定して起動します。

```
[root@ ~]# cassub /rMSSDEMO /jvsamwrt2.jcl
JCLCM0187I JOB01002 VSAMWRT2 JOB SUBMITTED (JOBNAME=VSAMWRT2,JOBNUM=01002) 16:04:41
JCLCM0180I JOB01002 VSAMWRT2 Job ready for execution. 16:04:41
Processed "vsamwrt2.jcl"
[root@ ~]#
```

3) スプールとカタログの確認

ESCWA から起動したインスタンスを選択します。画面上部の [JES] プルダウンメニューから [スプール] を選択し、JCL の実行結果を確認します。

- ① スプーラー一覧が表示されます。フィルタ機能で [すべて] が指定されていることを確認します。

JES 出力キュー

🔍

フィルタ

* 入力必須の項目です

*

ユーザー

*

ジョブ番号

*

クラス

*

COND

*

出力タイプ

キュー

☒ すべて

☒ 入力

☒ 入力の保留

☒ ディスパッチ

☒ アクティブ

☒ 完了

☒ 出力

☒ 出力の保留

☒ 印刷中

- ② 実行された VSAMWRT2 ジョブが現れています。条件コードが 0000 の場合は JCL が正常に終了したことを示しています。

☐ VSAMWRT2 J0001002 A mfuser 0000

- ③ VSAMWRT2 をダブルクリックすると以下のようにジョブ実行結果が表示されます。

DDエントリ							
DD名	ステップ	PROCステップ	状態	クラス			
状態	クラス	DD名	ステップ	ステップ番	PROCステ	レコード数	
 Hold	A	JESYSMSG		0		76	
 Ready	A	SYSPRINT	DEFVSAM1	1		18	
 Ready	A	SYSOUT	SORTSTEP	2	SORT1	12	
 Ready	A	SYSOUT	APPL1	3		1	
 Ready	A	PRINTER	APPL1	3		10	
 Ready	A	SYSPRINT	VERIFY1	4		41	

- ④ JESYSMSG をダブルクリックすると以下のようにジョブログが表示されます。

```

コードセット
表示 << < > >> ページ: 1 / 1 76 行 / 76 ASCII
☐ 詳細

*****
*** Micro Focus ESJCL ASCII JES2 Version ED7.0_PU2E ***
*** Copyright (C) Micro Focus 1997-2020. All rights reserved. ***
*** Job: 0001002 Name: VSAMWRT2 User: mfuser Date: 12/25/21 Time: 09:38:46 ***
*** File: $TXRFDIR/mfuser09384691.t ***
*** DSN: ***
*****

1 //VSAMWRT2 JOB CLASS=A,MSGCLASS=A
2 //*****
3 //DEFVSAM1 EXEC PGM=IDCAMS
4 //SYSPRINT DD SYSOUT=*
5 //SYSIN DD *
14 //SORTSTEP EXEC SORTD
   XXSORTD PROC
   XXSORT1 EXEC PGM=SORT

```

- ⑤ 以下のように各ステップが正常終了している履歴が確認できます。JCL が異常終了した場合にはここでエラーの原因を調査することができます。

```

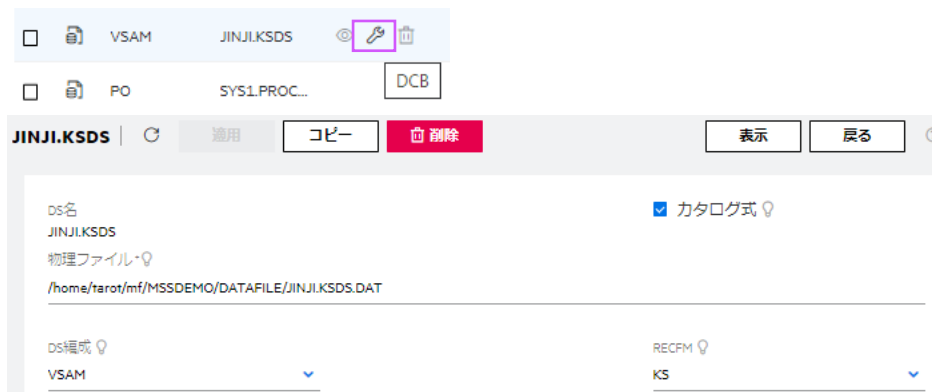
---> 15:21:30 JCLCM0191I STEP ENDED      VERIFY1 - COND CODE 0000
---> 15:21:30 JCLCM0182I JOB  ENDED      - COND CODE 0000

```

- ⑥ 再度実行結果スプール一覧を表示して各ステップの出力スプールが表示されますので、各リンクをダブルクリックして結果を確認してみてください。

- ⑦ この JCL 実行によってカタログされたデータセットを見えます。画面上部の [JES] プルダウンメニューから [カタログ] を選択し、[リスト] ボタンをクリックします。VSAM ファイル JINJI.KSDS がカタログされていることがわかります。

- ⑧ このファイルにカーソルを合わせ、[DCB] アイコンをクリックすると以下のように DCB 情報が表示されます。



- ⑨ カタログ一覧へ戻り [JINJI.KSDS] リンクをダブルクリックすると以下のようにデータセットの内容が表示されます。

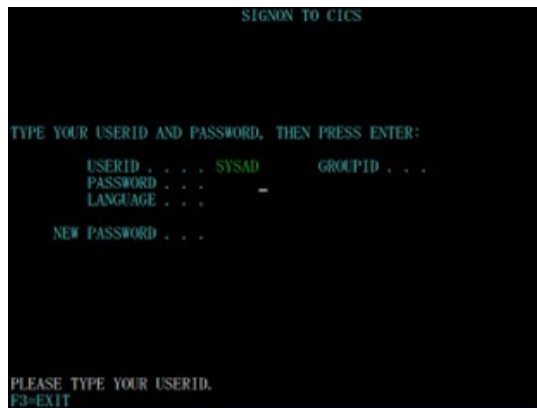


2.11 CICS トランザクションの実行

- 1) お使いの 3270 端末エミュレータを使用して、<ホスト名または IP アドレス>:9004 に接続します。以下は IBM パーソナル・コミュニケーションズを使用したものです。

補足) クリアキーを使用するため、TN3270 エミュレータのキーボード設定をご確認ください。





- 2) デフォルトでは初期トランザクションに CESN が設定されているため上記のようなサインオン画面が現れます。ユーザー ID SYSAD, パスワード SYSAD でサインオンすると下記の画面が表示されます。

```

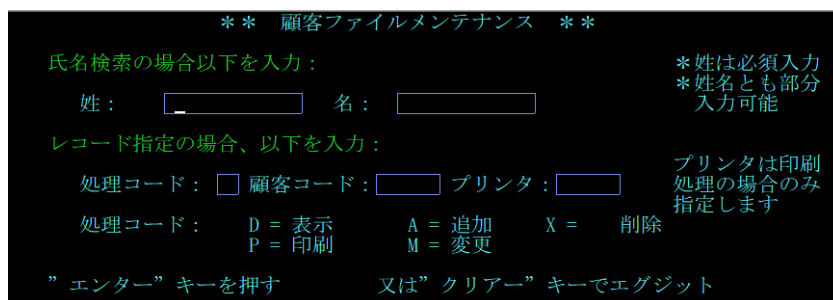
PASSE00121 SIGNON COMPLETE AT B000, FOR USER SYSAD. LOCAL SECURITY IS DISABLED.
15:47:47
  
```

- 3) クリアキーで画面をクリアし、トランザクション ACCT を実行します。

```

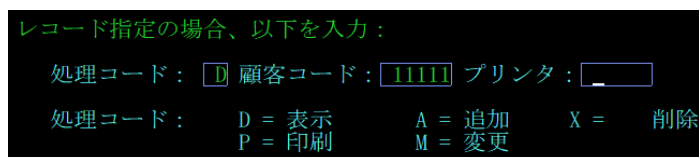
ACCT_
  
```

- 4) 以下のように例題プログラムの初期画面が現れます。



- 5) Tab キーでフィールドを移動し、以下のように入力してから実行キーを押下します。

処理コード:D 顧客コード:11111



- 6) 入力値でファイル照会を行い、マッチしたデータが詳細画面へ表示されます。

顧客ファイル レコード表示

顧客番号: 11111 姓: 柴
電話: 0771778888 住所: 式部 4-3、石山寺
大津市
滋賀県

MI: G 敬称: MRS

その他の請求先:

発行カード枚数: 1 発行日: 06 07 07 理由: N
カードコード: X 承認者: GNG 特別コード:

顧客状況: N 請求限度額: 1000.00

履歴:	残高	請求日	請求額	支払日	支払額
	0.00	00/00/00	0.00	00/00/00	0.00
	0.00	00/00/00	0.00	00/00/00	0.00
	0.00	00/00/00	0.00	00/00/00	0.00

表示終了なら”クリア”か”エンター”を押す

7) ターゲットマシンでの動作確認が完了しましたので、端末エミュレータを切断します。

2.12 Enterprise Server インスタンスの停止

MSSDEMO インスタンスを停止します。

コマンド例) `casstop /rMSSDEMO`

セキュリティが有効な場合は、権限があるユーザーの /u ユーザー名 /p パスワード を追加指定して停止します。

パラメータの詳細は製品マニュアルをご参照ください。

2.13 デバッグについて

アニメータ(対話式デバッガ)によるデバッグも可能ですが、Windows 環境の Eclipse IDE からターゲットマシンへのリモートデバッグを推奨しています。リモート開発のチュートリアルは Eclipse 版の製品マニュアルにございますので、こちらもご参照ください。

1) アニメーターの開始

実行形式ファイルを指定して開始します。

コマンド例) `anim ./AAAAA.int`

2) アニメーターの終了

[Esc] キーをクリックして、確認メッセージに Y を入力します。

3) アニメーター詳細に関しては製品マニュアルをご参照ください。

3. 免責事項

本チュートリアルの例題ソースコードは機能説明を目的としたサンプルであり、無謬性を保証するものではありません。例題ソースコードは弊社に断りなくご利用いただけますが、本チュートリアルに関わる全てを対象として、二次的著作物に引用する場合は著作権法の精神に基づき適切な扱いを行ってください。

本チュートリアルで学習した技術の詳細については製品マニュアルをご参照ください。