



Computer Associates

CA-EasytrievePlus

概説・使用方法及び
サンプル・プログラム集

目次

- **CA-Easytrieve Plusとは** P 3- 7
- **サンプル・プログラム** P 8-30
 - ・明細レポート(べた打ち)
 - ・明細レポート(合計出力)
 - ・サマリーレポート
 - ・複数レポート
 - ・専用帳票
 - ・マッチング処理
 - ・16進リスト作成
 - ・ファイル機能
 - ・テーブルファイル
 - ・サブルーチン・コール
 - ・DB2アクセス
 - ・IMSアクセス1
 - ・IMSアクセス2
 - ・ADABASアクセス
 - ・使用例
 - ・実行JCLサンプル(OS用)

CA-Easytrieve Plusとは

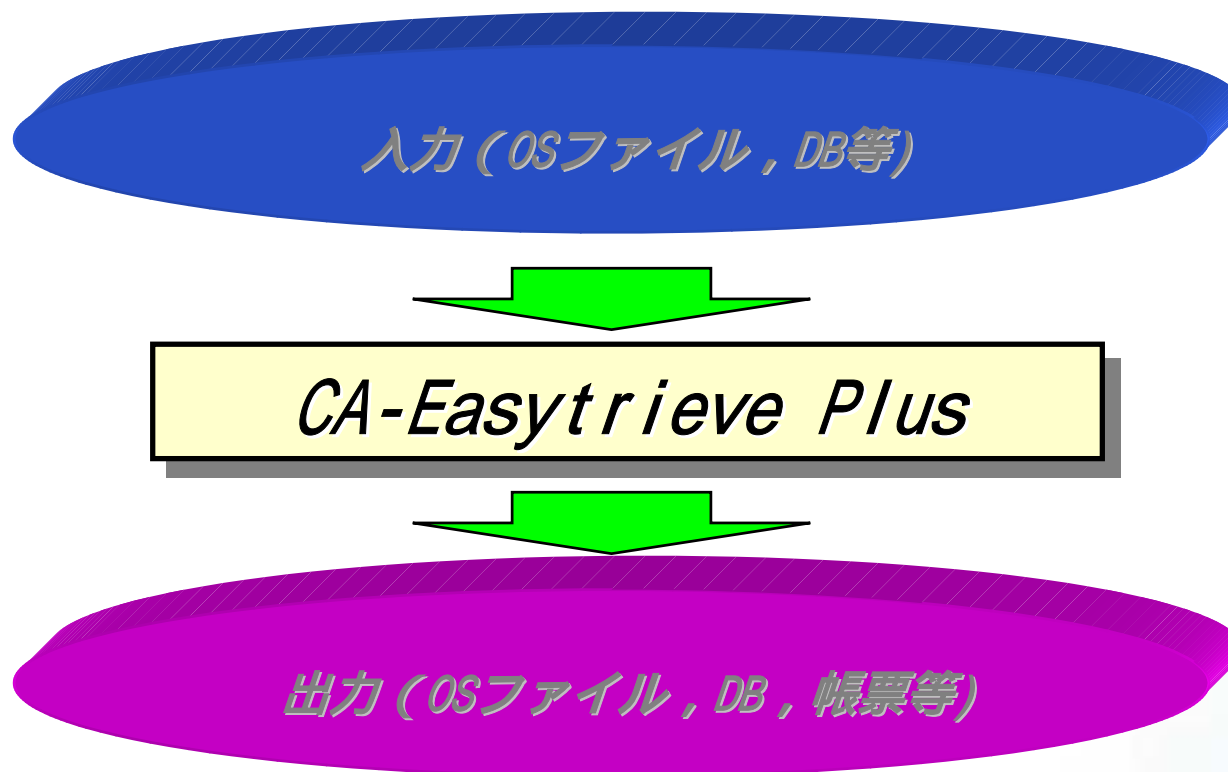
Computer Associates

CA-Easytrieve Plusはシステム開発の生産性を総合的に向上させるために設計された、情報検索 / アプリケーション開発システムです。CA-Easytrieve Plusの最大の特徴は、ロジックの自動化によりプログラムの標準化を実現している事です。これにより、簡単なプログラムから複雑で難易度の高いプログラムまで、シンプルな構造で広範囲に利用する事が可能になっています。CA-Easytrieve Plusは現在、以下のような場面で役立てられています。

- 急なレポート・ニーズが頻繁にある。
- データの整備を短期間で行ないたい。
- 単純な処理ではあるが、メーカー提供のユーティリティでは満足できなく、しかし高級言語 (COBOL, PL/I等) で作成するまでもない。
- 開発工程での検証ツールを統一化したい。
- 全社的なプログラムの標準化を行ないたい。
- メンテナンスが容易なプログラム開発を行ないたい。
- 短期間で習得でき、すぐに使いこなせるツールが欲しい。

CA-Easytrieve Plus稼働環境

CA-Easytrieve Plusはバッチ環境で稼働します



プログラム構成

3つのセクションから構成されています。

ENVIRONMENT

プログラムの環境の設定を行なう。 (*PARM*)

LIBRARY

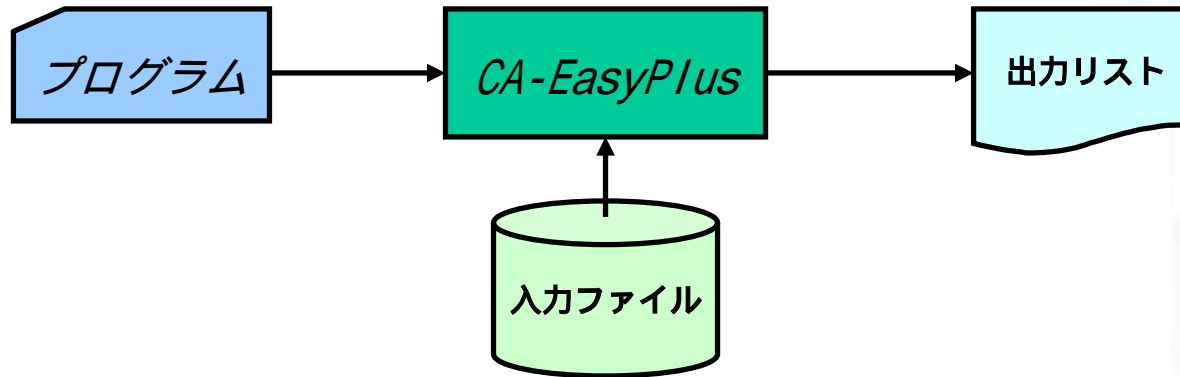
プログラムで使用するファイル・フィールドの定義を行なう。 (*FILE*)

ACTIVITIES

どのような処理を行なうかを記述する。
(*JOB*)
(*SORT*)
(*REPORT*)

プログラムの実行形式

a. インコアコンパイラ方式

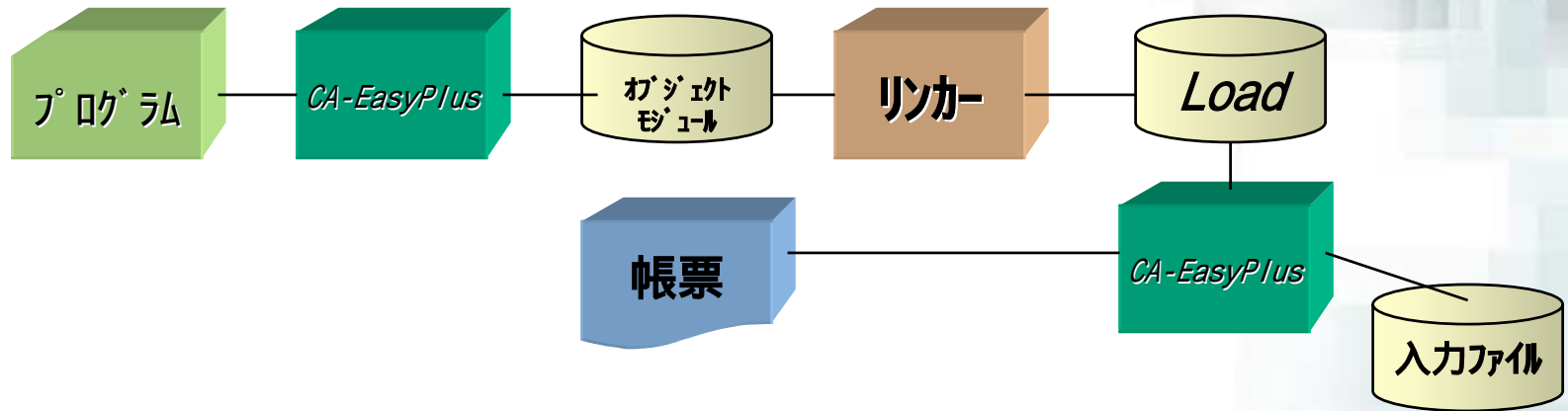


- プログラムを即時、コンパイル・実行
- テスト段階、非定例業務に効果大

(*)実行JCLサンプルの a.を参照。

プログラムの実行形式

b.コンパイラー方式



- コンパイラ言語(COBOL , PL/I)と同様の形態にて実行可能
- 本番運用への適用が容易

(*)実行JCLサンプルの b-1.b-2.を参照。

明細レポート(一覧表)

- 自動入力機能

ファイル入力、終了判定処理はコーディング不要

- 自動フォーマット機能

出力したいタイトル、項目のみを指定

- 自動改頁・改行機能

改行・改頁処理も完全に自動化

```
FILE FILEA
NAME 17 16 A
TEDORI 90 4 P
BANGO 9 5 N
SOSHIKYU 94 4 P
BU 98 3 N
KOJYOGAKU W 5 P
JOB INPUT(FILEA)
  IF BU = 911 THRU 914 921
    KOJYOGAKU = SOSHIKYU - TEDORI
  PRINT REPT1
END-IF
REPORT REPT1 LINESIZE 90
TITLE 1 '販売部門給与一覧表'
TITLE 2 '部門番号(911 912 913 914 921)'
LINE 1  NAME BU BANGO SOSHIKYU TEDORI KOJYOGAKU
```


出力結果 ~ 明細レポート(べた打ち)

- 出力項目を自動的にレイアウトして出力
レイアウトは任意に変更可能です
- 改ページも全て自動
- 見易いレポートを簡単に出力できます

96年01月30日

販売部門給与一覧表
部門番号(911 912 913 914 921)

1頁

NAME	BU	BANGO	SOSHIKYU	TEDORI	KOJYOGAKU
スズキ ミチコ	911	01963	44550	35687	08863
シミズ トシオ	914	11602	34480	25089	09391
ワタナベ ユウコ	911	11357	28392	21547	06845
ハガ ユミ	921	11376	36080	22371	13709
ハハチ スズコ	911	11701	24320	16796	07524
コイズミ サチオ	911	00445	29200	20600	08540
ヤマナカ ケイコ	921	00577	22080	15470	06610
トシヤマ シンタロウ	914	01795	27936	18906	09030
ハラダ トシロウ	911	03571	24240	18209	06031
カウ トシサブロー	912	04225	29520	23050	06470
イノウエ サチ	911	01730	31520	20243	11277
イガラシ ナオミ	911	12829	36560	23804	12756
イクダ ヤスコ	911	12641	31360	21191	09396
ヤマシタ ルミ	911	03890	38640	27253	11387
カサイ ユウジ	914	07231	100400	68523	31877
カキザワ ナオミ	914	08262	37600	21595	16005
ササキ シンコ	911	05805	17415	13403	04012
キタミ シュンコ	911	04589	31360	22969	08391
ヤマカワ スズコ	921	05482	18375	14147	04228
ミザカ ヨウコ	914	09764	12195	9664	02531
マルタニ チエコ	911	10961	39920	29170	10750

明細レポート(合計出力)

- **自動ブレイク機能**
合計を集計したい項目を指定するだけ
数種類の出力方法を標準で搭載
- **レポート中の並べ替え**
並べ替えしたい項目のみの指定するだけ
- **豊富な漢字機能**
漢字出力時のシフトコードの考慮は不要

```
FILE FILEA
NAME 17 16 K HEADING('社員氏名')
TEDORI 90 4 P 1 HEADING('手取額')
BANGO 9 5 N HEADING('社員番号')
SOSHIKYU 94 4 P 1 HEADING('総支給額')
BU 98 3 N HEADING('部')
KOJYOGAKU W 5 P 1 HEADING('控除額')
JOB INPUT(FILEA)
  IF BU = 911 THRU 914 921
    KOJYOGAKU = SOSHIKYU - TEDORI
  PRINT REPT1
END-IF
REPORT REPT1 LINESIZE 90 SUMCTL(TAG)
SEQUENCE BU BANGO
CONTROL BU
TITLE 1 '販売部門給与一覧表'
TITLE 2 '部門番号 (911 912 913 914 921)'
LINE 1 BU NAME BANGO SOSHIKYU TEDORI
      KOJYOGAKU
```

出力結果 ~ 明細レポート(合計出力)

- 合計行、総合計行を自動的に出力
- 並び替え処理もSEQUENCEコマンドを指定するだけ
- 漢字データも簡単に出力できます

96年01月30日 販売部門給与一覧表 1頁
部門番号(911 912 913 914 921)

部	社員氏名	社員番号	総支給額	手取額	控除額
911	小泉 幸雄	00445	2,920.0	2,060.0	854.0
	井上 左知	01730	3,152.0	2,024.3	1,127.7
	鈴木 美智子	01963	4,455.0	3,568.7	886.3
	原田 敏郎	03571	2,424.0	1,820.9	603.1
	山下 留美	03890	3,864.0	2,725.3	1,138.7
	北見 純子	04589	3,136.0	2,296.9	839.1
	佐々木 慎吾	05805	1,741.5	1,340.3	401.2
	丸谷 智恵子	10961	3,992.0	2,917.0	1,075.0
	渡辺 裕子	11357	2,839.2	2,154.7	684.5
	馬場 千須子	11701	2,432.0	1,679.6	752.4
	池田 康子	12641	3,136.0	2,119.1	939.6
	五十嵐 直美	12829	3,656.0	2,380.4	1,275.6
	BU 計		37,747.7	26,087.2	9,577.5
912	勝 富三郎	04225	2,952.0	2,305.0	647.0
	BU 計		2,952.0	2,305.0	647.0
913	富山 晋太郎	01795	2,793.6	1,890.6	903.0
	葛西 勇治	07231	10,040.0	6,852.3	3,187.7
	柿沢 直美	08262	3,760.0	2,159.5	1,600.5
	三笠 洋子	09764	1,219.5	966.4	253.1
	清水 敏夫	11602	3,448.0	2,508.9	939.1
	BU 計		21,261.1	14,377.7	6,883.4
921	山中 圭子	00577	2,208.0	1,547.0	661.0
	山川 静子	05482	1,837.5	1,414.7	422.8
	芳賀 裕美	11376	3,608.0	2,237.1	1,370.9
	BU 計		7,653.5	5,198.8	2,454.7
	総合計		6,9614.3	47,968.7	19,562.6

サマリーレポート

- 自動サマリー機能
集計行のみのレポートも簡単に出力

```

FILE FILEA
CHIKU 1 1 N HEADING('地区' '番号')
SHITEN 2 2 N HEADING('支店' '番号')
TEDORI 90 4 P 1 HEADING('手取り' '額') +
          MASK('¥¥¥,¥¥9.9')
BANGO 9 5 N HEADING('社員' '番号')
SOSHIKYU 94 4 P 1 HEADING('総' '支給' '額') +
          MASK('¥¥¥,¥¥9.9')
BU 98 3 N HEADING('部門' '番号')
KOJYOGAKU W 5 P 1 HEADING('控除' '額') +
          MASK('¥¥¥,¥¥9.9')
JOB INPUT(FILEA)
  IF CHIKU = 3 4 AND SHITEN = 2 3
    KOJYOGAKU = SOSHIKYU - TEDORI
  PRINT REPT1
  END-IF
REPORT REPT1 SUMMARY LINESIZE 90
SEQUENCE CHIKU SHITEN BU
CONTROL CHIKU SHITEN BU
TITLE 1 '販売地区 販売支店別 給与サマリー'
TITLE 2 '* * 地区番号 (3,4) 支店番号 (02,03)'
HEADING TALLY ('件数')
LINE 1 CHIKU SHITEN BU TALLY SOSHIKYU TEDORI
      KOJYOGAKU
  
```

複数レポート

- 複数レポート出力機能

選択されたデータを異なったレポートへ簡単に出力

```

FILE FILEA
NAME 17 16 A HEADING('社員' '氏名')
TEDORI 90 4 P 1 HEADING('手取り' '額') +
      MASK('###,###9.9')
BANGO 9 5 N HEADING('社員' '番号')
SOSHIKYU 94 4 P 1 HEADING('総' '支給' '額') +
      MASK('###,###9.9')
BU 98 3 N HEADING('部門' '番号')
KOJYOGAKU W 5 P 1 HEADING('控除' '額') +
      MASK('###,###9.9')

JOB
  IF BU = 911 THRU 914 921
    KOJYOGAKU = SOSHIKYU - TEDORI
    PRINT REPT1
    PRINT REPT2
  END-IF
*
REPORT REPT1 LINESIZE 90
SEQUENCE BU
CONTROL
TITLE 1 '販売部門 給与一覧表'
TITLE 2 '部門番号 (911 912 913 914 921)'
LINE 1 NAME BU BANGO SOSHIKYU TEDORI KOJYOGAKU
*
REPORT REPT2 SUMMARY LINESIZE 80
SEQUENCE BU
CONTROL BU
TITLE 1 '販売部門別 給与サマリー'
TITLE 3 '部門番号 911,912,913,914,921 のサマリー'
LINE 1 BU SOSHIKYU TEDORI KOJYOGAKU

```

専用帳票 (指定帳票)

- 専用帳票への出力

固定位置への項目データの出力も簡単な指定で可能

```
FILE FILEA
EMPNO 1 5 N
EMPNAMEK * 20 K
EMPNAMEC * 15 A
EMPADDR * 60 M
EMPZIP * 6 A
EMPTEL * 13 A
EMPSEX * 1 N
EMPBORN * 6 N MASK('Z9/99/99')
EMPDEPT * 3 N
EMPAGE * 2 N
EMPSAL * 5 P 0 MASK('###,###,##9')
JOB INPUT(FILEA)
  IF EMPSAL > 250000
    PRINT REPT1
  END-IF
*
REPORT REPT1 NOADJUST LINESIZE 100 PAGESIZE 40 +
  SPACE 0 NOHEADING
LINE 1 COL 12 EMPNO COL 31 EMPNAMEK +9 EMPNAMEC
LINE 2 POS 1 EMPZIP COL 21 EMPADDR
LINE 3 POS 1 EMPTEL COL 40 EMPBORN +10 EMPAGE +
  COL 73 EMPSAL
LINE 4 ''
```

マッチング処理

自動マッチング機能

マッチング処理時の複雑なファイル入出力処理
を完全に自動化(1:1, 1:n)

- ・全マッチング・ファイルの自動入力
- ・マッチング・キーによるキーの大小判定
- ・マッチング・キーに関するキー値の重複判定

プログラム作成者はマッチ・アンマッチ時の処理
のみを作成するのみ

ロジックの標準化

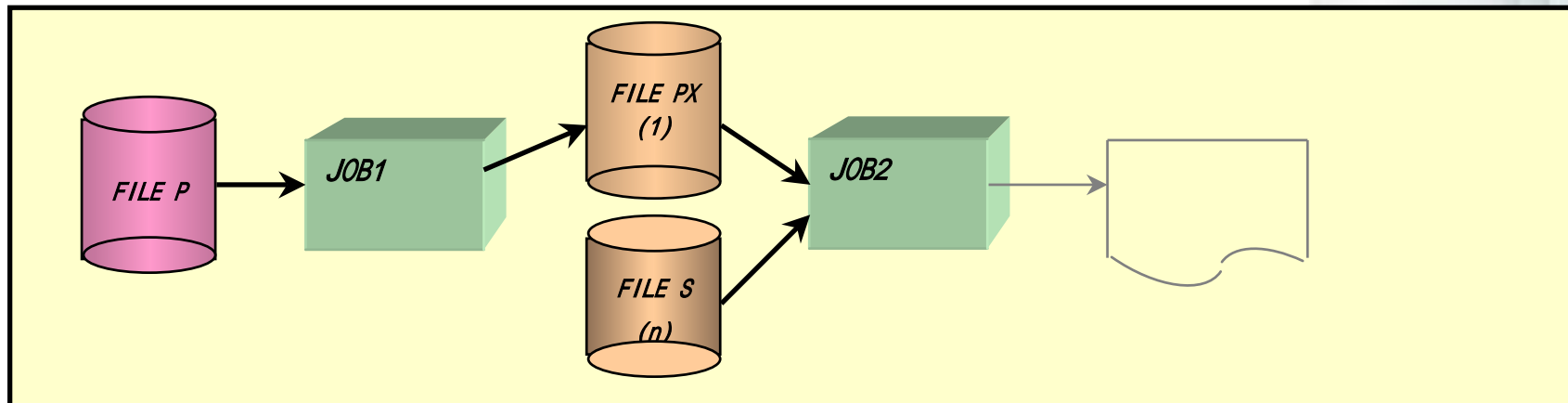
```

FILE JINJI
SYABAN 1 5 N
KIHON 80 4 P 0
TEATE 90 4 P 0
FILE TRANFL
BANGO 1 5 N HEADING('番号')
NEW-KIHON * 4 P 0 HEADING('基本給')
NEW-TEATE * 4 P 0 HEADING('手当')
JOB INPUT(JINJI KEY(SYABAN) +
          TRANFL KEY(BANGO))
  IF MATCHED
    PERFORM MATCH-RTN
  ELSE
    IF JINJI
      PERFORM MASTER-RTN
    ELSE
      PRINT ERROR-LIST
    END-IF
  END-IF
MATCH-RTN. PROC
.....
END-PROC
MASTER-RTN. PROC
.....
END-PROC
REPORT ERROR-LIST LINESIZE 80
TITLE 1 '基本給改定ファイル エラーリスト'
LINE 1 BANGO NEW-KIHON NEW-TEATE

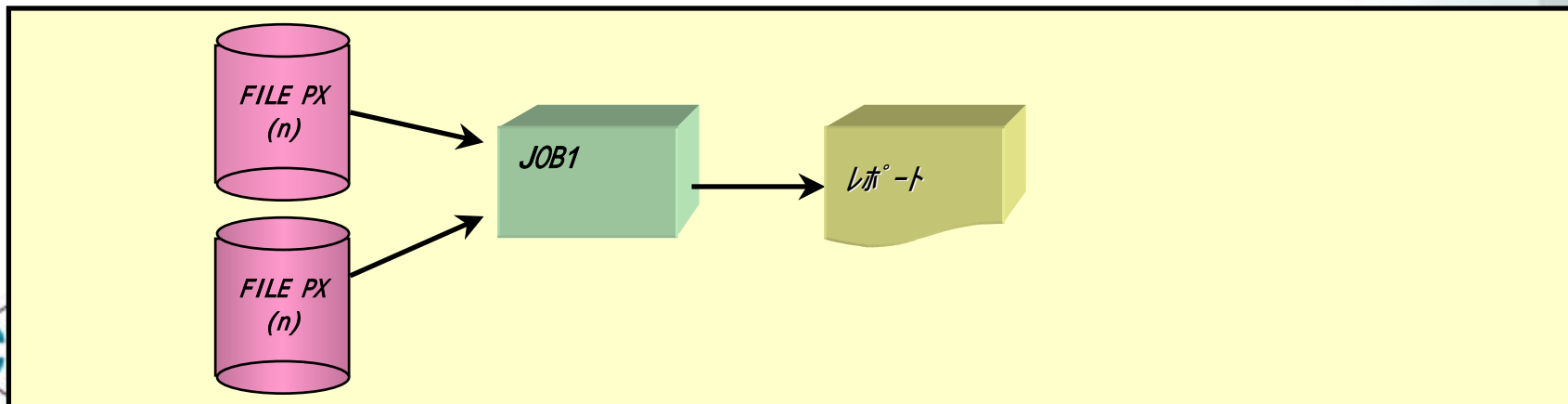
```

マッチング処理 (n:n)

パターン1: 自動機能を有効活用するために、JOBを分割する



パターン2: ユーザ・ロジックやワーク定義を追加して、1つのJOB内で処理する (*1)



マッチング処理(*1)

<入力レコードの判断>

- ・ マッチした場合、セカンダリ・ファイルのレコードを先に読み込む
 - ┌ セカンダリのキー割れが無い場合、プライマリは次レコードを読み込まない
 - └ セカンダリのキー割れが有る場合、プライマリは次レコードを読み込む
- ・ アンマッチの場合、キー値の小さい方のファイルのみ読み込み、キー値の大きい方のファイルは読み込まない

<JOB処理内で参照可能なファイル>

- ・ マッチした場合、セカンダリ・ファイルのレコードを先に読み込む
- ・ アンマッチの場合、キーの大小比較の結果、小さかった方のファイルのレコードのみ参照可能
 - ┌ プライマリの方が小さい場合、プライマリ・レコードのみ参照可能
 - └ セカンダリの方が小さい場合、セカンダリ・レコードのみ参照可能

<注意事項>

プライマリ側の2つ目以降の重複レコードは、アンマッチとして扱われる
 プライマリ側の2つ目以降の重複レコードは、アンマッチレコード処理ロジックで参照できるが、セカンダリ側のレコードは、既に同一キー値を持つレコードではない



16進リスト作成

- 16進リスト作成

フィールド、レコード単位に16進リストを簡単に出力できる

障害時の原因調査、プログラム開発時の検証作業等に威力を発揮する

```
FILE JINJI
NAME 17 12 K
KIHON 80 4 P 0
JOB INPUT(JINJI)
  IF NAME NOT KANJI
    DISPLAY SKIP2 '### レコード カウント ###' +
      JINJI:RECORD-COUNT
    DISPLAY HEX NAME
  END-IF
*
  IF KIHON NOT NUMERIC
    DISPLAY SKIP2 '@@@ レコード カウント @@@' +
      JINJI:RECORD-COUNT
    DISPLAY HEX KIHON
  END-IF
```

出力結果 ~ 16進リスト作成

- 16進リストも項目単位に簡単に出力できる
- DISPLAY ファイル名と指定すれば、レコードの16進リストを出力できる

```
### レコード カウント ### 104  
CHAR  
ZONE 000000000000  
NUMR 000000000000  
1...5...10..
```

```
### レコード カウント ### 105  
CHAR  
ZONE 000000000000  
NUMR 000000000000  
1...5...10..
```

```
@@@ レコード カウント @@@ 231  
CHAR ABCD  
ZONE CCCC  
NUMR 1234  
1...
```

ファイル機能

■ テストデータの作成

ファイル出力機能を応用すれば、テストデータの作成も簡単に行なえます。

中間ファイル(EZTVFM)を使用すれば、JCL上の指定は個々のファイルについては不要。

このプログラムでは、カードファイル(80BYTES)からデータを読み込む度に、320BYTESのデータを1件中間ファイルへ書き出し、最終的には2000件のデータを作成します。

```
FILE FILEA FB(320 3200) VIRTUAL
A-OCC 1 80 A OCCURS 4
A-NUMBER A-OCC 5 N
A-DATA A-OCC +5 75 A
A-ALLDATA 1 320 A
FILE FILEB
B-NUMBER 1 5 N
B-DATA * 320 A
FILE CARD
C-NUMBER 1 5 N
WK-CNT1 W 4 B VALUE 1
WK-CNT2 W 4 B VALUE 1
JOB INPUT(CARD)
DO WHILE WK-CNT1 LE 4
  A-NUMNER(WK-CNT1) = C-NUMBER
  WK-CNT1 = WK-CNT1 + 1
END-DO
DO WHILE WK-CNT2 LE 2000
  PUT FILEA
  WK-CNT2 = WK-CNT2 + 1
END-DO
JOB INPUT(FILEA)
B-NUMBER = FILEA:RECORD-COUNT
B-DATA = A-ALLDATA
PUT FILEB
```

テーブル・ファイル

・ テーブル・ファイル

ARG (引数)..... キーフィールド

DESC (記述)... 検索キーの情報

SEARCH ファイル名 WITH キー項目 GIVING 受取ワークエリア

テーブル・ファイルのファイル編成

- ・SAMファイル
- ・VSAMファイル
- ・ISAMファイル
- ・VFM

テーブル・ファイルの種類

- ・外部テーブル
- ・内部テーブル

```

FILE NOUHIN
S-CODE 5 5 N HEADING('製品' 'コード')
S-RYAKU 10 14 A HEADING('製品' '略称')
H-DATE 24 6 N HEADING('発注日') +
          MASK(D '99/99/99')
N-DATE 30 6 N HEADING('納品日') MASK(D)
H-SURYO 38 2 P 0 HEADING('発注' '数量')
S-TANKA 43 6 P 0 HEADING('製品' '単価')
T-CODE 49 6 A HEADING('取引先' 'コード')
*
FILE KAISHA TABLE
ARG 1 6 A
DESC 7 36 A
*
TABLE-WK W 36 A
SHAMEI TABLE-WK 24 K
TEL-NO TABLE-WK +24 12 A
*
JOB INPUT(NOUHIN)
IF N-DATE ZERO AND H-DATE >= 960430
SEARCH KAISHA WITH T-CODE GIVING TABLE-WK
IF NOT KAISHA
SHAMEI = '該当社名なし'
TEL-NO = ''
END-IF
PRINT REPT1
END-IF
REPORT REPT1
SEQUENCE T-CODE H-DATE
CONTROL T-CODE SHAMEI TEL-NO
TITLE 1 '**** 未納入製品一覧表 ****'
TITLE 2 '対象年月日 ==> 1996年4月30日以降'
LINE 1 T-CODE SHAMEI S-CODE H-DATE H-SURYO
LINE 2 POS 2 TEL-NO POS 3 S-RYAKU

```

サブルーチン・コール

- EXEC PARM取得

ユーティリティプログラムEZTPX01を使用する。

例)

CALL EZTPX01 USING(PARM-REGISTER フィールド名)

PARM-REGISTER:予約語

フィールド名:受け取り側フィールド

- 他言語のサブルーチン。

例)

CALL プログラム名 USING(フィールド1, ...)

フィールド1:パラメータ受け渡し用フィールド

```
//STEP01 EXEC PGM=PROG01,PARM='DATA01'
.....
FILE FILEA
.....
PARM-INFO W      12 A
PARM-LTH PARM-INFO  2 B
PARM-DATA PARM-INFO +2 10 A
WKFLD01 W      10 A
WKFLD02 W      10 A
WK-CODE W      15 A
JOB INPUT(FILEA) START INIT
.....
CALL PROG01 USING(WKFLD01 WKFLD02) RETURNS WK-
CODE
.....
INIT. PROC
  PARM-LTH = 10
  CALL EZTPX01 USING(PARM-REGISTER PARM-INFO)
END-PROC
```

DB2アクセス

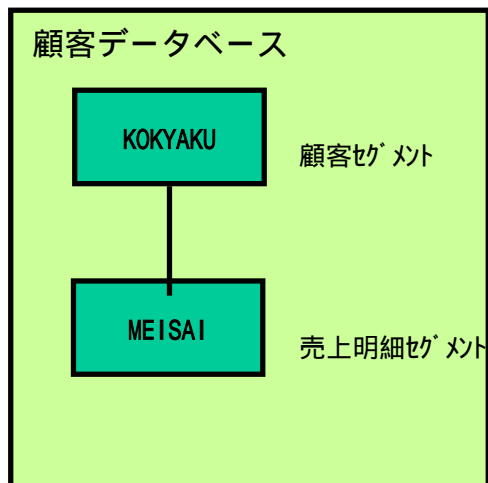
- DB2アクセス
自動検索

```
SQL INCLUDE FROM KYUYO.TABLE01
SQL INCLUDE FROM SYAIN.TABLE01
*
JOB INPUT SQL
SELECT K-SYAIN, K-TEDORI,           +
      S-ADDR1, S-FURIGANA, S-TEL   +
FROM KYUYO.TABLE01, SYAIN-TABLE01 +
WHERE K-SYAIN = S-SYAIN           +
ORDER BY K-SYAIN                   +
INTO :K-SYAIN, :K-TEDORI,          +
      :S-ADDR1, :S-FURIGANA, :S-TEL
PRINT REPT1
*
REPORT REPT1 LINESIZE 80
TITLE 1 '*** 社員給与一覧表 ***'
LINE 1 K-SYAIN S-FURIGANA S-ADDR1 S-TEL K-TEDORI
```

IMS アクセス 1

IMS データベースアクセス 1 自動アクセス

RETRIEVEステートメントによりIMS / DBを順次ファイルイメージで簡単に入力することができる



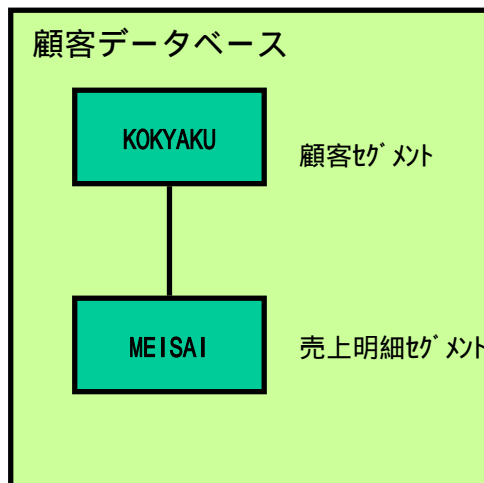
```

FILE KYADB DLI(KOKYAKUD 1)
RECORD KOKYAKU 150
KOCODE 1 5 A
KOENAME 6 20 A
KOKNAME 26 20 K
KOADDR 52 40 M
KOTELNO 103 12 A
RECORD MEISAI 50 KOKYAKU
MEDENNO 1 10 A HEADING('伝票NO')
MEKEIYMD 11 8 N HEADING('伝票日付') +
    MASK('9999-99-99')
MESHOC D 35 6 A HEADING('商品NO')
MESURYO 41 4 P 0 HEADING('数量') +
    MASK('-',---,--9')
MEURIAGE 45 4 P 0 HEADING('売上') +
    MASK('-',---,--9')
JOB INPUT(KYADB)
RETRIEVE KYADB +
    SELECT(KOKYAKU MEISAI)
*
PRINT REPT1
*
REPORT REPT1 LINESIZE 80
CONTROL FINAL NOPRINT KOCODE NOPRINT
NEWPAGE
TITLE 1 '***** 顧客別明細一覧表 *****'
TITLE 2 '顧客コード:' KOCODE '顧客名:' KOKNAME
LIEN 1 MEKEIYMD MEDENNO MESHOC D MEURIAGE
  
```


IMS アクセス 2

- IMSデータベースアクセス2
プログラム・アクセス

DLIステートメントによりCOBOL、PL/Iと同様の命令を使用できる



```

FILE KYADB DLI(KOKYAKUD 1)
RECORD KOKYAKU 150
KOCODE 1 5 A
KOKNAME 26 20 K
RECORD MEISAI 50 KOKYAKU
MEDENNO 1 10 A HEADING('伝票NO')
MESHOC 35 6 A HEADING('商品NO')
MEURIAGE 45 4 P 0 HEADING('売上') +
MASK('-',---,--9')
*
SSA1 W 9 A VALUE 'KOKYAKU '
SSA2 W 9 A VALUE 'MEISAI '
JOB INPUT(NULL)
DLI KYADB KOKYAKU 'GN ' SSA( SSA1 )
DO WHILE KYADB:FILE-STATUS = ' '
  DLI KYADB MEISAI 'GNP' SSA( SSA2 )
  IF KYADB:FILE-STATUS NOT = ' '
    IF KYADB:FILE-STATUS = 'GE' OR 'GB'
      GOTO JOB
    ELSE
      STOP
    END-IF
  END-IF
END-DO
IF KYADB:FILE-STATUS = 'GE' OR 'GB'
  STOP
ELSE
  DISPLAY '* KYADB ERR CODE=' KYADB:FILE-STATUS '****'
*
REPORT REPT1 LINESIZE 80
TITLE 1 '***** 顧客別明細一覧表 *****'
TITLE 2 '顧客コード:' KOCODE '顧客名:' KOKNAME
LIEN 1 MEDENNO MESHOC MEURIAGE
  
```

ADABASアクセス

- ADABASアクセスマクロを使用します。

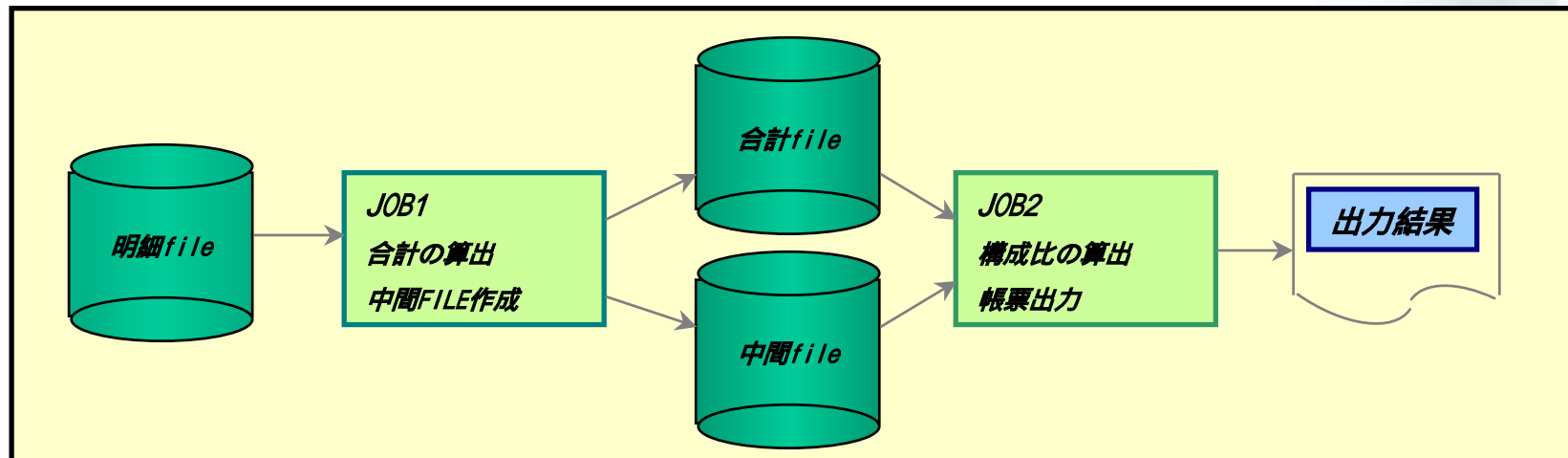
```
SQL INCLUDE FROM KYUYO.TABLE01
SQL INCLUDE FROM SYAIN.TABLE01
*
JOB INPUT SQL
SELECT K-SYAIN, K-TEDORI,           +
       S-ADDR1, S-FURIGANA, S-TEL   +
FROM KYUYO.TABLE01, SYAIN-TABLE01  +
WHERE K-SYAIN = S-SYAIN             +
ORDER BY K-SYAIN                    +
INTO :K-SYAIN, :K-TEDORI,           +
     :S-ADDR1, :S-FURIGANA, :S-TEL
PRINT REPT1
*
REPORT REPT1 LINESIZE 80
TITLE 1 '*** 社員給与一覧表 ***'
LINE 1 K-SYAIN S-FURIGANA S-ADDR1 S-TEL K-TEDORI
```

使用例

処理概要

1. ある商品に関する売上高・売り上げ数を出力する。
2. その際、全売上高・全売上数からの売上構成比・数量構成比を求める。

処理フロー



使用例(コーディング)

プログラム

```

FILE FILEA
  FLDA1 * 3 N
  FLDA2 * 5 P
  FLDA3 * 5 P
FILE FILEB F(100) VIRTUAL RETAIN
  FLDB1 * 3 N HEADING('カテゴリー')
  FLDBTAL * 10 P
  FLDB2 * 10 P 0 HEADING('数量')
  FLDB3 * 10 P 0 +
  MASK('###,###,###,###,###,###,###') HEADING('金額')
FILE FILEC F(100) VIRTUAL RETAIN
  FLDCTAL * 10 P
  FLDC2 * 10 P 0
  FLDC3 * 10 P 0
  WKFLD2 W 5 N 2 VALUE 0 HEADING('数量構成比')
  WKFLD3 W 5 N 2 VALUE 0 HEADING('売上構成比')
  WKFLDC2 S 10 P 0
  WKFLDC3 S 10 P 0
JOB INPUT(FILEA)
  PRINT REPT1
  PRINT REPT2
REPORT REPT1 SUMFILE FILEB
  SEQUENCE FLDA1
  CONTROL FLDA1
  LINE 01 FLDA1 FLDA2 FLDA3
REPORT REPT2 SUMFILE FILEC
  CONTROL
  LINE 01 FLDA2 FLDA3

```

```

JOB INPUT(FILEB) START INIT
  WKFLD2 ROUNDED = FLDB2 / WKFLDC2 * 100
  WKFLD3 ROUNDED = FLDB3 / WKFLDC3 * 100
  PRINT REPT3
INIT. PROC
  GET FILEC
  WKFLDC2 = FLDC2
  WKFLDC3 = FLDC3
END-PROC
REPORT REPT3 PAGESIZE 30 LINESIZE 100
CONTROL FINAL NOPRINT
TITLE 01 '売り上げ構成比リスト'
LINE 01 FLDB1 FLDB2 WKFLDB2 -2 '%' FLDB3 WKFLDB3 -2 '%'
AFTER-BREAK. PROC
  DISPLAY POS 1 '-----+'
  '-----'
  DISPLAY POS 1 '合計' POS 2 FLDB2 POS 5 FLDB3
END-PROC

```

出力結果 ~ 使用例

96年10月25日

売上構成比リスト

1頁

カテゴリー	数量	数量構成比	金額	売上構成比
001	110	.37 %	¥11,000	.37 %
002	1,000	3.45 %	¥100,000	3.45 %
003	220	.75 %	¥22,000	.75 %
004	340	1.17 %	¥34,000	1.17 %
005	3,100	10.70 %	¥310,000	10.70 %
006	1,230	4.24 %	¥123,000	4.24 %
007	2,000	6.90 %	¥200,000	6.90 %
008	90	.31 %	¥9,000	.31 %
009	2,300	7.93 %	¥230,000	7.93 %
010	1,110	3.83 %	¥111,000	3.83 %
011	2,220	7.66 %	¥222,000	7.66 %
012	2,330	8.04 %	¥223,000	8.04 %
013	6,600	22.78 %	¥660,000	22.78 %
014	780	2.69 %	¥78,000	2.69 %
015	980	3.38 %	¥98,000	3.38 %
016	750	2.58 %	¥75,000	2.58 %
017	2,300	7.93 %	¥230,000	7.93 %
018	450	1.55 %	¥45,000	1.55 %
019	830	2.86 %	¥83,000	2.86 %
020	230	.79 %	¥23,000	.79 %
合計	28,970		¥2,897,000	

実行JCLサンプル(OS用)

a. (コンパイル～実行)

```
//JOB CARD
//STEP名 EXEC PGM=EZTPA00
//STEPLIB DD DSN=CA-EasytrievePlus SYSTEMLIBRARY,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSOUT DD SYSOUT=*
//SORTWK01 DD UNIT=SYSDA,SPACE=(TRK,(3,1))
//PANDD DD DSN= CA-EasytrievePlus MACROLIBRARY,DISP=SHR
//EZTVFM DD UNIT=SYSDA,SPACE=(TRK,(3,1))
//FILE名 DD DSN=IN-PUTorOUT-PUT FILE,.....
//SYSIN DD *
~ CA-EasytrievePlus プログラム ~
/*
```

注) FILE名 : プログラムの FILEステートメントで指定したファイル名
IN-PUTorOUT-PUT FILE : 実際のデータセット名

b-1. (コンパイル～連携編集)

```
//JOB CARD
//STEP名 EXEC PGM=EZTPA00
//STEPLIB DD DSN=CA-EasytrievePlus SYSTEMLIBRARY,DISP=SHR
//SYSPRINT DD SYSOUT=*
//PANDD DD DSN= CA-EasytrievePlus MACROLIBRARY,DISP=SHR
//EZTVFM DD UNIT=SYSDA,SPACE=(TRK,(3,1))
//SYSLIN DD UNIT=SYSDA,SPACE=(TRK,(3,1)),DISP=(NEW,PASS),
// DSN=&&OBJECT名
//SYSIN DD *
PARM LINK(プログラム名 (R))
~ CA-EasytrievePlus プログラム ~
/*
//STEP名 EXEC PGM=連携編集プログラム名
//SYSPRINT DD SYSOUT=*
//SYSLIN DD DSN=&&OBJECT名,DISP=(OLD,DELETE)
//SYSLMOD DD DSN=LOAD MODULE格納ライブラリ名,DISP=SHR
//SYSUT1 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
```

b-2. (ロードモジュール実行)

```
//JOB CARD
//STEP名 EXEC PGM=プログラム名
//STEPLIB DD DSN=CA-EasytrievePlus SYSTEMLIBRARY,DISP=SHR
// DD DSN=LOAD MODULE格納ライブラリ名,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSOUT DD SYSOUT=*
//SORTWK01 DD UNIT=SYSDA,SPACE=(TRK,(3,1))
//EZTVFM DD UNIT=SYSDA,SPACE=(TRK,(3,1))
//FILE名 DD DSN=IN-PUTorOUT-PUT FILE,.....
```