

F*TRAN2007ver.2.0

操作説明書/マルチレコード編

第1版

株式会社富士通ビー・エス・シー

はじめに

F*TRAN2007 V2. 0をお買い上げいただき、誠にありがとうございます。

F*TRAN2007は、汎用機やオフコンなど、ホストコンピュータの標準フロッピーディスク形式であるIBM形式フロッピー(IBMファイル)と、パソコンの標準であるWindowsファイルを相互に変換するためのソフトウェアです。

おもな用途としては、汎用機やオフコンのCOBOLデータと、パソコンのCSV形式ファイルとの交換・プリント形式ファイルとの交換・WindowsCOBOLデータとの交換などがあります。また、COBOLのゾーン形式・パック形式をはじめ、豊富なコード変換機能をサポートしています。

日本で販売されたほとんどのホストの漢字をサポートしており、拡張漢字テーブルも標準提供しています。

F*TRAN2007 V2. 0では、Windows7上での動作をサポートしています。

F*TRAN2007のマニュアルには、導入編、解説編、コマンド編、マルチレコード編(本書)、プログラム応用編があります。

2010年4月 株式会社 富士通ビー・エス・シー

Copyright 1997-2010 株式会社 富士通ビー・エス・シー

] 次

	第1章 Atlas98とは?	
1.1	Atlas98とは?	2
1.2	何ができるようになるか?	3
	第2章 Atlas98の文法	
	A t l a s 9 8 の構文	
2.2	A t l a s 9 8 の文法······	7
	第3章 Atlas98の使用法	
n 1	At 1 as 98 = - F	1 7
3.1 3.2	Atlas98七一ト	
3. Z	Atlas98の使用伝 ■Atlas98を使わない場合	
	■ L o o p 文を使う方法	
	■E000pスを使う方法	
	■途中からマルチレイアウトになる場合	
	■複数の手続きを使う場合	
	■不要なレコードを捨てる場合 ····································	
	■小数項目による分岐を行う場合	
	■文字列による分岐を行う場合	
	■バイナリ項目による分岐を行う場合	3 3
	■レコード番号による分岐を行う場合	3 4
	■脱出文の使用法	3 5
	■ヘッダとトレーラの挿入方法	
	■COBOLのCopy句を読み込む	
	■Copy句読み込みの具体例	4 3
	第4章 Atlas98の利用例	
4 1		F. C
4.1	A t l a s 9 8 の利用例····································	
	■ 使来互換	
	■Attras98化	
	■ 早純なくり返し■ 2 重ループ ····································	
		0.0

	■マルチレイアウト変換を1つの手続きで設定する62
	■複数の手続きを活用する
	■ボディーレコードのみ変換する
	■変換を中断する
	■最初にレコード末尾付近の区分項目を見る 7 0
	■空白とゾーン形式を変換し分ける 7 2
	■999で変換を止める73
	■FF・・・(ハイバリュー)の詰まったレコードをスキップする74
	■負の売上を返品とみなす75
	■先頭レコードだけ別レイアウト7 6
	■範囲指定して変換77
	■ 4 レコード毎に同じパターン7 8
	■複合条件で抽出 7 9
	■複合条件と多重脱出8 1
	■ファイル区分別変換83
	■負の数のとき"▲"や"-"を後付けする85
	■Excelの式を埋め込む87
	第5章 マップ解析
5.1	マップ解析の機能9 1

本書で用いる表記法

●本文と画面のパラメータ類の表記法

 $\{A \mid B \mid C\}$ A、B、またはCのうち、どれか1つを選択します。

省略はできません。

A B 同上。

C

(A/B/C) 同上。

[A] Aは省略できます。

[A/B/C] A、B、またはCのうち、どれか1つを選択します。

省略が可能で、その場合は下線を引いたBを選択した

ものとみなします。

A B C

(A/[B]/C) 同上。ただし、[] でくくったBを選択したものと

みなします。

X ・・・ X類を A B C のように列挙します。

n、n n、< n > 10進数を指定します。

(<>は表記上の記号で、入力はしません)

xxH 16進でxxです。Hを省くこともあります。

→ 改行を意味します。リターンキーのシンボルです。

a 下線部を入力します。

abc ↓ 下線部を入力し、リターンキーを押します。

CTRL+A コントロール (CTRL) キーを押しながら、Aキー

を押します。コントロールAと読みます。

^ A 同上。

d: ドライブA:やC:など、任意のドライブ指定を表し

ます。

◆注意 ---- 実画面と少し差異がある

本書に示す画面と実際の画面には、若干の差異がある場合があります。あらかじめ、ご了承ください。

第1章

A t 1 a s 9 8 2 12?

1. 1 Atlas98とは?

COBOLのプログラムでは、よくOCCURS句を使ってくり返し項目をまとめたり、REDEFINES句を使って条件によって項目の定義を変えたりします。順ファイル処理ではごく普通の手法だといってよいでしょう。

従来のF*TRANでこれらに対応するのは大変でした。まず、くり返し項目を記述する方法がなく、必要回数マップの指定をくり返して書いてもらっていました。マルチレコードレイアウトはもっと大変で、レコードタイプ毎に別々に変換してから別プログラムで再合成してもらう、などの方法をとっていました。F*TRANだけでは変換できなかったのです。また、一部のレコードだけ抜き出して変換したいとか、逆に一部のレコードをスキップして変換したいなどの要望もありましたが、これも無理でした。

これらの問題を解決すべく生まれたのが、A t 1 a s 9 8 (F トラス 9 8) です。これは、今までの / M A P オプション(マップ設定)の拡張言語で、くり返しを記述する「L o o p 文」、多分岐を記述する「S e 1 e c t x」、処理をまとめる「手続き」があります。この 3 つを利用することで、ほとんどのパターンの順ファイルが変換可能になったといっても過言ではありません。

Atlas98は、マップ設定(/MAPオプション)の拡張言語です。つまり、マップ設定(/MAPオプション)では、Atlas98言語を使わずに、従来のパラメータをそのまま指定することもできます。つまり、従来のマップ設定(/MAPオプション)との上位互換が保たれています。そして、Atlas98を使用することにより、より拡張された機能を使うことができます。

1. 2 何ができるようになるか?

Atlas98の導入によって、以下のことが実現できるようになります。

- ■くり返し項目が記述できる。
- ■集団項目のくり返しや、多重にネストしたくり返しもできる。
- ■項目の内容に応じて、レイアウトを変えることができる。
- ■項目の内容に応じ、その項目を変換し分けることができる。
- ■条件に合ったレコードを抽出して変換することができる。
- ■条件に合ったレコードを除外して変換することができる。
- ■特定のレコード/項目で、変換を中断することができる(レコード単位、ファイル単位)。
- ■ヘッダレコードなど、特定位置のレコードを別レイアウトで変換することができる。
- ■ヘッダレコード、トレーラレコード、エンドレコードなどを削除することができる。
- ■レコード番号指定で、特定範囲のレコードだけ変換することができる。
- ■一定のサイクルで同じパターンのレコードがあるファイルを変換することができる。
- ■変換結果にヘッダ、トレーラを付加することができる。
- ■ファイルタイプ別の変換ができる。

第2章

A t l a s 9 8 の文法

2. 1 Atlas98の構文

■Atlas98の構文

Atlas98には、つぎの13の文があります。

Atlas	>	A t l a s宣言
Proc	>	手続き開始
EndProc	>	手続き終了
Loop	>	くり返し開始
EndLoop	>	くり返し終了
Select	>	多分岐開始
Case	>	条件指定
ElseCase	>	その他条件
EndSelect	>	多分岐終了
C a 1 1	>	手続きの呼び出し
Return	>	手続きからの復帰
Break	>	構造からの抜け出し
Quit	>	処理の中断

Atlas98には、くり返しを記述するための「Loop文」、多分岐を記述するための「Select文」、処理をまとめるための「手続き」があります。とくにSelect文は、Pascal言語のcase文、C言語のswitch文、Visual Basicのselect case文などに似ていますが、はるかに強力な構文として、Atlas98の中核機能をなしています。

Atlas98は、構造化された言語です。Atlas98には、Goto文がありません。 その代わり、柔軟な脱出機構があります(Break文、Return文、Quit文)。

Atlas98は、COBOLの仕様である最大10進18桁の数値の扱いを保証しています。

◆注意 ---- Atlas構文を指定する時の決まり事(コマンド型での指定)

- ・Atlas構文は、Map文との区別のため、ブレース({と}) でくくる必要があります。
- ・Atlas構文のキーワードは短縮できません。
- ・Atlas構文のキーワードの大文字、小文字は区別されません。

2. 2 Atlas98の文法

■Atlas文(Atlas宣言)

Atlas構文の使用を宣言します。

```
{Atlas 98}
```

マップ設定 (/MAPオプション) にAtlas構文を指定する場合は、必ずAtlas宣言から始めます。

■Proc文(手続き開始)

EndProc文(手続き終了)

Proc構造は、複数の文 (Map文、Atlas文) をまとめ、1つのまとまった手続きを構築します。

```
{手続き名:Proc}
Map文 または Atlas文
{EndProc}
```

マップ設定(/MAPオプション)にAtlas構文を指定する場合は、Atlas宣言をした後に、1つないし複数の手続きを指定します。手続きは再帰呼び出しができますが、入れ子にはできません。

1番最初の手続きが自動的に主手続きになります。Atlas指定の実行は、常に主手続きの 先頭から始まり、主手続きから復帰時に終了します。

手続き名は、英大文字 $(A \sim Z)$ 、英小文字 $(a \sim z)$ 、数字 $(0 \sim 9)$ 、全角文字で構成されます。ただし、先頭が数字であってはいけません。また、全角スペースを含んではいけません。英大文字と英小文字は区別されません。手続き名の長さは、16文字以内で指定します。2重定義はできません。手続き名は省略できません。手続き名は任意ですが、主手続きに与える手続き名は習慣的に「main」または「メイン」を用います。

EndProc文で、手続きを終了します。

■Loop文(くり返し開始)

EndLoop文(くり返し終了)

指定回数のくり返し構造を構築します。

{[構造名:]Loop 回数}
Map文 または Atlas文
{EndLoop}

構造名は、英大文字 $(A \sim Z)$ 、英小文字 $(a \sim z)$ 、数字 $(0 \sim 9)$ 、全角文字で構成されます。ただし、先頭が数字であってはいけません。また、全角スペースを含んではいけません。英大文字と英小文字は区別されません。構造名の長さは、16文字以内で指定します。2重定義はできません。構造名は省略できます。

Loop文で、くり返しの回数を指定します。整数値($0\sim65535$)、16進定数値(00H \sim ffffH)、またはシステム変数を指定することができます。くり返しの回数は省略できません。

システム変数は、つぎの5つです。

SysPhase フェーズ(Open=1、Main=2、Close=8)
SysRecNum レコード番号(0~)
SysReturn 返り値(Return文で代入される)
SysBreak ブレーク値(Break文で代入される)
SysQuit クイット値(Quit文で代入される)

EndLoop文で、くり返し指定を終了します。

■Select文(多分岐開始)

Case文(条件指定)

ElseCase文(その他条件)

EndSelect文(多分岐終了)

多分岐構造を構築します。Select構造の主な使い方は、マルチレイアウト変換への適用と、条件つき変換(抽出変換、除外変換)です。

構造名は、英大文字 $(A \sim Z)$ 、英小文字 $(a \sim z)$ 、数字 $(0 \sim 9)$ 、全角文字で構成されます。ただし、先頭が数字であってはいけません。また、全角スペースを含んではいけません。英大文字と英小文字は区別されません。構造名の長さは、16文字以内で指定します。2重定義はできません。構造名は省略できます。

Select文の選択型の指定は、つぎの4つです

```
Int整数型(Integer)Dec小数型(Decimal)Str文字列型(String)Binバイナリ型(Binary)
```

セレクタとは、多分岐の基準になる項目のことで、各Case値と比較されます。有効なMap p 文を 1 つ指定します。

セレクタに指定したMap文の変換結果は自動的に捨てられ、その変換はなかったことにされます。もし、変換結果も必要なら、同じMap文をCase文やElseCase文の後に再度、指定します。

選択型と有効なセレクタの組み合わせは、下表の組み合わせになります。

Мар文	Int/Dec	String	Binary
DispZone/ZoneDisp	0	×	×
DispPack/PackDisp	\bigcirc	×	×
DispBin /BinDisp	\bigcirc	×	×
ZonePack / PackZone	\bigcirc	×	×
ZoneBin /BinZone	\bigcirc	×	×
PackBin /BinPack	\bigcirc	×	×
ZoneZone / PackPack	\bigcirc	×	×
BinBin	\bigcirc	×	×
Numeric *	×	×	×
Ank	×	\circ	×
Kanji	×	\circ	×
KanjiMix	×	\circ	×
Binary	×	×	0
BinaryX	×	×	0
ВҮ	0	×	×

*) Numeric変換は数値比較も文字列比較もできない。

Case文は、変換する条件を指定します。Select文には、少なくとも1つのCase文を含まなければなりません。

{Case 範囲並び}

範囲並びは、Case文を実行すべき値、あるいは値の範囲を1つ、または、コンマで区切って複数指定します。範囲はA...Bの形式で示し、両端の値を範囲に含みます。たとえば、[A...B,C,D]は[AからB、または[C]、または[C]、または[C] を意味します。

整数型の範囲並び指定

[-] n	((-)はマイナス符号、nは最大18桁)
$0 \{X \mid x\} h \cdots$	hは0~9、A~F (a~f) で最大16桁 (8バイト)
Zero	0
Min	最小値
Max	最大値

小数型の範囲並び指定

[-] n . m	((-)はマイナス符号、nとmは合わせて最大18桁)
Zero	0. 0
Min	最小値
Max	最大値

文字列型の範囲並び指定

'文字列' 文字列は最大256文字で、半角(*)でくくる		
>	※('), (*), (?)は(¥'), (¥*), (¥?)で表し、(¥)自身は(¥¥)で表す	
Space	空白文字列(半角、全角の空白からなる文字列)	
LowValue	0 0 H…の文字列	
HighValue	F F H · · · の文字列 *	

バイナリ型の範囲並び指定

$\{X \mid x\}$ 'hh'	hは0~9、A~F (a~f)、2桁で1バイト
	任意のバイト境界に半角スペースを挿入できる
	半角(')でくくる、最大256バイト
LowValue	0 0 H…のバイナリデータ
HighValue	FFH…のバイナリデータ *

文字列の比較は、ホスト側文字列であれば、Win側文字列に変換した文字列の比較を行います。基本的に <math>JIS8/シフト JISコード順で大小比較をしますが、連続する空白は Case値、セレクタ値ともに 1 個の空白にまとめてから比較します。文字の大小関係はつぎのとおりです。

LowValue < ANK 制御文字(01~1FH) < 空白 < ANK < 全角文字(空白除く) < HighValue

Else Case 文は、その他の場合の変換方法を指定します。省略可能です。 End Select 文で、多分岐指定を終了します。

*) ANK変換表をFF→FFに変更する必要がある。

■BY変換

Map文の一種として、BY変換が新設されました。BY変換は、与えられたMap式の値をセレクタ値に変換します。Map式の中には「 * システム変数名」の形でシステム変数を記述することができます。これによって、システム変数の値によって分岐することができます。

BY Map式(システム変数を含む式)

指定できるシステム変数は、つぎの5つです。

~SysPhase フェーズ(Open=1、Main=2、Close=8) ~SysRecNum レコード番号(0~) ~SysReturn 返り値(Return文で代入された値) ~SysBreak ブレーク値(Break文で代入された値) ~SysQuit クイット値(Quit文で代入された値)

指定例: BY $^{\sim}$ SysRecNum¥¥4 レコード番号を4で割った余り値 (セレクタ値は $0\sim3$ の値になる)

■Call文(手続きの呼び出し)

手続きを呼び出します。

{Call 手続き名}

手続き名は、Proc文で指定した手続き名を指定します。

■Return文(手続きからの復帰)

手続きから復帰します。

{Return [整数値 | 16進定数値 | システム変数名]}

返り値として、整数値 $(-32768\sim32767)$ 、 $16進定数値 (00H\sim fff H)$ または、システム変数を返すことができます。返り値を省略すると、返り値は不定になります。

システム変数は、つぎの5つです。

SysPhase フェーズ(Open=1、Main=2、Close=8)
SysRecNum レコード番号(0~)
SysReturn 返り値(Return文で代入される)
SysBreak ブレーク値(Break文で代入される)
SysQuit クイット値(Quit文で代入される)

実行時に ${EndProc}$ に達した時は、 ${Return 0}$ が実行されたとみなされます。

■Break文(構造からの抜け出し)

Loop構造、Select構造から抜け出します。

{Break 「構造名」「,整数値 | 16進定数値 | システム変数名]}

構造名を省略すると、そのBreak文が直接属しているLoop構造/Select構造を抜け出し、構造名を指定すると、指定されたLoop構造/Select構造を抜け出します。

システム変数は、つぎの5つです。

SysPhase フェーズ(Open=1、Main=2、Close=8)
SysRecNum レコード番号(0~)
SysReturn 返り値(Return文で代入される)
SysBreak ブレーク値(Break文で代入される)
SysQuit クイット値(Quit文で代入される)

■Quit文(処理の中断)

処理を中断します。

{Quit [<u>RecSkip</u> | RecDone] [, <u>Continue</u> | EOF] [, 整数値 | 16進定数値 | システム変数名]} *

その時に、そのレコードの処置、継続/中断、クイット値を指定することができます。

RecSkip 処理中のレコードの処理をスキップします。

そのレコードのそこまでの変換結果は放棄されます。

RecDone 処理中のレコードの処理を完了します。

そのレコードのそこまでの変換結果は有効になります。

Continue そのファイルの処理を継続します。

EOF そのファイルの処理を中断・完了します。

そのファイルのそこまでの変換結果は有効になります。

*) RecSkip/RecDoneとContinue/EOFは順番を逆に書くこともできます。

クイット値は、全体の開始時(Openの直前)に自動的に<math>0にセットされます。Quitt では、クイット値として整数値(-32768~32767)、16 進定数値(00H~fff f f H)、または、システム変数の値を指定します。これを、その後のSelect と 文で判定することができます。

システム変数は、つぎの5つです。

SysPhase $J = -\vec{x}$ (Open=1, Main=2, Close=8)

SysRecNum レコード番号 (0~)

SysReturn 返り値(Return文で代入される)

SysBreak ブレーク値(Break文で代入される)

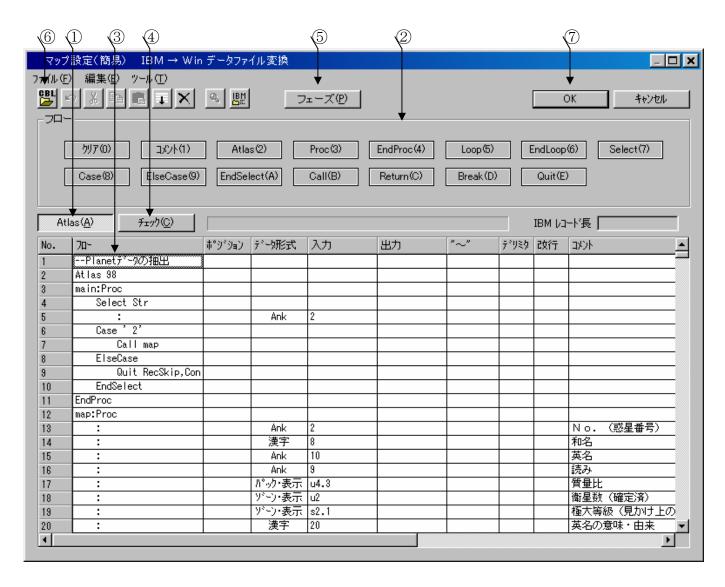
SysQuit クイット値(Quit文で代入される)

第3章

A t 1 a s 9 8 の使用法

3. 1 Atlas98モード

マップ設定(簡易)ウインドウでAtlas98を設定します。

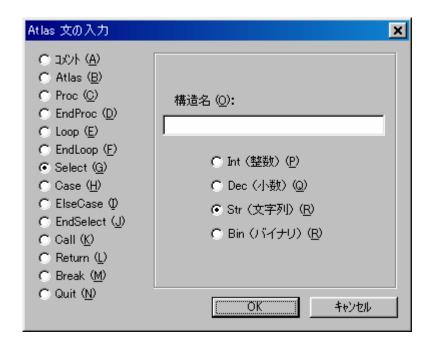


- ①Atlasボタンをクリックして、Atlasモードにします。
 - 再度、Atlasボタンをクリックすると、従来互換モードに戻ります。ただし、フロー欄にAtlas設定がある状態では、従来互換モードに戻ることはできません。マップ設定(簡易)ウインドウを開いた時に、すでにAtlas設定がある場合は、自動的にAtlasモードになります。
- ②セルポインタがフロー欄にある場合に、Atlas設定(コメントを含む)選択肢のボタンが表示されます。このボタンをクリック、または、割り付けられたキー入力を行うと、Atlas設定ウインドウが開きます。
- ③ A t l a s 設定の内容が表示されるフロー欄です。セルをダブルクリックすると、 A t l a s 設定ウインドウが開き、A t l a s の構文設定を簡単に行うことができます。

- ④このボタンをクリックすると、フロー欄の入力内容を検査します。 入力されたAtlas設定が正しければ、設定内容を整形して再表示します。 誤りがあれば、エラーメッセージを出力します。
- ⑤このボタンをクリックすると、フェーズ設定ウインドウが開きます。
- ⑥このボタンをクリックすると、COBOLのCopy句読み込みウインドウが開きます。
- ⑦OKボタンをクリックした時にも、フロー欄の入力内容を検査します。

<Atlas設定ウインドウ>

つぎのAtlas設定ウインドウで、Atlasの構文設定を行います。設定する構文のキーワードを選択すると、キーワードに沿った設定項目が右側に表示されます。



●マップ設定(詳細)でのAtlas設定

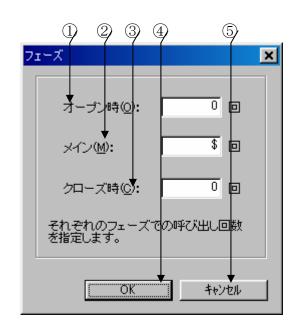
マップ設定(詳細)でAtlas設定を行う場合は、Atlas文をブレース($\{E\}$)でくくる記述を行います。F*TRAN2007をコマンド型で実行する際の/MAP指定の記述、または、その時に使用するパラメータファイルの記述の仕方も同様です。詳しくは、次章の使用法を参照してください。

<フェーズ設定ウインドウ>

変換時に \wedge ッダやトレーラを挿入したいときに、 ∇ ェーズの設定が必要になります。F*TRAN2007には、つぎの ∇ ェーズがあります。

Openフェーズヘッダの生成・付加を行うMainフェーズ本体の変換をするCloseフェーズトレーラやエンドレコードの生成・付加を行う

フェーズ設定ウインドウでは、各フェーズでのAtlas呼び出し回数を指定します。

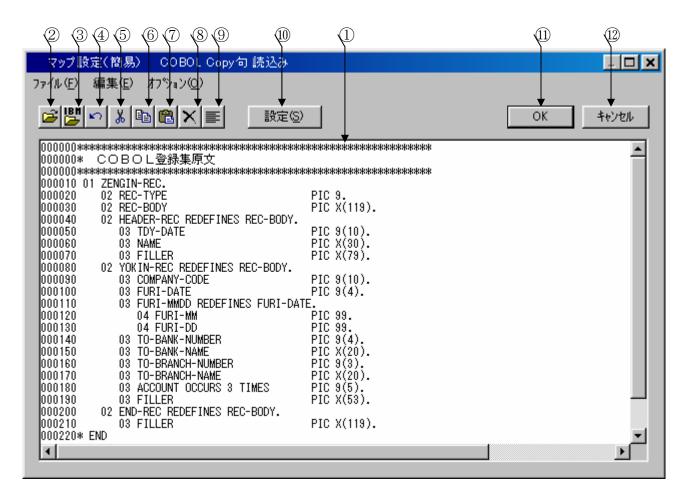


- ①OpenフェーズでのAtlas呼び出し回数を指定します。
- ②MainフェーズでのAtlas呼び出し回数を指定します。
- ③С10seフェースでのAtlas呼び出し回数を指定します。
- ④設定が完了したら、OKボタンをクリックします。
- ⑤キャンセルボタンをクリックすると、入力/編集作業が無効になります。

回数を 0 に指定すると、呼び出しは起こりません。初期値はそれぞれ、オープン時が 0 回、メインが \$ (全レコード件数)回、クローズ時が 0 回となっています。つまり、ヘッダやトレーラは付加せず、本体のみ変換します。

実際のフェーズごとの動作は、Atlas設定で指定する必要があります。詳しくは、次章の使用法を参照してください。

<COBOLのCopy句読み込みウインドウ>



- ①COBOLのソースを記述するエディットボックスです。
- ②~⑩はファイル (F)、編集 (E)、オプション (O) のメニューの機能がボタン化されています。
- ②既存のCOBOLソースファイルを読み込みます。
- ③既存のCOBOLソースファイルをIBM形式フロッピーディスクから読み込みます。
- ④直前の編集作業が無効になり、元に戻ります。
- ⑤選択した文字列が切り取られ、カットバッファに入ります。
- ⑥選択した文字列がカットバッファに入ります。
- ⑦⑤/⑥の操作でカットバッファに入った内容を、カーソルがある位置へ貼りつけます。 カーソルがある位置以降に文字列があれば、挿入になります。
- ⑧選択した文字列が削除されます.。
- ⑨①の文字列をすべて選択状態にします。
- ⑩設定ボタンをクリックすると、Copy句読み込み設定ウィンドウが開きます。変換設定ウインドウからも開くことができます。

⑪編集が完了したらOKボタンをクリックします。

マップ設定(簡易)画面に戻り、編集した内容がMap/Atlas展開されます。

⑫キャンセルボタンをクリックすると、すべての入力/編集作業が無効になります。

< Copy句読み込み設定ウインドウ>

COBOLのCopy句(登録集)を読み込み、マップ設定に展開するときの設定をします。



①出力項目の形式を指定します。

テキスト系 (T) を指定すると、入力形式に関わらずテキスト系で出力します。 入力ファイル形式に準じる (E) を指定すると、入力形式がテキスト系なら出力もテキスト 系にし、入力がランダム系なら出力もランダム系にします。 ②COBOLソースの正書法を指定します。"末尾8桁を $ID \cdot SEQ$ 欄とみなす (I)"の チェックボックスは、可変形式を指定したときのみ有効です。

[固定形式] 1行の初めから6桁目までを一連番号領域、73桁目から80桁目をID・SEQ欄(プログラム識別番号領域)と見なします。7桁目から72桁目が有効桁です。

[可変形式] 1行の初めから6桁目までを一連番号領域と見なします。7桁目以降が有効桁です。最大256桁とします。

ただし、"末尾8桁をID・SEQ欄とみなす(I)"をチェックすると、有効桁は7桁目から「行末-8」桁目までとなります。

[自由形式] 行全体が有効桁です。1行は最大256桁とします。

- ③COBOLソースで使用しているタブ間隔を指定します。
- ④COBOLソースがANKのみで書かれているか、漢字が含まれているかを指定します。
- ⑤COBOLソース中のREDEFINES句で再定義したデータを無視するか、有効にするかを指定します。

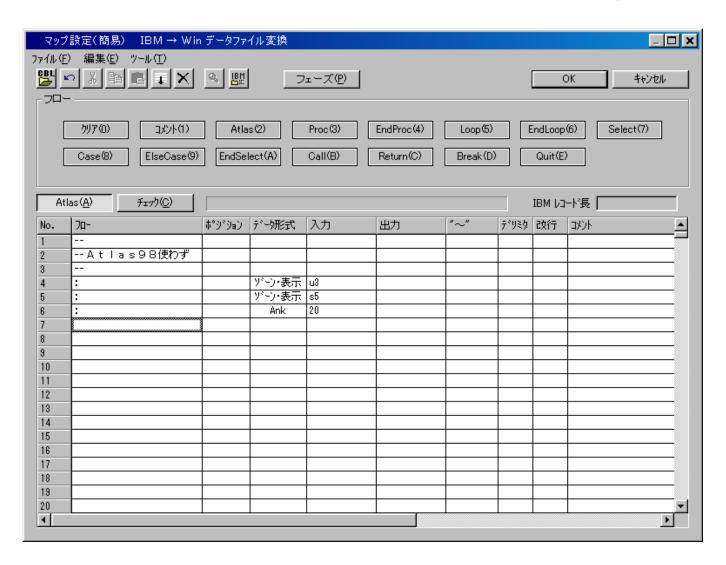
生成する(G)を指定すると、REDEFINES句はコメントとして生成されます。 具体的な使用方法は、次章の使用例を参照してください。

- ⑥見だしレコードを生成するかを指定します。⑤のREDEFINES句を生成すると指定したときは、見だしレコードを生成してもあまり意味がないため指定できません。 具体的な使用方法は、次章の使用例を参照して下さい。
- ⑦このチェックボックスをONにしてOKボタンをクリックすると、F*TRAN2007 終了時に、ここで設定した内容がコード変換表ファイルに書き込まれます。このウインドウを変換設定 $(S) \to Copy$ 句読み込み設定 (P) で開いたときは、デフォルトでチェックした状態になっていて変更できないようになっています。
- ⑧OKボタンをクリックすると修正したCopy句読み込み設定が有効になります。 ただし、あくまでもメモリ上の変更であり、コード変換表ファイルに書き込まれるわけでは ありません。
- ⑨キャンセルボタンをクリックすると修正した Copy 句読み込み設定を無効にし、 Copy 句読み込み設定のウィンドウを閉じます。

3. 2 Atlas98の使用法

■ A t | a s 9 8 を使わない場合

まず、AtlasモードでAtlas98を使わない方法を1つ見てみましょう。

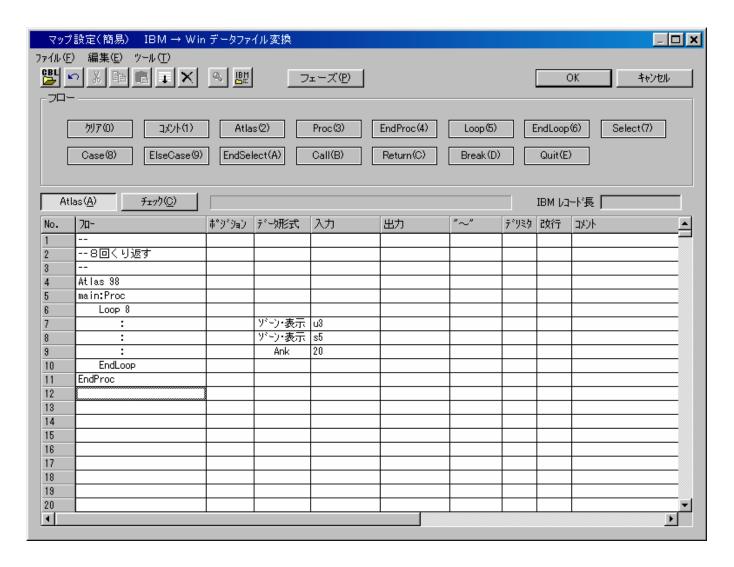


フロー欄というのが先頭にあります。ここにコメントやAtlas98の処理を書きます(コメントは右端のコメント欄にも書けます)。この例ではマップの設定がある行にコロン(:)が表示されているだけです。このコロンはレベルマークといって、入れ子の状態を分かりやすくする記号です。利用者が入力するものではないことを、最初に断っておきます。

そうすると、これは特に何も指定していないマップで、従来のマップ設定と意味は変わらないことが分かると思います。ここは重要なところです。つまり、積極的にAtlas98の機能を使わないかぎり、従来のマップ設定と互換性があるのです。今までの利用者の資産は、1つも無駄にはなりません。

■Loop文を使う方法

では手始めに、A t 1 a s 9 8の一番簡単な機能を使ってみましょう。A t 1 a s 9 8で一番簡単なのは、L o o p 文です。これは、指定回数だけマップをくり返すものです。最初の例を 8回くり返すとしたら、こんなふうになります。



Loop 8~EndLoopというのは、この中を8回くり返せという意味です。

◆注意 ---- Atlas98のルール

Atlas98は、次のルールに従わなければなりません。

コメントや空行を除く実質的な先頭に、Atlas 98と書くこと 実行したい部分をProc~EndProcでくくり、手続きという単位にすること

最低限これを守らないと、文法違反になります。 Procの行の「main」は手続き名で、 適当な名前を指定してもかまいませんが、習慣的に「main」または「メイン」を使います。

■簡単なSelect文を使う方法

さあ、これでAtlas98プログラムの最小限の構造は分かりました。次は、簡単なSe1e ct 文の使い方です。

最初に述べたとおり、Select 文は主にマルチレイアウト変換に使います。マルチレイアウトで一番多い例は、レコードの先頭1 バイトでレコード区分を示すものです。たとえば、こんなレコード区分になっているファイルを考えます。

- '1' ヘッダレコード
- '2' ボディーレコード
- '8' トレーラレコード
- '9' エンドレコード

このフォーマットのファイルを変換するには、

先頭1バイトを変換してみて、

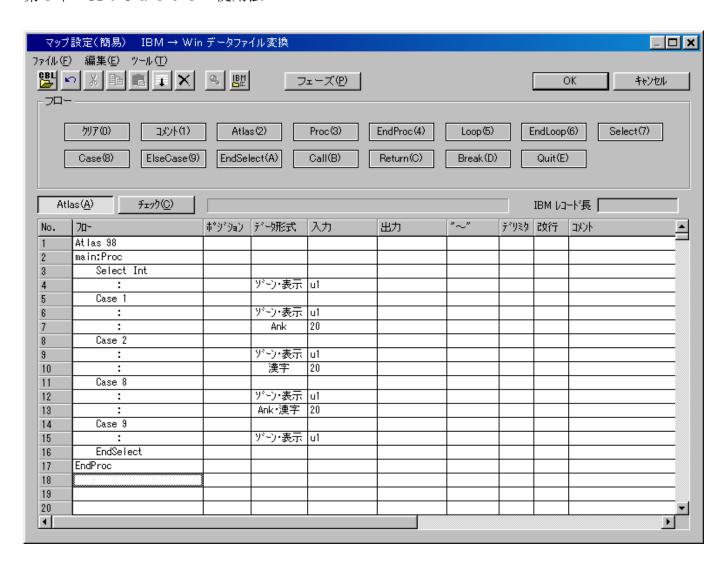
値が1なら、ヘッダレコード用のマップを実行し、

値が2なら、ボディーレコード用のマップを実行し、

値が8なら、トレーラレコード用のマップを実行し、

値が9なら、エンドレコード用のマップを実行する。

ということが書ければよいことが分かります。実際、この流れのとおり書けるのです。見てみましょう。



簡単ですね。他の高級言語を知っている方なら、とてもやさしいと思います。

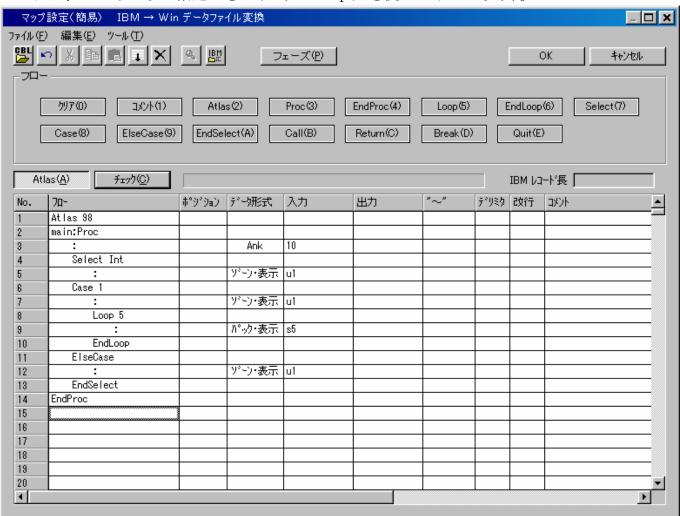
Select Int(Select Integer)というのは、整数値で場合分けするということです。そして、その次に書いた「ゾーン u1」が実行されます。といっても、テストのための実行ですから、その項目の値が得られたら実行はなかったことにされます。そして、各Caseの値とつぎつぎに比較していき、Caseの値が同じところを実行するのです。そのCaseが終わると、EndSelectsで飛んで終了ということになります。

■途中からマルチレイアウトになる場合

では途中まで同じレイアウトで、途中の区分項目によって後半のパターンが変化するという、 つぎのような例を考えてみましょう。

項 番	レコードタイプ 1	レコードタイプ 2
1 (共通)	Ank 10	Ank 10
2 (フラグ)	ゾーン u 1 (=1)	ゾーン u 1 (1以外の値)
3	パック s 5	なし
4	パック s 5	なし
5	パック s 5	なし
6	パック s 5	なし
7	パック s 5	なし

これは、こんなふうに指定できます(Loop文も使ってみましょう)。



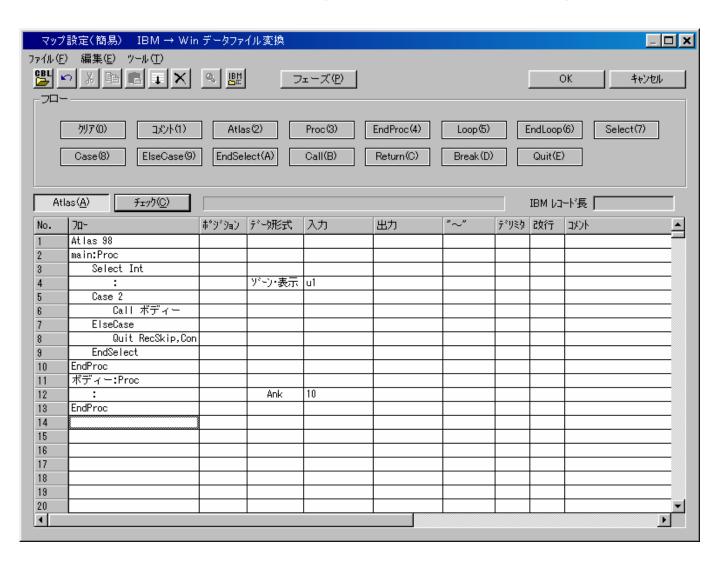
■複数の手続きを使う場合

ところで、ボディーレコード用のマップが何十行もあったらどうすべきでしょうか。そこに書けばいいだけだともいえますが、Selectoredistric

ボディー: Proc ボディーレコード用のマップ End Proc

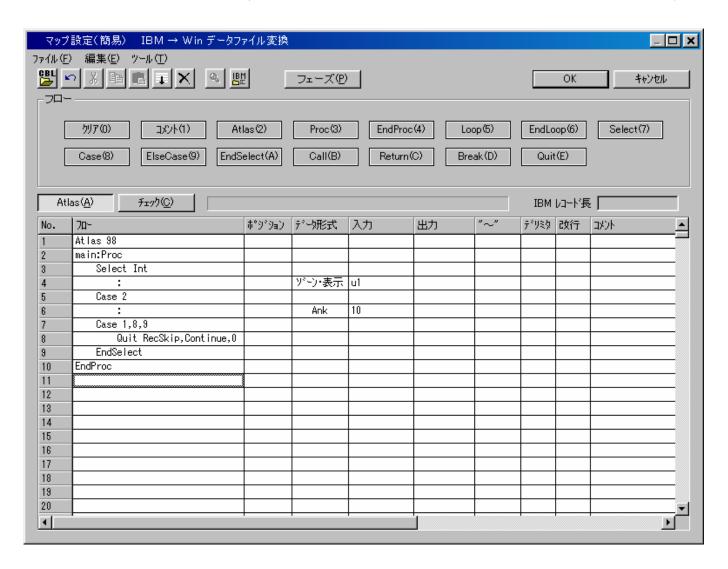
そしてこれを呼び出すには、元のところに

Call ボディー と書きます。全体は、こんな具合になります。



■不要なレコードを捨てる場合

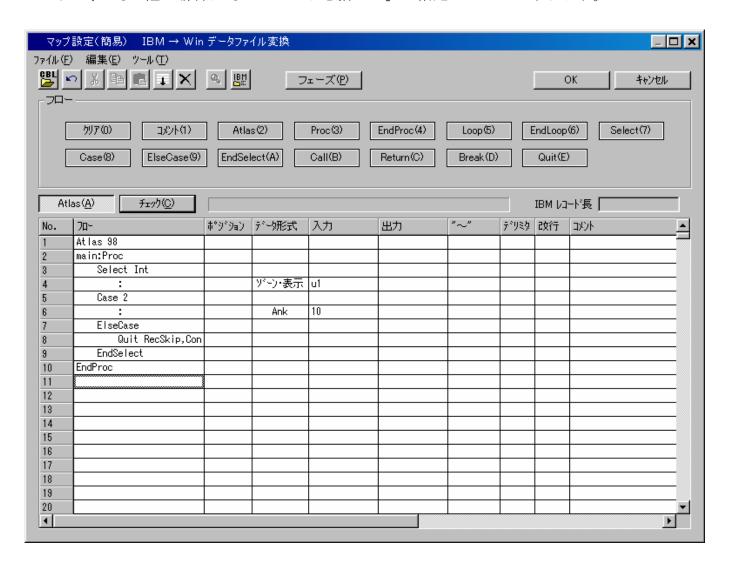
さて、最初の例に戻ってボディーレコードしか要らない場合、つまりヘッダレコード、トレーラレコード、エンドレコードは捨てるという処理が必要なら、どう書くのでしょうか。Caseには、どんな場合かをコンマ(,)で区切って列挙できるので、まずこんなふうに書けます。



Case 1, 8, 9は「1、8、または9なら」という意味です。その下に書いたQuit 文がポイントです。Quit文をデフォルトのまま書くと、「そのレコードを捨てよ」という意味になります。つまり上の例は、「値が1、8、または9のレコードは捨てよ」と指定したことになります。

Select文には「その他の場合」という指定もできます。具体的には、Caseの代わりにElseCaseと書いてやればよいのです。すると、前頁の例は、つぎのように書くこともできます。

これで、「その他の場合はそのレコードを捨てよ」と指定したことになります。



■小数項目による分岐を行う場合

ここまで、整数で分岐させる例を見てきましたが、この他に小数項目の値によって分岐させる 方法と、文字列項目の値によって分岐させる方法、そして、バイナリ項目の値によって分岐させ る方法があります。

小数項目によって分岐させるには、Select Dec (Select Decimal)を使います。COBOLのピクチャがS9V99のような項目の値によってレイアウトが分かれる、次のようなファイルを考えます。

範	井		レコード区分
-9.	9 9	\sim -0.01	マイナスレコード
0.	0 0		ゼロレコード
0.	0 1	~ 9.99	プラスレコード

範囲を「AからBまで」と指定するには、「A.. B」と書きます。また、 \sim 以上、 \sim 以下という指定をしやすくするM i n.. b.. M a x という指定方法があります。これを使ってみましょう。上の例は次のように書けます。ちなみにM i n 、M a x は整数のときにも使えます。



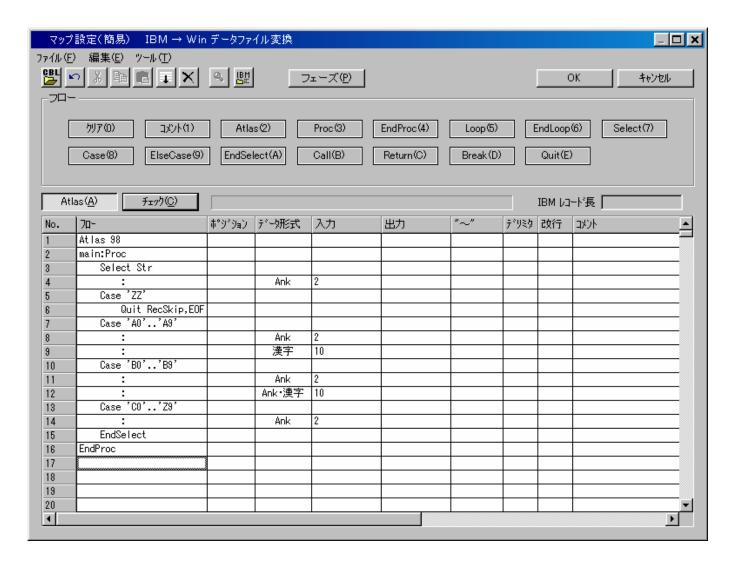
「Min...-0.01」は「-0.01以下なら」と読めばよいですし、「0.01...Max」は「0.01以上なら」と読めばよいのです。

■文字列による分岐を行う場合

ではSelect文を文字列で分岐させる、下表のような例を考えてみましょう。

```
    'A0' ~ 'A9'
    'B0' ~ 'B9'
    'C0' ~ 'Z9'
    'ZZ'
    タイプAレコード
    タイプBレコード
    タイプCZレコード
    エンドレコード(途中に現れたらそこで中断)
```

これは、こんなふうに書けます。



Select Str(Select String)と書くと、文字列による比較になります。'A0'..'A9'は「A0からA9までなら」という意味になります。

'ZZ'の場合の「Quit RecSkip, EOF, 0」とは、「ZZなら、ここでこのファイルの処理を終えよ」という意味です。

■バイナリ項目による分岐を行う場合

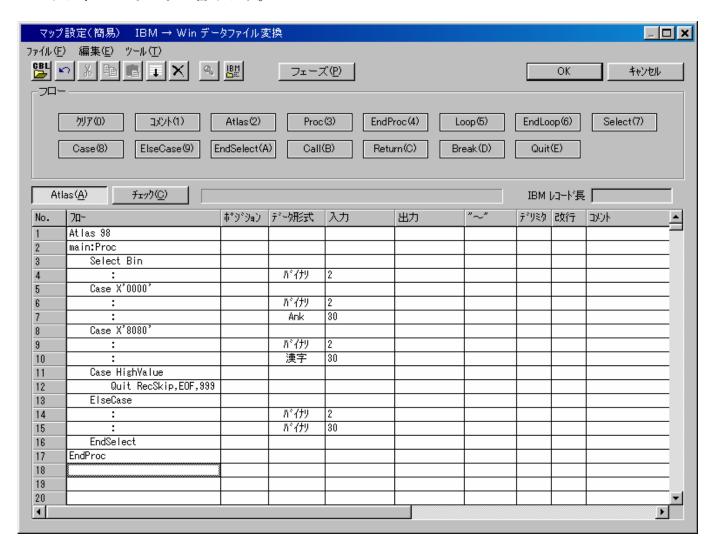
Select文をバイナリ項目によって分岐させる例を考えてみましょう。

X' 0000' 977AV=X' 8080' 977BV=

HighValue エンドレコード(途中に現れたらそこで中断)

上記以外 タイプCレコード

これは、こんなふうに書けます。



文字列による分岐とほとんど変わりませんが、Select Bin (Select Bin ary)と書くと、バイナリ項目による比較になります。

HighValueはバイナリFF・・・に一致します。「それ以外のどんなバイナリデータよりも大きい」という意味です。

■レコード番号による分岐を行う場合

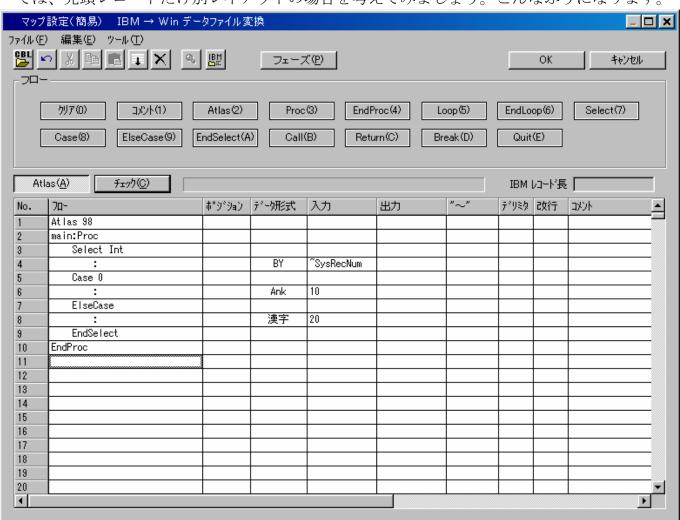
先頭1~数レコードだけ別レイアウトで、その後にはボディーレコードが続くといったファイルもよくあります。At1as98には、レコード番号をもとに変換し分ける方法も用意されています。

「レコード番号が何番か」と聞くには、

Select Int : BY SysRecNum

と書きます。このSysRecNumはシステム変数の1つで、F*TRAN2006が自動的にレコード番号をセットします。

では、先頭レコードだけ別レイアウトの場合を考えてみましょう。こんなふうになります。



ちなみに、先頭レコードのレコード番号は0です。

一定周期でくり返し同じレイアウトが出てくるようなファイルも、この応用で変換できます。

■脱出文の使用法

条件によって、くり返しを中断したいことがあります。このときは、Break文やReturn文を用います。Break文は、Loop構造やSelect 構造から抜け出すための文です。一方、Return 文は、手続きから抜け出すための文です。なお、先に述べたQuit 文も脱出文の1つです。

Break文は、おもにLoop構造からの抜け出しに使います。2重ループからの脱出もできます。脱出条件はSelect文で判定します。

7桁のパック形式項目(4バイト)の配列があるとします。64エントリで1レコードです。 このとき、値が999999ならそのレコードはそこまでで変換を中断する例を考えてみます。 全体をループにして、

XX: Loop 64

:

: XXループの中身

:

EndLoop

と書きます。このループにXXという名前(構造名)を付けていることに注意してください。 また、このループから脱出する部分は、

Select Int

: Pack u7

Case 9999999

--XXから脱出する

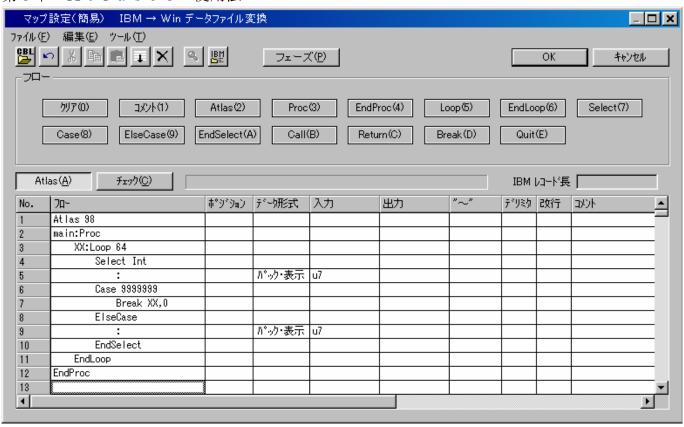
Break XX

ElseCase

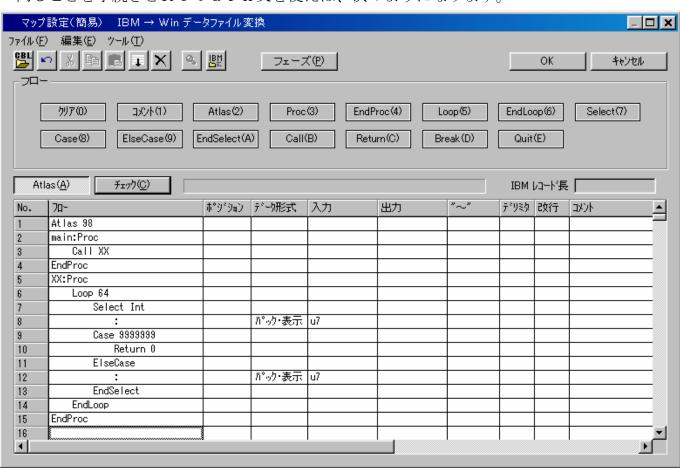
: Pack u7

EndSelect

と書きます。まとめると、つぎのようになります。



同じことを手続きとReturn文を使えば、次のようになります。



■ヘッダとトレーラの挿入方法

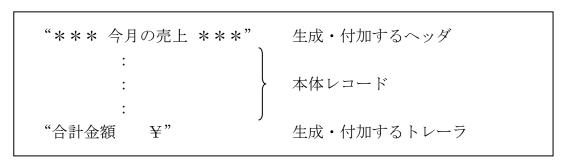
ボディレコードのみのファイルを変換する際に、ヘッダやトレーラを付けたいという場合がよくあります。F*TRAN2007では、ヘッダ/トレーラ挿入の機能をサポートしています。

ヘッダ/トレーラを生成・付加する場合は、Atlas98の設定の他に、フェーズの設定を行う必要があります。フェーズには、

Openフェーズヘッダの生成・付加を行うMainフェーズ本体の変換をするCloseフェーズトレーラやエンドレコードの生成・付加を行う

の3種類があり、フェーズの設定では各フェーズの呼び出し回数を指定します。

それでは、つぎのようなヘッダとトレーラを、それぞれ一行ずつ挿入する場合を考えてみましょう。



まず、A t l a s 9 8 の設定を行います。S e l e c t 文を使って、それぞれのフェーズごとのマップを指定します。

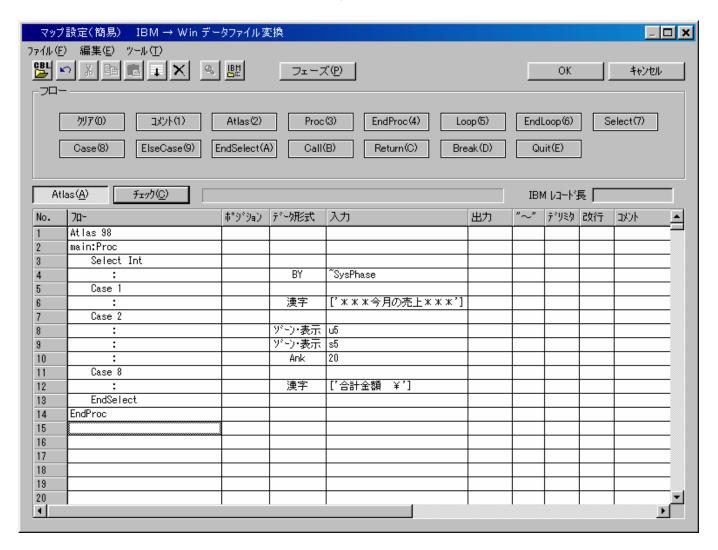
「どのフェーズか」による分岐は、

Select Int : BY "SysPhase

と書きます。このSysPhaseはシステム変数の1つで、F*TRAN2007があらかじめ値を設定しています。

SysPhaseの値は、つぎのようになっています。

1 = Openフェーズ 2 = Mainフェーズ 8 = Closeフェーズ 実際に書いてみると、このようになります。



Case 1でヘッダ「*** 今月の売上 ***」を、Case 8でトレーラ「合計金額 ¥」を挿入する指定をしています。以上で、Atlas98の設定は終わりです。

つぎに、フェーズの設定を行います。マップ設定(簡易)ウインドウのフェーズボタンをクリックして、フェーズ設定ウインドウを開きます。ここでは、それぞれのフェーズにおいて、 \mathbf{A} t $\mathbf{1}$ a s が呼び出される回数を設定します。

呼び出される順番は決まっていて、

Open $Jz-X \rightarrow MainJz-X \rightarrow CloseJz-X$

のようになっています。Openフェーズから始まり、設定した回数だけAtlasを呼び出すとつぎのフェーズに移り、<math>Closeフェーズで終了します。回数を0に指定すると、呼び出しは起こりません。

今回は、Openフェーズ、Closeフェーズをそれぞれ「1回」に指定して、<math>OKボタンをクリックします。Mainフェーズは初期値(<math>\$=全レコード件数)のまま、変更する必要はありません。



以上で設定は終わりです。この設定によって、つぎのように変換します。

- 1. まず、OpenフェーズでAtlasが呼び出され、SysPhase=1のときの変換を1回行う(ヘッダ挿入)。
- 2. つぎに、MainフェーズでAtlasが呼び出され、SysPhase=2のときの変換を全レコード件数回行う(ボディレコード変換)。
- 3. 最後に、CloseフェーズでAtlasが呼び出され、SysPhase=8のときの変換を1回行い(トレーラ挿入)、F*TRAN2006を終了する。

◆注意 ---- コマンドで実行する場合

F*TRAN2007をコマンド呼び出しする場合、フェーズの設定を行うには、つぎのオプションを付けて実行します。

/PHase [Open < n >], [Main < n >], [Close < n >]

<n>は呼び出しの回数で、それぞれのフェーズは順不同です。省略値は

/PHase Open0, Main\$, Close0

となっています。

■COBOLのCopy句を読み込む

COBOLのCopy句(COBOL登録集)などのファイルを直接読み込んで、データ記述項の文をMap文やAtlas文に自動展開することができます。ファイル中にデータ記述項が複数ある場合には、2つ目以降は無視します。登録集に限らず、COBOLの原始文中のデータ記述項も読み込み可能です。

データ記述項の中で、Map文やAtlas文に直接関係があるのは、つぎの4つの句です。

PICTURE句 Map文を生成する

USAGE句 Map文のデータ形式と入力幅に影響する

OCCURES句 Loop文を生成する

REDEFINES句 Select文を書くためのコメント文を生成する

SIGN句 Map文のデータ形式と入力幅に影響する

その他の句は、全て無視します。

◆注意 ---- 「PICTURE句」と「ピクチャ」の違い

Map文において入力幅や出力幅を入力する際に、小数点の有無や符号つきであるかどうかなどについても指定しますが、入出力幅および指定内容を表す形式をF*TRANでは「ピクチャ」と呼んでいます。COBOLの「PICTURE句」では符号つきかどうかだけでなく、データ形式(ゾーン、パック、漢字、ANKなどの形式)が何であるかについても指定しますので、その意味の違いにご注意ください。

ピクチャ (F*TRAN用語)

… Map 文の中で使用。入出力幅および符号の有無などを決めるための形式。データ形式については指定しない。

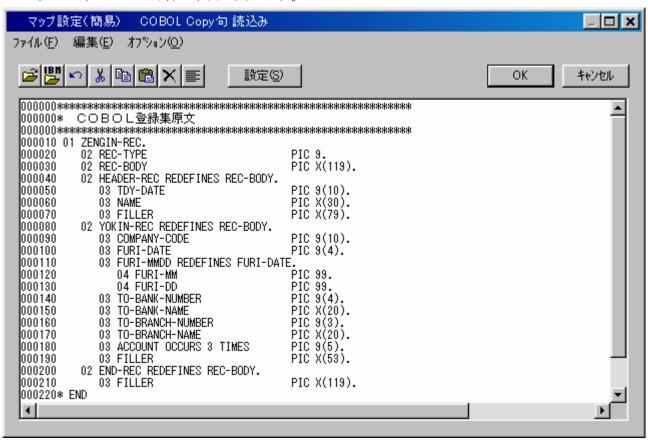
PICTURE句 (COBOL用語)

… ファイルのデータを読み書きする際にデータ形式およびデータ長を指定する句。

では実際に、どのような手順で行うか見てみましょう。

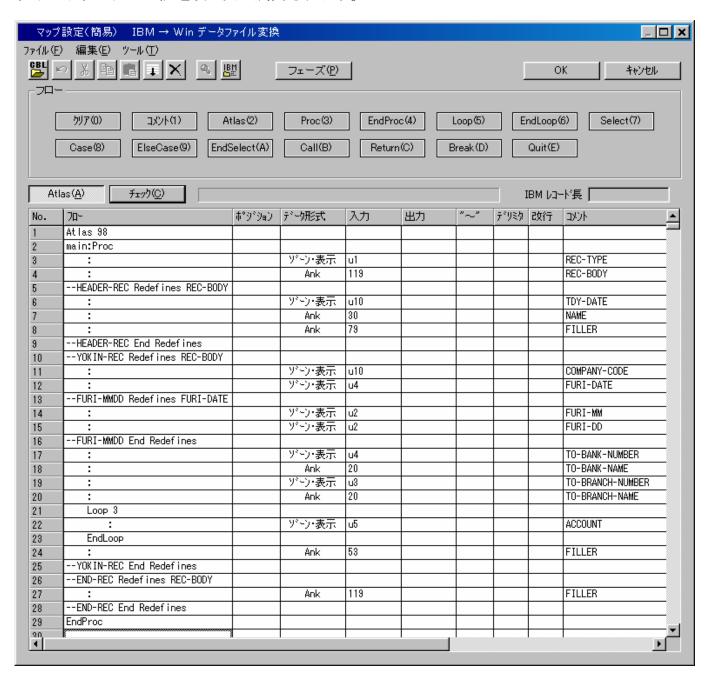
Copy句読み込みウインドウのメニューから、ファイル(F)→開く(O)(IBM形式のディスクから開くときは、ファイル(F) → IBMファイルを開く(I))を選択すると、ファイルの参照ウインドウが開きます。

COBOLのソースファイルを指定してOKボタンをクリックすると、Copy句読み込みウインドウにファイルの内容が表示されます。



COBOLの形式や変換方法に合わせて、読み込み設定を行う必要があります。設定ボタンをクリックして、Copy句読み込みウインドウから指定してください。詳細は前章のAtlas 98の使用法を参照してください。

内容を確認後(必要があれば編集して)、OKボタンをクリックすると、マップ設定(簡易) ウインドウにデータ記述項の文が展開されます。



◆注意 ---- データ名に含まれる全角スペースやタブ

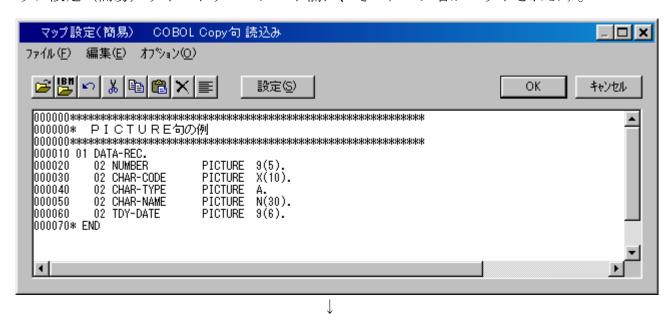
データ名などの利用者語に全角スペースやタブが含まれていた場合、正しく展開できない場合があります。あらかじめ、Copy句読み込みウインドウで編集/削除してください。

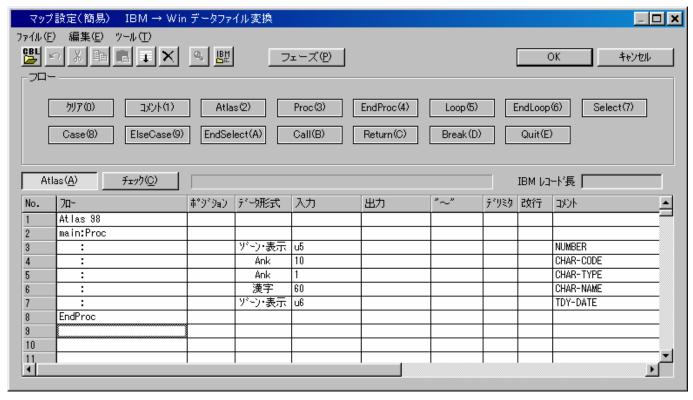
■Copy句読み込みの具体例

COBOLの句がどのようにAtlas/Map文に対応しているか、 $IBM \rightarrow Win$ データファイル変換を例にとって見てみましょう。

● PICTURE句

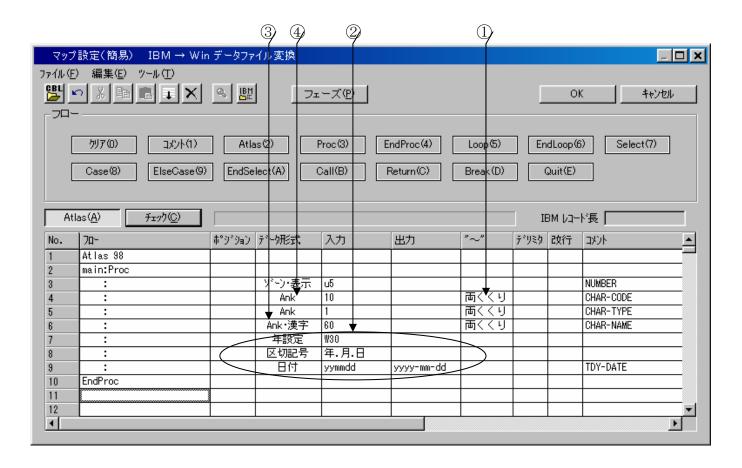
PICTURE句を含む1文が、マップ設定の1項目に対応しています。PICTURE句の文字列に応じたデータ形式と入力幅になります。PICTURE句でデータ名を指定した場合、マップ設定(簡易)ウインドウのコメント欄に、そのデータ名がセットされます。





自動展開によって、Map文のデータ形式と入力幅のみ生成されます。必要に応じてその他の 設定を手動で行ってください。

一般的に、次のような修正をする場合がよくあります。



- ①デリミタ形式へ変換するときは、データ形式がAnk、漢字、Ank・漢字などの文字列の項目に、引用符を付けることがよくあります。
- ②日付データ項目がある場合には、年設定、日付区切り設定、日付設定に変更してください。
- ③ PICTURE句の日本語項目 "N" は、漢字項目に自動展開されますが、KI/KOを含むときはAnk・漢字項目に変更してください。
- ④PICTURE句の英数字項目"X"は、Ank項目に自動展開されます。"X"に日本語文字列を含んでいるときは(おもにWindowsCOBOLの場合)、漢字項目(KI/KOを含むときはAnk・漢字項目)に変更してください。

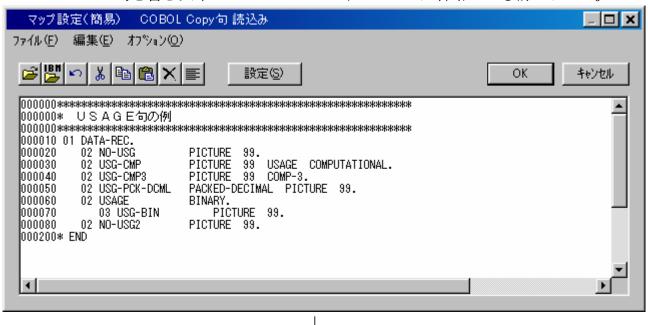
◆注意 ---- 数字項目の "P"

PICTURE句の数字項目の "P" はF*TRAN2007では扱えないデータ形式です。 COBOLのソースファイル中に数字項目の "P" が記述されている場合、数字項目の "9" に 置き換えて展開しています。そのため、正しいデータにならないことがあります。注意してくだ さい。

●USAGE句

PICTURE句から生成されたMap文のデータ形式と入力幅に、USAGE句の指定が反映されます。USAGE句を集団項目に指定した場合、その集団に含まれるPICTURE句のデータ全てに反映されます。

PICTURE句を含む文中の"USAGE"や"IS"は省略しても構いません。





<注意> USAGE句の指定では、ときどき、意図したものと違うデータ形式と入力幅に 展開されることがあります。その場合は、手動で変更してください。

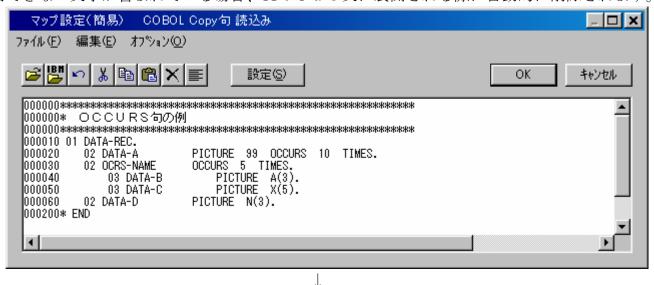
●OCCURS句

Atlas98のLoop文とくり返し回数、EndLoop文に対応しています。くり返し回数は、

OCCURS n TIMES
OCCURS m TO n TIMES

という記述の場合、両方とも同じ<n>回になります。

OCCURS句に指定したデータ名は、Loop文の構造名となります。ただし、構造名で使用できない文字が含まれている場合、Atlas文に展開される際に自動的に削除されます。



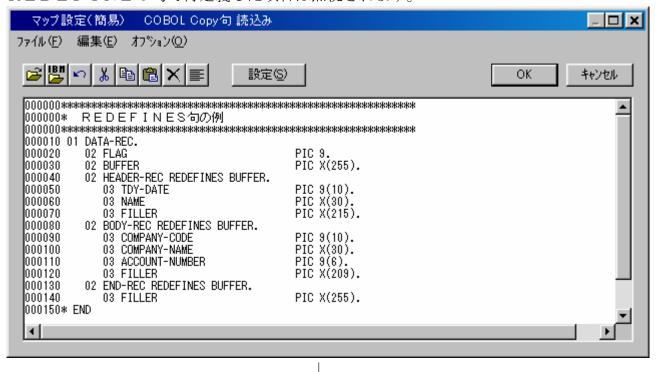


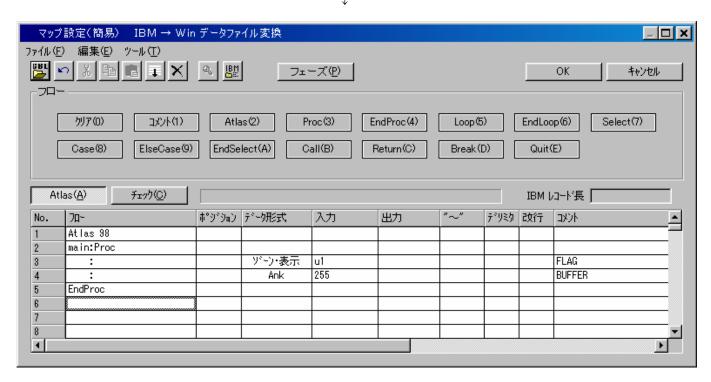
●REDEFINES句

REDEFINES句で再定義した項目を無視することも、マップに展開することもできます。 Copy句読み込み設定で、どちらかを指定してください。

REDEFINES句を生成しない場合

REDEFINES句で再定義した項目は無視されます。





REDEFINES句を生成する場合

REDEFINES句で再定義した項目をマップに展開します。再定義した項目の初めと終わりに、コメントが生成されます。たとえば、

A・・・REDEFINES句のデータ名(省略可)

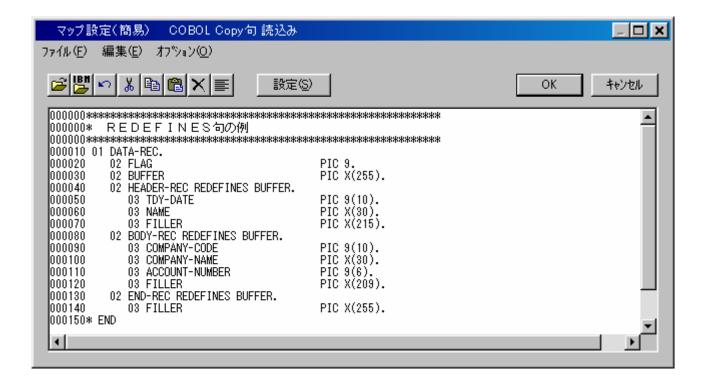
B・・・再定義される項目のデータ名

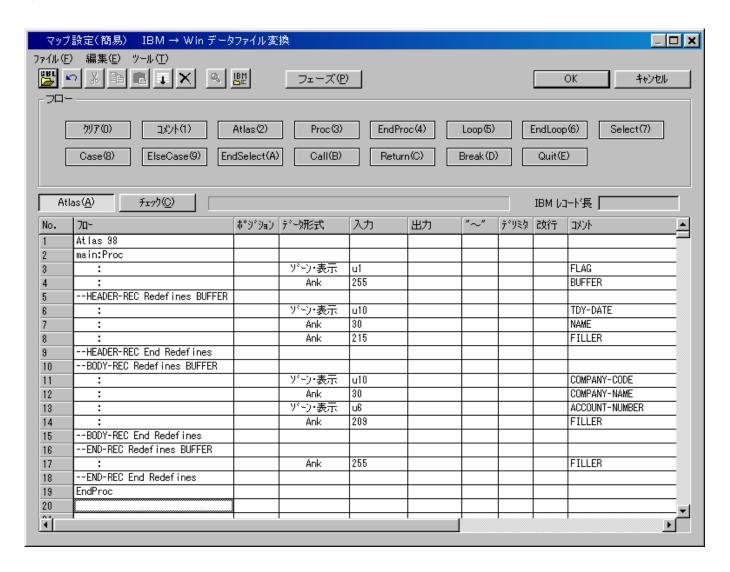
とすると、

```
-- A Redefines B
:
: 再定義した項目のマップ
:
-- A End Redefines
```

というコメントになります。

それでは、REDEFINES句を生成する場合を見てみましょう。例として使用しているCOBOLのソースは、REDEFINES句を生成しない場合の例と同じものです。





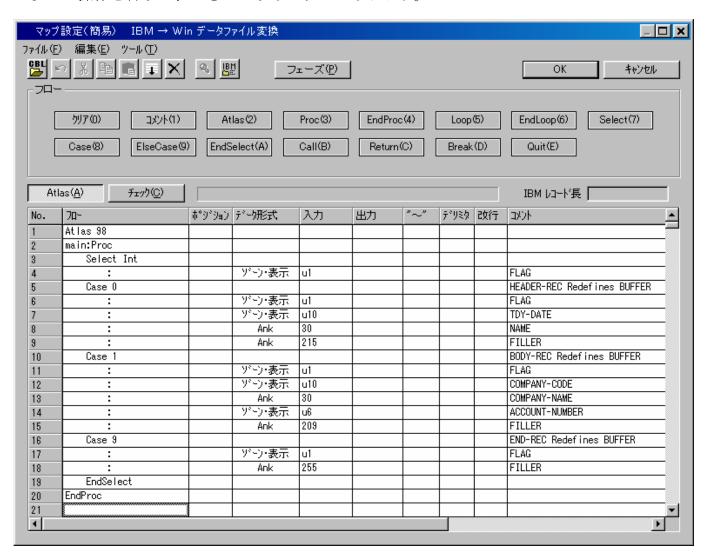
REDEFINES句の場合、コメントが生成されるだけなので、このままでは正しい設定にはなりません。このコメントを頼りに、マップ設定(簡易)ウインドウにて、手動で修正をする必要があります。

では実際に、先頭項目の「FLAG」の値を見て、ヘッダレコード/ボディレコード/エンドレコードに分岐するAtlas文にしてみましょう。手順はつぎのとおりです。

- ①行番号No. 2と3の間にSelect文を追加する。
- ②No. 4の「BUFFER」の行を削除する。(通常、再定義される項目は変換しない。)
- ③No. 5のRedefines句開始のコメント行を消して、変わりにCase文を 記述する。
- ④No. 5と6の間にセレクタとして使用した「FLAG」のMap文を追加する。
- ⑤No. 9のRedefines 句終了のコメント行を削除する。
- ⑥No. 10のRedefines句開始のコメント行を消して、変わりにCase文を 記述する。
- ⑦No. 10と11の間にセレクタとして使用した「FLAG」のMap文を追加する。

- ⑧No. 15のRedefines 句終了のコメント行を削除する。
- ⑨No. 16のRedefines句開始のコメント行を消して、変わりにCase文を 記述する。
- ⑩No. 16と17の間にセレクタとして使用した「FLAG」のMap文を追加する。
- ①No. 18のRedefines 句終了のコメント行を削除する。
- ⑫No. 19の前にEndSelect文を追加する。

以上の操作を行うと、つぎのようなマップになります。

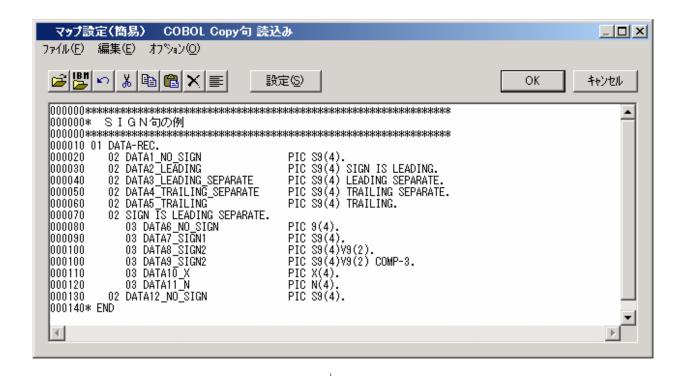


◆注意 ---- データ名の省略

REDEFINES句のデータ名は、省略されていても構いません。しかし、REDEFINES句が複数あったり、REDEFINES句で再定義した項目の中に2重でREDEFINES句が記述されていたりする場合には、分かりにくくなるので注意してください。

●SIGN句

PICTURE句から生成されたMap文の入力幅にSIGN句の指定が反映されます。 SIGN句は符号つきゾーン形式の項目にのみ有効です。それ以外の項目にSIGN句が指定されていても無視します。SIGN句を集団項目に指定した場合、その集団に含まれるPICTURE句の符号つきゾーン形式の項目全てに反映されます。PICTURE句を含む文中の "SIGN"や"IS"は省略しても構いません。





◆注意 ---- SEPARATE指定のないTRAILINGの形式**について**

SEPARATE指定のないTRAILINGの形式(符号が前づけで、分離していない形式) はサポートしていません。

●見だしレコード

見だしレコードとは、いわゆるヘッダのことです。通常、ヘッダを生成・付加する場合には、 そのためのマップを手動で書かなくてはなりません。しかし、COBOLのソースファイルを読 み込んでマップに展開する場合には、自動的に生成することができます。

これまでに見てきた例では、見だしレコード用のマップは生成されていません (本体レコード 用のマップのみ生成されています)。

見だしレコード用のマップを生成するには、Copy句読み込み設定ウインドウで、見だしレコードを生成する(I)に指定してください。そうすると、

```
{Select}
BY ~SysPhase *1
{Case 1}
:
: 見だしレコード用マップ
:
{Case 2}
:
: (本体レコード用マップ *2)
:
{EndSelect}
```

という文が自動的に追加されます。

見だしレコード用マップには、

Ank・漢字 「'PICTURE句のデータ名']

というMap文が、PICTURE句の項目数だけ生成されます。つまり、本体レコード用マップに展開された、すべてのPICTURE句のデータ名が挿入されます。ただし、データ名が省略されている場合、

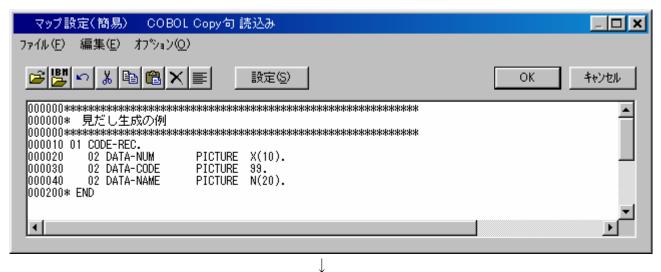
Ank·漢字 「'']

となります。また、OCCURS句でくり返しが指定されている場合は、本体レコード用マップと同様に、Loop文が生成されます。

- *1) SysPhaseの使い方についての詳細は、使用法のヘッダとトレーラの挿入方法を 参照してください。
- *2) 本体レコード用マップは、見だしレコードを生成しない場合と共通のマップです。

見だしレコードを生成する場合

それでは実際に、例を見てみましょう。





◆注意 ---- REDEFINES句と見だしレコード

REDEFINES句を生成する場合は、整合性がなくなるため、見だしレコードを生成しても意味がありません。Copy句読み込み設定で、REDEFINES句を生成する(G)と指定したときは、見だしレコードを生成する(I)は指定できないようになっています。

◆注意 ---- フェーズの設定

見だしレコードを生成するには、マップ設定の他にフェーズの設定を行う必要があります。見だしレコードは先頭に1行付け加えるのが一般的なため、オープン時の呼び出し回数を自動的に1に変更しています。見出しレコードを複数行付けたいときなどは、Copy句読み込み後、手動でフェーズの設定を行ってください。

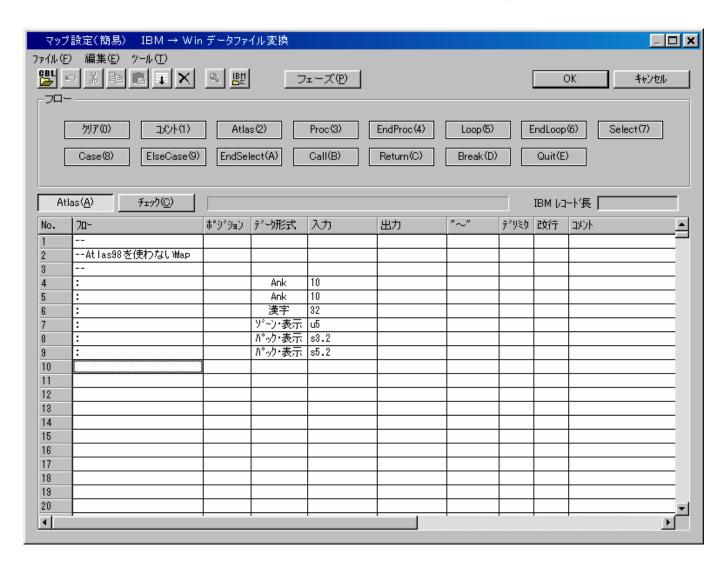
第4章

A t 1 a s 9 8 の利用例

4. 1 Atlas98の利用例

■従来互換

Atlas98モードで、Atlas文を使わない例を示します。

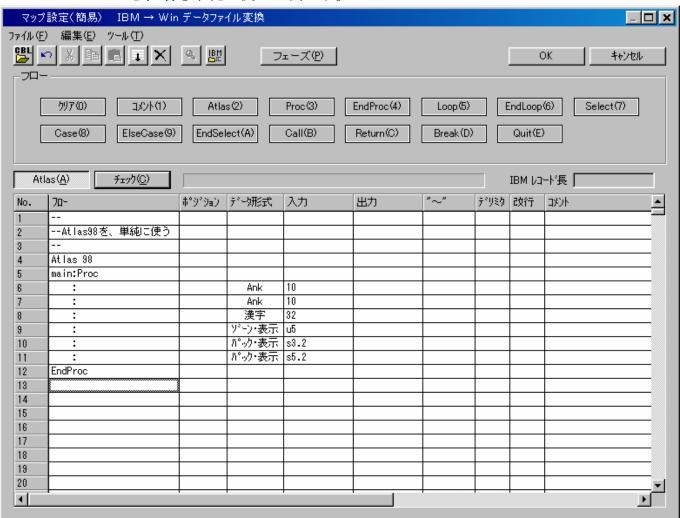


<パラメータファイルの記述>

```
--
-- Atlas98を使わないMap
--
ank 10
ank 10
kanji 32
zonedisp u5
packdisp s3. 2
packdisp s5. 2
```

■Atlas98化

Atlas98を、最も単純に使った例です。



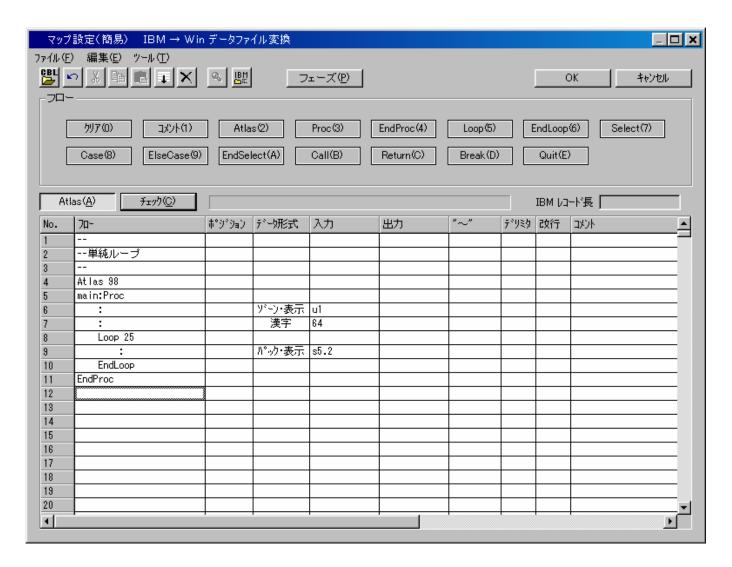
<パラメータファイルの記述>

```
--
-- Atlas98を、単純に使う
--
{Atlas 98}

{main: Proc}
ank 10
ank 10
kanji 32
zonedisp u5
packdisp s3. 2
packdisp s5. 2
{EndProc}
```

■単純なくり返し

レコードの後半が、同じパターンを25回くり返します。



<パラメータファイルの記述>

```
--

--単純ループ

--

{Atlas 98}

{main: Proc}

zonedisp u1

kanji 64

{Loop 25}

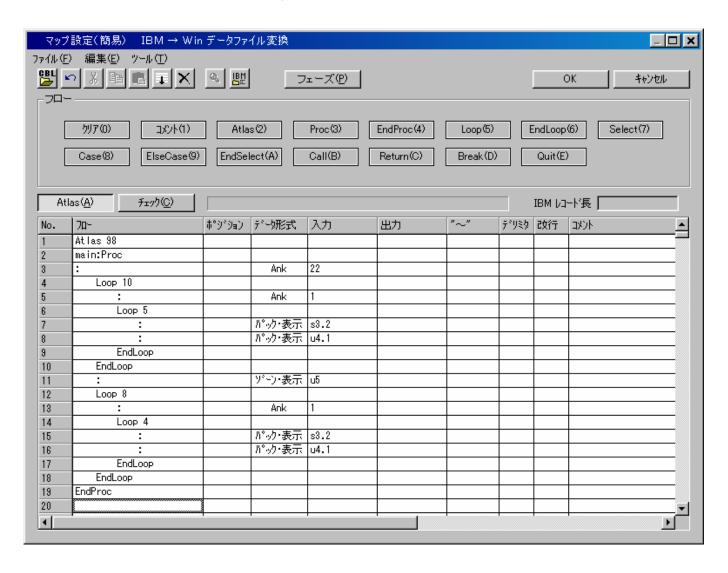
packdisp s5. 2

{EndLoop}

{EndProc}
```

■2重ループ

単純な2重ループが2回あります。

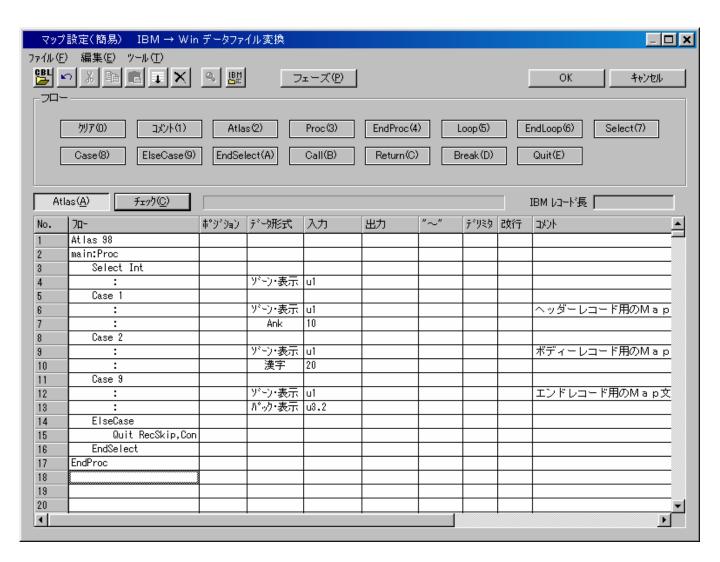


<パラメータファイルの記述>

```
{Atlas 98}
{main: Proc}
 ank
            2 2
 {Loop 10}
    ank
           1
    {Loop 5}
     packdisp s3.2
    packdisp u4.1
    {EndLoop}
 {EndLoop}
 zonedisp u5
  {Loop 8}
    ank
              1
    {Loop 4}
     packdisp s3. 2
     packdisp u4.1
    {EndLoop}
  {EndLoop}
{EndProc}
```

■マルチレイアウト変換を1つの手続きで設定する

IBMからWinへ、先頭の項目が、1、のときヘッダレコードとして、、2、のときボディーレコードとして、、9、のときエンドレコードとして変換します。その他のレコードは無視します。

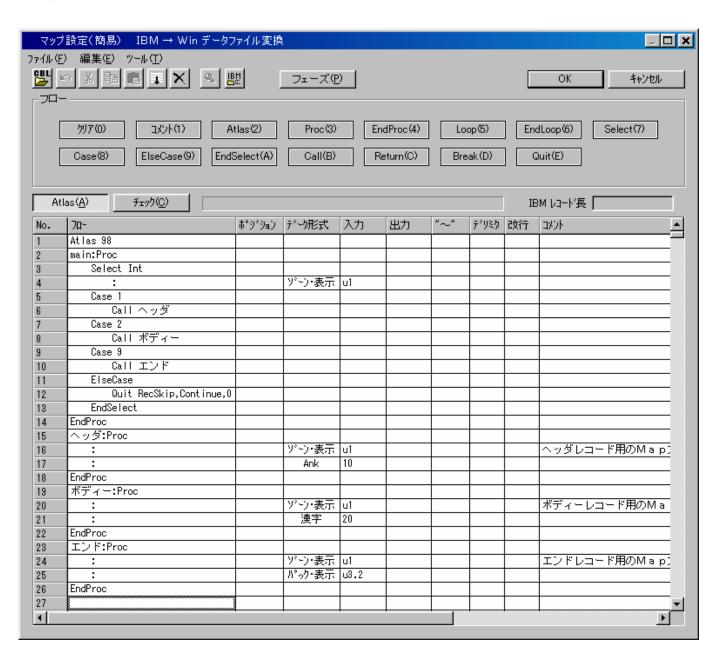


<パラメータファイルの記述>

```
{Atlas 98}
{main:Proc}
  {Select Int}
    zonedisp u1
  {Case 1}
    zonedisp
                     --ヘッダレコード用のMap文
              u 1
    ank
               1 0
  {Case 2}
                     --ボディーレコード用のMap文
    zonedisp
              u 1
    kanji
               2 0
  {Case 9}
    zonedisp u1 --エンドレコード用のMap文
    pack
               s 3. 2
  {ElseCase}
    \{Quit RecSkip, Continue, 0\}
  {EndSelect}
{EndProc}
```

■複数の手続きを活用する

IBMからWinへ、先頭の項目が、1、のときヘッダレコードとして、、2、のときボディーレコードとして、、9、のときエンドレコードとして変換します。その他のレコードは無視します。

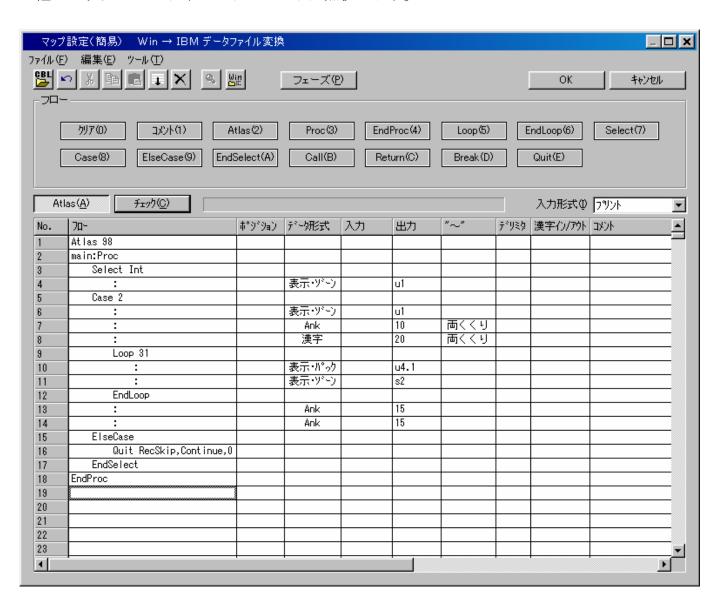


<パラメータファイルの記述>

```
{Atlas 98}
{main:Proc}
  {Select Int}
   zonedisp u1
  {Case 1}
    [Call ヘッダ]
  {Case 2}
    {Call ボディー}
  {Case 9}
    【Call エンド】
  {ElseCase}
    {Quit RecSkip, Continue, 0}
  {EndSelect}
{EndProc}
[ヘッダ: Proc]
                   --ヘッダレコード用のMap文
 zonedisp
            u 1
             10
 ank
{EndProc}
{ボディー: Proc}
                  --ボディーレコード用のMap文
 zonedisp
            u 1
 kanji
             20
{EndProc}
{エンド: Proc}
           и 1 --エンドレコード用のМар文
 zonedisp
             s 3. 2
 pack
{EndProc}
```

■ボディーレコードのみ変換する

Winから IBMへ、先頭の項目が、2、のときだけボディーレコードとして変換します。その他のヘッダレコードやエンドレコードは無視します。

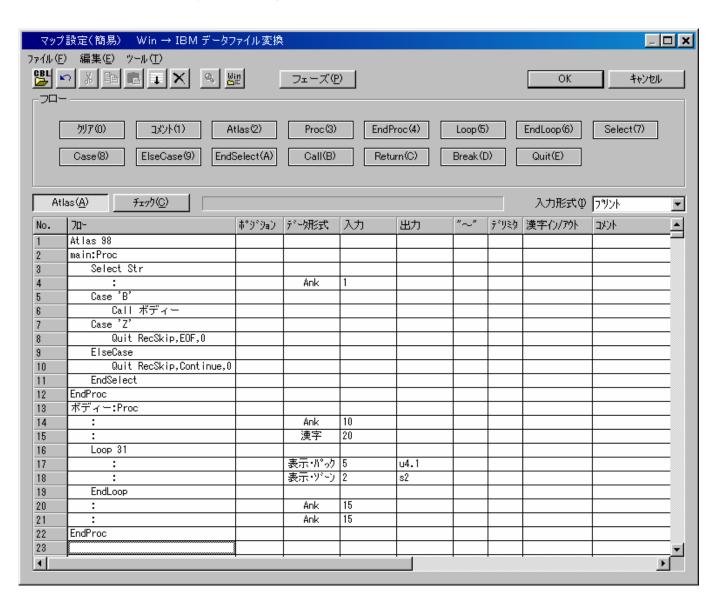


<パラメータファイルの記述>

```
[Atlas 98]
{main:Proc}
  {Select Int}
    dispzone : u1
  {Case 2}
     dispzone
               : u 1
    "ank
                : 10"
    "kanji
                : 20"
    [Loop 31]
      disppack
                : u 4. 1
      dispzone : s 2
    {EndLoop}
     ank
                 : 15
    ank
                : 15
  {ElseCase}
    {Quit RecSkip, Continue, 0}
  {EndSelect}
{EndProc}
```

■変換を中断する

WinhらIBMへ、先頭の項目が B'のときだけボディーレコードとして変換します。ただし、途中で Z'のレコードが現れたら、そこでこのファイルの変換を中断します。この Z'のレコードは変換結果に含めません。



```
{Atlas 98}
{main:Proc}
  {Select Str}
   ank
               1
  {Case 'B'}
    [Call ボディー]
  {Case 'Z'}
    {Qiut RecSkip, EOF, 0}
  {ElseCase}
    {Quit RecSkip, Continue, 0}
  {EndSelect}
{EndProc}
{ボディー: Proc}
 ank
             10
 kanji
             20
 {Loop 31}
   disppack 5:u4.1
   dispzone 2:s2
 {EndLoop}
 ank
             1 5
             15
 ank
{EndProc}
```

■最初にレコード末尾付近の区分項目を見る

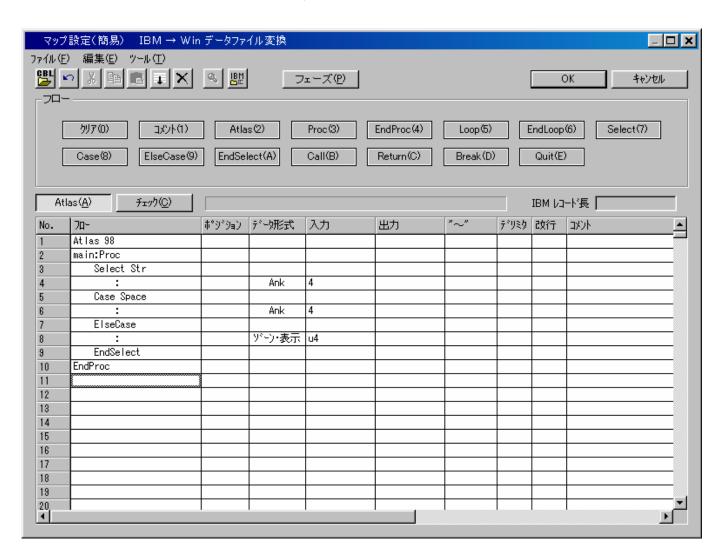
IBMからWinへ、レコードの末尾付近(128桁目)にある区分項目が、1、のときヘッダレコードとして、、2、のときボディーレコードとして、、9、のときエンドレコードとして変換します。その他のレコードは無視します。



```
{Atlas 98}
{main:Proc}
  {Select Int}
   @127 zonedisp
                      u 1
  {Case 1}
    @0
         kanji
                      2 0
                             ーヘッダ用のMap文
          ank
                      107
          zonedisp
                      u 1
  {Case
         2 }
                             --ボディ用のMap文
    @0
        kanjimix
                      2 0
                      107
          ank
          zonedisp
                      u 1
        9 }
  {Case
    @0
                      127
                            --トレーラ用のMap文
         ank
          zonedisp
                      u 1
  {ElseCase}
    {Quit RecSkip, Continue, 0}
  {EndSelect}
{EndProc}
```

■空白とゾーン形式を変換し分ける

空白(空項目)と0を区別して変換します(空項目にゾーン変換をかけると0になってしまうため、文字列型を指定して判定します)。



<パラメータファイルの記述>

```
{Atlas 98}

{main: Proc}

{Select Str}

ank 4

{Case Space}

ank 4

{ElseCase}

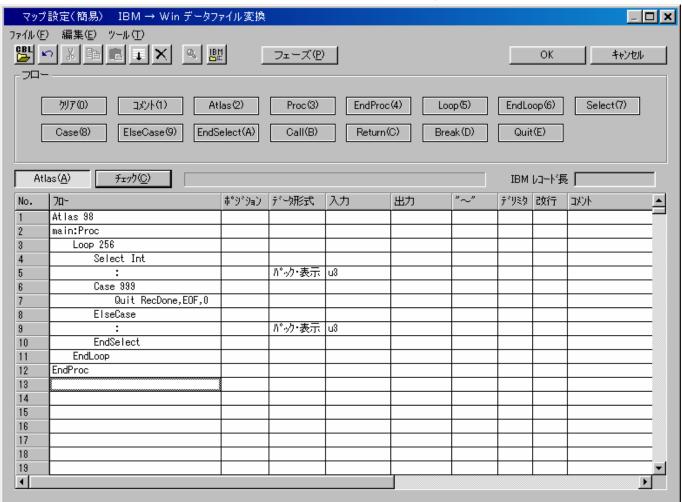
zonedisp u4

{EndSelect}

{EndProc}
```

■999で変換を止める

パック形式の配列(256個/レコード)を変換します。そのとき、値が999の項目が見つかったら、その直前まででそのファイルの変換を中断します。



<パラメータファイルの記述>

```
{Atlas 98}

{main: Proc}

{Loop 256}

{Select Int}

packdisp u3

{Case 999}

{Quit RecDone, EOF, 0}

{ElseCase}

packdisp u3

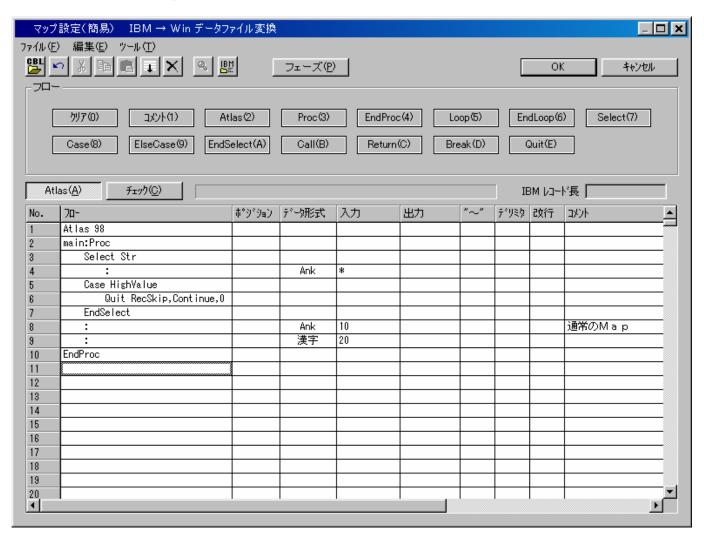
{EndSelect}

{EndLoop}

{EndProc}
```

■ F F … (ハイバリュー) の詰まったレコードをスキップする

オールFF(ハイバリュー)のレコードは削除レコードの意味としているシステムがあります。 これをスキップします。



<パラメータファイルの記述>

```
{Atlas 98}

{main:Proc}

{Select Str}

ank *

{Case HighValue}

{Quit RecSkip, Continue, 0}

{EndSelect}

ank 10 一通常のMap

kanji 20

{EndProc}
```

■負の売上を返品とみなす

返品レコードなどの場合、項目分けを変えたいことがあります。 Min_Max を利用すると、正負の指定ができます。

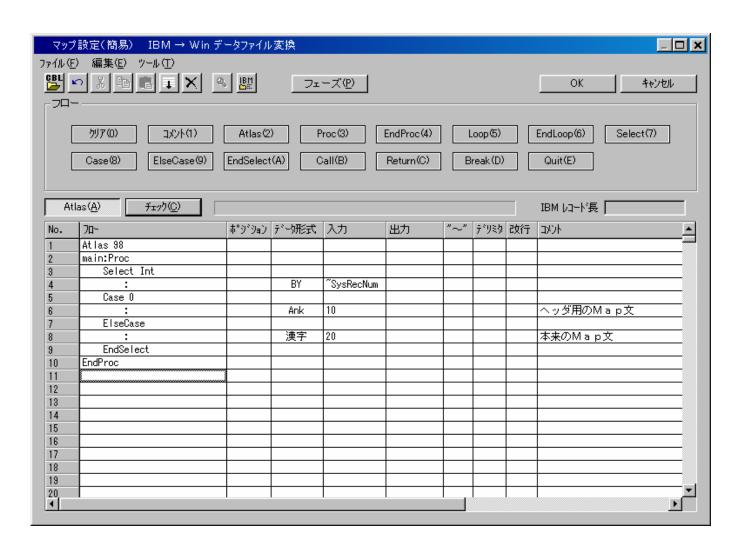


<パラメータファイルの記述>

```
{main: Proc}
  {Select Int}
    zonedisp s10
  {Case O. Max}
    zonedisp s10
                        ──売上用Map
                10
    ank
  \{Case Min.. -1\}
    zonedisp
                s 10
                        --返品用M a p
                20
    kanji
  {EndSelect}
{EndProc}
```

■先頭レコードだけ別レイアウト

IBMからWinへ、先頭レコードだけヘッダレコードとして変換し、それ以降は本来のレコードとして変換します。レコードタイプを示す項目はありません。



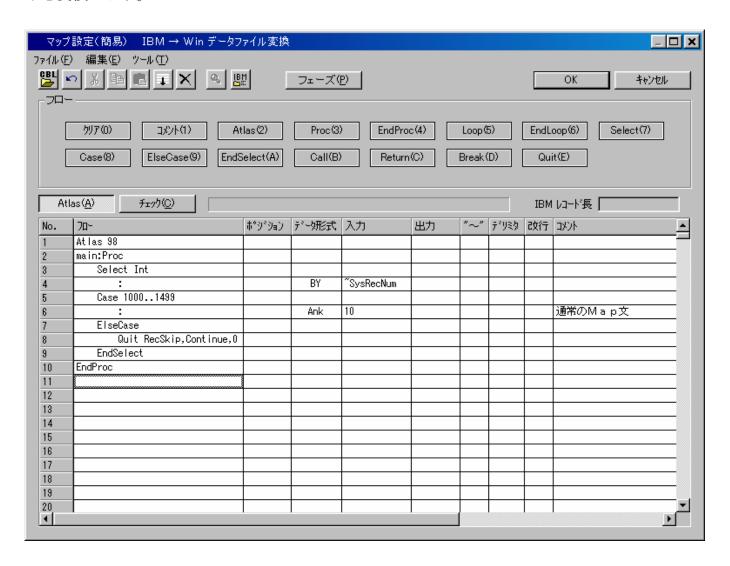
<パラメータファイルの記述>

```
{Atlas 98}

{main:Proc}
{Select Int}
BY ~SysRecNum
{Case 0}
ank 10 --ヘッダ用のMap文
{ElseCase}
kanji 20 --本来のMap文
{EndSelect}
```

■範囲指定して変換

IBM→Winファイル変換で、0から数えて1000番目のレコードから500レコードだけを変換します。



<パラメータファイルの記述>

```
{Atlas 98}

{main:Proc}
{Select Int}
BY ~SysRecNum
{Case 1000.. 1499}
ank 10 一通常のMap文
{ElseCase}
{Quit RecSkip, Continue, 0}
{EndSelect}
```

■ 4 レコード毎に同じパターン

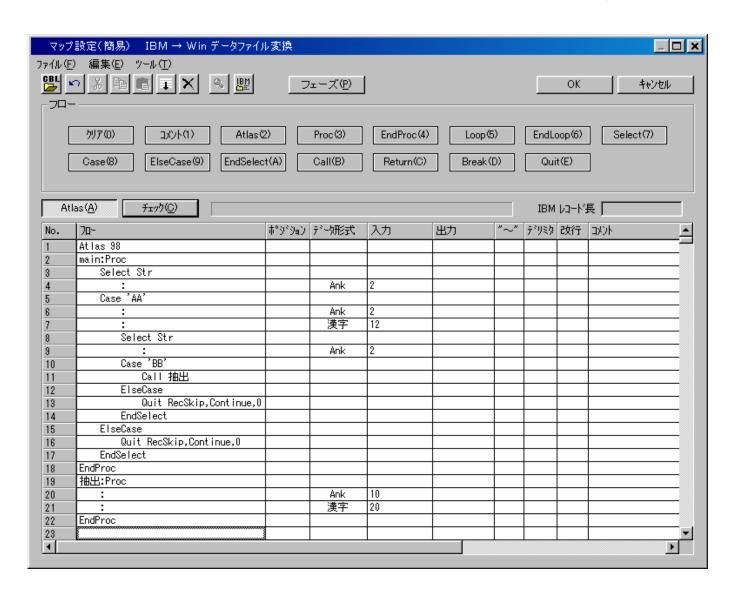
4 レコード毎に同じパターンのレコードをくり返すファイルを変換します。



```
{Atlas 98}
{main: Proc}
  {Select Int}
    ВΥ
                ~SysRecNum¥¥4
  Case
         0 }
                          --0番目のレコードのマップ
    ank
                10
  {Case 1}
                          -- 1 番目のレコードのマップ
    kanji
                2 0
  {Case 2}
                          --2番目のレコードのマップ
    zonedisp
                u 5
  {Case 3}
                         --3番目のレコードのマップ
    packdisp
                s 3. 2
  {EndSelect}
{EndProc}
```

■複合条件で抽出

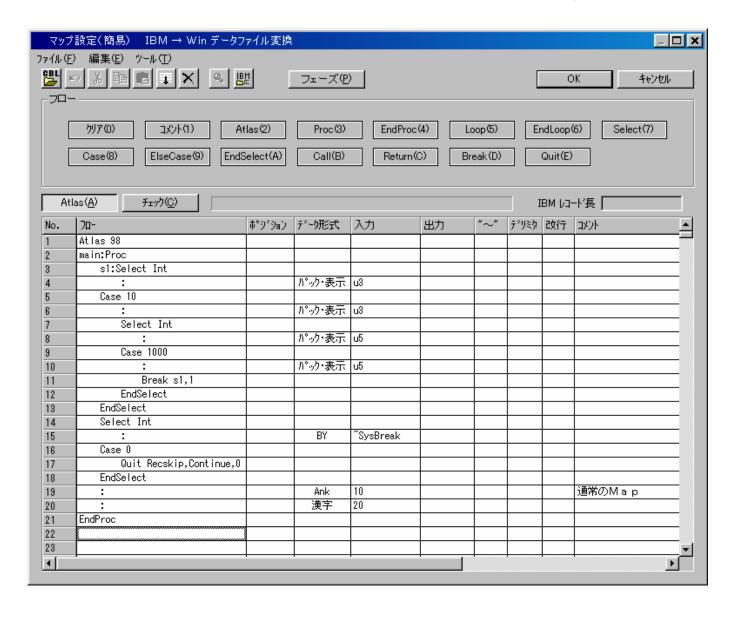
先頭の項目が'AA'で次々項目が'BB'のレコードだけを抽出変換します。



```
{Atlas 98}
{main:Proc}
  {Select Str}
    ank
                2
  {Case 'AA'}
    ank
                2
    kanji
               1 2
    {Select Str}
      ank
                  2
    {Case 'BB'}
      {Call 抽出}
    {ElseCase}
       {Quit RecSkip, Continue, 0}
    {EndSelect}
  {ElseCase}
    {Quit RecSkip, Continue, 0}
  {EndSelect}
{EndProc}
{抽出: Proc}
 ank
             10
 kanji
             20
{EndProc}
```

■複合条件と多重脱出

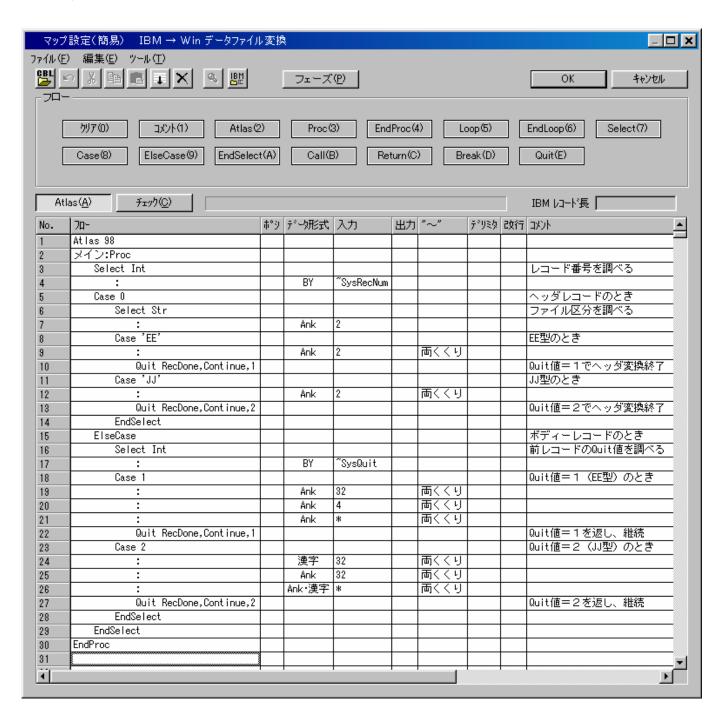
先頭の項目が10で次の項目が1000のレコードだけを抽出変換します。



```
{Atlas 98}
{main:Proc}
  {s1:Select Int}
    packdisp u3
  {Case 10}
    packdisp u3
    {Select Int}
      packdisp
                 u 5
    {Case 1000}
      packdisp
      {Break s1, 1}
    {EndSelect}
  {EndSelect}
  {Select Int}
    ВΥ
               ~SysBreak
  {Case 0}
    {Quit RecSkip, Continue, 0}
  {EndSelect}
                 --通常のMap
 ank
             10
 kanji
             2 0
{EndProc}
```

■ファイル区分別変換

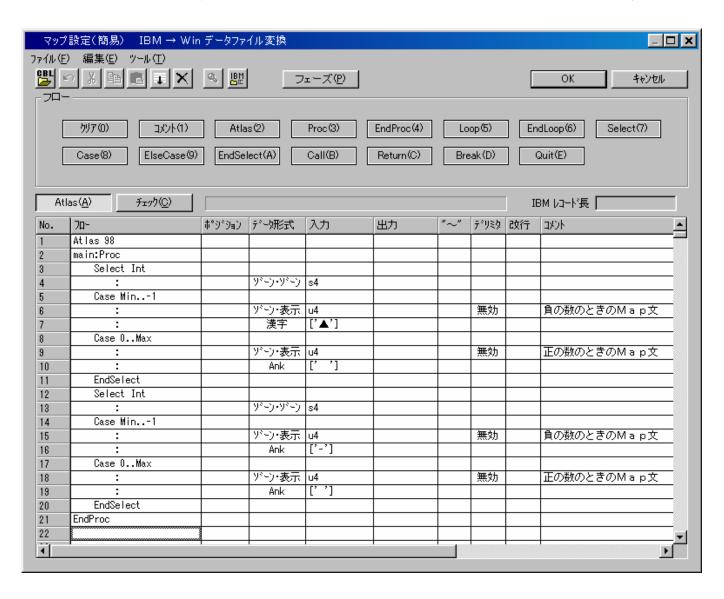
先頭レコードのファイル区分が $^{'}$ E E $^{'}$ のときと $^{'}$ J $^{'}$ のときで、本体レコードを変換し分けます。



```
{Atlas 98}
{main:Proc}
                            --レコード番号を調べる
  {Select Int}
                 ~SysRecNum
    ВΥ
                            --ヘッダレコードのとき
  {Case 0}
    {Select Str}
                            --ファイル区分を調べる
       ank
                            --EE 型のとき
     {Case 'EE'}
       "ank
       {Quit RecDone, Continue, 1}
                            --Quit 値= 1 でヘッダ変換終了
    {Case 'JJ'}
                            --JJ 型のとき
       "ank
                 2 "
       {Quit RecDone, Continue, 2}
                            --Quit 値= 2 でヘッダ変換終了
    {EndSelect}
                            --ボディーレコードのとき
 {ElseCase}
                            --前レコードの Quit 値を調べる
    {Select Int}
                    ~SysQuit
       ВΥ
     {Case 1}
                            --Quit 値= 1 (EE 型) のとき
       "ank
                 32"
       "ank
                 4 "
       "ank
                 * "
       {Quit RecDone, Continue, 1}
                            --Quit 値= 1 を返し、継続
    {Case 2}
                            --Quit 値= 2 (JJ 型) のとき
               32"
       "kanii
       "ank
                32"
       "kanjimix *"
       {Quit RecDone, Continue, 2}
                            --Quit 値= 2 を返し、継続
    {EndSelect}
  {EndSelect}
{EndProc}
```

■負の数のとき "▲"や "ー"を後付けする

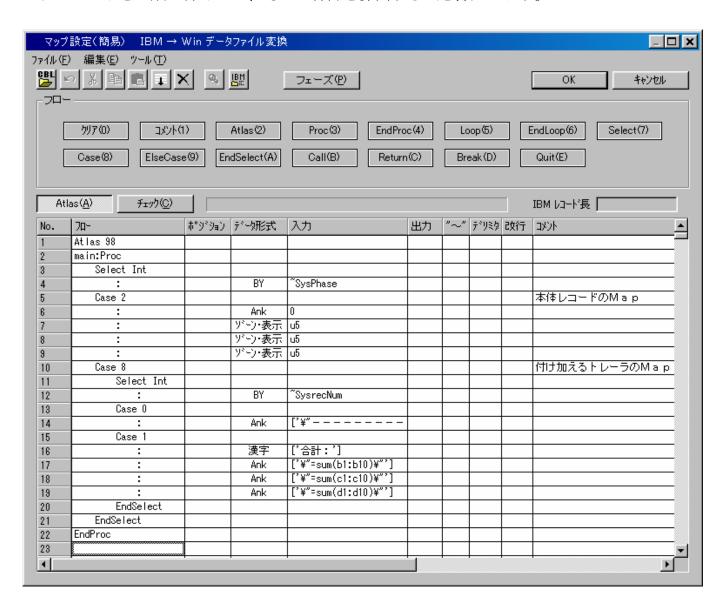
IBMからWinへ、値が負の数の項目に"▲"や"ー"を後付けして変換します。



```
{Atlas 98}
{main:Proc}
  {Select Int}
   zonezone
                 s 4
  \{Case\ Min...-1\}
                 u 4 &
                           ──負の数のときのMap文
    zonedisp
                 ['▲']
    kanji
  {Case O. Max}
    zonedisp
                 u 4 &
                           ──正の数のときのMap文
                 [''],
    ank
  {EndSelect}
  {Select Int}
    zonezone
                 s 4
  \{Case\ Min...-1\}
    zonedisp
                 u 4 & --負の数のときのMap文
                 [' -'] ,
    ank
  {Case O. Max}
                          ―正の数のときのMap文
    zonedisp
                 u 4 &
                 [''],
    ank
  {EndSelect}
{EndProc}
```

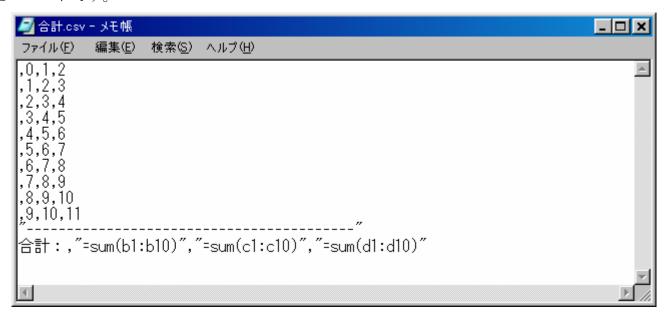
■Excelの式を埋め込む

IBMからWinへCSV形式に変換するときに、Excelon の計算式を埋め込みます。トレーラレコードを生成・付加して、そこに合計を算出する式を挿入します。

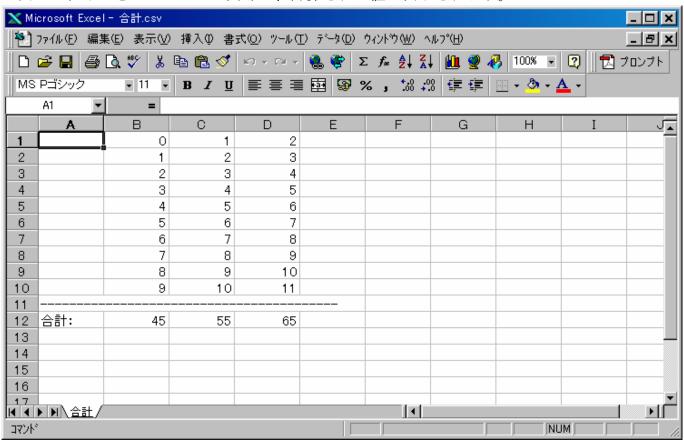


```
{Atlas 98}
{main:Proc}
 {Select Int}
            ~SysPhase
   ВΥ
                     --本体レコードのMap
 {Case 2}
   ank
             Ο,
   zonedisp
             u5,
   zonedisp
             u5,
   zonedisp
             u5,
 {Case 8}
                     --付け加えるトレーラのMap
   {Select Int}
              ~SysRecNum ,
     ВΥ
   \{Case 0\}
     ank
   {Case 1}
               ['合計:'],
     kanji
               ank
              ank
               ank
   {EndSelect}
 {EndSelect}
{EndProc}
```

変換後のファイルをテキストエディタで開くと、このようになります。下の2行が、付け加えたレコードです。



同じファイルをExcelで開くと、計算された値が表示されます。



第5章

マップ解析

5. 1 マップ解析の機能

F*TRAN2007には、マップ解析(マップ設定の動作を確認する)の機能が実装されています。通常は、この機能を使う必要はありませんが、意図した変換結果が得られない場合に、確認することができます。

マップ解析の機能を使うには、つぎの手順で行います。メニューバーの変換設定 (\underline{S}) →変換設定 (S) を実行し、つぎのウインドウを開きます。



- ①変換設定の、□マップ解析のチェックボックスをONにします。
- ②①のチェックボックスONの状態でOKボタンをクリックすると、メニューバーの ツール (T) の中にある "マップ解析 (J)" メニューが選択できるようになります。

マップ解析には、つぎの機能があります。

■Amダンプ (Atlas/Map Dumper)

Amダンプは、変換実行前のAmコード(マップ設定でコンパイルされた中間コード)をファイルに出力して確認する機能です。つぎの4つの中から目的に合ったメニューを選択し、出力するファイル名を指定して、実行します。

●Amダンプバイナリモード (<u>B</u>) → I BM→W i n (<u>G</u>) Amダンプバイナリモード (<u>B</u>) →W i n → I BM (<u>P</u>)

Amコードをそのままバイナリでファイルに出力します。

● Amダンプテキストモード (\underline{T}) → I BM→W i n (\underline{G}) Amダンプテキストモード (T) →W i n → I BM (P)

Amコードを逆コンパイルリストにしてファイルに出力します。

<出力例>マップ設定(詳細)IBM→Winで"MultiRec. P"を読み込み、 Amダンプテキストモード(IBM→Win)で出力する。

```
{Atlas 98}
{main:Proc}
     {Select Str}
          Ank 2
     {Case ' 2'}
          {Call map}
     {ElseCase}
          {Quit RecDone, Continue, 0}
     {EndSelect}
     {EndProc}
          Ank 2
          Kanji 8
          {EndProc}
```

■Amトレース (Atlas/Map Tracer)

Amトレースは、変換実行時に蓄えられたAmコード(マップ設定でコンパイルされた中間コード)を、1ステップ毎のテキストリストにしてファイルに出力します。Amトレースを採取する手順はつぎのとおりです。

- ①マップ解析メニューのAmトレースON(O)の項目を選択し、AmトレースONの状態にします。
- ②Amトレースを採取するファイル変換を実行します。
- ③マップ解析メニューのAmトレース(<u>A</u>)の項目を選択し、 出力するファイル名を指定して、実行します。

<出力例>マップ設定(詳細) I BM→Winで"MultiRec.P"を読み込み、サンプルファイル"Planet"を変換した後、Amトレースで出力する。

A:PLANE	T -	→ C:¥PLANET データファイル変換.
000000	0	{Atlas 98}
00000D	0	SysRecNum=0 SysReturn=0 SysBreak=0 SysPhase=2 SysQuit=0
00001A	1	{Select Str} SysRecNum=0 SysReturn=0 SysBreak=0 SysPhase=2 SysQuit=0
000027	2	Ank 2 0 - 4F 01 1
00002E	2	{Case ' 2'} (不成立) SysRecNum=0 SysReturn=0 SysBreak=0 SysPhase=2 SysQuit=0

F*TRAN2007 V2. 0 操作説明書/マルチレコード編

2010年 4月 第1版発行

編集・著作 株式会社 富士通ビー・エス・シー 所 在 地 〒135-8300 東京都港区台場 2-3-1 トレードピアお台場

- ・Windows、MS-DOS、Visual Basic、Access、Visual C++、Excel は 米国 Microsoft Corp.の米国およびその他の国における登録商標または商標です。
- ・Acrobat は Adobe Systems Inc. (アドビシステムズ社) の商標です。
- ・F*TRAN は富士通ビー・エス・シーの登録商標です。
- ・会社名および製品名はそれぞれ各社の商標または登録商標です。
- ・本書およびシステムは、改善のため事前連絡なしに変更することがあります。
- ・無断複製、および転載を禁じます。