

F * T R A N +

V 3 . 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 * T R A N + (エフトラン・プラス) は、汎用機 / オフコン / U n i x などのホストのファイル転送データと、パソコンの標準である W i n d o w s ファイルとのデータ交換をする汎用性の高いファイル変換ユーティリティです。W i n d o w s ファイル間のデータ変換もできます。

W i n d o w s 9 8 / 9 5 または W i n d o w s N T / 2 0 0 0 上の 3 2 ビットアプリケーションとして動作します。

ソースプログラム、バイナリファイル、ランダムファイル、プリント形式ファイルなどを変換する基本機能に加え、コンマ区切り (C S V) 形式対応など、市販ソフトとのデータ交換に適した強力なデータ加工・編集機能を備えています。また、各種漢字コードに対応し、拡張漢字にも本格対応しています。さらに、C O B O L の数値項目等 (ゾーン形式、パック形式、2 進形式、B C D 形式) にも対応しています。

F * T R A N + は、さまざまなホストとパソコンの連携利用を強力に支援します。その高い汎用性、高性能、高機能が有効に活用されることを願ってやみません。

F * T R A N + 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。マルチレコード編 (本書) の構成はつぎのとおりです。

第 1 章 A t l a s 9 8 とは ?

A t l a s 9 8 の概要を説明しています。

第 2 章 A t l a s 9 8 の文法

A t l a s 9 8 の文法を説明しています。

第 3 章 A t l a s 9 8 の使用法

A t l a s 9 8 の使用方法を説明しています。

第 4 章 A t l a s 9 8 の利用例

A t l a s 9 8 の利用例を説明しています。

第 5 章 マップ解析

マップ解析の機能と操作を説明しています。

目 次

第 1 章 A t l a s 9 8 とは？

1.1	A t l a s 9 8 とは？	2
1.2	何ができるようになるか？	3

第 2 章 A t l a s 9 8 の文法

2.1	A t l a s 9 8 の構文	6
2.2	A t l a s 9 8 の文法	8

第 3 章 A t l a s 9 8 の使用法

3.1	A t l a s 9 8 モード	16
3.2	A t l a s 9 8 の使用法	18
	■ A t l a s 9 8 を使わない場合	18
	■ L o o p 文を使う方法	19
	■ 簡単な S e l e c t 文を使う方法	20
	■ 途中からマルチレイアウトになる場合	22
	■ 複数の手続きを使う場合	23
	■ 不要なレコードを捨てる場合	24
	■ 小数項目による分岐を行う場合	26
	■ 文字列による分岐を行う場合	28
	■ レコード番号による分岐を行う場合	29
	■ 脱出文の使用法	30

第 4 章 A t l a s 9 8 の利用例

4.1	A t l a s 9 8 の利用例	34
	■ 従来互換	34
	■ A t l a s 9 8 化	35
	■ 単純なくり返し	36
	■ 2 重ループ	38
	■ マルチレイアウト変換を 1 つの手続きで設定する	40
	■ 複数の手続きを活用する	42
	■ ボディーレコードのみ変換する	44
	■ 変換を中断する	46

■ 空白とゾーン形式を変換し分ける	4 8
■ 9 9 9 で変換を止める	4 9
■ F F ・ ・ ・ (ハイバリュー) の詰まったレコードをスキップする	5 0
■ 負の売上を返品とみなす	5 1
■ 先頭レコードだけ別レイアウト	5 2
■ 範囲指定して変換	5 3
■ 4 レコード毎に同じパターン	5 4
■ 複合条件で抽出	5 6
■ 複合条件と多重脱出	5 8

第 5 章 マップ解析

5.1 マップ解析の機能	6 2
--------------------	-----

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 B 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。						
<table><tr><td>A</td><td> </td></tr><tr><td>B</td><td> </td></tr><tr><td>C</td><td> </td></tr></table>	A		B		C		同上。
A							
B							
C							
(A / B / C)	同上。						
[A]	Aは省略できます。						
[A / <u>B</u> / 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、下線を引いたBを選択した ものとみなします。						
<table><tr><td>A</td><td> </td></tr><tr><td><u>B</u></td><td> </td></tr><tr><td>C</td><td> </td></tr></table>	A		<u>B</u>		C		同上。
A							
<u>B</u>							
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、nn、< 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)						
x x H	16進でx x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。						
<u>a</u>	下線部を入力します。						
<u>a b c</u> ↓	下線部を入力し、リターンキーを押します。						
C T R L + A	コントロール (C T R L) 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 l a s 9 8とは？

1 . 1 A t l a s 9 8 とは？

COBOLのプログラムでは、よくOCCURS句を使って繰り返し項目をまとめたり、REDEFINE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 * T R A N + でこれらに対応するのは大変でした。まず、繰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 * T R A N + 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 l a s 9 8 (アトラス98)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 l e c t 文」、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。

A t l a s 9 8 は、マップ設定 (/ M A P オプション) の拡張言語です。つまり、マップ設定 (/ M A P オプション) では、A t l a s 9 8 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 (/ M A P オプション) との上位互換が保たれています。そして、A t l a s 9 8 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？

A t l a s 9 8 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。

第2章

A t l a s 9 8の文法

2.1 Atlas 98の構文

■ Atlas 98の構文

Atlas 98の構文は、つぎの13文です。

Atlas	--->	Atlas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
Call	--->	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

Atlas 98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、Atlas 98の中核機能をなしています。

Atlas 98は、構造化された言語です。Atlas 98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）。

Atlas 98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 ---- A t l a s 構文を指定する時の決まり事 (コマンド型での指定)

- A t l a s 構文は、M a p 文との区別のため、ブレース ({ と }) でくくる必要があります。
- A t l a s 構文のキーワードは短縮できません。
- A t l a s 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 A t l a s 9 8 の文法

■ A t l a s 文 (A t l a s 宣言)

A t l a s 構文の使用を宣言します。

```
{ A t l a s 9 8 }
```

マップ設定 (/ M A P オプション) に A t l a s 構文を指定する場合は、必ず A t l a s 宣言から始めます。

■ P r o c 文 (手続き開始)

E n d P r o c 文 (手続き終了)

P r o c 構造は、複数の文 (M a p 文、A t l a s 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 : P r o c }  
    M a p 文 または A t l a s 文  
{ E n d P r o c }
```

マップ設定 (/ M A P オプション) に A t l a s 構文を指定する場合は、A t l a s 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。A t l a s 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 ~ Z)、英小文字 (a ~ z)、数字 (0 ~ 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はいけません。また、全角スペースを含んでははいけません。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。

E n d P r o c 文で、手続きを終了します。

■ L o o p 文 (くり返し開始)

E n d L o o p 文 (くり返し終了)

回数の決まったくり返し構造を構築します。

```
{[構造名:]L o o p 回数}
    Map文 または A t l a s 文
{E n d L o o p }
```

構造名は、英大文字 (A ~ Z)、英小文字 (a ~ z)、数字 (0 ~ 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はいけません。また、全角スペースを含んでははいけません。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。

L o o p 文で、くり返しの回数 (0 ~ 3 2 7 6 7) を指定します。

E n d L o o p 文で、くり返し指定を終了します。

■ S e l e c t 文 (多分岐開始)

C a s e 文 (条件指定)

E l s e C a s e 文 (その他条件)

E n d S e l e c t 文 (多分岐終了)

多分岐構造を構築します。S e l e c t 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:]S e l e c t 選択型}
    セレクタ ( Map文 )
    { C a s e 範囲並び1 }
        Map文 または A t l a s 文
    [ { C a s e 範囲並び2 }
        Map文 または A t l a s 文 ] ...
    [ { E l s e C a s e }
        Map文 または A t l a s 文 ]
{E n d S e l e c t }
```

構造名は、英大文字（A～Z）、英小文字（a～z）、数字（0～9）、全角文字で構成されます。ただし、先頭が数字であってははいけません。また、全角スペースを含んでははいけません。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。

S e l e c t 文の選択型の指定は、つぎの3つです

I n t 整数型 (I n t e g e r)
 D e c 小数型 (D e c i m a l)
 S t r 文字列型 (S t r i n g)

セレクトとは、多分岐の基準になる項目のことで、各 C a s e 値と比較されます。有効な M a p 文を1つ指定します。

セレクトに指定した M a p 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 M a p 文を C a s e 文や E l s e C a s e 文の後に再度、指定します。

選択型と有効なセレクトの組み合わせは、下表の組み合わせになります。

M a p 文	I n t / D e c	S t r i n g
DispZone / ZoneDisp	○	×
DispPack / PackDisp	○	×
DispBin / BinDisp	○	×
ZonePack / PackZone	○	×
ZoneBin / BinZone	○	×
PackBin / BinPack	○	×
ZoneZone / PackPack	○	×
BinBin	○	×
N u m e r i c	×	×
A n k	×	○
K a n j i	×	○
K a n j i M i x	×	○
B Y	○	×

N u m e r i c 変換は数値比較も文字列比較もできない。

C a s e 文は、変換する条件を指定します。S e l e c t 文には、少なくとも1つのC a s e 文を含まなければなりません。

{ C a s e 範囲並び }

範囲並びは、C a s e 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D」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)
M i n	最小値
M a x	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)
M i n	最小値
M a x	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	
S p a c e	空白文字列(半角、全角の空白からなる文字列)	
L o w V a l u e	00H...の文字列	
H i g h V a l u e	FFH...の文字列	ANK変換表をFF→FF に変更する必要がある。

文字列の比較は、ホスト側文字列であれば、W i n 側文字列に変換した文字列の比較を行います。基本的にJ I S 8 / シフトJ I S コード順で大小比較をしますが、連続する空白はC a s e 値、セレクト値ともに1個の空白にまとめてから比較します。文字の大小関係の法則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

E l s e C a s e 文は、その他の場合の変換方法を指定します。E l s e C a s e 文は省略できます。

E n d S e l e c t 文で、多分岐指定を終了します。

■ B Y 变换

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)

指定できるシステム変数は、つぎの3つです。

~S y s R e c N u m	レコード番号 (0 ~)
~S y s R e t u r n	返り値 (R e t u r n 文で代入された値)
~S y s B r e a k	ブレイクコード (B r e a k 文で代入された値)

指定例 : B Y ~ S y s R e c N u m ¥ ¥ 4 レコード番号を4で割った余り値
(セレクト値は0～3の値になる)

■ C a l l 文（手続きの呼び出し）

手続きを呼び出します。

```
{ C a l l   手続き名 }
```

手続き名は、P r o c 文で指定した手続き名を指定します。

■ R e t u r n 文（手続きからの復帰）

手続きから復帰します。

```
{ R e t u r n   [整数値 | システム変数名] }
```

返り値として、整数値（ - 3 2 7 6 8 ~ 3 2 7 6 7 ） 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3つです。

S y s R e c N u m	レコード番号（ 0 ~ ）
S y s R e t u r n	返り値（ R e t u r n 文で代入される）
S y s B r e a k	ブレークコード（ B r e a k 文で代入される）

実行時に { E n d P r o c } に達した時は、{ R e t u r n 0 } が実行されたとみなされます。

■ B r e a k 文 (構造からの抜け出し)

L o o p 構造、S e l e c t 構造から抜け出します。

```
{ B r e a k  [構造名][ , 整数値 | システム変数名 ] }
```

構造名を省略すると、その B r e a k 文が直接属している L o o p 構造 / S e l e c t 構造を抜け出し、構造名を指定すると、指定された L o o p 構造 / S e l e c t 構造を抜け出します。

その時にブレークコードを指定することができます。ブレークコードは、手続きが開始時に自動的に 0 にセットされます。B r e a k 文では、ブレークコードとして整数値 (- 3 2 7 6 8 ~ 3 2 7 6 7) または、システム変数の値を指定します。これを、その後の S e l e c t 文で判定することができます。なお、ブレークコードは C a l l 文の実行に影響されません。

システム変数は、つぎの 3 つです。

S y s R e c N u m	レコード番号 (0 ~)
S y s R e t u r n	返り値 (R e t u r n 文で代入される)
S y s B r e a k	ブレークコード (B r e a k 文で代入される)

■ Q u i t 文 (処理の中断)

処理を中断します。

```
{ Q u i t  [ R e c S k i p | R e c D o n e ][ , C o n t i n u e | E O F ] }
```

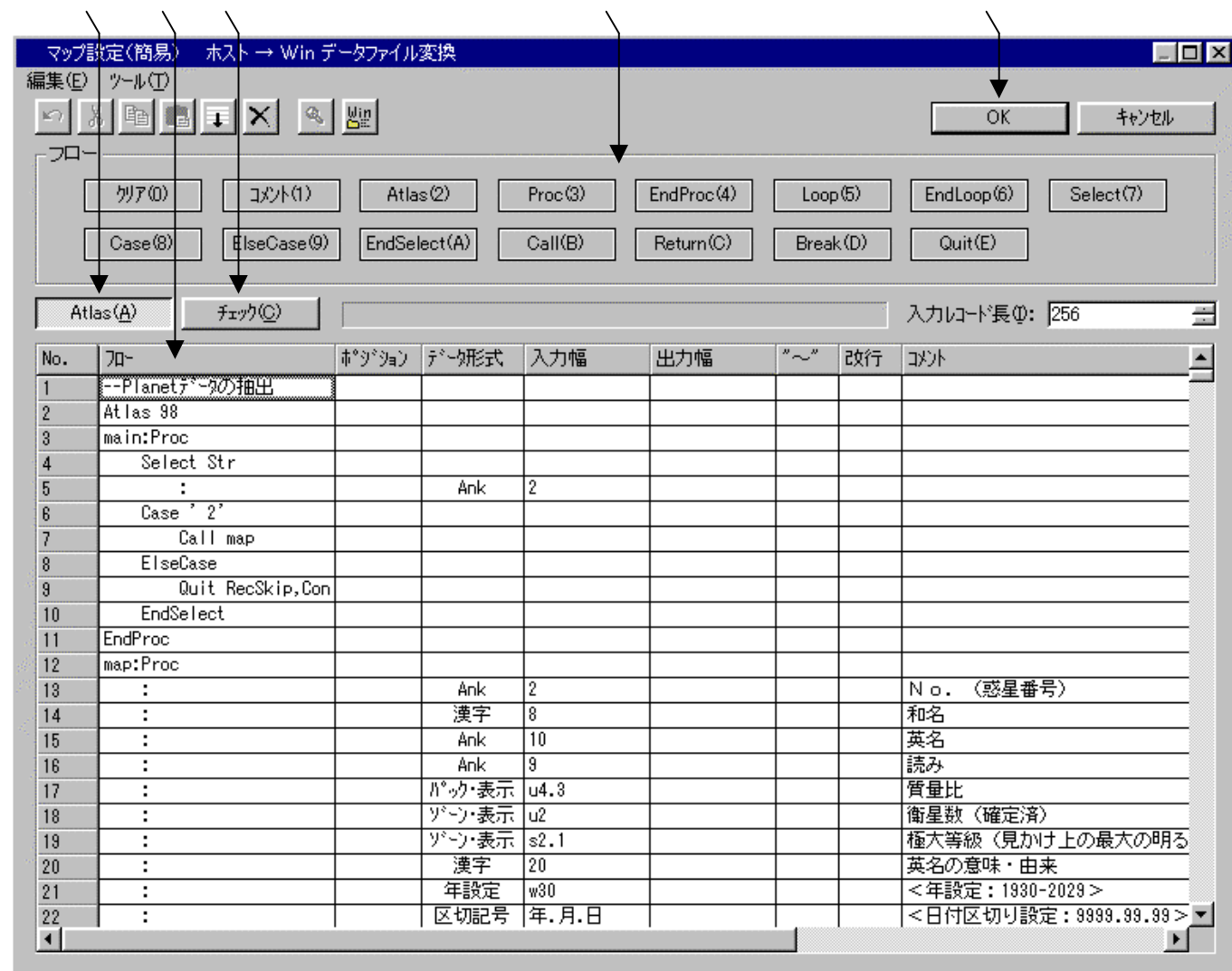
R e c S k i p	処理中のレコードの処理をスキップします。 そのレコードのそこまでの変換結果は放棄されます。
R e c D o n e	処理中のレコードの処理を完了します。 そのレコードのそこまでの変換結果は有効になります。
C o n t i n u e	そのファイルの処理を継続します。
E O F	そのファイルの処理を中断・完了します。 そのファイルのそこまでの変換結果は有効になります。

第3章

A t l a s 9 8 の使用法

3.1 A t l a s 9 8 モード

マップ設定（簡易）ウインドウで A t l a s 9 8 を設定します。



A t l a s ボタンをクリックして、A t l a s モードにします。

再度、A t l a s 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 A t l a s 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 A t l a s 設定がある場合には、自動的に A t l a s モードになります。

セルポイントがフロー欄にある場合に、A t l a s 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なうと、A t l a s 設定ウインドウが開きます。

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して、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なうことができます。

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 A t l a s 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。

O K 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。

< A t l a s 設定ウインドウ >

つぎの A t l a s 設定ウインドウで、A t l a s 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 A t l a s 設定

マップ設定 (詳細) で A t l a s 設定を行う場合は、A t l a s 文をブレース ({ と }) でくくる記述を行います。F * T R A N + をコマンド型で実行する際の / M A P 指定の記述、または、その時に使用するパラメ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。

3.2 A t l a s 9 8 の使用法

■ A t l a s 9 8 を使わない場合

まず、A t l a s モードで A t l a s 9 8 を使わない方法を 1 つ見てみましょう。

No.	フロー	ホスト名	データ形式	入力幅	出力幅	変数	改行	コメント
1	--							
2	-- A t l a s 9 8 使わず							
3	--							
4	:		ゾーン表示	u3				
5	:		ゾーン表示	s5				
6	:		Ank	20				
7								
8								
9								
10								
11								
12								
13								
14								
15								
16								
17								
18								
19								
20								
21								
22								

フロー欄というのが先頭にあります。ここにコメントや A t l a s 9 8 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判ると思います。ここは重要なところですよ。つまり、積極的に A t l a s 9 8 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。

■ L o o p 文を使う方法

では手始めに、A t l a s 9 8 の一番簡単な機能を使ってみましょう。A t l a s 9 8 で一番簡単なのは、L o o p 文です。これは、固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。

No.	フロー	ホップション	データ形式	入力幅	出力幅	"~"	改行	コメント
1	--							
2	-- 8回くり返す							
3	--							
4	Atlas 98							
5	main:Proc							
6	Loop 8							
7	:		データ表示	u8				
8	:		データ表示	s5				
9	:		Ank	20				
10	EndLoop							
11	EndProc							
12								
13								
14								
15								
16								
17								
18								
19								
20								
21								
22								

L o o p 8 ~ E n d L o o p というのは、この中を 8 回くり返せという意味です。

注意 ---- A t l a s 9 8 のルール

A t l a s 9 8 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、A t l a s 9 8 と書くこと
実行したい部分を P r o c ~ E n d P r o c でくり、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。P r o c の行の「m a i n」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「m a i n」または「メイン」を使います。

■ 簡単な S e l e c t 文を使う方法

さあ、これで A t l a s 9 8 プログラムの最小限の構造は分かりました。次は、簡単な S e l e c t 文の使い方です。

最初に述べたとおり、S e l e c t 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。

マップ設定(簡易) ホスト → Win データファイル変換

編集(E) ツール(T)

OK キャンセル

フロー

クリア(0) コメント(1) Atlas(2) Proc(3) EndProc(4) Loop(5) EndLoop(6) Select(7)

Case(8) ElseCase(9) EndSelect(A) Call(B) Return(C) Break(D) Quit(E)

Atlas(A) チェック(Q) 入力コード長φ: 256

No.	フロー	ホスト・アクション	データ形式	入力幅	出力幅	"~"	改行	コメント
1	Atlas 98							
2	main:Proc							
3	Select Int							
4	:		ゾーン・表示	u1				
5	Case 1							
6	:		ゾーン・表示	u1				
7	:		Ank	20				
8	Case 2							
9	:		ゾーン・表示	u1				
10	:		漢字	20				
11	Case 8							
12	:		ゾーン・表示	u1				
13	:		Ank・漢字	20				
14	Case 9							
15	:		ゾーン・表示	u1				
16	EndSelect							
17	EndProc							
18								
19								
20								
21								
22								

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ことです。そして、その次に書いた「ゾーン・表示 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、EndSelectま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・表示 u1 (= 1)	ゾーン・表示 u1 (1以外の値)
3	パック・表示 s5	なし
4	パック・表示 s5	なし
5	パック・表示 s5	なし
6	パック・表示 s5	なし
7	パック・表示 s5	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)。

何でも入れ子状に書ける様子がよく分かります。「ゾーン・表示 u1」をCase、ElseCaseのところに再指定しているのに注意してください。そうしないと、そのゾーン形式の項目がSelectに「食われて」しまうと考えてください。

■ 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、`Select ~ End Select` 構造が短くまとまっていたほうが、判りやすくなります。ボディーレコード用のマップを、外にくくり出してみましよう。そして、それを呼び出すようにします。マップのくくり出しには、`Proc ~ End Proc`を使います。手続きをこんな具合にもう1つ作るのです。

```
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。

マップ設定(簡易) ホスト → Win データファイル変換

編集(E) ツール(T)

フロー

Atlas(A) チェック(Q) 入力レコード長: 256

No.	フロー	機能	テンプレート形式	入力幅	出力幅	改行	コメント
1	Atlas 98						
2	main:Proc						
3	Select Int						
4	:		データ表示	10			
5	Case 2						
6	Call ボディー						
7	ElseCase						
8	Quit RecSkip, Con						
9	EndSelect						
10	EndProc						
11	ボディー:Proc						
12	:		Ank	10			
13	EndProc						
14							
15							
16							
17							
18							
19							
20							
21							
22							

■ 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、まずこんなふうを書けます。

マップ設定(簡易) ホスト → Win データファイル変換

編集(E) ツール(T)

フロー

クリア(O) コメント(1) Atlas(2) Proc(3) EndProc(4) Loop(5) EndLoop(6) Select(7)

Case(8) ElseCase(9) EndSelect(A) Call(B) Return(C) Break(D) Quit(E)

Atlas(A) チェック(Q) 入力レコード長: 256

No.	フロー	ホスト位置	データ形式	入力幅	出力幅	"~"	改行	コメント
1	Atlas 98							
2	main:Proc							
3	Select Int							
4	:		32-bit 表示	ul				
5	Case 2							
6	:		Ank	10				
7	Case 1,8,9							
8	Quit RecSkip,Con							
9	EndSelect							
10	EndProc							
11								
12								
13								
14								
15								
16								
17								
18								
19								
20								
21								
22								

Case 1,8,9は「1、8、または9なら」という意味です。その下にした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。

マップ設定(簡易) ホスト → Win データファイル変換

編集(E) ツール(T)

OK キャンセル

フロー

クリア(0) コメント(1) Atlas(2) Proc(3) EndProc(4) Loop(5) EndLoop(6) Select(7)

Case(8) ElseCase(9) EndSelect(A) Call(B) Return(C) Break(D) Quit(E)

Atlas(A) チェック(C) 入力レコード長: 256

No.	フロー	ホスト名	データ形式	入力幅	出力幅	"～"	改行	コメント
1	Atlas 98							
2	main:Proc							
3	Select Int							
4	:		ザン・表示	ul				
5	Case 2							
6	:		Ank	10				
7	ElseCase							
8	Quit RecSkip, Con							
9	EndSelect							
10	EndProc							
11								
12								
13								
14								
15								
16								
17								
18								
19								
20								
21								
22								

■ 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 囲	レコード区分
- 9 . 9 9 ~ - 0 . 0 1	マイナスレコード
0 . 0 0	ゼロレコード
0 . 0 1 ~ 9 . 9 9	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、～以上、～以下という指定をしやすくするMin ..と..Max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Min、Maxは整数のときにも使えます。

No.	フロー	ホストリジョン	データ形式	入力幅	出力幅	〃〃	改行	コメント
1	Atlas 98							
2	main:Proc							
3	Select Dec							
4	:		リレーン・表示	s1.2				
5	Case Min..-0.01							
6	:		リレーン・表示	u1.2				
7	Case 0.00							
8	:		リレーン・表示	s1.2				
9	Case 0.01..Max							
10	:		リレーン・表示	s1.2				
11	EndSelect							
12	EndProc							
13								
14								
15								
16								
17								
18								
19								
20								
21								
22								

「M i n . . - 0 . 0 1 」は「 - 0 . 0 1 以下なら」と読めばよいですし、「0 . 0 1 . . M a x 」は「0 . 0 1 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 S e l e c t 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。

' A 0 ' ~ ' A 9 '	タイプ A レコード
' B 0 ' ~ ' B 9 '	タイプ B レコード
' C 0 ' ~ ' Z 9 '	タイプ C Z レコード
' Z Z '	エンドレコード (途中に現れたらそこで中断)

これは、こんなふう書けます。

マップ設定(簡易) ホスト → Win データファイル変換

編集(E) ツール(T)

フロー

Atlas(A) チェック(C) 入力レコード長: 256

No.	フロー	ホスト	データ形式	入力幅	出力幅	"~"	改行	コメント
1	Atlas 98							
2	main:Proc							
3	Select Str							
4	:		Ank	2				
5	Case 'ZZ'							
6	Quit RecSkip,EOF							
7	Case 'A0'..'A9'							
8	:		Ank	2				
9	:		漢字	10				
10	Case 'B0'..'B9'							
11	:		Ank	2				
12	:		Ank・漢字	10				
13	Case 'C0'..'Z9'							
14	:		Ank	2				
15	EndSelect							
16	EndProc							
17								
18								
19								
20								
21								
22								

S e l e c t S t r (S e l e c t S t r i n g) と書くと、文字列による比較になります。' A 0 ' .. ' A 9 ' は「A 0 から A 9 までなら」という意味になります。

' Z Z ' の場合の「Q u i t R e c S k i p , E O F」とは、「Z Z 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。

■ 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。A t l a s 9 8 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
      :          B Y ~ S y s R e c N u m
```

と書きます。この S y s R e c N u m はシステム変数の1つで、F * T R A N + 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。

No.	フロー	ポジション	データ形式	入力幅	出力幅	"〜"	改行	コメント
1	Atlas 98							
2	main:Proc							
3	Select Int							
4	:		B Y	~SysRecNum				
5	Case 0							
6	:		Ank	10				
7	ElseCase							
8	:		漢字	20				
9	EndSelect							
10	EndProc							
11								
12								
13								
14								
15								
16								
17								
18								
19								
20								
21								
22								

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9999999ならそのレコードはそこま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、

```

X X : L o o p   6 4
      :
      :   X Xループの中身
      :
E n d L o o p

```

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、

```

S e l e c t   I n t
      :   P a c k D i s p   u 7
C a s e   9 9 9 9 9 9 9
      -- X Xから脱出する
      B r e a k   X X
E l s e C a s e
      :   P a c k D i s p   u 7
E n d S e l e c t

```

と書きます。まとめると、つぎのようになります。

マップ設定(簡易) ホスト → Win データファイル変換

編集(E) ツール(T)

OK キャンセル

フロー

クリア(0) コメント(1) Atlas(2) Proc(3) EndProc(4) Loop(5) EndLoop(6) Select(7)

Case(8) ElseCase(9) EndSelect(A) Call(B) Return(C) Break(D) Quit(E)

Atlas(A) チェック(C) 入力ワード長: 256

No.	フロー	ホスト名	データ形式	入力幅	出力幅	"~"	改行	コメント
1	Atlas 98							
2	main:Proc							
3	XX:Loop 64							
4	Select Int							
5	:		バック表示	u7				
6	Case 9999999							
7	Break XX,0							
8	ElseCase							
9	:		バック表示	u7				
10	EndSelect							
11	EndLoop							
12	EndProc							
13								

同じことを手続きとReturn文を使えば、次のようになります。

マップ設定(簡易) ホスト → Win データファイル変換

編集(E) ツール(T)

OK キャンセル

フロー

クリア(0) コメント(1) Atlas(2) Proc(3) EndProc(4) Loop(5) EndLoop(6) Select(7)

Case(8) ElseCase(9) EndSelect(A) Call(B) Return(C) Break(D) Quit(E)

Atlas(A) チェック(C) 入力ワード長: 256

No.	フロー	ホスト名	データ形式	入力幅	出力幅	"~"	改行	コメント
1	Atlas 98							
2	main:Proc							
3	Call XX							
4	EndProc							
5	XX:Proc							
6	Loop 64							
7	Select Int							
8	:		バック表示	u7				
9	Case 9999999							
10	Return 0							
11	ElseCase							
12	:		バック表示	u7				
13	EndSelect							
14	EndLoop							
15	EndProc							
16								

第4章

A t l a s 9 8の利用例

4.1 A t l a s 9 8 の利用例

■ 従来互換

A t l a s 9 8 モードで、A t l a s 文を使わない例を示します。

No.	フロー	ホッピング	データ形式	入力幅	出力幅	~	改行	コメント
1	--							
2	--Atlas98を使わないMap							
3	--							
4	:		Ank	10				
5	:		Ank	10				
6	:		漢字	32				
7	:		ゾーン表示	u5				
8	:		パック表示	s3.2				
9	:		パック表示	s5.2				
10								
11								
12								
13								
14								
15								
16								
17								
18								
19								
20								
21								
22								

< パラメータファイルの記述 >

```
--
-- A t l a s 9 8 を使わないMap
--
ank          10
ank          10
kanji        32
zonedisp     u5
packdisp     s3.2
packdisp     s5.2
```

■ A t l a s 9 8 化

A t l a s 9 8 を、最も単純に使った例です。

< パラメータファイルの記述 >

```
--
--  A t l a s 9 8 を、単純に使う
--
{ A t l a s   9 8 }

{ m a i n : P r o c }
    a n k           1 0
    a n k           1 0
    k a n j i       3 2
    z o n e d i s p u 5
    p a c k d i s p s 3 . 2
    p a c k d i s p s 5 . 2
{ E n d P r o c }
```

■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。

マップ設定(簡易) ホスト → Win データファイル変換

編集(E) ツール(T)

フロー

Atlas(0) コメント(1) Atlas(2) Proc(3) EndProc(4) Loop(5) EndLoop(6) Select(7)

Case(8) ElseCase(9) EndSelect(A) Call(B) Return(C) Break(D) Quit(E)

Atlas(A) チェック(Q) 入力レコード長: 256

No.	フロー	ポジション	データ形式	入力幅	出力幅	～	改行	コメント
1	--							
2	--単純ループ							
3	--							
4	Atlas 98							
5	main:Proc							
6	:		ゾーン表示	u1				
7	:		漢字	64				
8	Loop 25							
9	:		パック表示	s5.2				
10	EndLoop							
11	EndProc							
12								
13								
14								
15								
16								
17								
18								
19								
20								
21								
22								

<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。

マップ設定(簡易) ホスト → Win データファイル変換

編集(E) ツール(T)

OK キャンセル

フロー

クリア(0) コメント(1) Atlas(2) Proc(3) EndProc(4) Loop(5) EndLoop(6) Select(7)

Case(8) ElseCase(9) EndSelect(A) Call(B) Return(C) Break(D) Quit(E)

Atlas(A) チェック(C) 入力レコード長: 256

No.	フロー	ホスト・ジョブ	データ形式	入力幅	出力幅	"～"	改行	コメント
1	Atlas 98							
2	main:Proc							
3	:		Ank	22				
4	Loop 10							
5	:		Ank	1				
6	Loop 5							
7	:		パック・表示	s3.2				
8	:		パック・表示	u4.1				
9	EndLoop							
10	EndLoop							
11	:		ゾーン・表示	u5				
12	Loop 8							
13	:		Ank	1				
14	Loop 4							
15	:		パック・表示	s3.2				
16	:		パック・表示	u4.1				
17	EndLoop							
18	EndLoop							
19	EndProc							
20								
21								
22								

<パラメータファイルの記述>

```
{ A t l a s   9 8 }

{ m a i n : P r o c }
  a n k           2 2
  { L o o p   1 0 }
    a n k         1
    { L o o p   5 }
      p a c k d i s p   s 3 . 2
      p a c k d i s p   u 4 . 1
    { E n d L o o p }
  { E n d L o o p }
  z o n e d i s p   u 5
  { L o o p   8 }
    a n k         1
    { L o o p   4 }
      p a c k d i s p   s 3 . 2
      p a c k d i s p   u 4 . 1
    { E n d L o o p }
  { E n d L o o p }
{ E n d P r o c }
```

■マルチレイアウト変換を1つの手続きで設定する

ホストから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。

マップ設定(簡易) ホスト → Win データファイル変換

編集(E) ツール(T)

OK キャンセル

フロー

クリア(0) コメント(1) Atlas(2) Proc(3) EndProc(4) Loop(5) EndLoop(6) Select(7)

Case(8) ElseCase(9) EndSelect(A) Call(B) Return(C) Break(D) Quit(E)

Atlas(A) チェック(C) 入力レコード長: 256

No.	フロー	ホスト機能	データ形式	入力幅	出力幅	"~"	改行	コメント
1	Atlas 98							
2	main:Proc							
3	Select Int							
4	:		ワープ・表示	u1				
5	Case 1							
6	:		ワープ・表示	u1				ヘッダレコード用のMap
7	:		Ank	10				
8	Case 2							
9	:		ワープ・表示	u1				ボディーレコード用のMap
10	:		漢字	20				
11	Case 9							
12	:		ワープ・表示	u1				エンドレコード用のMap
13	:		パック・表示	u3.2				
14	ElseCase							
15	Quit RecSkip,Continue							
16	EndSelect							
17	EndProc							
18								
19								
20								
21								
22								

<パラメータファイルの記述>

```
{ A t l a s   9 8 }

{ m a i n : P r o c }
  { S e l e c t   I n t }
    z o n e d i s p    u 1
  { C a s e   1 }
    z o n e d i s p    u 1      --ヘッダレコード用のM a p 文
    a n k              1 0
  { C a s e   2 }
    z o n e d i s p    u 1      --ボディーレコード用のM a p 文
    k a n j i          2 0
  { C a s e   9 }
    z o n e d i s p    u 1      --エンドレコード用のM a p 文
    p a c k d i s p    s 3 . 2
  { E l s e C a s e }
    { Q u i t   R e c S k i p , C o n t i n u e }
  { E n d S e l e c t }
{ E n d P r o c }
```

■ 複数の手続きを活用する

ホストから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。

マップ設定(簡易) ホスト → Win データファイル変換

編集(E) ツール(T)

OK キャンセル

フロー

Call(0) コメント(1) Atlas(2) Proc(3) EndProc(4) Loop(5) EndLoop(6) Select(7)
Case(8) ElseCase(9) EndSelect(A) Call(B) Return(C) Break(D) Quit(E)

Atlas(A) チェック(C) 入力レコード長: 256

No.	フロー	オプション	データ形式	入力幅	出力幅	"~"	改行	コメント
1	Atlas 98							
2	main:Proc							
3	Select Int							
4	:		ワット・表示	u1				
5	Case 1							
6	Call ヘッダ							
7	Case 2							
8	Call ボディー							
9	Case 9							
10	Call エンド							
11	ElseCase							
12	Quit RecSkip,Continue							
13	EndSelect							
14	EndProc							
15	ヘッダ:Proc							
16	:		ワット・表示	u1				ヘッダレコード用のMap
17	:		Ank	10				
18	EndProc							
19	ボディー:Proc							
20	:		ワット・表示	u1				ボディーレコード用のMap
21	:		漢字	20				
22	EndProc							
23	エンド:Proc							
24	:		ワット・表示	u1				エンドレコード用のMap
25	:		ワット・表示	u8.2				
26	EndProc							
27								

<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}
  {EndSelect}
{EndProc}

{ヘッダ:Proc}
  zonedisp    u1      --ヘッダレコード用のMap文
  ank         10
{EndProc}

{ボディー:Proc}
  zonedisp    u1      --ボディーレコード用のMap文
  kanji       20
{EndProc}

{エンド:Proc}
  zonedisp    u1      --エンドレコード用のMap文
  packdisp   s3.2
{EndProc}

```

■ ボディーレコードのみ変換する

Winからホストへ、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。

マップ設定(簡易) Win → ホストデータファイル変換

編集(E) ツール(T)

OK キャンセル

フロー

クリア(0) コメント(1) Atlas(2) Proc(3) EndProc(4) Loop(5) EndLoop(6) Select(7)

Case(8) ElseCase(9) EndSelect(A) Call(B) Return(C) Break(D) Quit(E)

Atlas(A) チェック(C) 入力形式Φ: フォント

No.	フロー	オプション	データ形式	入力幅	出力幅	"~"	漢字イン/アウト	コメント
1	Atlas 98							
2	main:Proc							
3	Select Int							
4	:		表示・ツブシ		u1			
5	Case 2							
6	:		表示・ツブシ		u1			
7	:		Ank		10	あり		
8	:		漢字		20	あり		
9	Loop 31							
10	:		表示・ブロック		u4.1			
11	:		表示・ツブシ		s2			
12	EndLoop							
13	:		Ank		15			
14	:		Ank		15			
15	ElseCase							
16	Quit RecSkip,Continue							
17	EndSelect							
18	EndProc							
19								
20								
21								
22								

<パラメータファイルの記述>

```
{ A t l a s   9 8 }

{ m a i n : P r o c }
  { S e l e c t   I n t }
    d i s p z o n e      : u 1
  { C a s e   2 }
    d i s p z o n e      : u 1
    " a n k              : 1 0 "
    " k a n j i          : 2 0 "
  { L o o p   3 1 }
    d i s p p a c k      : u 4 . 1
    d i s p z o n e      : s 2
  { E n d L o o p }
    a n k                : 1 5
    a n k                : 1 5
  { E l s e C a s e }
    { Q u i t   R e c S k i p , C o n t i n u e }
  { E n d S e l e c t }
{ E n d P r o c }
```

■変換を中断する

Winからホストへ、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。

マップ設定(簡易) Win → ホスト データファイル変換

編集(E) ツール(T)

OK キャンセル

フロー

クリア(0) コメント(1) Atlas(2) Proc(3) EndProc(4) Loop(5) EndLoop(6) Select(7)

Case(8) ElseCase(9) EndSelect(A) Call(B) Return(C) Break(D) Quit(E)

Atlas(A) チェック(Q) 入力形式Φ: フォント

No.	フロー	ホスト・ジョブ	データ形式	入力幅	出力幅	"~"	漢字イン/アウト	コメント
1	Atlas 98							
2	main:Proc							
3	Select Str							
4	:		Ank	1				
5	Case 'B'							
6	Call ボディー							
7	Case 'Z'							
8	Quit RecSkip,EOF							
9	ElseCase							
10	Quit RecSkip,Continue							
11	EndSelect							
12	EndProc							
13	ボディー:Proc							
14	:		Ank	10				
15	:		漢字	20				
16	Loop 31							
17	:		表示・バック	5	u4.1			
18	:		表示・ツーン	2	s2			
19	EndLoop							
20	:		Ank	15				
21	:		Ank	15				
22	EndProc							
23								

<パラメータファイルの記述>

```
{Atlas 98}

{main:Proc}
  {Select Str}
    ank          1
  {Case 'B'}
    {Call ボディー}
  {Case 'Z'}
    {Quit RecSkip, EOF}
  {ElseCase}
    {Quit RecSkip, Continue}
  {EndSelect}
{EndProc}

{ボディー:Proc}
  ank          10
  kanji        20
  {Loop 31}
    disppack    5:u4.1
    dispzone    2:s2
  {EndLoop}
  ank          15
  ank          15
{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・表示変換をかけると 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 >

```
{ A t l a s   9 8 }

{ m a i n : P r o c }
  { S e l e c t   S t r }
    a n k                4
  { C a s e   S p a c e }
    a n k                4
  { E l s e C a s e }
    z o n e d i s p      u 4
  { E n d S e l e c t }
{ E n d P r o c }
```

■ 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。

マップ設定(簡易) ホスト → Win データファイル変換

編集(E) ツール(T)

フロー

Atlas(0) コメント(1) Atlas(2) Proc(3) EndProc(4) Loop(5) EndLoop(6) Select(7)

Case(8) ElseCase(9) EndSelect(A) Call(B) Return(C) Break(D) Quit(E)

Atlas(A) チェック(C) 入力レコード長: 256

No.	フロー	ポジション	データ形式	入力幅	出力幅	"~"	改行	コメント
1	Atlas 98							
2	main:Proc							
3	Loop 256							
4	Select Int							
5	:		パック表示	u3				
6	Case 999							
7	Quit RecSkip,EOF							
8	ElseCase							
9	:		パック表示	u3				
10	EndSelect							
11	EndLoop							
12	EndProc							
13								
14								
15								
16								
17								
18								
19								
20								
21								
22								

<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 }
    { ElseCase }
      packdisp u3
    { EndSelect }
  { EndLoop }
{ EndProc }
```

■ FF... (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。

No.	フロー	オプション	データ形式	入力幅	出力幅	〜	改行	コメント
1	Atlas 98							
2	main:Proc							
3	Select Str							
4	:		Ank	*				
5	Case HighValue							
6	Quit RecSkip,Continue							
7	EndSelect							
8	:		Ank	10				通常のMap
9	:		漢字	20				
10	EndProc							
11								
12								
13								
14								
15								
16								
17								
18								
19								
20								
21								
22								

< パラメータファイルの記述 >

```
{ Atlas 98 }
```

```
{ main:Proc }
```

```
{ Select Str }
```

```
ank *
```

```
{ Case HighValue }
```

```
{ Quit RecSkip,Continue }
```

```
{ EndSelect }
```

```
ank 10 --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。

No.	フロー	ポジション	データ形式	入力幅	出力幅	～	改行	コメント
1	Atlas 98							
2	main:Proc							
3	Select Int							
4	:		ゾーン表示	s10				
5	Case 0..Max							
6	:		ゾーン表示	s10				売上用Map
7	:		Ank	10				
8	Case Min..-1							
9	:		ゾーン表示	s10				返品用Map
10	:		漢字	20				
11	EndSelect							
12	EndProc							
13								
14								
15								
16								
17								
18								
19								
20								
21								
22								

<パラメータファイルの記述>

```
{ Atlas 98 }

{ main:Proc }
  { Select Int }
    zonedisp    s10
  { Case 0..Max }
    zonedisp    s10      --売上用Map
    ank         10
  { Case Min..-1 }
    zonedisp    s10      --返品用Map
    kanji       20
  { EndSelect }
{ EndProc }
```

■先頭レコードだけ別レイアウト

ホストから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。

マップ設定(簡易) ホスト → Win データファイル変換

編集(E) ツール(T)

フロー

クリア(0) コメント(1) Atlas(2) Proc(3) EndProc(4) Loop(5) EndLoop(6) Select(7)

Case(8) ElseCase(9) EndSelect(A) Call(B) Return(C) Break(D) Quit(E)

Atlas(A) チェック(C) 入力レコード長: 256

No.	フロー	ホスト関数	データ形式	入力幅	出力幅	～	改行	コメント
1	Atlas 98							
2	main:Proc							
3	Select Int							
4	:		BY	~SysRecNum				
5	Case 0							
6	:		Ank	10				ヘッダ用のMap文
7	ElseCase							
8	:		漢字	20				本来のMap文
9	EndSelect							
10	EndProc							
11								
12								
13								
14								
15								
16								
17								
18								
19								
20								
21								
22								

<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 }
{ EndProc }
```

■ 範囲指定して変換

ホスト→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。

マップ設定(簡易) ホスト → Win データファイル変換

編集(E) ツール(T)

フロー

Atlas(A) チェック(C) 入力レコード長: 256

No.	フロー	オプション	データ形式	入力幅	出力幅	"~"	改行	コメント
1	Atlas 98							
2	main:Proc							
3	Select Int							
4	:		BY	~SysRecNum				
5	Case 1000..1499							
6	:		Ank	10				通常のMap文
7	ElseCase							
8	Quit RecSkip,Continue							
9	EndSelect							
10	EndProc							
11								
12								
13								
14								
15								
16								
17								
18								
19								
20								
21								
22								

<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 --通常のMap文
```

```
{ElseCase}
```

```
{Quit RecSkip,Continue}
```

```
{EndSelect}
```

```
{EndProc}
```

■ 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。

マップ設定(簡易) ホスト → Win データファイル変換

編集(E) ツール(T)

OK キャンセル

フロー

Atlas(2) Proc(3) EndProc(4) Loop(5) EndLoop(6) Select(7)

Case(8) ElseCase(9) EndSelect(A) Call(B) Return(C) Break(D) Quit(E)

Atlas(A) チェック(C) 入力レコード長: 256

No.	フロー	ホスト	データ形式	入力幅	出力幅	変換	改行	コメント
1	Atlas 98							
2	main:Proc							
3	Select Int							
4	:		BY	~SysRecNum¥¥4				
5	Case 0							
6	:		Ank	10				0 番目のレコードのマップ
7	Case 1							
8	:		漢字	20				1 番目のレコードのマップ
9	Case 2							
10	:		ゾーン表示	u5				2 番目のレコードのマップ
11	Case 3							
12	:		パック表示	s3.2				3 番目のレコードのマップ
13	EndSelect							
14	EndProc							
15								
16								
17								
18								
19								
20								
21								
22								

< パラメータファイルの記述 >

```
{ Atlas 98 }
{ main:Proc }
  { Select Int }
    BY ~SysRecNum¥¥4
  { Case 0 }
    ank 10 -- 0 番目のレコードのマップ
  { Case 1 }
    kanji 20 -- 1 番目のレコードのマップ
  { Case 2 }
    zonedisp u5 -- 2 番目のレコードのマップ
  { Case 3 }
    packdisp s3.2 -- 3 番目のレコードのマップ
  { EndSelect }
{ EndProc }
```


■ 複合条件で抽出

先頭の項目が 'AA' で次々項目が 'BB' のレコードだけ抽出変換します。

マップ設定(簡易) ホスト → Win データファイル変換

編集(E) ツール(T)

OK キャンセル

フロー

クリア(0) コメント(1) Atlas(2) Proc(3) EndProc(4) Loop(5) EndLoop(6) Select(7)

Case(8) ElseCase(9) EndSelect(A) Call(B) Return(C) Break(D) Quit(E)

Atlas(A) チェック(C) 入力レコード長: 256

No.	フロー	ポジション	データ形式	入力幅	出力幅	"~"	改行	コメント
1	Atlas 98							
2	main:Proc							
3	Select Str							
4	:		Ank	2				
5	Case 'AA'							
6	:		Ank	2				
7	:		漢字	12				
8	Select Str							
9	:		Ank	2				
10	Case 'BB'							
11	Call 抽出							
12	ElseCase							
13	Quit RecSkip,Continue							
14	EndSelect							
15	ElseCase							
16	Quit RecSkip,Continue							
17	EndSelect							
18	EndProc							
19	抽出:Proc							
20	:		Ank	10				
21	:		漢字	20				
22	EndProc							
23								

<パラメータファイルの記述>

```
{Atlas 98}

{main:Proc}
  {Select Str}
    ank          2
  {Case 'AA'}
    ank          2
    kanji        12
    {Select Str}
      ank        2
    {Case 'BB'}
      {Call 抽出}
    {ElseCase}
      {Quit RecSkip,Continue}
    {EndSelect}
  {ElseCase}
    {Quit RecSkip,Continue}
  {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。

マップ設定(簡易) ホスト → Win データファイル変換

編集(E) ツール(T)

OK キャンセル

フロー

別ア(0) コメント(1) Atlas(2) Proc(3) EndProc(4) Loop(5) EndLoop(6) Select(7)
 Case(8) ElseCase(9) EndSelect(A) Call(B) Return(C) Break(D) Quit(E)

Atlas(A) チェック(C) 入力レコード長: 256

No.	フロー	オプション	データ形式	入力幅	出力幅	"~"	改行	コメント
1	Atlas 98							
2	main:Proc							
3	s1:Select Int							
4	:		ハック・表示	u3				
5	Case 10							
6	:		ハック・表示	u3				
7	Select Int							
8	:		ハック・表示	u5				
9	Case 1000							
10	:		ハック・表示	u5				
11	Break s1,0							
12	EndSelect							
13	EndSelect							
14	Select Int							
15	:		BY	~SysBreak				
16	Case 0							
17	Quit RecSkip,Continue							
18	EndSelect							
19	:		Ank	10				通常のMap
20	:		漢字	20				
21	EndProc							
22								

<パラメータファイルの記述>

```
{Atlas 98}

{main:Proc}
  {s1:Select Int}
    packdisp u3
  {Case 10}
    packdisp u3
    {Select Int}
      packdisp u5
    {Case 1000}
      packdisp u5
      {Break s1, 1}
    {EndSelect}
  {EndSelect}

  {Select Int}
    BY ~SysBreak
  {Case 0}
    {Quit RecSkip, Continue}
  {EndSelect}
  ank 10 --通常のMap
  kanji 20
{EndProc}
```


第5章

マップ解析

5.1 マップ解析の機能

F * T R A N + 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（S）→変換設定（S）を実行し、つぎのウィンドウを開きます。



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。

のチェックボックスONの状態でも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 (B) → ホスト → Win (G)
Amダンプバイナリモード (B) → Win → ホスト (P)

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 (T) → ホスト → Win (G)
Amダンプテキストモード (T) → Win → ホスト (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) ホスト → Win で " MultiRec.P " を読み込み、
Amダンプテキストモード (ホスト → Win) で出力する。

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

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。

Amトレースを採取するファイル変換を実行します。

マップ解析メニューのAmトレース (A) 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) ホスト→Winで " M u l t i R e c . P " を読み込み、
サンプルファイル " P l a n e t " を変換した後、Amトレースで出力する。

C:\¥FTRANP¥Planet → C:\¥FTRANP¥Planet.get データファイル変換 .

000000 0 {Atlas 98}

00000D 0 {main:Proc}
SysRecNum=0 SysReturn=0 SysBreak=0

00001A 1 {Select Str}
SysRecNum=0 SysReturn=0 SysBreak=0

000027 2 Ank 2
0 -
4F
01
1
..
..

00002E 2 {Case ' 2' } (不成立)
SysRecNum=0 SysReturn=0 SysBreak=0

.....

F * T R A N + V 3 . 0 操作説明書・マルチレコード編

2 0 0 0 年 1 0 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 〒108-8531 東京都港区芝浦 4 - 15 - 33 芝浦清水ビル
プロダクツ&サービス事業部 TEL 03 - 5445 - 2101
FAX 03 - 5445 - 2109

- ・ Windows、MS-DOS は米国 Microsoft Corporation 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。
- ・ 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。
- ・ 落丁、乱丁はお取り替えいたします。