

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 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. 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.sjis
LC_CTYPE=ja_JP.sjis
LC_NUMERIC=ja_JP.sjis
LC_TIME=ja_JP.sjis
LC_COLLATE=ja_JP.sjis
LC_MONETARY=ja_JP.sjis
LC_MESSAGES=ja_JP.sjis
LC_PAPER=ja_JP.sjis
LC_NAME=ja_JP.sjis
LC_ADDRESS=ja_JP.sjis
LC_TELEPHONE=ja_JP.sjis
LC_MEASUREMENT=ja_JP.sjis
LC_IDENTIFICATION=ja_JP.sjis
LC_ALL=
```

② 使用可能な日本語ロケールの確認

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の設定

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例]

- 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] > [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、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] > [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

Directory Server 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/ED11PU2/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/ED11PU2/bin/cobsetenv
COBDIR set to /opt/mf/ED11PU2
```

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 64bit 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 mfd の一括管理できる 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 mfd 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 &
```

上記の & 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。

製品バージョン 10.0 以降では、製品が提供する VSAM 外部セキュリティマネージャー (以降 ESM と称す) 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。

まずはデフォルトユーザーと初期パスワードを取得します。

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
```

```
{"mfUser": "SYSAD", "mfPassword": "hi_wvpvJ"}
```

上記例の場合、デフォルトユーザーは SYSAD、初期パスワードは hi_wvpvJ です。

この情報は Linux マシンで稼働する ESCWA ログオン時に使用しますので、記憶しておいてください。

デフォルトセキュリティを無効にすることもできますが、安全を確認後に実施してください。

詳しくは製品マニュアルの [ここからはじめよう] > [Getting Started] にある [デフォルトセキュリティの構成] チュートリアルをご参照ください。

Windows マシンの Web ブラウザから Linux マシンで起動した 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。

TN3270 エミュレータに設定されたコードページによって、自動的に半角英小文字が半角英大文字に変換されることがあります。この場合は、パスワードに含まれる半角英小文字と CICS サインオン時のパスワードが不一致となり、CICS にサインオンできません。

これを回避するために、新パスワードの英字はすべて大文字で入力してください。

良い新パスワードの例) SYSAD123

悪い新パスワードの例) sysad123

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。

Enterprise Server Administration

⚠ Rocket Software Enterprise Serverでは、インストール時に基本的なセキュリティ機能がデフォルトで有効になっています。


[詳細情報](#)

ユーザー名

パスワード

パスワード変更

認証情報は、次のセキュリティ マネージャを使用して検証されます: VSAM ESM



Enterprise Server Administration

⚠ Rocket Software Enterprise Serverでは、インストール時に基本的なセキュリティ機能がデフォルトで有効になっています。

[詳細情報](#)


ユーザー名

パスワード

新しいパスワード* パスワードの確認*

キャンセル

認証情報は、次のセキュリティ マネージャを使用して検証されます: VSAM ESM



ESCWA が表示されます。

Rocket Software Enterprise Server | ダッシュボード オペレーション メインフレーム セキュリティ

ナビゲーション

- グループ
 - 論理
 - PAC
 - Directory Server
 - Default
 - CICS1
 - CICS2
 - ESDEMO
 - ESDEMO64
 - ESIMTK
 - RJCLDEMO
 - SOR

リージョンおよびサーバー プロパティ セキュリティ ジャーナル

このDirectory ServerホストではTLSが有効ではありません。データはローカル ネットワーク上で送信されますが、他の接続が公開される可能性があります。

リージョンおよびサーバー リスト ※ 新規作成 すべて削除 エクスポート インポート コピー 更新

フィルタ

ア...	名前	タ...	ステータス	64ビ...	MSS有効	セキュリティ	アクション
<input type="checkbox"/>	CICS1	Region	Stopped	✓	✓	デフォルト	編集 設定 削除
<input type="checkbox"/>	CICS2	Region	Stopped	✓	✓	デフォルト以外	編集 設定 削除
<input type="checkbox"/>	ESDEMO	Region	Stopped			デフォルト	編集 設定 削除

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 Directory Server(mfds)の起動

root ユーザーで mfds(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 32bit 環境用に mfds32 コマンド、64bit 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)mfds &

上記の&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。
<https://download.boulder.ibm.com/ibmdl/pub/software/dw/java/cdctables.zip>
- 2) 製品インストールフォルダ配下の etc フォルダに CCSID フォルダがない場合はこれを作成します。
Linux 例) /opt/mf/ED11PU2/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、両マシンの CCSID フォルダ配下へ貼り付け、14 ファイルが含まれていることを確認します。

```
#ls /opt/mf/ED11PU2/etc/ccsid/IBM-932
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) Linux マシンでは権限のあるユーザーで ESCWA を再起動し、インストールした CCSID をロードさせます。

停止コマンド) `escwa --shutdown SYSAD hi_wvpvj`

開始コマンド)

`cd $COBDIR/bin`

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

- 7) IBM-932 を利用するために、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

各ファイルの配置を決定します。

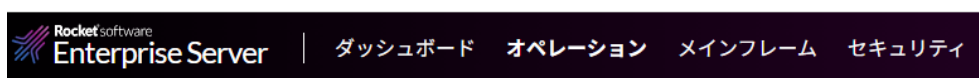
2) リソース定義ファイルの移行

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

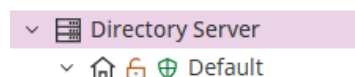
ESCWA からインスタンスを作成します。パスの形式は Windows 環境とは異なり、/(スラッシュ)を使用します。¥(またはバックスラッシュ)は使用できません。複数パスを指定する場合は:(半角コロン)で区切ります。

- ① Web ブラウザで ESCWA を表示し、画面上部の [オペレーション] をクリックします。



- ② Linux マシンの mfds 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)



Directory Server を追加する場合は、[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*

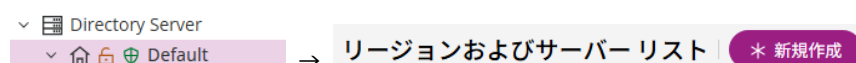
ポート*

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稼働させるために【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*

説明

☒ 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

* 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。

ア...	名前	タ...	ステータス	64ビット	MSS有効	セキュリティ	アクション
	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】へチェックし、【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】

MFACCCGI_CHARSET=Shift_JIS

MFCODESET=9122

リージョンの機能

☒ MSS有効 ?

☒ JES有効 ?

☐ IMS有効 ?

☐ MQ有効 ?

追加設定

構成情報 ?

【ES-Environment】
MFACCCGI_CHARSET=Shift_JIS
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 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) ⓘ リソース定義ファイル パス ⓘ

トランザクション パス ⓘ

ファイル パス ⓘ

マップ パス ⓘ

☐ EZASOCKET サポート ⓘ

- 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。

JESの構成 適用

JES プログラム パス ⓘ システム カタログ ⓘ

データセットの省略時ロケーション ⓘ システム プロシージャ ライブラリ ⓘ

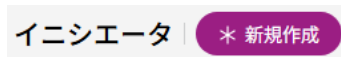
Fileshare 構成ロケーション ⓘ



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 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。



- 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* 

クラス 

説明 

* 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

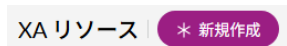
ステータス ステータスの設定  実際のアドレス

Stopped


また、各リスナーの【tcp】の値を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。




プロトコル 
tcp ホスト名またはIP アドレス*  ポート 

- 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。



- 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  <input type="text" value="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 から 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

開始権限を持つユーザーで開始します。

コマンド例) casstart /r<インスタンス名> /u ユーザー名 /p パスワード

```
#casstart /rMSSDEMO /uSYSAD /pSYSAD
..
CASCD0167I ES Daemon successfully auto-started 1
CASCD1005I /var/mfcobol/es/MSSDEMO/console.log 1
CASCD0050I ES "MSSDEMO" initiation is starting 1
```

[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

開始権限を持つユーザーで実行します。

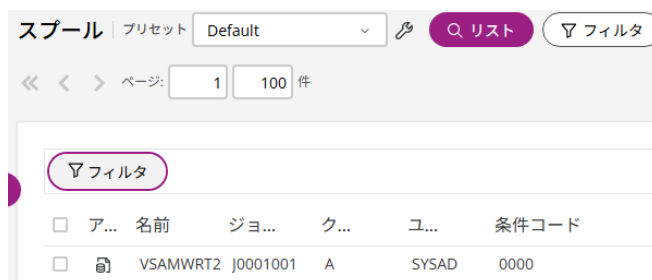
コマンド例) cassub /r<インスタンス名> /j<JCL ファイル名> /u ユーザー名 /p パスワード

```
#cassub /rMSSDEMO /jvsamwrt2.jcl /uSYSAD /pSYSAD
JCLCM0187I J0001000 VSAMWRT2 JOB SUBMITTED (JOBNAME=VSAMWRT2,JOBNUM=0001000)
JCLCM0180I J0001000 VSAMWRT2 Job ready for execution. 09:33:17
Processed "vsamwrt2.jcl"
```

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、画面上部の [JES] プルダウンメニューから [スプール] を選択後、[リスト] ボタンをクリックして JCL の実行結果を確認します。

- ① スプルー一覧が表示され、実行した VSAMWRT2 ジョブが表示されています。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。



- ② VSAMWRT2 をダブルクリックするとメッセージが表示されます。

ジョブ: J0001001

適用

保留

削除

一般

メッセージ

DDエントリ

ジョブ ステップ

メッセージ

JCLCM0188I J0001001 VSAMWRT2 JOB STARTED 09:45:23

JCLCM0182I J0001001 VSAMWRT2 JOB ENDED - COND CODE 0000 09:45:23

DD エントリタブを選択すると一覧が表示されます。

一般 メッセージ DDエントリ ジョブ ステップ

DDエントリ

▼フィルタ

ア...	状態	ク...	DD名	ステップ	ステップ番号	PROCステップ	レコード数	アクション
	Hold	A	JESYSMSG		0		83	
	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	

- ③ 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。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。

表示 < < > > ページ: 1 / 1 83 行 / 83 コードセット ☐ 詳細

```
*****  
**-* Rocket Software ESJCL ASCII JES2 Version ED11.0_PU2B **-*  
**-* Copyright (C) Rocket Software 1997-2024. All rights reserved. **-*  
**-* Job: 0001001 Name: VSAMWRT2 User: SYSAD Date: 01/14/26 Time: 09:45:23 **-*  
**-* File: $TXRFDIR/t000000024.t **-*  
**-* DSN: **-*  
*****  
  
1 //VSAMWRT2 JOB CLASS=A,MSGCLASS=A  
2 //*****  
3 //DEFVSAM1 EXEC PGM=IDCAMS  
4 //SYSPRINT DD SYSOUT=*  
5 //SYSIN DD *  
6 //SORTSTEP EXEC SORTD  
**** JCLCM030I Procedure SORTD was expanded using the library SYS1.PROCLIB  
    XXSORTD PROC  
    XXSORT1 EXEC PGM=SORT  
15 //SORT1.SORTIN DD *  
27 //SORT1.SORTOUT DD DSN=*&1IN1DAT DISP=(NEW,PASS)
```

- ④ 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。
- ⑤ この JCL 実行によってカタログされたデータセットを確認します。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 ファイルの JINJI.KSDS がカタログされていることがわかります。

カタログ | 🔍 リスト | 🗑️ フィルタ | 🗑️ 削除 | * 新規作成

◀ < > ▶ ページ: 1 100 個のエントリ

🗑️ フィルタ

<input type="checkbox"/>	ア...	DS編成	DS名	アクション
<input type="checkbox"/>	📄	VSAM	JINJI.KSDS	👁️ 🔧 🗑️
<input type="checkbox"/>	📄	PO	SYS1.PROCLIB	👁️ 🔧 🗑️

- ⑥ 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。

<input type="checkbox"/>	ア...	DS編成	DS名	アクション
<input type="checkbox"/>	📄	VSAM	JINJI.KSDS	👁️ 🔧 🗑️
<input type="checkbox"/>	📄	PO	SYS1.PROCLIB	👁️ 🔧 🗑️ DCB

JINJI.KSDS | 適用 | コピー | リネーム | 🗑️ 削除

* 入力必須の項目です

DS名 JINJI.KSDS 🔗 カタログ式

物理ファイル*
 /home/tarot/mf/MSSDEMO/DATAFILE/JINJI.KSDS.DAT

DS編成 🔗 VSAM コードセット 🔗 ASCII LRECL 🔗 71 バイト BLKSIZE 🔗 0 バイト

- ⑦ カタログ一覧へ戻り [JINJI.KSDS] の [表示] アイコンをクリックするとデータセットの内容が表示されます。

<input type="checkbox"/>	ア...	DS編成	DS名	アクション
<input type="checkbox"/>	📄	VSAM	JINJI.KSDS	👁️ 🔧 🗑️
<input type="checkbox"/>	📄	PO	SYS1.PROCLIB	👁️ 🔧 🗑️ 表示

表示 | ◀ < > ▶ ページ: 1 1000 行 コードセット 🔗 A... 🔗 ☐ 詳細 ☐ ページ方向を反転

JINJI.KSDS

```

00001Soseki Natsume      1-1,Koishikawa,Bunkyo-ku,Tokyo-to      1886
00002Ryotaro Shiba      2-3,Sonezaki,Kita-ku,Osaka-shi,Osaka-fu  1900
00003Hideyo Noguchi     5-1,Inawashiro,Aizu-shi,Fukushima-ken   1911
00004Osamu Dazai        2-6,Tsugaru,Tsugaru-gun,Aomori-ken     1911
00005Eiji Yoshikawa     9-3,Miyatomomura,Mimasaka-gun,Okayama-ken 1920
00006Jirocho Shimizu    6-6,Jiro-cho,Shimizu-shi,Shizuoka-ken   1800
00007Gai Mori          3-1,Rintaro-cho,Tsuwano-shi,Shimane-ken  1886
00008Ryoma Sakamoto     1-1,Harimayabashi,Kochi-shi,Kochi-ken   1820
00009Shiki Masaoka      5-5,Dogo Onsen,Matsuyama-shi,Ehime-ken   1870
00010Yukichi Fukuzawa   8-8,Keio-cho,Nakatsu-shi,Oita-ken      1835
***END OF FILE***

```

2.11 CICS トランザクションの実行

- 1) お使いの TN3270 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 ポートに接続します。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。



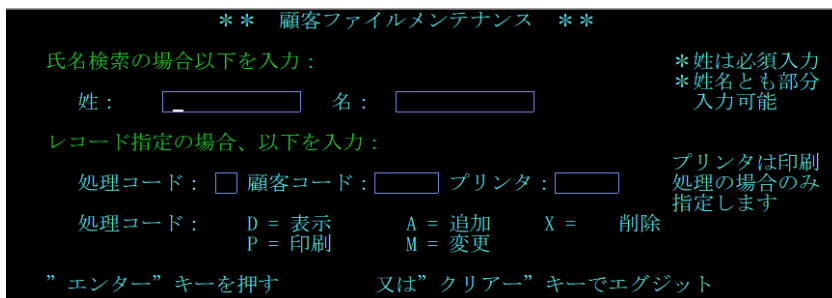
- 2) デフォルトでは初期トランザクションに CESN が設定されているため CICS サインオン画面が表示されます。ユーザー ID に SYSAD を、パスワードに自身で設定したパスワードを入力してサインオンすると下記の画面が表示されます。



- 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。

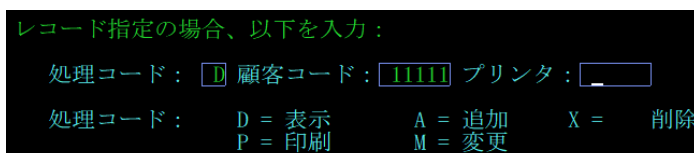
ACCT_

- 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 名: 式部 MI: G 敬称: MRS
 住所: 4-3、石山寺
 大津市
 滋賀県

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、TN3270 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、停止権限のあるユーザーで停止します。

コマンド例)casstop /rMSSDEMO /u ユーザー名 /p パスワード

```
#casstop /rMSSDEMO /uSYSAD /pSYSAD
CASST0005I Shutdown of ES MSSDEMO starting
Return code: 0
```

2.13 デバッグについて

アニメータ(対話式デバッグ)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、こちらをご参照ください。

- 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。