Micro Focus 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になりますが、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 を使用することになります。

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 Red Hat Enterprise Linux Server release 7.5 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. Micro Focus Directory Server の起動
6. JES / CICS インスタンスの作成
7. COBOL プログラムのコンパイル
8. JCL のサブミット
9. CICS トランザクションの実行
10. Enterprise Server インスタンスの停止
11. 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 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 を使用し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。コンパイルはコマンドラインからの作業となります。

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はターゲットマシンに合わせる必要があります。

3) 実行ファイルの配備

Linux/UNIX 上でプログラムを再コンパイルし、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。

4) 生成済み BMS マップ（.MOD）の転送

CICS の場合、Windows 開発環境で生成した BMS マップ（.MOD）をターゲットマシンへ転送し、Enterprise Server インスタンスに指定したマップパスへ配備します。BMS マップは Linux/UNIX 環境ではコンパイルされませんので注意してください。

.MOD ファイル等の生成済みファイルはファイル名、拡張子ともに大文字にしてください。(AAA.MOD)

2.2 Enterprise Developer のインストール

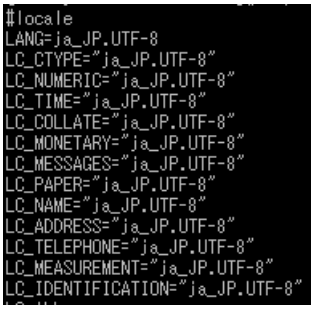
1) SJIS ロケールの設定

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



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

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ので、その場合は root ユーザーで次のコマンドを実行することで追加が可能です。

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

Micro Focus 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 Micro Focus 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）。

詳細は下記 URL の Readme [ライセンス] – [Linux および UNIX で製品のライセンスを有効にする場合]

http://www.microfocus.co.jp/support/fixpacks/mfenterprisedeveloper-dised-50-install-readme.html

および下記 URL の [UNIX/Linux 環境 における SafeNet ライセンスの認証・移行方法] をご参照ください。

http://www.microfocus.co.jp/support/fixpacks/SafeNet_U_R7.html

4) ファイアウォールの解除

このチュートリアルでは先ず下記ポートのアクセスを許可しておいてください。

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/ED50/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。

```
#. /opt/mf/ED50/bin/cobsetenv
COBDIR set to /opt/mf/ED50
```

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/sqllib/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詳細に関しては製品マニュアルを参照してください。

2.5 Micro Focus Directory Server の起動

root ユーザーで mfds (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 mfds32 コマンド、64 bit 環境用に mfds64 コマンドを実行することもできます。

コマンド例）mfds64 &

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 mfds のプロセスが起動されます。

2.6 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

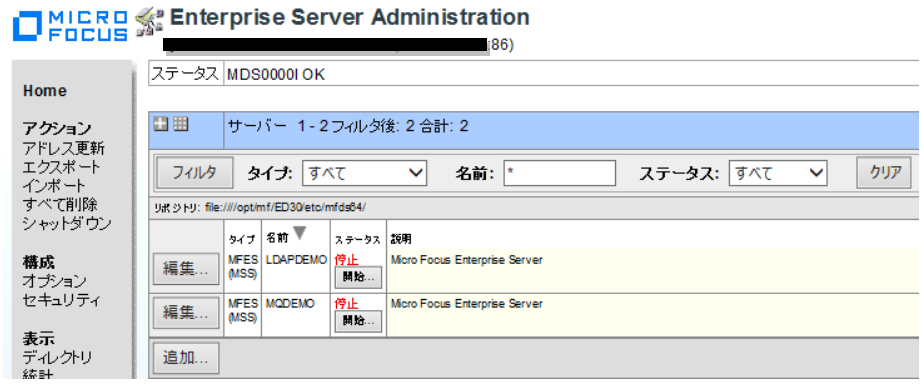
CICS 環境（IMS 機能でも使用）に必要なリソース定義は dfhdrdat（dfhdrdat.idx 付き形式も可能）という固有の索引ファイルで管理されています。 Windows 環境でテスト済みのファイルをそのままバイナリ転送し、上記で構成した適切なディレクトリに配置します。ファイル名は小文字です。

3) 管理画面へのアクセス

Enterprise Server Administration （管理画面）へのアクセス権はインストール時の –ESadminID 指定による管理者ユーザー ID の権限が反映されます。

Enterprise Server Administration は、mfds が起動している環境のホスト名または IP アドレスとポート番号（デフォルトは 86）を組み合わせた URL をブラウザに指定し、表示します。

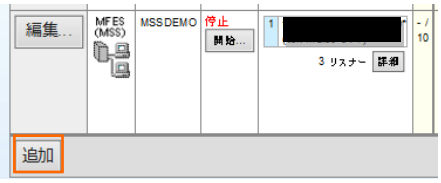
http://<ホスト名または IP アドレス>:86



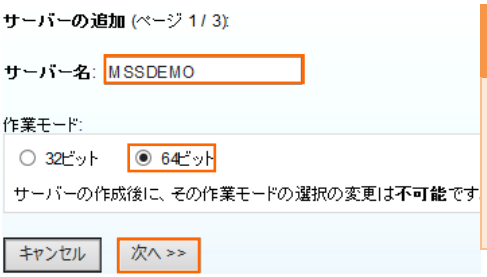
4） インスタンスの作成

管理画面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 ： （コロン）で区切ります。

① 管理画面左下の [追加] ボタンをクリックします。



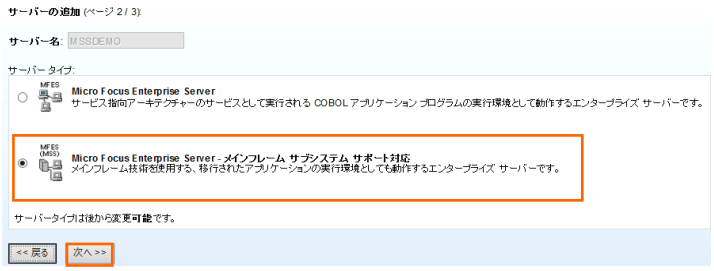
② 追加画面へ遷移しますので、[サーバー名] に MSSDEMO を入力し、動作モードに稼働ビット数を選択して [次へ] ボタンをクリックします。



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。

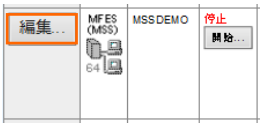
③ 次の画面では [Micro Focus Enterprise Server - メインフレーム サブシステム サポート対応] が選択されていることを確認して [次へ] ボタンをクリックします。



④ 3 番目の画面では [TN3270 リスナーの作成] のチェックをオンにし、ポート番号として 9004 を入力します。

最後に [追加] ボタンをクリックします。

⑤ 一覧に作成した MSSDEMO インスタンスが表示されますので、[編集] ボタンをクリックします。



⑥ 表示された画面下部の [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、[Apply] ボタンをクリックします。

設定値)

[ES-Environment]

MFCODESET=9122

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 [サーバー] > [プロパティ] > [MSS] > [CICS] タブで表示される画面の各項目を設定します。入力後は [Apply] ボタンをクリックします。

項目名	説明
メインフレーム サブシステム サポート有効	[MSS] タブ配下の設定をオン、オフ指定します。ここではオンを指定します。
システム初期化テーブル（SIT）	CICS 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。ここではサンプルに含まれている DBCS 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。ここでは .gnt ファイルが存在するパスを指定します。
File Path	データセットのデフォルトパスを指定します。ここではサンプルで用意されている 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 ファイルが存在するパスを指定します。
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。転送したリソース定義ファイルが存在するパスを指定します。

設定例)

重要

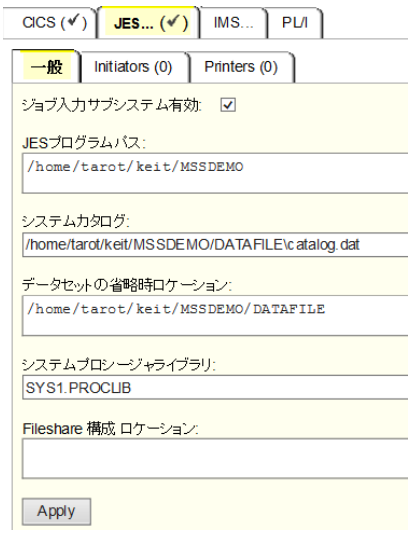
入力値は全て半角英数字で指定してください。

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 [サーバー] > [プロパティ] > [MSS] > [JES] タブで表示される画面の各項目を設定します。入力後は [Apply] ボタンをクリックします。

項目名	説明
ジョブ入力サブシステム 有効	[JES] タブ配下の設定をオン、オフ指定します。ここではオンに指定します。
JES プログラム パス	COBOL アプリケーション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	カタログファイルが存在するパスと、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）

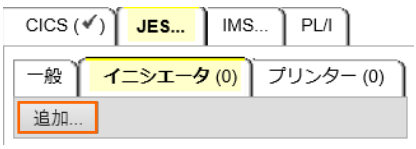


重要

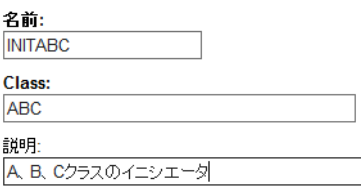
入力値は全て半角英数字で指定してください。

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。

⑨ [サーバー] > [プロパティ] > [MSS] > [JES] > [イニシエータ] タブを表示し、左下の [追加] ボタンをクリックします。



⑩ 下記画面のように入力して [追加] 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。



⑪ 前項で作成した XA スイッチモジュールを定義します。[サーバー] > [プロパティ] > [XA リソース] タブを表示して、左下の [追加] ボタンをクリックします。

⑫ 必要項目を入力後 [追加] 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 DSN TSO コマンドの SYSTEM パラメータへ指定する 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: Oracle_XA+SesTm=100+SqlNet=tok-par+Acc=P/scott/tiger 【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: DB=SAMPLE,uid=db2inst1,pwd=ibmdb2,AXLIB=casaxlib
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。

⑬ 画面左上の [Home] をクリックして一覧画面に戻ります。

5) 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へディレクトリ（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。

6) 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。

データファイル名は大文字、拡張子は小文字にしてください。(AAA.dat)

2.7 COBOL プログラムのコンパイル

Micro Focus の COBOL コンパイラは多くのコンパイラ指令を用意しており、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しています。 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。

COBOL ソースおよび COPY メンバのファイル名は大文字、拡張子は小文字にしてください。(AAA.cbl)

注意

コンパイルの方言指定で “DIALECT(ENTCOBOL)” を使用する場合は拡張子が大文字になる仕様のため、コンパイルに使用する COPY メンバファイルの拡張子は大文字にしておいてください。(.CPY)
ただし、コンパイル指令「COPYEXT=CPY,cpy」指定で小文字も有効にできます。

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)"
```

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
```

注意

コンパイルの方言指定で “DIALECT(ENTCOBOL)” を使用する場合は、CHARSET(ASCII) を指定しないと EBCDIC モードでコンパイルされます。

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 3.0.00
#
```

3) 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)"

```
[root@                    ]# cob -u TBLCRTE.cbl -C"DB2(DB==SAMPLE,VALIDATE==RUN,PASS==tarot.password)"
[root@                    ]#
```

2.8 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>

```
[root@t                   ]# casstart /rMSSDEMO
..
CASCD0167I ES Daemon successfully auto-started 15:41:12
CASCD0050I ES "MSSDEMO" initiation is starting 15:41:12
[root@t                   ]#
```

[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

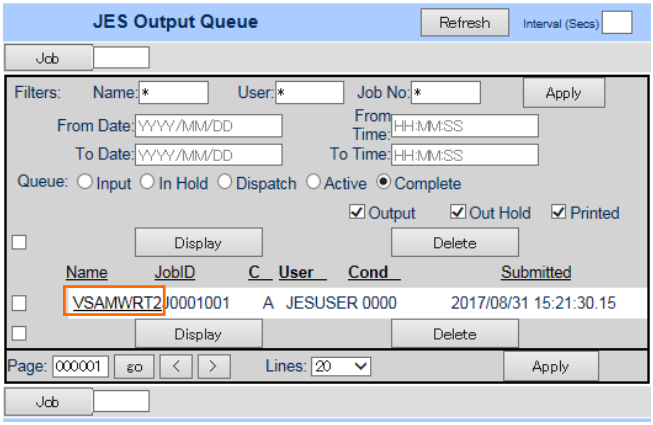
コマンド例）cassub /r<インスタンス名> /j<JCL ファイル名>

```
[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@                    ]#
```

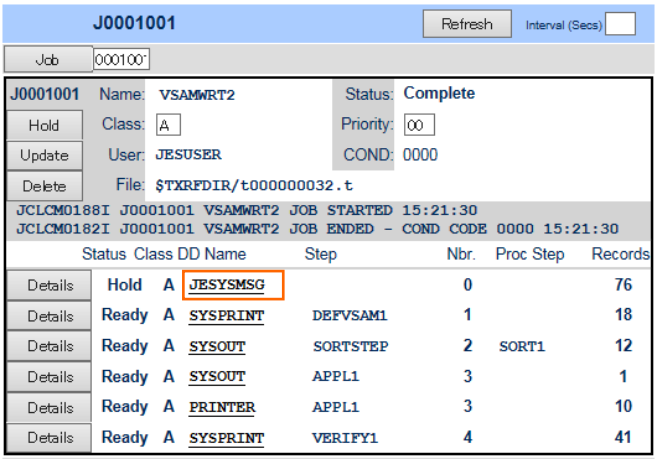
3) スプールとカタログの確認

ジョブの実行結果を確認します。

① 以下のようにスプールビューの Out Hold キューを開くと、実行された VSAMWRT2 ジョブが現れています。Cond = 0000 はジョブが正常に終了したことを示しています。なお、1 回目の VSAMWRT2 のジョブ結果では COND: 00008 が出るはずです。これは、このジョブの STEP1 において一旦 JINJI.KSDS ファイルを削除する処理が入っているにも関わらず、最初の実行であるため、そのファイルが存在していないことが理由ですので問題はありません。2 度目以降の実行であれば、0000 でジョブが正常に終了するはずです。



② VSAMWRT2 を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。



③ JESYSMSG を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。

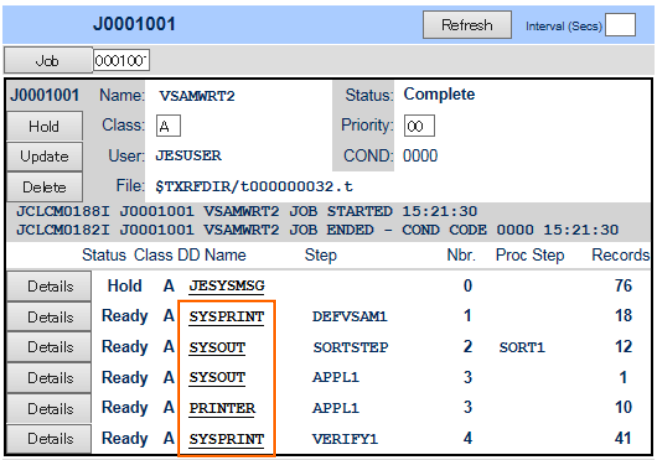
```
*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
*-*-*  Micro Focus ESJCL  ASCII  JES2 Version ED3.0_022                          *-*-*
*-*-*  Copyright (C) 1997-2017 Micro Focus. All rights reserved.                 *-*-*
*-*-*  Job: J0001001 Name: VSAMWRT2 User: JESUSER  Date: 08/31/17 Time: 15:21:30 *-*-*
*-*-*  File: $TXRFDIR/t000000032.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
15 //SORT1.SORTIN DD *
27 //SORT1.SORTOUT DD DSN=&&JINJIDAT,DISP=(NEW,PASS),
28 // SPACE=(800,(10,10)),DCB=(RECFM=FB,LRECL=71,DSORG=PS),UNIT=SYSDA
29 //SORT1.SYSIN DD *
32 //APPL1 EXEC PGM=KSDSWRT2
33 //SYSOUT DD SYSOUT=*
34 //KSDSFILE DD DSN=JINJI.KSDS,DISP=SHR
35 //INDD DD DSN=&&JINJIDAT,DISP=(OLD,DELETE)
36 //PRINTER DD SYSOUT=*
37 //*****************************************
38 //VERIFY1 EXEC PGM=IDCAMS
39 //SYSPRINT DD SYSOUT=*
40 //SYSIN DD *
*** JCLCM0180I Job ready for execution.
*** Execution on Server MQDEMO    Process       9589
```

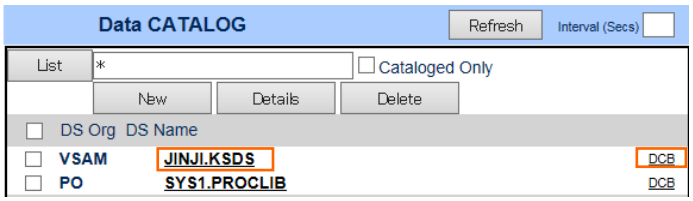
④ 以下のように各ジョブステップが正常終了している履歴が確認できます。ジョブが異常終了した場合にはここでエラーの原因を調査することができます。

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

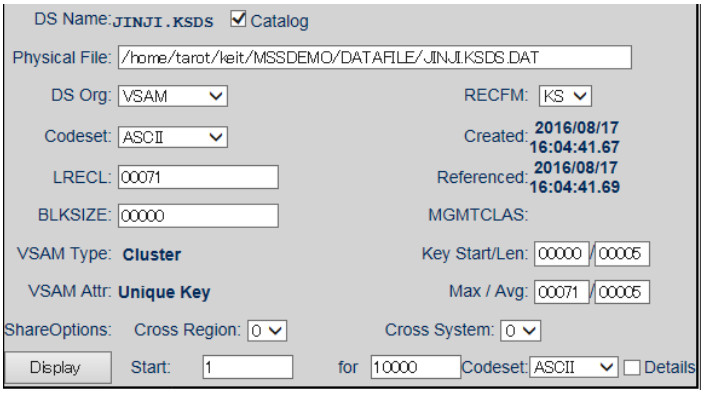
⑤ 再度実行結果スプール一覧を表示して各ジョブステップの出力スプールが表示されますので、各 STEP のリンクをクリックして結果を確認してみてください。



⑥ このジョブの実行によってカタログされたデータセットを見てみます。以下のようにカタログビューを開き、[List] 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。



⑦ 右端の [DCB] をクリックすると以下のように DCB 情報が表示されます。



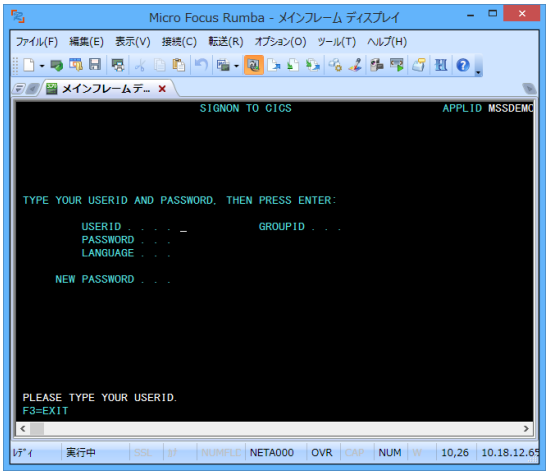
⑧ カタログ一覧へ戻り [JINJI.KSDS] リンクをクリックすると以下のようにデータセットの内容が表示されます。

2.9 CICS トランザクションの実行

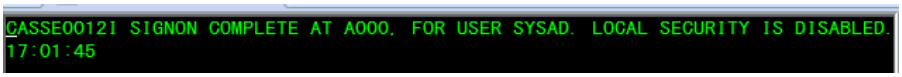
1) お使いの 3270 端末エミュレータを使用して、<ホスト名または IP アドレス>:9004 に接続します。以下は Micro Focus 純正の RUMBA を使用したものです。

補足）TN3270 エミュレータで、使用しているキーボード設定をご確認ください。

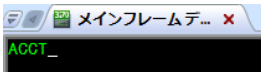
Rumba の例）



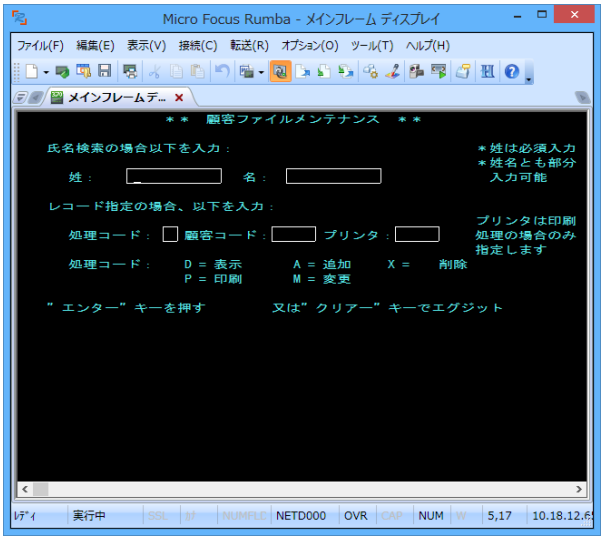
2) デフォルトでは初期トランザクションに CESN が設定されているため上記のようなサインオン画面が現れます。ユーザー ID SYSAD, パスワード SYSAD でサインオンすると下記の画面が表示されます。



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。

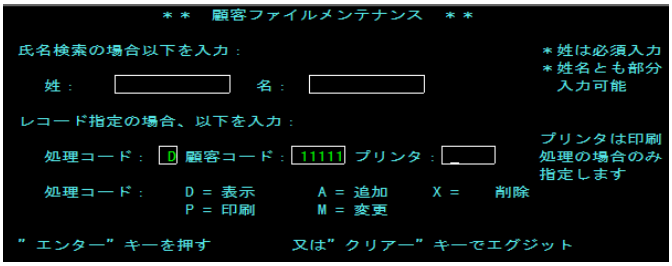


4) 以下のように例題プログラムの初期画面が現れます。

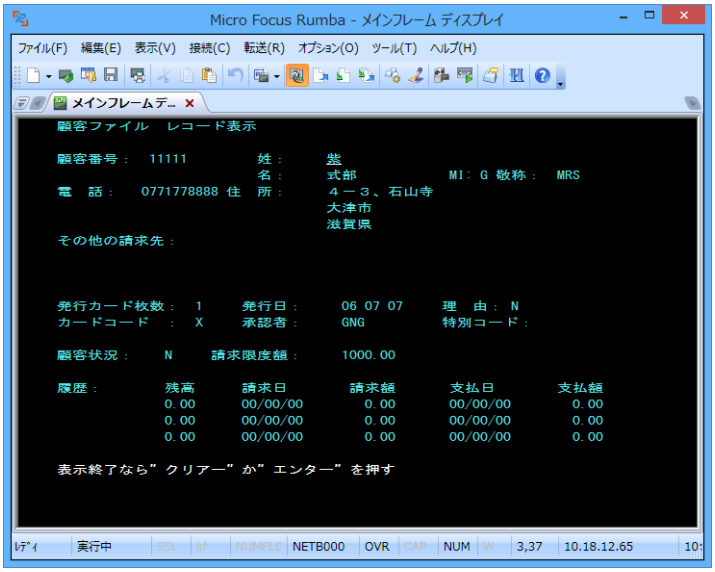


5) Tab キーでフィールドを移動し、以下のように入力してから実行キーを押下します。

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。



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。

2.10 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

casstop コマンドにはパラメータも用意されています。詳細は製品マニュアルをご残照ください。

2.11 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クを推奨しています。

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。

WHAT'S NEXT

- リモート メインフレーム COBOL 開発：JCL　Eclipse 編
- 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。