Micro Focus メインフレームソリューション

スターターズキット

5. Enterprise Developer JCL チュートリアル

5.1 チュートリアルの準備

本チュートリアルを実行する前に「4. Enterprise Developer CICS チュートリアル」を終了している必要があります。ここで使用する例題プログラムは、CICS チュートリアルとともに Tutorials.zip に圧縮されており、すでに C:¥Tutorials の下に解凍されており、MSSDEMO プロジェクトに追加されています。

5.2 JES リージョンの構成

本チュートリアルでは例題 JCL を Enterprise Server にサブミットして実行します。ここでは CICS チュ ートリアルで使用した MSSDEMO リージョンをそのまま使用しますので、JES を使用可能とするため の準備を行います。

- 1) MSSDEMO が開始状態の場合にはいったん停止します。
- CICS チュートリアルで作成された MSSDEMO プロジェクトのフォルダ
 C:¥work¥MSSDEMO¥MSSDEMO下に DATAFILE フォルダを作成しておきます。



3) Enterprise Server Admin にて MSSDEMO の [編集] ボタンをクリックします。

編集	MSSDEMO	停止 開始	1 top:*:*	3リスナー	i¥≆⊞ (/ Default	Server: CP 1: CASST00011

4) [サーバー...] > [プロパティ...] > [MSS...] > [JES...] > [General] タブを開きます。以下のように
 [Job Entry Subsystem enabled] のチェックをオンにします。

<mark>フロパティ</mark> 構成 診断 過去の統計	
般 XAリソース (0) MSS (✔) スクリプト アクセス権	セキュリティ
ainframe Subsystem Support enabled: 🗹	
CICS (🗸) JES (🖌) IMS PL/I	
General Initiators (0) Printers (0)	
Job Entry Subsystem enabled:	

5) 以下のように各フィールドに入力し、[Apply] ボタンをクリックします。

ここで、[JES プログラムパス] はジョブステップで実行される COBOL アプリケーションの探索先パスで すので、開発プロジェクトの bin ディレクトリを指定しています。

[システムカタログ] は、JES リージョンで仮定されるマスターカタログの置き場所です。

[データセットの省略時ロケーション] はジョブの実行ともに生成されるスプールデータやカタログされる データセットの置き場所です。

[システムプロシージャライブラリ]は、ジョブの実行時に使用されるプロシージャライブラリの名前です。

なお、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。

JESプログラムパス・	
C:\work\MSSDEMO\MSSDEMO\New_Configuration.bin	*
	4
<u>システム力々ログ・</u>	
C:\work\MSSDEMO\MSSDEMO\DATAFILE\CATALOG.dat	*
	*
データセットの省略時ロケーション	
C:\work\MSSDEMO\MSSDEMO\DATAFILE	*
	*
<u>シュテムプロシージ</u> ャライブラリ:	
SYS1.PROCLIB	
Fileshare Configuration Location:	
	*
	*
Арру	

6) 「JES...」> [Initiators] タブを開き、[追加] をクリックします。

-ภ-	- <u></u> ע	ナー <mark>(</mark> 3)	サ <i>ー</i> ビス (4)	ハンドラ (3)	パッケージ (0)]
プロパ	ティ │ キ	構成 】 詞	診断…) 過去の	統計		
一般]	XAリソーフ	τ (0))	MSS (✔)	スクリプト	アクセス権	セキュリティ
Mainfra	ame Subsy	stem Sur	oport enabled:	2		
CIC	s(*) J	ES (₹) MS I			
一般	t Initia	tors (0)	Printers (0)			

7) 以下のように入力し [追加] ボタンをクリックします。

一般	Initiators (0)	Printers (0)
Add	Initiator	
名前:		_
INITAB	C	
Class:		
ABC		
説明:		
257 A	ABCのイニシエータ	-
キャン	セル 追加]

8) 以下のようにジョブクラス A, B, C に対する JES イニシエータが定義されます。

一般 】	Initiate	ors (1)	Printers (0)
	名前	クラス	說明
編集	INITABC	ABC	クラスABCのイニシェータ

以上で JES リージョンの設定は完了しました。

5.3 簡単な JCL の実行

まずもっとも簡単な JCL をこの JES リージョンにサブミットして実行してみます。

- 1) MSSDEMO を再起動します。
- 2) プロジェクト内の COBOL エクスプローラで copy1.jcl をダブルクリックし、エディタでその内容を確認します。

COB X 월 카~	📑サーバー: ローカル 💿 copylici 🗙
 ■ ● MSSDEMO ● ● COBOL プログラム ● ● DC-ファイル ● ● BMS ソースファイル ● ● BMS ソースファイル ● ● copy1jcl ● vsamwrt2jcl ● ● New_Configuration bin 	<pre>//COPY1 JOB MSGCLASS=A //S1 EXEC PGM=IEBGENER //SYSUT1 DD * 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,Miyamotomura,Mimasaka-gun,Okayama-ken 1920 000065Jirocho Shimizu 6-6,Jiro-cho,Shimizu-shi,Shizuoka-ken 1800 00007Ogai 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 0001Yukichi Fukuzawa 8-8,Keio-cho,Nakatsu-shi,Oita-ken 1835 /* //SYSIN DD DUMMY</pre>

- 3) このジョブは IEBGENER ユーティリティを起動して JCL 内に書かれたインラインデータを SYSOUT に書き出しているだけのものです。
- 4) COBOLエクスプローラ内で copy1.jcl を右クリックし「Enterprise Server へのサブミット」を選択します。



5) コンソールにジョブがサブミットされたことを示すメッセージが表示されますので、ジョブ番号をクリ



6) 以下のようにスプールビューが開きます。

DB01039	Name:	COPY1	Sta	tus: Output Hold		
Release	Class:	Δ	Prio	rity: 00		
Jpdate	User: a	JESUSER	co	ND: 00000		
CLCM0188 CLCM0182	I JOB01039 I JOB01039 Status C	O COPY1 JOB STAR O COPY1 JOB ENDE Lass DD Name	TED 18:20:50 D - COND CODE 000 Step	0 18:20:51 Nbr.	Proc Step	Records
Details	Hold	A JESYSMSG		0		31
		an meanstraneau	C1	1		4
Details	Ready	A <u>SYSPRINT</u>	51	<u> </u>		

7) JESYSMSG をクリックすると以下のように COPY1 ジョブのジョブログが表示されます。



おクリックし[前に戻る]を選択し、SYSPRINT をクリックすると以下のように COPY1 ジョブから起動された IEBGENER ユーティリティの実行ログが記録されていることが確認できます。



9) 前画面に戻り、SYSUT2 をクリックします。以下のように出力されたスプールの内容が表示されま

す。

ヨサーバー: ローカル (copy 1 jcl	🔜 サーバー ローカル/MSSDEMO Job 🗙	- 8
Content-Type: text/plain		<u> </u>
00001Soseki Natsume 00002Ryotaro Shiba 00003Hideyo Noguchi 00004Osamu Dazai 00005Eiji Yoshikawa 00006Jirocho Shimizu 00007Ogai Mori 00008Ryoma Sakamoto 00009Shiki Masaoka 00010Yukichi Fukuzawa	1-1,Koishikawa,Bunkyo-ku,Tokyo-to 2-3,Sonezaki,Kita-ku,Osaka-shi,Osaka-fu 5-1,Inawashiro,Aizu-shi,Fukushima-ken 2-6,Tsugaru,Tsugaru-gun,Aomori-ken 9-3,Miyamotomura,Mimasaka-gun,Okayama-ken 6-6,Jiro-cho,Shimizu-shi,Shizuoka-ken 3-1,Rintaro-cho,Tsuwano-shi,Shimane-ken 1-1,Harimayabashi,Kochi-shi.Kochi-ken 5-5,Dogo Onsen,Matsuyama-shi,Ehime-ken 8-8,Keio-cho,Nakatsu-shi,Oita-ken	1886 1900 1911 1911 1920 1800 1886 1820 1870 1835

以上で簡単なジョブの実行が確認できました。

5.4 プロシージャライブラリの作成

本チュートリアルでは使用する例題 JCL ではプロシージャを使用しています。Enterprise Server では ジョブプロシージャはパーティションドデータセットのメンバーとして配置します。このためまずプロシー ジャライブラリを作成し、プロシージャを配備しておきます。

1) Enterprise Server 管理コンソールで MSSDEMO の [詳細] をクリックします。

新集 MSSDEMO 間的 「第一日」 「第一日 「第一日 「第一日」 「第一日 「第一日」 「第一日 「第一日」 「第一日 「第一日」 「第一日 「第一日」 「第一日 「第一日 「第一日 「第一日 「第一日 「 「 「 「 「 「 「 「 「 「 「 「 「	ver: 1: DS38011 :11:45 1 ninutes ` Started e under s
---	--

2) 以下の[ES モニター&コントロール] ボタンをクリックします。

▲ I Server MSSDEMO [開始 ✓]										
<mark>サーバー…</mark> リスナー (3) サービス (4) ハンドラ (3) パッケージ (0) [
ブロバティ コントロール 診断 過去の統計										
ESモニター&コントロール										
Count Type PID TR Count State Executing Time Time Duration										
	1	Normal	3936	9	Idle					
	2	Normal	2448	1	Idle					

3) ESMAC 画面が表示されますので、左下の [Resources] プルダウンで [JES] を選択します。

	0	Server: N Host: 192 Address:	ISSDEMO .168.17.139 192.168.17.139		User ID: mfuser Group: mfuser Time: 12/19/2013 18:52:30
Home		Server Informa	tion	Ref	resh Interval (Secs)
Server Monitor 1 15	Times:	Start: Elapsed:	12/19/20	013 - 18:11:43 0 - 0:40:47	Number: 3
Control SEPs Clients	Counts:	Transactions: Per Hour: Active Clients:	237 348 1	Dumps Trace Blocks: Limit:	0 0 10
Diagnostics C Log C A C Dump C B C Trace C C/x	Sizes:	Max Tasks: Diagnostics Size: Shared Memory: SM Cushion:	2 (admin:1) 4190208 k 512 pages (4k) 131 k	HTTP Out ID Timeout SM Total SM Free	63 k segments 0 (minutes) 2,048 k 1,809 k (3)
100 Blods Display Resources Services	States:	Working mode: Force Phase In: Dump on ABEND: Active trace:	32 Yes Sys:Yes Tran:No api, kcp	Perf Enabled: Dump: Trace:	No A A
Active by Group by Start L by Type Users	Startup:	SIT: SYSID: JES/JCL: Protected TS/TD: Fileshare User:	DBCS \$IVP Yes Warm/Warm	PLT-PI: PLT-SD: IMS: unP. TS/TD:	No Warm/Warm

4) [Catalog] ボタンをクリックすると右側ペインに以下のようなカタログビューが表示されます。ここで [List]ボタンをクリックします。

Home	Data CATALOG	Refresh Interval (Secs)
Server Monitor 1 15	List *	Cataloged Only
Control SEPs	Data CATALOG	Refresh
Clients Diagnostics C Log C A C Dump C B C Trace C C/x 100 Blocks Display Resources JES	casrdo45: p 573947	
Catalog		

5) 現在カタログされているデータセットは何もありません。そこで [New] ボタンをクリックします。

	Data CATA	LOG		Refresh Interval (Secs)
List	*		Cataloge	ed Only
	New	Details	Delete	
DS O	rg DS Name			
	Data CATA	LOG		Refresh

 6) 以下のカタログエントリの新規作成ダイアログが現れます。以下のように入力して [Apply] をクリ ックします。「PO」はパーティションドデータセットであることを示します。また、この PO が拡張
 子.PRC のテキストファイルをフォルダ配下に保持する動的 PDS であることを指定しています。

CAT	ALOG Entry		Refresh	Interval (Secs)
Apply	Сору	Delete		
DS Name:	YS1.PROCLIE			Catalog
Physical File:	:¥work¥MSSDEMO¥N	MSSDEMO¥DATAFILI	E¥SYS1.PROCLI	B¥
DS Org	°0 💽	RECFM	LSEQ 🔽	
Codeset: A	ASCII 🔽	Created	2013/12/20 12:	:01:45.49
LRECL:	0000	Referenced	2013/12/20 12:	:01:45.49
BLKSIZE:	0000			
E C	Dynamic PDS	PDS Exts	PRC	
Display St	art: 1 for	10000 Codeset		Details
CAT	ALOG Entry		Refresh	

7) 以下のようにカタログエントリ SYS1.PROCLIB が作成されました。

	Data CATA	LOG	Ì	Refresh	Interval (Secs)
List *	ĸ		Cataloged	d Only	
	New	Details	Delete		
DS Or	rg DS Name				
П РО	SYS	1.PROCLIB			DCB
DS Or	rg DS Name				
	Data CATA	LOG		Refresh	

8) 指定した物理フォルダ C:¥work¥MSSDEMO¥MSSDEMO¥DATAFILE¥SYS1.PROCLIB を Windows エクスプローラで作成し、その下に C:¥Tutorials にある SORTD.prc をコピーします。

🔒 DATAFILE				_ _ ×
GO⊽ 📕 • work • I	MSSDEMO + MSSDEMO + DATAFILE +	👻 🔯 🛛 DATAFIL	Eの検索	<u></u>
整理 👻 🥽 開く うイ:	ブラリに追加 🔻 共有 👻 新しいフォルダー			:= 👻 🔟 📀
🚺 ダウンロード 🖉	」名前 ▲	更新日	時	種類
■ デスクトップ	SYS1.PROCLIB	2013/1	2/20 15:52	ファイル フォルダー
12	📄 casspool.dat	2013/1	2/20 10:28	DAT ファイル
🍃 ライブラリ	CATALOG.DAT	2013/1	2/20 10:27	DAT ファイル
SYS1.PROCLIB				_ 🗆 ×
	MO + DATAFILE + SYS1.PROCLIB	👻 🔯 SYS1.PRO	OCLIBの検索	
整理 ▼ ライブラリに追加	▼ 共有 ▼ 新しいフォルダー			🖽 👻 🛄 🔞
👉 お気に入り	名前 ▲	更新日時	種類	サイズ
ダウンロード	SORTD prc	2012/05/23 16:49	PRC 77	าน

9) カタログビューで SYS1.PROCLIB をクリックします。

	Data CATA	LOG		Refresh	Interval (Secs)
List	*		Cataloged	Only	
	New	Details	Delete		
	Org DS Name				
P	0 <u>SYS</u>	1.PROCLIB			DCB
	Org DS Name				
	Data CATA	LOG		Refresh	

10) 以下のようにメンバー SYS1.PROCLIB(SORTD) が登録されたことが確認できます。

	Data CATA	LOG		Refresh	Interval (Secs)
List	ĸ		Cataloged	Only	
	New	Details	Delete		
DS O	rg DS Name				
Г РО	SYST	I.PROCLIB			DCB
D PD	SM SYS1	I.PROCLIB(SOR	TD)		DCB
DS O	rg DS Name				
	Data CATA	LOG		Refresh]

11) SYS1.PROCLIB(SORTD) をクリックすると以下のようにその内容を表示させることができます。

САТА	LOG Entry	Refresh	<
Content-Type: tex	t/plain		
//SORTD PROC //SORT1 EXEC PGI //SYSOUT=* //SORTWK01 DD UI //SORTWK02 DD UI // PEND	M=SORT NIT=SYSDA,SPACE=(CYL,(10,10)) NIT=SYSDA,SPACE=(CYL,(10,10))		

5.5 バッチプログラムのコンパイル

本チュートリアルで使用する COBOL バッチプログラム KSDSWRT2.cbl は既に MSSDEMO プロジェクトに追加されています。しかし、JCL から実行されるためにはメインフレーム方言にてコンパイルする必要があるため、設定を追加して再コンパイルしておきます。

 MSSDEMO プロジェクトの COBOL エクスプローラ内で KSDSWRT2.cbl を右クリックし [プロパ ティー] を選択します。



2) 以下のダイアログにてプロジェクト内の個別のソースファイルに対する設定を行うことができます。
 ここでの設定はプロジェクトに対する設定より優先して採用されます。以下のように設定します。

<u>₩</u> プロパティー: KSDSWRT2.cbl			_ 🗆 🗙
フィルター入力	COBOL		(⇒
ー・リソース 田・COBOL 上・実行/デバッグ設定	マファイルの固有な設定を可能に	93(F)	
	文字集合: 言語方言: ソース フォーマット: メインフレームのコピー処理: □ 指令ファイルの生成 □ リストファイルを生成 □ デバッグ用にコンパイル(D) □ 出力の表示 □ GNT (こコンパイル	ASCII Enterprise COBOL for z/OS 固定 COPY	
	警告レベル: 最大エラー数: 追加指令: DIALECT(ENTCOBOL)	回復可能なエラーを含める(レベル E) <u>・</u> 100]
		デフォルトの復元	 (D) /適用(A)
?		ОК	キャンセル

3) 同様に CICS プリプロセッサも不要ですので以下のように外します。



4) [OK] をクリックすると自動ビルドが実行され、以下のように再コンパイルが完了します。



5.6 COBOL バッチプログラムの実行

COBOL プログラムを含むより実践的なジョブを実行してみます。

1) COBOL エクスプローラ内で vsamwrt2.jcl をダブルクリックしエディタで開きます。

ೀcob 🕱 ९६ ナ೮೫ _5. +	<u> サーバー: ローカル</u> 🚺 vsam	wrt2jcl 🗙		
COB Cob	<pre> th -/i -: D -thi</pre>	<pre>ww(2;cl X MSGCLASS=A MSGCLASS=A MGE (JINJI.KSDS)) - ACE(20 10) KEYS(5 0)</pre>	1870 1886 1886 1900 1920 1911 1820 1911	
	// SORTI.SURTOF DD DD = // SPACE=(800,(10,10)), //SORTI.SYSIN DD * SORT FIELDS=(1,5,CH,A	DCB=(RECFM=FB,LRECL=71,DSORG=PS),UNIT=SYSDA		

このジョブは4つのステップから構成され下記の様に連携されています。

• STEP1: DEFVSAM1

IDCAMS を使用して KSDS クラスター"JINJI.KSDS"を削除し再作成

• STEP2: SORTSTEP

事前に登録したカタログ式プロセージャ SORTD を使用して JINJI.KSDS ファイルへの書き込み用デ ータをソート

• STEP3: APPL1

アプリケーション KSDSWRT2 を呼び出し STEP2 でソートされたデータを使って STEP1 で定義された VSAM クラスターに書き込みを行う。同時に書き込まれたデータを DD=PRINTER へ出力

- STEP4: 書き込まれた内容の確認のため AMS の REPRO で内容を出力
- COBOL エクスプローラ内で vsamwrt2.jcl を右クリックして [Enterprise Server へのサブミット] を選択します。



3) 以下のように VSAMWRT2ジョブが実行された旨のメッセージがコンソールに表示されますので、

ジョブ番号をクリックします。



4) ジョブの実行結果を確認します。実行された VSAMWRT2 のジョブ結果で COND: 00008 が出ています。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、今回が最初の実行であるため、そのファイルが存在していなかった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。

JOB01001	Name	e: VSA	AMWRT2	Status:	Output Hold		
Release	Class	a: A		Priority:	00		
Update	Use	r: JES	USER	COND: (80008		
JCLCM018 JCLCM018	8I JOB01 2I JOB01	001 V 001 V	SAMWRT2 JOB STA SAMWRT2 JOB EN	ARTED 18:08:38 DED - COND CODE 0008	8 18:08:38		
	Status	Class	DD Name	Step	Nbr.	Proc Step	Record
Details	Hold	A	JESYSMSG		0		76
Details	Ready	A	SYSPRINT	DEFVSAM1	1		18
Details	Ready	A	SYSOUT	SORTSTEP	2	SORT1	12
Details	Ready	A	SYSOUT	APPL1	3		1
Details	Ready	A	PRINTER	APPL1	3		10

5) 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。



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。



7) 以下、順番に内容を確認します。最初は IDCAMS の SYSPRINT です。

🧾サーバー: ローカル 💿 vsamwrt2.jcl 🛛 🔚 サーバー ローカル/MSSDEMO Job 🗙	- 0
Content-Type: text/plain	A
Micro Focus MFJAMS Utility Version ED22.00.00_022 Copyright (C) 1997-2013 Micro Focus. All rights reserved.	
DELETE JINJI.KSDS PURGE	
JCLAM0115E(08) - ENTRYNAME NOT CATALOGED [JINJI.KSDS]	
SET LASTCC=0 JCLAM0140I(00) - LASTCC set to 0.	
DEFINE CLUSTER (NAME(JINJI.KSDS)) - DATA - (RECORDS(10) FREESPACE(20 10) KEYS(5 0) - RECORDSIZE(71 71)) - INDEX (RECORDS(50 50)) JCLAMO113I(00) - ENTRYNAME DEFINED [JINJI.KSDS]	

8) 次は SORTD プロシージャ経由で起動される SORT ステップの SYSOUT です。

黒サーバー:ローカル	- 0)
Content-Type: text/plain	<u>*</u>
Micro Focus MFJSORT ユーティリティ 3.0.00	
SORT FIELDS=(1,5,CH,A)	
SORT204I: ************************************	
使用レコード 10件 SORT2061: OUTPUT ファイル 'SORTOUT'	
使用レコード 10 件 出力レコード 10 件	
SORT399I: Micro Focus MFJSORT ユーティリティ終了	

9) 次は COBOL プログラム KSDSWRT2 の DISPLAY 文による出力です。



10) 次は COBOL プログラム KSDSWRT2 の PRINTER 出力です。

🧾サーバー: ローカル 🛛 💽 vsamwrt:	2jcl 📃サーバーローカル/MSSDEMO Job 🗙	C	
Content-Type: text/plain			
00001Soseki Natsume 00002Ryotaro Shiba 00003Hideyo Noguchi 00004Osamu Dazai 00005Eiji Yoshikawa 00006Jirocho Shimizu 00006Jirocho Shimizu 00007Ogai Mori 00008Ryoma Sakamoto 00009Shiki Masaoka 00010Yukichi Fukuzawa	1-1,Koishikawa,Bunkyo-ku,Tokyo-to 2-3,Sonezaki,Kita-ku,Osaka-shi,Osaka-fu 5-1,Inawashiro,Aizu-shi,Fukushima-ken 2-6,Tsugaru,Tsugaru-gun,Aomori-ken 9-3,Miyamotomura,Mimasaka-gun,Okayama-ken 6-6,Jiro-cho,Shimizu-shi,Shizuoka-ken 3-1,Rintaro-cho,Tsuwano-shi,Shimane-ken 1-1,Harimayabashi,Kochi-shi,Kochi-ken 5-5,Dogo Onsen,Matsuyama-shi,Ehime-ken 8-8,Keio-cho,Nakatsu-shi,Oita-ken	1886 1900 1911 1911 1920 1800 1886 1820 1870 1835	

11) 次は最後の IDCAMS のステップの SYSPRINT です。

🛃サーバー: ローカル 🛛 💽 vsamwrt2j	cl 🔄 サーバー ローカル/MSSDEMO Job 🗙		
Content-Type: text/plain			-
Micro Focus MFJAMS Utility Version ED22.00.00_022 Copyright (C) 1997-2013 Micro Focus. All rights reserved.			
REPRO INDATASET(JINJI.KSDS) - OUTFILE(SYSPRINT)			
LISTING OF DATASET - JI	NJI.KSDS		
KEY OF RECORD - 00001 00001Soseki Natsume	1-1,Koishikawa,Bunkyo-ku,Tokyo-to	1886	
KEY OF RECORD - 00002 00002Ryotaro Shiba	2-3,Sonezaki,Kita-ku,Osaka-shi,Osaka-fu	1900	
KEY OF RECORD - 00003 00003Hideyo Noguchi	5-1,Inawashiro,Aizu-shi,Fukushima-ken	1911	
KEY OF RECORD - 00004 000040samu Dazai	2-6,Tsugaru,Tsugaru-gun,Aomori-ken	1911	
VEV OF PEOOPD AGAAF			-

12) 続いて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 ESMAC 画面で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。

Home	Data CATALOG	Refresh Interval (Secs)
Server Monitor 1 15	List *	Cataloged Only
Control SEPs		
Clients	PO <u>SYS1.PROCLIB</u>	
© Log C A C Dump C B	Data CATALOG	Refresh
C Trace C/x	casrdo45: p 573947	
Display Resources		
JES 💌 Spool Catalog		

13) 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。

CAT	ALOG Entry		Refresh	<
Apply	Сору	Delete		
DS Name:JI	NJI.RSDS 🔽 Cat	alog		
Physical File:	:¥WORK¥MSSDEMO	¥MSSDEMO¥DATAFI	LE¥JINJI.KSDS.DAT	
DS Org: 🔽	/SAM 🔽	RECFM	KS	
Codeset: A	Codeset: ASCI Created: 2013/12/20 18:08:38.16			38.16
LRECL:	LRECL: 00071 Referenced: 2013/12/20 18:08:38.32			38.32
BLKSIZE: 00000				
VSAM Type: Cluster Key Start/Len: 00000 /00005				
VSAM Attr: Unique Key Max / Avg: 00071 /00005				
Display St	art: 1 for	10000 Codeset	ASCI 🔽 🗖 Det	ails
CAT	ALOG Entry		Refresh	<

14) [Display]ボタン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。

CATALOG E	ntry	Refresh	<
Content-Type: text/plain			
00001SOSEKI NATSUME 00002RYOTARO SHIBA 00003HIDEYO NOGUCHI 00004OSAMU DAZAI 00005EIJI YOSHIKAWA 00006JIROCHO SHIMIZU 00007OGAI MORI 00008RYOMA SAKAMOTO 00009SHIKI MASAOKA 00010YUKICHI FUKUZAWA	1-1,KOISHIKAWA,BUNKYO-KU,T 2-3,SONEZAKI,KITA-KU,OSAKA 5-1,INAWASHIRO,AIZU-SHI,FU 2-6,ISUGARU,TSUGARU-GUN,AC 9-3,MIYAMOTOMURA,MIMASAKA- 6-6,JIRO-CHO,SHIMIZU-SHI,S 3-1,RINTARO-CHO,TSUWANO-SH 1-1,HARIMAYABASHI,KOCHI-SH 5-5,DOGO ONSEN,MATSUYAMA-S 8-8,KEIO-CHO,NAKATSU-SHI,C	OKYO-TO -SHI,OSAKA-FU KUSHIMA-KEN MORI-KEN GUN,OKAYAMA-KEN HIZUOKA-KEN I,SHIMANE-KEN I,KOCHI-KEN HI,EHIME-KEN ITA-KEN	1886 1900 1911 1911 1920 1800 1880 1820 1870 1870
CATALOG E	ntry	Refresh	<

5.7 COBOL バッチプログラムのデバッグ

JES 配下で実行される COBOL プログラムをステップ実行でデバッグすることができます。

1) [実行] > [デバッグの構成...]を選択します。



 以下のダイアログで、COBOL プロジェクトとして MSSDEMO を選択し、[デバッグの種類] の JCL タブでデバッグ対象ジョブ名 VSAMWRT2 を入力します。[デバッグ] ボタンをクリックしま す。

MI デバッグ構成 構成の作成、管理、お Enterprise Server アプリケーショ	メ うよび実行 シへの接続とデバッグ
Correction Correction	名前(N) 新規構成 COBOL プログラムの起動を待機しながら Enterprise Server 上でデバッグセッションを開始します。 ・ COBOL プログラムの起動を待機しながら Enterprise Server 上でデバッグセッションを開始します。 ・ COBOL プログラムの起動を待機しながら Enterprise Server 上でデバッグセッションを開始します。 ・ COBOL プログラム(P) MSSDEMO 参照 ・ Enterprise Server 接続 [ローカル ・ Enterprise Server 接続 [ローカル ・ デバッグの種類 グロン 「MS Web サービス」Java OLCS JOL IMS Web サービス Java JOL 設定(空白の場合はすべての JOL ジョブをデバッグ) ジョブ名: 「VSAMWRT2 ジョブ者号: ジョゴオラ・) 泉上位 ウログラム: 」 適用(Y) 前回保管した状態に戻す(V)
?	デバッグ(D) 閉じる

3) 以下のダイアログに対して [いいえ] をクリックします。



4) デバッガがアタッチの待機状態で起動します。



5) COBOL エクスプローラ内で vsamwrt2.jcl を右クリックし [Enterprise Server へのサブミット] を 選択します。



6) しばらくして以下のダイアログが現れますので、今度は [はい] をクリックします。



7) 以下のようにデバッグセッションが開始し、COBOL プログラム KSDSWRT2 の手続き部の先頭 でハイライトされています。

<mark>III デバッグ - MSSDEMO/KSDSWRT2.cb1 - Eclipse</mark> ファイル(F) 編集(E) ナビゲート(N) Search プロジェクト(P) 実行(R) ウィンドウ(W) ヘルプ(H)	
」 🗈 • 🗄 🐘 📥 🔤 🔅 • 🍳 • 💁 • 🕽 🔌 № 11 🔳] 철 • 월 • ← 🔶 • → •	N 3. 7 . e 5. 1
参デバッグ 🕄 🦛 Servers] 🦓 🚽 🕾 🔅 🎽 🗖	🛙 😡= 変数 🙁 🗣 ブレークポイント
 □ い 新規構成 [COBOL Enterprise Server] □ い COBOL デバッガ: (一時停止) □ い COBOL スレッド2184 (一時停止) 	名前 ◆ INDATA
ー 量 C#work#MSDEMO¥MSSDEMO¥New_Configuration bin¥KSDSWRI2gnt: 許法之(3 〒-19 COBOL スレッド4000(→時有序止) 〒-19 COBOL スレッド3196(一時有序止)	
■ サーバー: ローカル ● vsamwrt2.jcl ■ KSDSWRT2.cbl ※	- D)
KSDSWBT2ch	
••••*A·1·8·•••·2·•••·3·•••4·••5·•••6·•••• 01 FSTAT-K PIC XX. 01 FSTAT-I PIC XX. 01 FSTAT-P PIC XX. 01 FSTAT-P PIC XX. 01 VSTAT-K. 01 VSTAT-K. 05 VSAM-RETURN-CODE 06 VSAM-RETURN-CODE 07 PROCEDURE DIVISION. 00 PEN NUPUT INDATA. 00 PEN OUTPUT INDEXFILE.	··7··I·•···8
OPEN OUTPUT PRTFILE.	-

8) デバッグビュー内の [ステップイン] ボタンまたは F5 キーでプログラムをステップ実行することが できます。



9) ステップ実行を進めてゆくと、以下のように変数ビュー内に更新されたデータ項目の内容が自動的に表示されます。以下は READ 文でレコード内にデータファイルから読み込まれた内容が表示されている様子を示しています。

<mark>四 デバッグ - MSSDMEO/KSDSWRT2.cbl - Eclipse</mark> ファイル(F) 編集(E) ナビゲート(N) Search プロジェクト(P) 実行(R)	ウィンドウ(W) ヘルプ(H)	×
📬 • 🔛 🐘 🕒 🎄 • 🔕 • 🍇 • 😕 🛷 • 🗾	😰 🗉 🖬 🛛 🖢 + 🗟 + 🍫 🔶 • -	 ・
参 デバッグ 🛛 🖓 🗖	🕪= 変数 🙁 📀 ブレークポイント	(in the second s
	名前 ◆ INDATA □ ◆ INREC ◆ INAKEY ◆ INAME1 ◆ INADDR1 ◆ INBKEY ◆ KRFC ◆ KRFC ◆ KRFC	fit Open Input Last status:0/0 00001Soseki Natsume 1-1, 00001 Soseki Natsume 1-1,Koishikawa,Bunkyo-ku,Tok 1886
vsamwrt2.jcl KSDSWRT2.cbl ⊠ ■ サーバー: Local		۲ ۲۰۱۰-۲۰۰۶ ۲۰۰۶ ۲۰۱۰-۲۰۰۶ ۲۰۰۶ ۲۰۰۶
KSDSWBT2cbl		
····*A·1·B······························	···5····•6····•7··I·•··	Data Division
PROC1. PROC1. PERFORM UNTIL LOOP1 READ INDATA AT END SET LOOP1 TO TRUE NOT AT END MOVE INREC TO KREC MOVE INREC TO PREC WRITE PREC WRITE KREC INVALID KEY DISPLAY "INVALID K		P → I NOEXTILE → • KREC GROUP, サイズ: 71 → • AKEY PIC X(5), サイズ → • AME1 PIC X(20), サ- → • ADDR1 PIC X(42), サ- → • BKEY PIC X(4), サイズ ⊕ → INDATA ⊕ → PRTFILE ⊕ → Working-Storage Section ⊕ → Procedure Division

10) 以下のように PROCEND1 段落の CLOSE 文の左端の領域をダブルクリックすると、この行にブ レークポイントが設定されます。



11) [再開] ボタンをクリックするとブレークポイントまで処理が進みます。



以上で JCL チュートリアルを終了します。