

JAERI - M
82-097

可変配列サイズ・プログラムのための
プリプロセッサ・システム"EOS"

1982年8月

竹田 辰興・常松 俊秀・栗田 源一

JAERI-Mレポートは、日本原子力研究所が不定期に公刊している研究報告書です。
入手の問い合わせは、日本原子力研究所技術情報部情報資料課（〒319-11茨城県那珂郡東海村）あて、お申しこしてください。なお、このほかに財団法人原子力弘済会資料センター（〒319-11茨城県那珂郡東海村日本原子力研究所内）で複写による実費頒布をおこなっております。

JAERI-M reports are issued irregularly.

Inquiries about availability of the reports should be addressed to Information Section, Division of Technical Information, Japan Atomic Energy Research Institute, Tokai-mura, Naka-gun, Ibaraki-ken 319-11, Japan.

©Japan Atomic Energy Research Institute, 1982

編集兼発行 日本原子力研究所
印 刷 いばらき印刷(株)

可変配列サイズ・プログラムのための
プリプロセッサ・システム "EOS"

日本原子力研究所東海研究所核融合研究部

竹田辰興・常松俊秀・栗田源一

(1982 年 7 月 13 日受理)

TRITON 計画の一環として、配列サイズに変数および算術式を許したプログラム (EOS ソース・プログラム) をフォートラン・ソース・プログラムに変換するプリプロセッサ (EOS プリプロセッサ) が開発された。これは、フォートラン・ソース・プログラムを EOS ソース・プログラムに変換する逆 EOS プリプロセッサと合せて EOS プリプロセッサ・システムを構成する。EOS プリプロセッサを利用することによりプログラム内の配列サイズの変更が著しく容易になるので、このプリプロセッサ・システムは大規模計算機コードを開発し利用する上で極めて有効である。本報告書は、EOS プリプロセッサ・システムの使用手引書としても利用可能である。

JAERI-M 82-097

Preprocessor System "EOS" for a Variable-Array-Size Program

Tatsuoki TAKEDA, Toshihide TSUNEMATSU and Gen-ichi KURITA

Division of Thermonuclear Fusion Research

Tokai Research Establishment

JAERI

(Received July 13, 1982)

As a part of the project TRITON a preprocessor (EOS preprocessor) has been developed. This preprocessor converts an EOS source program in which variables and arithmetic expressions are usable to define sizes of arrays into a usual fortran source program in which only constants are usable for the purpose. The EOS preprocessor forms a preprocessor system with an inverse preprocessor (the inverse EOS preprocessor) which converts a fortran source program into an EOS source program. As the alteration of sizes of arrays becomes extremely easy by using the EOS preprocessor, this system is very useful in development and use of large-scale computer codes. This report can be used as a user's manual of the EOS preprocessor system.

Keywords: Preprocessor, EOS, TRITON, Numerical Code System,
Variable-Array-Size Program, COMMON block,
Fortran Program, Nuclear Fusion Code

目 次

1.	はじめに	1
2.	EOS 概要	3
3.	EOS 文法	6
3.1	基本事項	6
3.2	サイズ・ブロック	6
3.3	テキスト・ブロック	7
3.3.1	テキスト・ブロックの規約	7
3.3.2	サイズ式の数字列への変換	8
3.4	プログラム・ブロック	9
3.4.1	プログラム・ブロックの規約	9
3.4.2	テキストの挿入	10
3.4.3	変数名の変更	11
4.	EOS プリプロセッサ	12
4.1	ジョブ制御文の構成	12
4.2	実行時オプション・パラメータ	14
4.3	JCL プロシジャ	16
4.4	TSS コマンド	18
4.5	EOS プリプロセッサの使用例	22
5.	逆 EOS プリプロセッサ	27
5.1	逆 EOS プリプロセッサの機能	27
5.2	ジョブ制御文の構成	29
5.3	実行時オプション・パラメータ	31
5.4	JCL プロシジャ	33
5.5	逆 EOS プリプロセッサの使用例	33
6.	まとめ	38
謝 辞	38	
文 献	39	
付録 A	データ・セット一覧	40
付録 B	ユーティリティ一覧	42
付録 C	ロード・モジュールの作成	43
付録 D	EOS プリプロセッサのエラー・メッセージ	49
付録 E	逆 EOS プリプロセッサのエラー・メッセージ	58

Contents

1.	Introduction	1
2.	Outline of EOS preprocessor	3
3.	EOS syntax	6
3.1	Basic concepts	6
3.2	Size block	6
3.3	Text block	7
3.3.1	Rule on text blocks	7
3.3.2	Transformation of a size expression into a string of numerals.	8
3.4	Program block	9
3.4.1	Rule on program blocks	9
3.4.2	Insertion of a text	10
3.4.3	Alteration of names of variables	11
4.	EOS preprocessor	12
4.1	Job control program for the EOS preprocessor	12
4.2	Option parameters for the EOS preprocessor	14
4.3	JCL procedures	16
4.4	TSS commands	18
4.5	Example of use of the EOS preprocessor	22
5.	Inverse EOS preprocessor	27
5.1	Functions of the inverse EOS preprocessor	27
5.2	Job control program for the inverse EOS preprocessor	29
5.3	Option parameters for the inverse EOS preprocessor	31
5.4	JCL procedures	33
5.5	Example of use of the inverse EOS preprocessor	33
6.	Summary	38

Acknowledgments	38
References	39
Appendix A Data sets concerning the EOS preprocessor system	40
Appendix B JCL procedures for utility program EOSUTY	42
Appendix C Generation of load modules of the EOS and inverse EOS preprocessors	43
Appendix D Error messages of the EOS preprocessor	49
Appendix E Error messages of the inverse EOS preprocessor	58

1. はじめに

限られた計算機資源を有効に利用することは、大規模計算コードを開発し利用する際には、特に重要なことである。大規模計算を行う上でその有限性が大きな制限となるのは、CPU時間、主記憶容量である。プログラムの性質や計算センターの運用形態によっては、この他に、補助記憶容量、入出力チャンネルにおけるデータ転送量、経過時間等の有限性も利用者の要求を強く制限する計算機資源である。これらの資源を有効に利用する上で最も大切なことは、言うまでもなく、最適のアルゴリズムを採用しプログラム構成について十分配慮することである。これは、個々の問題に応じて解決すべきことである。これに対し、各プログラムを実行する際にプログラム内に用意した配列のサイズを必要最少限の大きさにすることは、一般的に役立つ最も基本的な省資源の方法である。しかし、通常のフォートラン・プログラムを使って、入力パラメータに応じた必要最少限の記憶容量で計算を実行するためには、配列サイズを毎回書き替える必要がある。これは、煩雑な作業であって、計算コードの開発・利用の能率を著しく低下させる。特に、共通領域(COMMON)中の配列サイズの変更は煩雑である。これを避けるためには配列のサイズを、数値としてではなく、変数または式として与えて計算に際して自動的に数値に変換することができればよい。最も理想的には、このような機能が計算の実行時に利用可能なことである。しかし、現在のフォートラン言語のもとでは、これは不可能である*。そこで、TRITON計画¹⁾の一環として、コンパイル前にプリプロセッサを通すことによってこのような機能を利用可能とするようなシステムEOSを開発した。

EOSで対象とするような可変配列サイズのプログラムをフォートラン言語を使って作成することに対する要求は古くからある。K. V. RobertsらによるOLYMPUSシステム^{2, 3)}は、核融合研究分野でしばしばあらわれる初期値問題解析用の計算コードに対し、プログラム構造を規定し各種のユーティリティ・プログラムを用意したコード開発・利用のためのシステムであるが、このシステムの標準的規約としてコモン・ブロックは各サブルーチン内に作らず、独立に用意するようになっている(これは、OLYMPUSが最初に開発されたCDC計算機において使いやすい機能が存在したことによる)。このため、コモン配列のサイズ変更は各サブルーチン毎に行わなくてよいのでコードの開発・利用にあたって大巾な能率向上をもたらしている。配列サイズのプリプロセス機能は、必ずしもOLYMPUSシステムの主要な機能ではないが、前述したようにコード開発・利用に際しこの機能はきわめて有効である。原研では、独立したプログラムとしてのコモン文処理プリプロセッサCMMNG⁴⁾が滝塚、岡本により作成されたことがある。次章以降の記述からもわかるようにEOSシステムはOLYMPUS、CMMNGの可変配列サイズ機能の多くを備えていると共に、新機能を加え、全体がなるべく一般的なシステムになるようにまとめあげたものである。特に、EOSでは、配列サイズとして算術式を書くことを認めているので、配列サイズを定めるための入力パラメータの数は極めて少なくすることができる。これによって、例えば、偏微分方程式を解くプログラムの差分メッシュの数を変更した時に必要な配列サ

* FORTRAN 77には一部この機能があるが数値の代入だけに限られる。

イズを毎回計算してからプログラムを実行しなければならないという煩雑さから解放される。また、通常のフォートラン・プログラムを EOS 形式のプログラムに変更するための逆 EOS プリプロセッサも用意してあるので、EOS を意識せずに開発したフォートラン・プログラムを EOS 形式のプログラムに変更することも比較的容易である。

次章に、EOS システムの概要を記す。EOS 形式のプログラムを EOS 基本機能だけを使って作成するためにはこの概要の理解だけでほとんど十分である。しかし、高度の機能を利用し、また、制限事項等について詳細を理解するため EOS の文法を第 3 章に記す。EOS プリプロセッサおよび逆 EOS プリプロセッサの使用方法については、それぞれ、第 4 章、第 5 章に述べる。EOS システムの今後の課題等については第 6 章にまとめる。

2. E O S 概 要

最初に、第 2.1 図に示すようなフォートラン・プログラムを考える。このプログラムは、1つのメイン・プログラムと2つのサブルーチンから構成されており、2つのコモン領域 COMA および COMB が使われている。コモン領域には配列 V1, V2, V3 がとられており、必要な配列のサイズは、パラメータ N を使って、次のようにあらわされるものである。即ち

$$V1(i), \quad i = 1 \sim N,$$

$$V2(i), \quad i = 1 \sim 2N,$$

$$V3(i, j), \quad i = 1 \sim N^2, \quad j = 1 \sim N^2 + 2N.$$

このフォートラン・プログラムでは、 $N = 10$ としてある。このプログラムを必要最少限の記憶容量で動作させるためには、入力データに応じて上記配列のサイズを書き替える必要が生じてくる。このプログラムの場合、6個所のコモン文を書き替えねばならない。サブルーチンの数とコモン文の数が増大すると、このような作業は利用者にとってきわめて大きな負担となるばかりか、書き間違いによる能率低下が生ずる可能性がますます増大する。

そこで、コモン文をプログラム本体から引き抜き、一個所に集めておき、プリプロセッサを用いることによりコンパイル実行前にこれらコモン文を必要個所に挿入する事が考えられる。この場合、第 2.1 図のプログラムは第 2.2 図のようになる。挿入するコモン文は *COMDECK 文により、また挿入されるプログラム単位は *DECK 文により区別される。*COMDECK に続いて書かれたフォートラン文は、プログラム本体で *CALL 文の個所に挿入される。このプログラムの場合、6つの変更個所が2つのコモン文書き替えだけでよいことになる。

しかし、ラベル・コモンの数が増加し、サイズの異なる配列が増加してくると、この方法もまだ十分能率的とは言えない。特に、科学技術計算においては、元来比較的少数の基本パラメータのみで種々の配列サイズがあらわせることを考えるならば、全てのコモン配列のサイズを書き替えるのは無駄である。そこで、*COMDECK 文に続くコモン文中の配列サイズは、定数でなく変数や算術式として、この変数や算術式の値を別の個所で設定することが考えられる。このようにして得られたプログラムが、第 2.3 図に示すものである。ここで示した場合においては、全ての配列サイズの値が1個のパラメータ N によってあらわされるので書き替えは1個所のみでよいことになる。

第 2.1 図に示すフォートラン・ソース・プログラムに対し、第 2.2 図、第 2.3 図に示すものを EOS ソース・プログラムと称する。EOS ソース・プログラムは、EOS プリプロセッサを通すことによりフォートラン・ソース・プログラムに変換される。また、逆に、通常のフォートラン・ソース・プログラムは、逆 EOS プリプロセッサを通すことにより第 2.2 図の形の EOS ソース・プログラムに変換される。この形の EOS ソース・プログラムから第 2.3 図の形の EOS ソース・プログラムを得るためには若干の手作業を必要とする。これは仕方の無いことであろう。コモン変数は全てのサブルーチンにおいて、同一の名称をもつことが望ましいが、同一変数に対して異った名称を使うことも可能である。これらについて、次章以下に詳しく記述される。

```

C***MAIN PROGRAM***
COMMON/COMA/V1(10),V2(20)
COMMON/COMB/V3(100,120)
.
.
END

SUBROUTINE SUBA
COMMON/COMA/V1(10),V2(20)
COMMON/COMB/V3(100,120)
.
.
END

SUBROUTINE SUBB(X,Y)
COMMON/COMA/V1(10),V2(20)
COMMON/COMB/V3(100,120)
.
.
END
    
```

(プログラム・ブロック)

第 2.1 図 フォートラン・ソース・プログラムの構成

```

*COMDECK COMA
COMMON/COMA/V1(10),V2(20)
*COMDECK COMB
COMMON/COMB/V3(100,120)
    
```

(テキスト・ブロック)

```

*DECK MAIN
C***MAIN PROGRAM***
*CALL COMA
*CALL COMB
.
.
END
*DECK SUBA
SUBROUTINE SUBA
*CALL COMA
*CALL COMB
.
.
END
*DECK SUBB
SUBROUTINE SUBB(X,Y)
*CALL COMA
*CALL COMB
.
.
END
    
```

(プログラム・ブロック)

第 2.2 図 EOS ソース・プログラム
(サイズ・ブロック無し)の構成

<pre>N =10 N1=N N2=2*N N3=N*N</pre>	<p>(サイズ・ブロック)</p>
<pre>*COMDECK COMA COMMON/COMA/V1(N1),V2(N2) *COMDECK COMB COMMON/COMB/V3(N3,N2+N3)</pre>	<p>(テキスト・ブロック)</p>
<pre>*DECK MAIN C***MAIN PROGRAM*** *CALL COMA *CALL COMB . . END *DECK SUBA SUBROUTINE SUBA *CALL COMA *CALL COMB . . END *DECK SUBB SUBROUTINE SUBB(X,Y) *CALL COMA *CALL COMB . . END</pre>	<p>(プログラム・ブロック)</p>

第2.3図 EOS ソース・プログラム
(サイズ・ブロック付)の構成

3. E O S 文 法

3.1 基本事項

(1) 用語の定義

- * EOS ソース・プログラム (EOS source program) : EOS プリプロセッサの処理の対象となるソース・プログラム。また、逆 EOS プリプロセッサの出力ソース・プログラム。
EOS ソース・プログラムは、サイズ・ブロック、テキスト・ブロック、プログラム・ブロックの3種のブロック(任意の個数)から構成される。
- * サイズ・ブロック (Size block) : テキスト・ブロックで使われるサイズ変数を定義する。
- * テキスト・ブロック (Text block) : プログラム・ブロックに挿入される一群のテキストを定義する。コモン文、ディメンション文を対象とするのが EOS 本来の目的であるが、他の文でもよい。テキスト・ブロックに書かれた配列名のサイズの部分にサイズ変数およびサイズ変数を含んだサイズ式を書くことができる。
- * プログラム・ブロック (Program block) : テキスト挿入文 (*CALL 文、または、INCLUDE 文)を含むフォートラン形式のプログラム部分。
- * サイズ変数 (Size variable) : 配列のサイズを定めるパラメータ。正整数。
- * サイズ式 (Size expression) : 配列のサイズを定める式。サイズ変数と定数の四則演算式。
- * EOS プリプロセッサ (EOS preprocessor) : EOS ソース・プログラムをフォートラン・プログラムに変換するプリプロセッサ。
- * 逆 EOS プリプロセッサ (Inverse EOS preprocessor) : フォートラン・プログラムを EOS ソース・プログラムに変換するプリプロセッサ。
- * テキスト名 (Name of text) : 1つのテキスト挿入文に対応するテキストの単位につけられた名称。

(2) 各ブロックの順序

EOS ソース・プログラムにおいて各ブロックは、次のような順序で書かれなければならない。

- ① サイズ・ブロックは、必ずプログラムの先頭になければならない。但し、必要がなければ省略してもよい。
- ② テキスト・ブロックとプログラム・ブロックは任意の順序で書くことができる。但し、プログラム・ブロックの挿入命令で引用されるテキスト・ブロックは、引用前に定義されていなければならない。

3.2 サイズ・ブロック

サイズ・ブロックは、サイズ変数の定義式とフォートラン形式のコメント行とから構成される。

・例：

```
.....*.....*.....*...
C === BLOCK1 SIZE ===
N=5,M=100*N
C === BLOCK2 SIZE ===
L=20,LL=N+1
```

[規約]

- 1) サイズ変数名は、英字ではじまる6文字以内の英数字で全て整数型として扱われる。
- 2) サイズ変数の定義式には、「値による定義」と「式による定義」がある。「値による定義」では正整数値のみが許される。「式による定義」では右辺にサイズ変数を含む算術式が許される。
- 3) 「式による定義」の右辺は、「値による定義」が全て登録された後に評価される。
- 4) 値による定義を2回以上行くと最後の値が有効となる。
- 5) 使用フィールドは、1～71カラムで第72カラムが継続印として使われる。継続は19行まで許される。
- 6) 第1カラムの「C」は無条件にコメント行として扱われるので、サイズ変数名の付け方に注意すること。
- 7) 後に述べるように、サイズ変数はEOSプリプロセッサに対する入力データによって変更可能である。この場合サイズ式の値はこの入力データの値に基づいて評価される。

3.3 テキスト・ブロック

3.3.1 テキスト・ブロックの規約

テキスト・ブロックは、次に示すような構成をとる。

<第1行>

```
*COMDECK   テキスト名
```

<第2行以降>

テキスト名によって引用される文の集りで、次のものを含む

- ・任意のフォートラン文(含コメント)
- ・サイズ式を含むE宣言文(規約1)
- ・¥付サイズ式(規約5)を含むE宣言文以外のフォートラン文

・例：

```
.....*.....*.....*...
*COMDECK TEXT1
COMMON /A/ X(L+M,N)
INTEGER   Y(L**2)
KK= ¥(L+M+2000)
```

[規約]

- 1) 次の8つの宣言文を総称して「E宣言文」と呼ぶ。

INTEGER, REAL, LOGICAL, COMPLEX, DOUBLE PRECISION,
COMMON, DIMENSION, EQUIVALENCE

E 宣言文が通常のフォートラン宣言文と異なるのは E 宣言文においては配列サイズを変数または算術式で与え得ることである。

- 2) *COMDECK 文は第 1 カラムから左づめで書かれる。
- 3) テキスト名は英字で始まる 6 字以内の英数字か、その後「#」と数字列を付けたものである。

・例： *COMDECK TEXT 1 # 3

このテキスト名は、プログラム・ブロックの *CALL 文で参照される。

- 4) サイズ式は、サイズ変数を含むフォートランの算術式で、E 宣言文の配列宣言子の「(」と「)」の間に書くことができる。

・例： INTEGER A(L+M, N)

- 5) ¥付サイズ式は、「¥」の後の「(」と「)」ではさまれた部分に書くことのできるサイズ式である。¥付サイズ式は、テキスト・ブロック内の E 宣言文以外の文に書けるか、1～6 カラムには書けない。¥付サイズ式が単項式でかつ文の末尾に書かれる時に限ってカッコを省くことができる。

・例：*.....*.....*....

```

N= ¥(N**2)
DATA M/ ¥(M)/
K= ¥ K

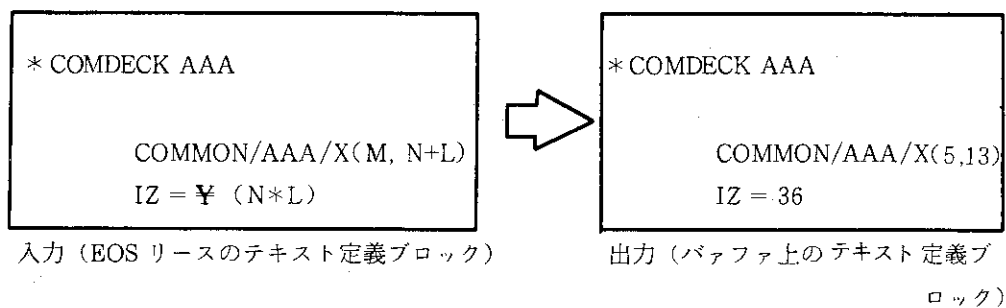
```

- 6) 継続行の規約や途中の空白の無視はフォートランと同様である。

3.3.2 サイズ式の数字列への変換

テキスト定義文で定義されているテキスト本文に対する編集機能である。テキスト本文に含まれる E 宣言文は配列名の後の宣言子添字の中の「サイズ式」を含むことができる。また、それ以外の文は ¥付サイズ式を含むことができる。

EOS プリプロセッサは、サイズ式を読み込むと、サイズ変数テーブルを参照しながら、式の値の評価を行い、式の値を表わす数字列にサイズ式を置き換える。ここで、サイズ変数テーブルとは、サイズ式中に現われる変数(サイズ変数)とその値の参照テーブルで、EOS ソース・プログラムの先頭の部分で入力されるサイズ変数データから作られる。具体的なサイズ変数の変換の例は第 3.1 図に示す。



サイズ変数テーブルは下に示すようになっている。

サイズ変数名	値
L	4
M	5
N	9

第 3.1 図 サイズ変数・サイズ式の変換例

3.4 プログラム・ブロック

3.4.1 プログラム・ブロックの規約

プログラム・ブロックは、次に示すような構成をとる。

<第 1 行>

*DECK デッキ名

<第 2 行以降>

通常のフォートラン文またはテキスト挿入文

・例：

```

.....*.....*.....*...
*DECK      SUB1
           SUBROUTINE SUB1
           INTEGER A
*CALL TEXT1
           A=1
           RETURN
           END
    
```

[規約]

- 1) 「*DECK」は第 1 カラムから左づめで書く。
- 2) デッキ名は英字で始まる 6 字以内の英数字で、印刷制御に用いられる。
- 3) 「*CALL」は第 1 カラムから左づめで書く。*CALL 文の位置に引用されたテキストが展開される。

4) 「*CALL」のかわりに「INCLUDE」を用いることができる。「INCLUDE」は第7カラムから左づめで書く。下の例に示す2つの文は同じ内容をもつ。

・例：
*.....*.....*....
 *CALL TEXT1①
 INCLUDE TEXT1②

5) テキスト・ブロックのE宣言文の変数名を変更したい時には、*CALL文のテキスト名に続けて、変数名変更リストを書けばよい。即ち、*CALL テキスト名/旧変数名/新変数名/[, ...]ここで、「/」は、英数字および「,」以外の記号ならばよい。リストは「,」で区切って30組まで書ける。また、継続行も許される。

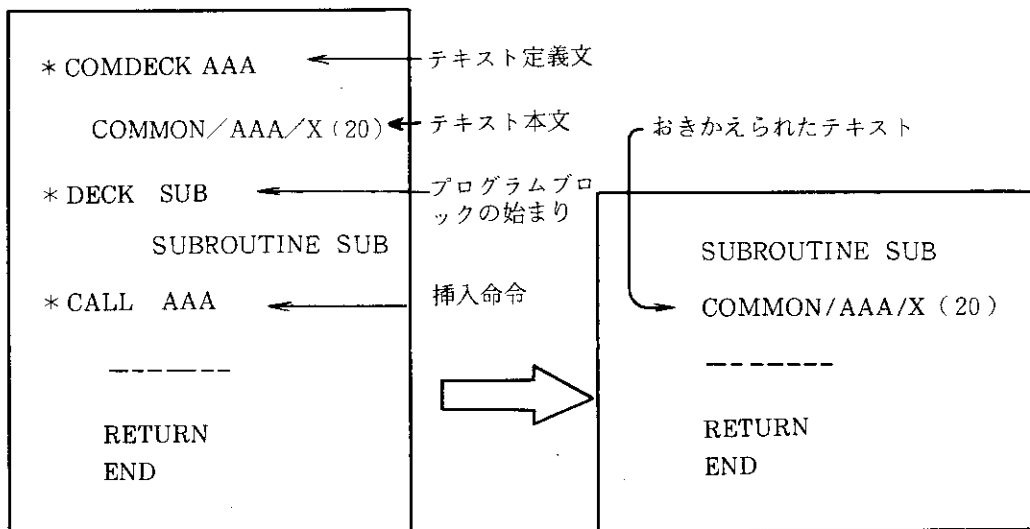
6) 継続行の規約や途中の空白の無視はフォートランと同様である。

7) EOS ソース・プログラムでは、通常のフォートランのコメント行以外の所にもコメントを書くことができる。これを行コメントと呼ぶ。行コメントは、標準的には「;」を区切り記号としてフォートラン文の後に続けて書く。区切り記号は変更可能である。EOS プリプロセッサは、処理の過程で「;」以降の文字列を消去する。行コメントの例を示す。

・例： A = 12.9 ; KG/M**3

3.4.2 テキストの挿入

テキストの挿入は、EOS プリプロセッサの主機能である。EOS ソース・プログラム中のプログラム・ブロックはテキスト名を伴ったテキスト挿入文を含んでいる。EOS プリプロセッサは、テキスト挿入文を読み込むとテキスト名を認識して、対応するテキストをテキスト挿入文と置き換える。具体的なテキスト挿入の例を第3.2図に示す。



入力 (EOS ソース)

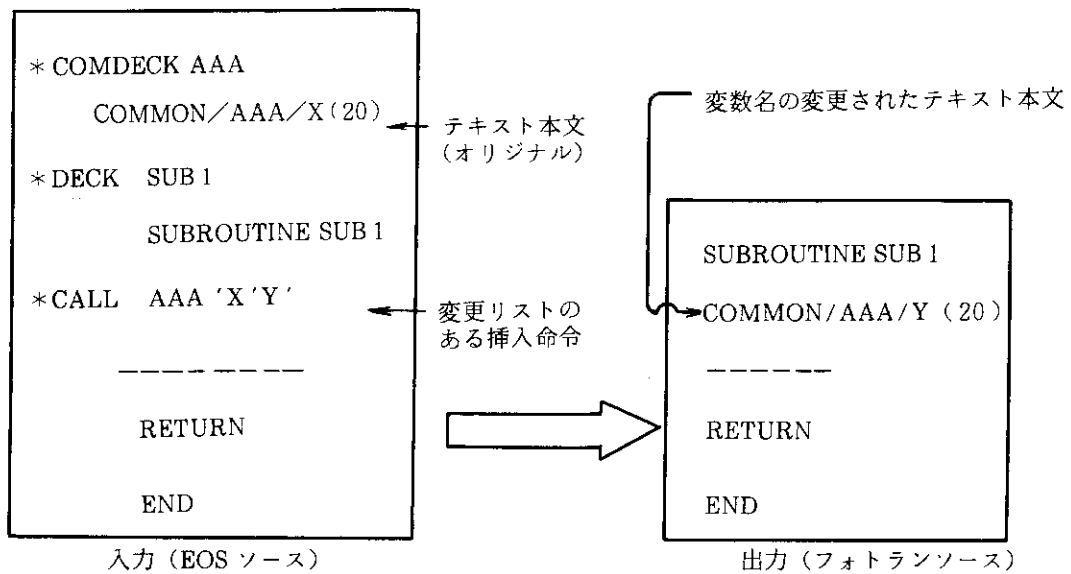
出力 (フォートラン・ソース)

第3.2 テキスト挿入の例

3.4.3 変数名の変更

テキスト文をフォートラン・ソース・プログラム上に展開する時に、テキスト本文を再度変更することができる。但し、変更できるのは、テキスト本文中の E 宣言文の中にある変数名、配列名に限られる。

既に 3.4.1 の規約 5 で述べたように、EOS ソース・プログラム中のテキスト挿入文は、テキスト名の他に変数名変更リストを含むことができる。EOS プリプロセッサは、変数名変更リストを認識すると、テキスト本文中の E 宣言文の中の変数名の中から変数名変更リストの変数に対応するものを探し出して新しい変数名に置き換えてフォートラン・ソース・プログラム上に展開する。変数名変更の具体例を第 3.3 図に示す。



第 3.3 図 変数名変更の件

4. EOSプリプロセッサ

4.1 ジョブ制御文の構成

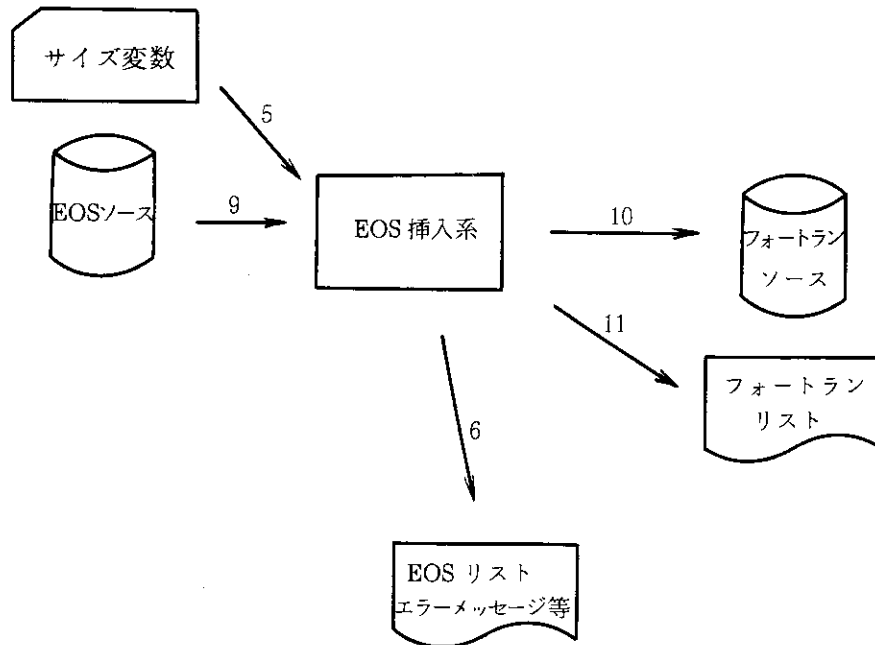
EOS プリプロセッサを起動する際のジョブ制御文の例を第 4.1 図に示す。この例は、EOS ソース・プログラム "EOS" をプリプロセスしてフォートラン・ソース・プログラム "FORT" を作成するための JCL の例である。なお、実際の利用にあたっては、4.3, 4.4 節の JCL プロシジャ、TSS コマンドを使うのが便利である。第 4.1 図において、第 1 行目の「PARM =」に示したリストは、実行時オプション・パラメータ・リストで詳細は 4.2 に述べる。「E1」はロード・モジュール・データ・セット J2636, EOS 200, LOAD の中の EOS プリプロセッサを表すメンバー名である。EOS プリプロセッサを起動するに際し必要な入出力機番は、5, 6, 9, 10, 11 の 5 種であるが、それぞれの役割については、第 4.1 表にまとめてある。また、各入出力機器の関係図は、第 4.2 図に示した。

第 4.1 表 プリプロセスに際しての各入出力機番の役割

入出力機番	役 割
5.	サイズ変数の入力機番 (1 RECORD = 80 BYTE) EOSソースのサイズブロックと同一の文法で書かれたカード 入力が行える。第 1 カラムから第 71 カラムまで使用出来る。 第 72 カラムは継続カラムである。 例) <div style="border: 1px solid black; padding: 5px; margin: 10px auto; width: fit-content;"> $M = N * 2, L = N + 1$ $K = 100$ </div>
6.	EOS = 挿入系の処理情報が出力される。(SYSOUT データセット) 出力される情報には次の様なものがある。 <ol style="list-style-type: none"> 1. 頁のヘッダー 2. EOS ソースリスト 3. EOS の出すエラーメッセージ 4. ダンプテーブル類 5. 処理単位毎のエラーカウント 6. 全体の会計情報
9.	EOS ソースの入力 (1 RECORD = 80 BYTE) 順編成ファイルでなくてはならない。
10.	フォートラン・ソースの出力 (1 RECORD = 80 BYTE) 順編成ファイルでなくてはならない。
11.	フォートラン・ソースのプリント出力 SYSOUT データセットである。

```
// EXEC PGM=E1,PARM='SLIST=1,LINECM=0,SDUMP=1'
//STEPLIB DD DSN=J2636.EOS200.LOAD,DISP=SHR
//SYSPRINT DD SYSOUT=*
//FT05F001 DD *
M=5,L=70
/*
//FT06F001 DD SYSOUT=*
//FT09F001 DD DSN=EOS,DISP=OLD
//FT10F001 DD DSN=FORT,DISP=(NEW,CATLG,DELETE),
              DCB=(RECFM=FB,LRECL=80,BLKSIZE=1200,DS=PS),
              UNIT=USER,SPACE=(TRK,(30,10))
//FT11F001 DD DUMMY
```

第 4.1 図 EOS プリプロセッサ JCL の例



第 4.2 図 EOS プリプロセッサに関する入力機器の関係図

4.2 実行時オプション・パラメータ

EOS プリプロセッサの利用に際しては、実行時オプション・パラメータとして第4.2表に示すものを指定することができる。

第4.2表 EOS プリプロセッサの実行時オプション・パラメータ

No	パラメータ	標準	意味
1	PREC	'*'	EOS コマンド記号
2	SEPRAT	';'	行コメント用セパレーター
3	SLIST	0	入力ソースリストのプリント (SLIST>0の時出力)
4	SAVE	1	出力フォートランファイルの書込み (SAVE>1の時書込む)
5	FPR 1	0	出力フォートランプリント出力 (FPR 1>0の時出力)
6	LINECM	1	行コメントの使用/不使用 (LINECM>0で使用)
7	SDUMP	0	サイズシンボルテーブルダンプ (SDUMP>0で出力)
8	TDUMP	0	テキストテーブルダンプ (TDUMP>0で出力)
9	RDUMP	0	変数名変更テーブルダンプ (RDUMP>0で出力)

(1) オプション・パラメータの詳細

- ① PREC: EOS ソース・プログラム中の EOS プリプロセッサに対するコマンドを、他の文から識別する為に EOS コマンドは第1カラムから左づめに書かれ、かつその先頭は「PREC」に与えられた記号でなくてはならない。標準値は「*」。
例) *COMDECK *DECK *CALL
- ② SEPRAT: EOS ソースは各行毎に行末の空白にコメントを書くことができる。コメントと文の本体の間の区切り番号が SEPRAT である。行コメントを使用する場合、SEPRAT からは無条件にコメントとみなされるので注意すること。標準値は「;」。
例) X = 12.0 ; CM**3
- ③ SLIST: EOS プリプロセッサは、入力 EOS ソースイメージを「機番6」に出力する。SLIST=0 ならば出力は抑制される。但しエラーメッセージは必ず出力される。標準値は0 (出力なし)。
- ④ SAVE: EOS 挿入系の生成するフォートラン・ソースを「機番10」に出力するかどうかのスイッチ。EOS ソース・リストのみが必要な時等に SAVE = 0 とするとよい。標準値は1 (出力あり)。

- ⑤ FPR1: EOS 挿入系の生成する フォートラン・ソースのプリントイメージの出力スイッチ。標準値は0 (出力なし)。
FPR1 = 1 とすると出力される。
- ⑥ LINECM: EOS ソース中で行コメントを使用するかどうかのスイッチ。LINECM=1 とすると、各「行」の中の「SEPRAT」で指定された区切番号以降の文字は空白と見なされる。但し、ソースイメージ出力の際は保存されている。標準値は1 (使用する)。
- ⑦ SDUMP: EOS ソース中のサイズブロックの解析の結果得られた、各サイズ変数の名前と値の対照表をダンプ出力する。標準値は0 (ダンプしない)。
- ⑧ TDUMP: EOS ソース中のテキストブロックを編集し、挿入用テキストとした時のテキストイメージをダンプ出力する。各 COMDECK 毎に出力される。標準値は0 (ダンプしない)。
- ⑨ RDUMP: EOS ソース中の挿入命令の後の変数名変更リストの解析結果として得られる新、旧変数名の対照表を出力する。標準値は0 (ダンプしない)。

(2) オプション・パラメータの入力方法

EOS プリプロセッサに対するオプション・パラメータの入力はプログラム実行時パラメータの文字列としてプログラムに渡される。

・例:*.....*.....*
 バッチ処理 // EXEC PGM=E1,PARM='SLIST=1'
 TSS 処理 CALL EOS200.LOAD(E1) 'SLIST=1'

この際、一般に、複数個のパラメータを指定できる。即ち、入力文字列は次のようになる。

'パラメータ名=値 [, パラメータ名=値……]'

また、入力文字列が引用符を必要とする時は、引用符を2回続ければよい。即ち、

・例:*.....*.....*.....*..
 // EXEC PGM=E1,PARM='PREC='%'',SLIST=1'

次節で示す JCL マクロ・プロシジャでは、SDUMP, TDUMP, RDUMP を除く 6 個のパラメータが与えられるようになっている。

4.3 JCLプロシジャ

EOS プリプロセッサを起動するために、プロシジャ EOSFOR が用意されている。これは、PS 形式の EOS ソース・プログラムを変換して、PS 形式のフォートラン・プログラムを作成するためのものである。ソース・プログラム・ファイルとして PS, PO, GEM 形式のファイルについてはいずれも利用可能なように変換プログラム (J2636, EOSUTY, PREP, LOAD) と JCL マクロプロシジャ GEMPO, POGEM, POPS, PSFPO, PSEPO を用意してある。これらについては、付録 B を参照のこと。第 4.3 表にプロシジャ EOSFOR のパラメータ・リストを示す。また、第 4.3 図には EOSFOR のプロシジャ・リストを示す。なお、EOS 固有のストリーム・プロシジャは、ディスク・ファイル J2636, EOS 200, CNTL にプロシジャ名をメンバ名として格納されている。

第 4.3 表 プロシジャ EOSFOR パラメータ・リスト

パラメータ	意 味	デフォルト値
DSN	入力データ・セット名 (EOS, PS)	
ODSN	出力データ・セット名 (フォートラン, PS)	
UNIT	出力データ・セット・ユニット指定	TSSWK
SYSOUT	SYSOUT パラメータ	' * '
OLD	出力データ・セット・DISP パラメータ	NEW
CATLG	出力データ・セット・DISP パラメータ	CATLG
SPC	出力データ・セット・スペース指定	' 30, 10 '
PREC	EOS コマンド記号	*****
SLIST	入力ソース・リスト・プリント・スイッチ	1 (プリント)
SAVE	出力ソース・ファイル書込みスイッチ	1 (書込み)
FPR 1	出力ソース・ファイル・プリント・スイッチ	1 (プリント)
LINECM	行コメント使用スイッチ	1 (使用)

```

00000010
00000020
00000030
00000040
00000050
00000060
00000070
00000080
00000090
00000100
00000110
00000120
00000130
00000140
00000150
00000160
00000170
00000180
00000190
00000200
00000210
00000220
00000230
00000240
00000250

*****
/*      EOSFOR      :  EOS(PS) TO FORTRAN(PS)
*****
/*      :  DSN      :  EOS SOURCE (PS)      (      )
/*      :  ODSN     :  FORT SOURCE(PS)     (  NULLFILE )
/*      :  UNIT     :  UNIT                  (  TSSWK   )
/*      :  SYSOUT   :  SYSOUT CLASS         (  *       )
/*      :  OLD      :  DISP PARAMETER      (  NEW     )
/*      :  CATLG    :  DISP PARAMETER      (  CATLG   )
/*      :  SPC      :  SPACE FOR PS        (  '30,10' )
*****
/*      :  DSN      :  ODSN=NULLFILE,UNIT=TSSWK,SYSDOUT='*',OLD=NEW,
/*      :  CATLG=CATLG,SPC='30,10',
/*      :  PREC='*',*','*',SLIST=1,SAVE=1,FPR1=1,LINECM=1
/*      :  EXEC PGM=E1,
/*      :  PARM='PREC=&PREC,SLIST=&SLIST,SAVE=&SAVE,FPR1=&FPR1,LINECM=&LINECM'
/*      :  STEPLIB DD DSN=J2636.EOS200.LOAD,DISP=SHR
/*      :  FT05F001 DD DDNAME=SYSIN
/*      :  FT06F001 DD SYSOUT=&SYSOUT
/*      :  FT09F001 DD DSN=&ODSN,DISP=OLD
/*      :  FT10F001 DD DSN=&ODSN,DISP=(&OLD,&CATLG,DELETE),
/*      :  UNIT=&UNIT,SPACE=(TRK,(&SPC)),
/*      :  DCB=(RECFM=FB,LRECL=80,BLKSIZE=3200,DSORG=PS)
/*      :  FT11F001 DD SYSOUT=&SYSOUT,DCB=(BLKSIZE=13700,LRECL=137,RECFM=FBA)
*****

```

第4.3図 EOS プリプロセッサ利用のためのプロシジャ・リスト (EOSFOR)

4.4 TSSコマンド

EOS プリプロセッサを TSS で使用するために、コマンド EFL が用意されている。EFL は GEM 形式の EOS ソース・プログラムをモジュール毎にコンパイル・リンクを行うためのコマンドである。EOS ソース・プログラム中では、サイズ・ブロック、テキスト・ブロックは、「#」ではじまるモジュール名で格納されていることが必要であり、また、プログラムを格納している各モジュールはすべて先頭に「*DECK」文が置かれていなければならない。なお、EFL 実行前に 1 日 1 回だけ FILE 2 を実行しておくことが必要である。EFL および FILE 2 は、J 2636、EOS 200、CLIST のメンバである。

コマンド EFL の使い方は次のとおりである。

→ EFL

▶ GROUP @@COM WAS DEFINED ▶

▶ ▶

▶ GEM3 ×××× START. WHEN UPDATE END, ENTER 'END' ▶

→ U AA

修正作業

①

→ EOM S

→ END

▶ \$\$ MODULE NAME OR GROUP NAME ==>モジュール名を入力 ②③

▶ <<< EOS START >>>▶

▶ <<< EOS END >>>▶

▶ \$\$ IF COMPILE, ENTER 'C' ==>'C' を入力 ④

▶ \$\$ COMPILER START ▶

▶ \$\$ COMPILE O.K. ▶

▶ \$\$ IF LINK, ENTER 'L' ==>'L' を入力 ④

▶ \$\$ LINKAGE END. NOW ××××. LOAD UPDATED ▶

おわり

注意 ① 通常の GEM のモジュールの修正作業

② 'END' の入力前に、任意のグループを定義しておいて、グループ名を入力できる。

③ '==>' のプロンプトに対し '0' を入力すると EFL コマンドから脱出。

④ '0' を入力すると脱出、指定文字以外なら GEM にもどる。

なお、コマンド EFL のキーワード・パラメータおよびリストを第 4.4 表および第 4.4 図に、FILE 2 のリストを第 4.5 図に示す。

第4.4表 TSSコマンドEFLのキーワード・パラメータとその意味

No.	パラメータ	意味	デフォルト値
1	DSN	ソースファイル	—
2	ELOAD	EOS ロード・モジュール	'J2636.EOS200.LOAD(E1)'
3	EOSPARAM	EOSの実行時パラメータ	'LINECM=1, SLIST=0'
4	EOF	EOSへのカード入力ファイル	'*'
5	HE	COMPILER種別	空 白
6	SOURCE	COMPILEリスト・スイッチ	NOSOURCE
7	FIX	フォートランファイル形式	FIXED
8	GO	GE-COMPILERオプション	NOGO
9	SEQ	GE-COMPILERオプション	NOSEQ
10	DBL	倍精度オプション	空 白
11	DATAFILE	LINKERへのコントロールデータ	ソースリスト参照
12	LD	INCLUDEする機番	FLOAD
13	IMPLOAD	ロードモジュール	—
14	PRINTF	プリントファイル	'EOSPRINT.LIST'
15	ID	ID	J2636.

```

00000010PROC O DSN /* SOURCE FILE */-
00000020 ELOAD('J2636.EOS200.LOAD(E1)') /* EOS LOAD MODULE */-
00000030 EOSPARAM('LINECM=1') /* EOS PARAMETER */-
00000040 EOF('*' ) /* EOS INPUT DATA */-
00000050 HE() /* COMPILER */-
00000051 SOURCE(NOSOURCE) /* COMPELE LIST */-
00000060 FIX(FIXED) /* FORTRAN FILE FORMAT */-
00000070 GO(NOGO) /* COMPILER OPTION */-
00000080 SEQ(NOSEQ) /* COMPILER OPTION */-
00000090 DBL() /* COMPILER OPTION */-
00000100 DATAFILE('SYSDATA.DATA(LINK)') /* CONTENT IF AS FOLLOWS */-
00000110 /* INCLUDE FLOAD(A) */-
00000120 /* ENTRY MAIN */-
00000130 /* NAME A(R) */-
00000140 LD(FLOAD) /* DDNAME OF LOAD MODULE */-
00000150 IMPLOAD /* IMPURITY LOADMODULE */-
00000160 PRINTF('*' ) /* PRINT FILE */-
00000170 ID(J2636) /* ID NUMBER */
00000180CONTROL NOFLUSH
00000190 FREE F(FT05F001 FT06F001 FT09F001 FT10F001 FT11F001 FT30F001)
00000200 FREE F(DD1 DD2)
00000330STEP1:SET &STEP=STEP1
00000340 GEM3 &DSN OLD
00000350 GROUP @&COM,MODNAME=#/1
00000360 ALLOC DA(PS1.FORT) F(DD1)
00000370 PUT @&COM,OUT=DD1
00000380 END
00000390 FREE F(DD1)
00000400STEP2:SET &STEP=STEP2
00000420 ATTN OFF
00000430 GEM3 &DSN OLD
00000440 GROUP A
00000441 WRITE $$ GEM3 &DSN START. WHEN UPDATE END, ENTER 'END'
00000450 TERMIN END
00000460 WRITENR $$ MODULE NAME OR GROUP NAME ==>
00000470 READ &B
00000480 IF &B EQ 0 THEN GOTO STEP7
00000490 ALLOC DA(PS2.FORT) F(DD2)
00000500 PUT &B,OUT=DD2
00000510 END
00000520 FREE F(DD2)
00000530 ATTN DO
00000540 WRITE $$ COMMAND INTERUPTED
00000550 GOTO STEP8
00000560 END
00000570STEP3:SET &STEP=STEP3
00000580 ALLOC DA(&EOF) F(FT05F001)
00000590 ALLOC DA(&PRINTF) F(FT06F001)
00000600 ALLOC DA(PS1.FORT PS2.FORT) F(FT09F001)
00000610 ALLOC DA(PS3.FORT) F(FT10F001)
00000620 ALLOC DU F(FT11F001)
00000630 ALLOC DU F(FT30F001)
00000631 IF &LASTCC NE 0 THEN DO
00000632 WRITE %% ERROR CODE =:&LASTCC
00000633 WRITE %% FILE ALLOCATION ERROR FOR EOS
00000634 GOTO STEP8
00000635 END
00000640 WRITE <<< EOS START >>>
00000650 CALL '&ELOAD' '&EOSPARAM'
00000660 WRITE <<< EOS END >>>
00000670 FREE F(FT05F001 FT06F001 FT09F001 FT10F001 FT11F001 FT30F001)
00000671STEP4:SET &STEP=STEP4
00000680 WRITENR $$ IF COMPILE, ENTER 'C' ==>
00000690 READ &COMP
00000700 IF &COMP EQ 0 THEN GOTO STEP7
00000710 IF &COMP NE C THEN GOTO STEP2
00000720 SET &SOURCE=NOSOURCE

```

第 4.4 図 EOS プリプロセッサ利用のための TSS コマンド EFL

```

00000730STEP5:SET &STEP=STEP5
00000740      WRITE $$ COMPILER START
00000750      FORT&HE PS3.FORT &FIX &GO &SEQ &DBL &SOURCE OBJ(PS3.OBJ) PRINT(*)
00000760      IF &LASTCC GT 4 THEN DO
00000770          WRITENR ¥¥ IF COMPILE LIST, ENTER 'S'
00000780          READ &SRC
00000790          IF &SRC EQ 0 THEN GOTO STEP7
00000800          IF &SRC NE S THEN GOTO STEP2
00000810          SET &SOURCE=SOURCE
00000820          GOTO STEP5
00000830      END
00000840      WRITE $$ COMPILE O.K.
00000841STEP6:SET &STEP=STEP6
00000850      WRITENR $$ IF LINK, ENTER 'L' ==>
00000860      READ &LINK
00000870      IF &LINK EQ 0 THEN GOTO STEP7
00000880      IF &LINK NE L THEN GOTO STEP2
00000900      ALLOC DA(&IMpload..LOAD) F(&LD)
00000901      IF &LASTCC NE 0 THEN DO
00000902          WRITE ¥¥ ERROR CODE =:&LASTCC
00000903          WRITE ¥¥ FILE ALLOCATION ERROR FOR LINKAGE
00000904          GOTO STEP8
00000905      END
00000910      LINK (PS3.OBJ &DATAFILE) F LOAD(&IMpload) PRINT(*)
00000920      IF &LASTCC EQ 0 THEN WRITE $$ LINKAGE END. NOW &IMpload UPDATED
00000921      IF &LASTCC NE 0 THEN DO
00000922          WRITE $$ LINKAGE ERROR
00000923          WRITENR $$ IF PROC END, ENTER '0' ==>
00000924          READ &PEND
00000925          IF &PEND NE 0 THEN GOTO STEP2
00000926      END
00000930      GOTO STEP8
00000940STEP7:SET &STEP=STEP7
00000950      END
00000960STEP8:SET &STEP=STEP8
00000970      EXIT Q
00000980      END

```

第4.4図 (続き)

```

00000010FREE ATTR(EOSATTR)
00000020FREE ATTR(EOSPR)
00000030DELETE PS1.FORT
00000040DELETE PS2.FORT
00000050DELETE PS3.FORT
00000060DELETE EOSPRINT.LIST
00000100ATTR EOSATTR RECFM(F B)  LRECL(80)  BLKSIZE(3120) DSORG(PS)
00000110ATTR EOSPR  RECFM(V B A) LRECL(125) BLKSIZE(4840) DSORG(PS)
00000120ALLOC DA(EOSPRINT.LIST) NE CA SP(30,10) T US(EOSPR)
00000130ALLOC DA(PS1.FORT)      NE CA SP(30,10) T US(EOSATTR)
00000150ALLOC DA(PS2.FORT)      NE CA SP(30,10) T US(EOSATTR)
00000170ALLOC DA(PS3.FORT)      NE CA SP(30,10) T US(EOSATTR)
00000190FREE  DA(EOSPRINT.LIST)
00000200FREE  DA(PS1.FORT)
00000220FREE  DA(PS2.FORT)
00000240FREE  DA(PS3.FORT)
00000260WRITE ** ALLOCATION END ( EOSPRINT.LIST, PS1.FORT, PS2.FORT, PS3.FORT )

```

第4.5図 TSS コマンド FILE2

4.5 EOSプリプロセッサの使用例

EOSプリプロセッサの使用例を第4.6図, 第4.7図に示す。それぞれの図で(a)は入力のEOSソース・プログラム, (b)が変換情報, (c)は出力のフォートラン・ソース・プログラムである。また, 第4.6図は, EOSコマンド記号として標準値の*を, 第4.7図は%を使用した例である。第4.6図の入力のEOSソース・プログラムは, J2636.EOS200.DATA (E1TEST)であり, JCLは第4.3図のものを利用した。

```

SOURCE( ) EOS PREPROCESSOR V2.0 DATE 82-06-18 PAGE 1
C----- SIZE BLOCK -----00000010
M=2 00000020
N=3,N2=N**M,L=M+N,A= /00000030
L=5 00000040
C----- SIZE BLOCK END -----00000050

```

```

SOURCE( ) EOS PREPROCESSOR V2.0 DATE 82-06-18 PAGE 2
*COMDECK C1 00000060
COMMON/ C1 / X1(M,N),X2(L,A+L) 00000070

*COMDECK C1#2 00000080
COMMON/ C1 / XD(156) 00000090

*COMDECK EXECUT 00000100
A=*(N**M+A) 00000110

*COMDECK DATA 00000120
DATA Y1/*(M+A)/, Y2/*(L)/ 00000130

*COMDECK FORMAT 00000140
100 FORMAT(1H,*(M)HAB,*(A+L)X,'X2(L,5)=' ,1PE15.7) 00000150

```

```

SOURCE( ) EOS PREPROCESSOR V2.0 DATE 82-06-18 PAGE 3
*DECK SUB1 00000160
C----- TEST SUBROUTINE -----00000170
SUBROUTINE SUB1 00000180
*CALL C1 00000190
X1(1,1)=0.0 00000200
RETURN 00000210
E N D 00000220

```

```

SOURCE( ) EOS PREPROCESSOR V2.0 DATE 82-06-18 PAGE 4
*DECK FTMAIN 00000230
C----- TEST FTMAIN -----00000240
*CALL C1#2 00000250
*CALL DATA 00000260
DO 10 I=1,156 00000270
XD(I)=0.0 00000280
10 CONTINUE 00000290
CALL SUB1 00000300
STOP 00000310
END 00000320
*COMDECK C2 00000330
C----- INSERTED COMDECK -----00000340
COMMON/ C2 / Z1(N,L) 00000350
L = *L 00000360

```

```

SOURCE( ) EOS PREPROCESSOR V2.0 DATE 82-06-18 PAGE 5
*DECK FUNC 00000370
C----- TEST FUNCTION -----00000380
REAL FUNCTION FUNC*B 00000390
*CALL C1 /X1/XR1/, 'X2'XR2' 00000400
*CALL C2 00000410
DO 20 I=1,L 00000420
XR2(I,5)=Z1(5,I) 00000430
20 CONTINUE 00000440
*CALL EXECUT 00000450
WRITE(6,100) XR2(L,5) 00000460
*CALL FORMAT 00000470
RETURN 00000480
E N D 00000490

```

第 4.6 図 EOS プリプロセッサ使用例(EOS コマンド記号：*)
(a) EOS ソースプログラム

```

===== E O S   T O T A L   A C C O U N T   T A B L E   =====
TOTAL CONDITION =      0
CARD             =     44
ERROR NUMBER    =      0
FTMAIN          CPU TIME =          138 (MS)

```

(b) 変換情報

```

C----- TEST SUBROUTINE -----00000170
SUBROUTINE SUB1
COMMON/ C1 / X1(2,3),X2(5,30) 00000180
X1(1,1)=0.0 C1000070
RETURN 00000200
E N D 00000210
C----- TEST FTMAIN -----00000220
COMMON/ C1 / XD(156) 00000240
DATA Y1/27/, Y2/5/ C1000090
DO 10 I=1,156 DAT00130
XD(I)=0.0 00000270
10 CONTINUE 00000280
CALL SUB1 00000290
STOP 00000300
END 00000310
C----- TEST FUNCTION -----00000320
REAL FUNCTION FUNC*B 00000380
COMMON/ C1 /XR1(2,3),XR2(5,30) 00000390
C1000070
C----- INSERTED COMDECK -----C2000340
COMMON/ C2 / Z1(3,5) C2000350
L = 5 C2000360
DO 20 I=1,L 00000420
XR2(I,5)=Z1(5,I) 00000430
20 CONTINUE 00000440
A=34 EXE00110
WRITE(6,100) XR2(L,5) 00000460
100 FORMAT(1H ,2HAB,30X,'X2(L,5)=' ,1PE15.7) FOR00150
RETURN 00000480
E N D 00000490

```

(c) フォートラン・ソース・プログラム

第4.6図 (続き)

```

SOURCE( ) EOS PREPROCESSOR V2.0 DATE 82-06-18 PAGE 1
C----- SIZE BLOCK -----00000010
*----- ASTER OPTION TEST LINE 1/3 -----00000011
M=2 00000020
N=3,N2=N**M,L=M+N,A= /00000030
L=5 00000040
C----- SIZE BLOCK END-----00000050

SOURCE( ) EOS PREPROCESSOR V2.0 DATE 82-06-18 PAGE 2
XCOMDECK C1 00000060
*----- ASTER OPTION TEST LINE 2/3 -----00000061
COMMON/ C1 / X1(M,N),X2(L,A+L) 00000070

XCOMDECK C1#2 00000080
COMMON/ C1 / XD(156) 00000090

XCOMDECK VACANT 00000091

XCOMDECK EXECUT 00000100
A=(N**M+A) 00000110

XCOMDECK DATA 00000120
DATA Y1/(M+A)/, Y2/(L)/ 00000130

XCOMDECK FORMAT 00000140
100 FORMAT(1H,*(M)HAB,(A+L)X,'X2(L,5)=' ,1PE15.7) 00000150

SOURCE( ) EOS PREPROCESSOR V2.0 DATE 82-06-18 PAGE 3
XDECK SUB1 00000160
C----- TEST SUBROUTINE -----00000170
*----- ASTER OPTION TEST LINE 3/3 -----00000171
SUBROUTINE SUB1 00000180
XCALL C1 00000190
X1(1,1)=0.0 00000200
RETURN 00000210
END 00000220

SOURCE( ) EOS PREPROCESSOR V2.0 DATE 82-06-18 PAGE 4
XDECK FTMAIN 00000230
C----- TEST FTMAIN -----00000240
XCALL C1#2 00000250
XCALL DATA 00000260
XCALL VACANT 00000261
DO 10 I=1,156 00000270
XD(I)=0.0 00000280
10 CONTINUE 00000290
CALL SUB1 00000300
STOP 00000310
END 00000320
XCOMDECK C2 00000330
C----- INSERTED COMDECK -----00000340
COMMON/ C2 / Z1(N,L) 00000350
L = *L 00000360

SOURCE( ) EOS PREPROCESSOR V2.0 DATE 82-06-18 PAGE 5
XDECK FUNC 00000370
C----- TEST FUNCTION -----00000380
REAL FUNCTION FUNC*B 00000390
XCALL C1 /X1/XR1/,'X2'XR2' 00000400
XCALL C2 00000410
DO 20 I=1,L 00000420
XR2(I,5)=Z1(5,I) 00000430
20 CONTINUE 00000440
XCALL EXECUT 00000450
WRITE(6,100) XR2(L,5) 00000460
XCALL FORMAT 00000470
RETURN 00000480
END 00000490

```

第 4.7 図 EOS プリプロセッサ使用例 (EOS コマンド記号: %)

(a) EOS ソースプログラム


```

===== E O S   T O T A L   A C C O U N T   T A B L E   =====
TOTAL CONDITION = 0
CARD             = 48
ERROR NUMBER    = 0
FTMAIN          CPU TIME = 145 (MS)

```

(b) 変換情報

```

C----- TEST SUBROUTINE -----00000170
*----- ASTER OPTION TEST LINE 3/3 -----00000171
SUBROUTINE SUB1 00000180
*----- ASTER OPTION TEST LINE 2/3 -----C1000061
COMMON/ C1 / X1(2,3),X2(5,30) C1000070
X1(1,1)=0.0 00000200
RETURN 00000210
E N D 00000220
C----- TEST FTMAIN -----00000240
COMMON/ C1 / XD(156) C1000090
DATA Y1/27/, Y2/5/ DAT00130
DO 10 I=1,156 00000270
XD(I)=0.0 00000280
10 CONTINUE 00000290
CALL SUB1 00000300
STOP 00000310
END 00000320
C----- TEST FUNCTION -----00000380
REAL FUNCTION FUNC*8 00000390
*----- ASTER OPTION TEST LINE 2/3 -----C1000061
COMMON/ C1 /XR1(2,3),XR2(5,30) C1000070
C----- INSERTED COMDECK -----C2000340
COMMON/ C2 / Z1(3,5) C2000350
L = 5 C2000360
DO 20 I=1,L 00000420
XR2(I,5)=Z1(5,I) 00000430
20 CONTINUE 00000440
A=34 EXE00110
WRITE(6,100) XR2(L,5) 00000460
100 FORMAT(1H ,2HAB,30X,'X2(L,5)=' ,1PE15.7) FOR00150
RETURN 00000480
E N D 00000490

```

(c) フォートラン・ソース・プログラム

第4.7図 (続き)

5. 逆EOSプリプロセッサ

5.1 逆EOSプリプロセッサの機能

逆EOSプリプロセッサは、任意のフォートラン・ソース・プログラムを読み込んで、EOSソース・プログラムを生成する処理系である。逆EOSプリプロセッサの基本的な機能はフォートラン・ソース・プログラム中のコモン文をEOSのテキスト挿入文におきかえ、かつ、対応する「テキスト・ブロック」を生成することである。「テキスト・ブロック」の生成の際には、コモン変数を含む他の宣言文（型宣言文、デイメンション文等）の情報も取り込んだ形式のテキストが作られる。なお、コモン文の取り出し（テキスト挿入文の挿入）に際しては、① 取り出しの対象になるコモン文の指定（取り出し処理ラベルの選択）、および、② 同一ラベル・コモンに対する複数の「テキスト・ブロック」の作成（同一ラベル名コモン文の別名）、が可能である。以下、これらについて詳述する。

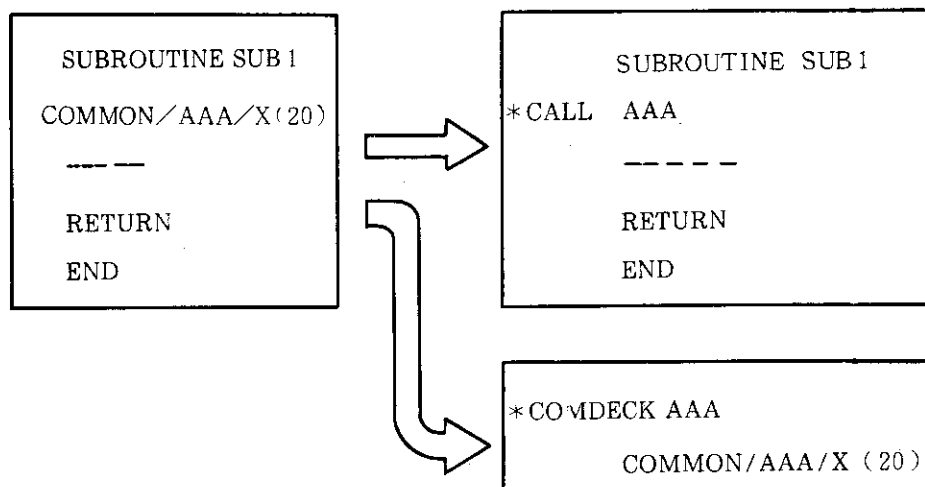
(1) コモン文のおきかえとテキスト・ブロックの生成

逆EOSプリプロセッサは、フォートラン・ソース・プログラム中のコモン文を読み込むと、ラベル名をテキスト名とするテキスト挿入文を生成し、原プログラム中のコモン文と置換する。同時に、ラベル名をCOMDECK名とするテキスト・ブロックを別のファイル（テキスト・ブロック・ファイル）上に生成する。この様子は、第5.1図に示すとおりである。

(2) 取り出し処理ラベルの選択

逆EOSプリプロセッサを利用するに際し、入力データによって、取り出すべきコモン文のラベルを指定することができる。指定の方法は、第5.1表に示した3種類が可能である。

(3) 同一ラベル名コモン文の別名



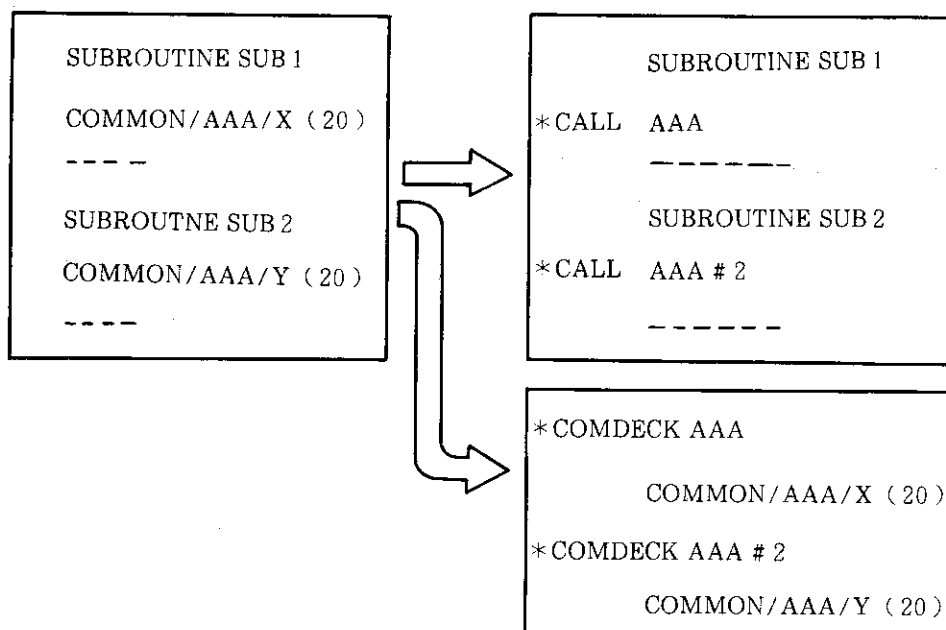
第5.1図 フォートラン・ソース・プログラムから
EOSソース・プログラムの作成

一般のフォートラン・ソース・プログラム中においては、同一ラベル名のコモン文であっても、サブルーチンによってその内容が異なる場合がある。このような場合のために、テキスト・ブロック・ファイルに登録される「テキスト・ブロック」は、同一ラベル名であってもテキストの内容の異なるものについては、別の名前のテキスト・ブロックとして全て登録されるようになっている。この際、ラベル名「AAA」に対応するCOMDECK名は、「AAA #n」の形で表わされ、番号nはプログラム全体の中で出現頻度の高いものから順に1, 2, ……と付けられる。具体例を第5.2図に示す。

第5.1表 取り出し処理ラベルの指定方法

ここで、ラベル名リストは、ラベル名 [, ラベル名, …]
の形をとる。

- | | |
|----|-----------------------------------------------------------------------------------------------------------------------------------|
| 1. | <ul style="list-style-type: none"> • EXTRACT, ラベル名リスト <p>ラベル名リストに書かれた COMMON ラベルのみが取出しの対象となる。</p> |
| 2. | <ul style="list-style-type: none"> • HOLD, ラベル名リスト <p>ラベル名リストに書かれた COMMON ラベルは、取出されず、それ以外のすべての COMMON 文が取出される。</p> |
| 3. | <p>指定なし</p> <p>すべての COMMON 文が取出される。</p> |



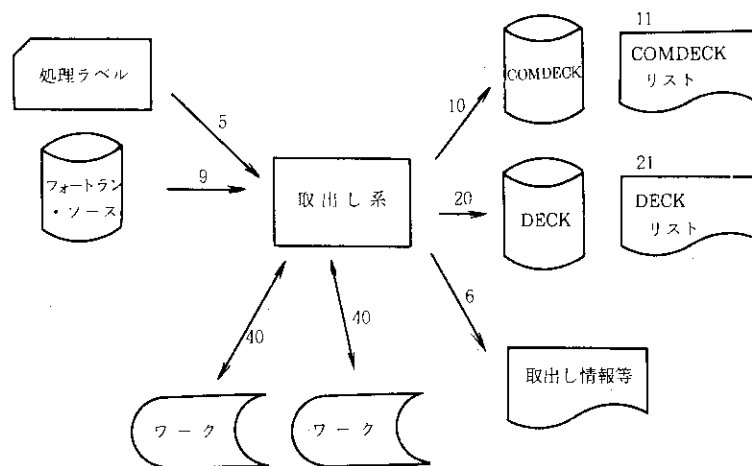
第5.2図 同一ラベル名コモン文に対する別名付与の例

5.2 ジョブ制御文の構成

逆EOSプリプロセッサを起動する際のジョブ制御文の例を第5.3図に示す。この例は、フォートラン・ソース・プログラム“FORT”を処理し、テキスト・ブロック・ファイル“COMDECK”とプログラム・ファイル“DECK”を作成するためのJCLの例である。なお、実際の利用にあたっては、5.4、5.5節のJCLマクロ、TSSコマンドを使うのが便利である。第5.3図において、第1行目の「PARM=」に示したリストは、オプション・パラメータ・リストで詳細は次節で述べる。また「E2」は、ロード・モジュール・データ・セットJ2636.EOS200.LOADの中の逆EOSプリプロセッサを表すメンバー名である。逆EOSプリプロセッサを起動する際に必要な入出力機番は、5、6、9、10、11、20、21、30、40の9種であるが、それぞれの役割については、第5.2表にまとめてある。また、各入出力機器の関係は第5.4図に示した。

```
// EXEC PGM=E2,PARM='SAVE=1,FPR2=1'
//STEPLIB DD DSN=J2636.EOS200.LOAD,DISP=SHR
//SYSPRING DD SYSOUT=*
//FT05F001 DD *
.EXTRACT AAA
/*
//FT06F001 DD SYSOUT=*
//FT09F001 DD DSN=FORT,DISP=OLD
//FT10F001 DD DSN=COMDECK,DISP=(NEW,CATLG),UNIT=USER,
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=1200,DSORG=RS),
//          SPACE=(TRK,(30,10))
//FT11F001 DD SYSOUT=*
//FT20F001 DD DSN=DECK,DISP=(NEW,CATLG),UNIT=USER,
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=1200,DSORG=PS),
//          SPACE=(TRK,(30,10))
//FT30F001 DD DSN=&&WK1,DISP=NEW,UNIT=SYSDA,
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=1200,DS=PS),
//          SPACE=(TRK,(30,10))
//FT40F001 DD DSN=&&WK2,DISP=NEW,UNIT=SYSDA,
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=1200,DS=PS),
//          SPACE=(TRK,(10,10))
```

第5.3図 逆EOSプリプロセッサのJCLの例



第5.4図 逆EOSプリプロセッサに関する入出力機器の関係図

第 5.2 表 プリプロセスに際しての各入出力機番の役割

入出力機番	役 割
5	<p>ラベル名指定入力, (1 RECORD = 80 BYTE)</p> <p>第 1 カラムから第 71 カラムまでが使える。第 72 カラムは継続行カラムである。</p> <p>各カードは先頭に KEY WORD ".EXTRACT" 又は ".HOLD" が書かれ、次にラベル名リストが並ぶ。</p> <p>例</p> <div style="border: 1px solid black; padding: 5px; margin: 10px auto; width: fit-content;"> <pre>.EXTRACT DDD</pre> </div> <div style="border: 1px solid black; padding: 5px; margin: 10px auto; width: fit-content;"> <pre>.EXTRACT AAA , BBB</pre> </div>
6	<p>逆 EOS プリプロセッサの取出し処理情報が出力される。 (SYSOUT データセット)</p> <p>出力される情報には次の様なものがある。</p> <ol style="list-style-type: none"> 1. ヘッダー 2. 入力ソースリスト (OPTION) 3. エラーメッセージ 4. ルーチン毎の処理情報 <p style="padding-left: 40px;">関連する COMMON 文や他の宣言文, 編集の影響を受ける変数名</p> <ol style="list-style-type: none"> 5. 全体の会計情報
9	<p>フォートランソースの入力 (1 RECORD = 80 BYTE), 順編成ファイルでなくてはいけない。</p>
10	<p>COMDECK ソースの出力 (1 RECORD = 80 BYTE), 順編成ファイルでなくてはいけない。</p>
11	<p>COMDECK ソースのリスティング (SYSOUT データセット)</p>
20	<p>DECK ソースの出力 (1 RECORD = 80 BYTE), 順編成ファイルでなくてはいけない。</p>
21	<p>DECK ソースのリスティング (SYSOUT データセット)</p>
30	<p>ワーク・ファイル (1 RECORD = 80 BYTE), 順編成ファイルでなくてはいけない。</p>
40	<p>ワーク・ファイル (1 RECORD = 80 BYTE), 順編成ファイルでなくてはいけない。</p>

5.3 実行時オプション・パラメータ

逆EOSプリプロセッサの利用に際しては、実行時オプション・パラメータとして第5.3表に示すものを指定することができる。

第5.3表 逆EOSプリプロセッサの実行時オプション・パラメータ

No.	パラメータ	標準	意味
1	PREC	▼*▼	EOS コマンド記号
2	SLIST	0	入力ソースリストのプリント (SLIST=1で出力)
3	SAVE	1	出力EOSファイルへの書込み (SAVE=1で書込む)
4	FPR 1	1	COMDECKソースリスティング (FPR 1=1で出力)
5	FPR 2	0	DECKソースリスティング (FPR 2=1で出力)

(1) オプション・パラメータの詳細

- ① PREC 出力されるEOS形式ソースで使用されるEOSコマンド記号。
標準値は▼*▼。
- ② SLIST 入力されるフォートラン・ソースのリスティング制御パラメータ。
SLIST > 0の時、機番6に入力ソース・イメージが出力される。
標準値は0。
- ③ SAVE 出力機番への書込み指定。
SAVE > 0の時、機番10と機番20に各々、COMDECKとDECKが出力される。
標準値は1。
- ④ FPR 1 COMDECKファイルのリスティング制御パラメータ。
FPR 1 > 0の時、機番11に機番10と同一のリストが出力される (制御コード付で出力される)。
標準値は1。
- ⑤ FPR 2 DECKファイルのリスティング制御パラメータ。
FPR 2 > 0の時、機番21に機番20と同一のリストが出力される (制御コード付で出力される)。
標準値は0。

(2) オプション・パラメータの入力方法

逆 EOS プリプロセッサに対するオプション・パラメータの入力はプログラム実行時パラメータの文字列としてプログラムに渡される。

・例:*.....*.....*.....

バッチ処理 // EXEC PGM=E2,PARM='SAVE=0,FPR2=1'

TSS 処理 CALL EOS200.LOAD(E2) 'SAVE=0,FPR2=1'

この際、一般に、複数個のパラメータを指定できる。即ち、入力文字列は次のようになる。

パラメータ名=値 [, パラメータ名=値, ……]

また、入力文字列が引用符を必要とする時は、引用符を2回続ければよい。即ち、

・例:*.....*.....*.....*.....

// EXEC PGM=E2,PARM='PREC='%'',SLIST=1'

次節の JCL プロシージャでは、全てのオプション・パラメータの指定が可能である。

5.4 JCLプロシジャ

逆EOSプリプロセッサを起動するために、プロシジャFOREOSが用意されている。これは、PS形式のフォートラン・ソース・プログラムを変換して、PS形式のEOSソース・プログラムを生成するためのものである。実際に使う時は、4.3節に記したように、GEMPO, POGEM, POPS, PSEPO等のマクロ・プロシジャと組合わせて利用する。プロシジャFOREOSのパラメータ・リストおよびプロシジャ・リストは、それぞれ、第5.4表および第5.5図に示す。

5.5 逆EOSプリプロセッサの使用例

逆EOSプリプロセッサの使用例について示す。第5.6図に示したものが逆EOSプリプロセッサに対する入力データ（フォートラン・ソース・プログラム）の例で、第5.7図には、これに対応する出力データ（EOSソース・プログラム）および取出し処理情報が示されている。

表 5.4 表 プロシジャFOREOSパラメータ・リスト

パラメータ	意 味	デフォルト値
DSN	入力データ・セット名（フォートラン, PS）	
ODSN	出力データ・セット名（EOS, PS）	
UNIT	出力データ・セット・ユニット指定	TSSWK
SYSOUT	SYSOUTパラメータ	'*'
OLD	出力データ・セット DISPパラメータ	NEW
CATLG	出力データ・セット DISPパラメータ	CATLG
SPC	出力データ・セット・スペース指定	'30,10'
PREC	EOSコマンド記号	*****
SLIST	入力ソース・リスト・プリント・スイッチ	1（プリント）
SAVE	出力ソース・ファイル書込みスイッチ	1（書込み）
FPR 1	出力COMDECKプリント・スイッチ	1（プリント）
FPR 2	出力DECKプリント・スイッチ	1（プリント）


```

//*****
//*      FOREOS      :  FORTRAN(PS) TO EOS(PS)      * 00000010
//*****
//*      DSN        :  FORTRAN SOURCE (PS) (      ) * 00000020
//*      ODSN       :  EOS SOURCE (PS) ( NULLFILE ) * 00000030
//*      UNIT       :  OUTPUT UNIT ( TSSWK ) * 00000040
//*      SYSOUT     :  SYSOUT CLASS ( '*' ) * 00000050
//*      OLD        :  DISP PARAMETER ( NEW ) * 00000060
//*      CATLG      :  DISP PARAMETER ( CATLG ) * 00000070
//*      SPC        :  SPACE FOR ODSN ( '30,10' ) * 00000080
//*      SPCCM      :  SPACE FOR &&COMDSN ( '10,5' ) * 00000090
//*      SPCDK      :  SPACE FOR &&DEKDSN ( '30,10' ) * 00000100
//*****
//FOREOS PROC DSN=,ODSN=NULLFILE,UNIT=TSSWK,SYSOUT='*',OLD=NEW,
//          CATLG=CATLG,SPC='30,10',SPCCM='10,5',SPCDK='30,10',
//          PREC='''*''',SLIST=1,SAVE=1,FPR1=1,FPR2=1
//STEP1 EXEC PGM=E2,
//          PARM='PREC=&PREC,SLIST=&SLIST,FPR1=&FPR1,FPR2=&FPR2'
//STEPLIB DD DSN=J2636.EOS200.LOAD,DISP=SHR
//SYSPRINT DD SYSOUT=&SYSOUT
//FT05F001 DD DDNAME=SYSIN
//FT06F001 DD SYSOUT=&SYSOUT
//FT09F001 DD DSN=&DSN,DISP=OLD
//FT10F001 DD DSN=&&COMDSN,UNIT=WK10,
//          DISP=(NEW,PASS,DELETE),SPACE=(TRK,(&SPCCM)),
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=3200,DSORG=PS)
//FT11F001 DD SYSOUT=&SYSOUT,
//          DCB=(BLKSIZE=13700,LRECL=137,RECFM=FBA)
//FT20F001 DD DSN=&&DEKDSN,UNIT=WK10,
//          DISP=(NEW,PASS,DELETE),SPACE=(TRK,(&SPCDK)),
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=3200,DSORG=PS)
//FT21F001 DD SYSOUT=&SYSOUT
//FT30F001 DD DSN=&&WK1,DISP=(NEW,DELETE),SPACE=(TRK,(30,10)),
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=3200,DSORG=PS),UNIT=WK10
//FT40F001 DD DSN=&&WK2,DISP=(NEW,DELETE),SPACE=(TRK,(30,10)),
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=3200,DSORG=PS),UNIT=WK10
//STEP2 EXEC PGM=JSDGENER
//SYSIN DD DUMMY
//SYSPRINT DD SYSOUT=&SYSOUT
//SYSUT1 DD DSN=&&COMDSN,DISP=(OLD,DELETE)
//          DD DSN=&&DEKDSN,DISP=(OLD,DELETE)
//SYSUT2 DD DSN=&ODSN,DISP=(OLD,&CATLG,DELETE),SPACE=(TRK,(&SPC)),
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=3200,DSORG=PS),UNIT=&UNIT
//*****

```

第 5.5 図 逆EOS プリプロセッサ利用のためのプロシジャ・リスト (FOREOS)

```

C----- TEST SUBROUTINE -----00000010
SUBROUTINE SUB1(P)
COMMON X1(100,5), X2(2,2,7)
COMMON/ C1 / Y1,Y2,Y3(80)
COMMON/ C2 / Z, ZP
REAL*16 Y1,YD1,X2
DOUBLE PRECISION Y2
READ(5,100,END=10) Y3
100 FORMAT(80A1)
10 RETURN
E N D
C----- TEST FTMAIN -----000000110
COMMON/ C1 /YY1 / C2 / ZZ(2)
COMMON// XD
DIMENSION XD(528)
REAL*4 YY1(82)
CALL SUB1(5)
STOP
E N D
C----- TEST FUNCTION -----000000120
INTEGER FUNCTION FUNC*2(P1,P2)
COMMON/ C1 /YY1
COMMON/ C2 /ZP, Z
REAL*4 YY1
DIMENSION YY1(82)
EQUIVALENCE( WK(1),YY1(3))
INTEGER*2 WK(2)
FUNC = 0.1
RETURN
E N D
C----- TEST BLOCK DATA -----000000190
BLOCK DATA
COMMON/ C1 / YD1, YD2, YD3(80)//XD / C2 / ZZ(2)
REAL*4 XD(528)/528*0.0/
DATA (YD3(J),J=1,80)/1.0,79*0.0/
E N D

```

第 5.6 図 逆EOS プリプロセッサ入力データとしてのフォートラン・ソース・プログラムの例

```

C----- TEST SUBROUTINE -----00000010
      SUBROUTINE SUB1(P)
      00000020
C----- TEST SUBROUTINE -----00000010
      SUBROUTINE SUB1(P)
      00000020
      COMMON X1(100,5), X2(2,2,7)
      00000030
      COMMON/ C1 / Y1,Y2,Y3(80)
      00000040
      COMMON/ C2 / Z, ZP
      00000050
      REAL*16 Y1,YD1,X2
      00000060
      DOUBLE PRECISION Y2
      00000070
      READ(5,100,END=10) Y3
      00000080
***** // SUB1 // *****

## REPLACED STATEMENT(S)
      COMMON X1(100,5), X2(2,2,7)
      00000030
      COMMON/ C1 / Y1,Y2,Y3(80)
      00000040
      COMMON/ C2 / Z, ZP
      00000050
      REAL*16 Y1,YD1,X2
      00000060
      DOUBLE PRECISION Y2
      00000070
100 FORMAT(80A1)
      00000090
10 RETURN
      00000100
      E N D
      00000110

## WARNING
      1 Y1
      2 X2
      3 Y2

C----- TEST FTMAIN -----00000120
      COMMON/ C1 /YY1 / C2 / ZZ(2)
      00000130
C----- TEST FTMAIN -----00000120
      COMMON/ C1 /YY1 / C2 / ZZ(2)
      00000130
      COMMON// XD
      00000140
      DIMENSION XD(528)
      00000150
      REAL*4 YY1(82)
      00000160
      CALL SUB1(5)
      00000170
      STOP
      00000180
      E N D
      00000190
***** // FTMAIN // *****

## REPLACED STATEMENT(S)
      COMMON/ C1 /YY1 / C2 / ZZ(2)
      00000130
      COMMON// XD
      00000140
      DIMENSION XD(528)
      00000150
      REAL*4 YY1(82)
      00000160

## WARNING
      1 XD
      2 YY1

C----- TEST FUNCTION -----00000200
      INTEGER FUNCTION FUNC*2(P1,P2)
      00000210
C----- TEST FUNCTION -----00000200
      INTEGER FUNCTION FUNC*2(P1,P2)
      00000210
      COMMON/ C1 /YY1
      00000220
      COMMON/ C2 /ZP, Z
      00000230
      REAL*4 YY1
      00000240
      DIMENSION YY1(82)
      00000250
      EQUIVALENCE( WK(1),YY1(3))
      00000260
      INTEGER*2 WK(2)
      00000270
      FUNC = 0.1
      00000280
***** // FUNC // *****

## REPLACED STATEMENT(S)
      COMMON/ C1 /YY1
      00000220
      COMMON/ C2 /ZP, Z
      00000230
      REAL*4 YY1
      00000240
      DIMENSION YY1(82)
      00000250
      EQUIVALENCE( WK(1),YY1(3))
      00000260
      RETURN
      00000290
      E N D
      00000300

```

(a) 変換情報

第 5.7 図 第 5.6 図の入力データに対する逆 EOS プリプロセッサの出力

WARNING
1 YY1

```
C----- TEST BLOCK DATA -----00000310
      BLOCK DATA                                00000320
C----- TEST BLOCK DATA -----00000310
      BLOCK DATA                                00000320
      COMMON/ C1 / YD1, YD2, YD3(80)//XD / C2 / ZZ(2) 00000330
      REAL*4 XD(528)/528*0.0/                    00000340
      DATA (YD3(J),J=1,80)/1.0,79*0.0/         00000350
      E N D                                       00000360
***** // BLOCKD // *****
```

```
## REPLACED STATEMENT(S)
      COMMON/ C1 / YD1, YD2, YD3(80)//XD / C2 / ZZ(2) 00000330
      REAL*4 XD(528)/528*0.0/                    00000340
```

WARNING
1 XD

```
*DECK SUB1
C----- TEST SUBROUTINE -----00000010
      SUBROUTINE SUB1(P)                          00000020
*CALL BLANKC                                    00000030
*CALL C1                                        00000040
*CALL C2                                        00000050
      READ(5,100,END=10) Y3                       00000080
      100 FORMAT(80A1)                            00000090
      10 RETURN                                    00000100
      E N D                                       00000110
*DECK FTMAIN
C----- TEST FTMAIN -----00000120
*CALL C1#2                                      00000130
*CALL C2#2                                      00000130
*CALL BLANKC#2                                  00000140
      CALL SUB1(5)                                00000170
      STOP                                         00000180
      E N D                                       00000190
*DECK FUNC
C----- TEST FUNCTION -----00000200
      INTEGER FUNCTION FUNC*2(P1,P2)             00000210
*CALL C1#2                                      00000220
*CALL C2#3                                      00000230
      EQUIVALENCE( WK(1),YY1(3))                 00000260
      INTEGER*2 WK(2)                             00000270
      FUNC = 0.1                                  00000280
      RETURN                                       00000290
      E N D                                       00000300
*DECK BLOCKD
C----- TEST BLOCK DATA -----00000310
      BLOCK DATA                                00000320
*CALL C1#3                                      00000330
*CALL BLANKC#3                                  00000330
*CALL C2#2                                      00000330
      DATA (YD3(J),J=1,80)/1.0,79*0.0/         00000350
      E N D                                       00000360
```

```
===== EOS REVERSE TOTAL ACCOUNT TABLE =====
TOTAL CONDITION = 0
CARD = 72
ERROR NUMBER = 0
```

第 5.7 図(a) (続き)

```

*COMDECK BLANKC
COMMON//
1 X1(100,5), X2(2,2,7)
*COMDECK BLANKC#2
COMMON//
1 XD(528)
*COMDECK BLANKC#3
COMMON//
1 XD(528)
*COMDECK C1#2
COMMON/ C1 /
1 Y1, Y2, Y3(80)
*COMDECK C1
COMMON/ C1 /
1 YY1(82)
*COMDECK C1#3
COMMON/ C1 /
1 YD1, YD2, YD3(80)
*COMDECK C2#2
COMMON/ C2 /
1 Z, ZP
*COMDECK C2
COMMON/ C2 /
1 ZZ(2)
*COMDECK C2#3
COMMON/ C2 /
1 ZP, Z

```

BLANKCOO
 CONTINUE
 CONTINUE
 BLANKCOO
 CONTINUE
 CONTINUE
 BLANKCOO
 CONTINUE
 CONTINUE
 C1000000
 CONTINUE
 CONTINUE
 C1000000
 CONTINUE
 CONTINUE
 C1000000
 CONTINUE
 CONTINUE
 C2000000
 CONTINUE
 CONTINUE
 C2000000
 CONTINUE
 CONTINUE
 C2000000
 CONTINUE
 CONTINUE
 C2000000
 CONTINUE
 CONTINUE

(b) EOS ソース・プログラムのテキストブロック

```

*DECK SUB1
C----- TEST SUBROUTINE -----00000010
SUBROUTINE SUB1(P)
00000020
*CALL BLANKC
00000030
*CALL C1#2
00000040
*CALL C2#2
00000050
READ(5,100,END=10) Y3
00000080
100 FORMAT(80A1)
00000090
10 RETURN
00000100
E N D
00000110
*DECK FTMAIN
C----- TEST FTMAIN -----00000120
*CALL C1
00000130
*CALL C2
00000130
*CALL BLANKC#2
00000140
CALL SUB1(5)
00000170
STOP
00000180
E N D
00000190
*DECK FUNC
C----- TEST FUNCTION -----00000200
INTEGER FUNCTION FUNC*2(P1,P2)
00000210
*CALL C1
00000220
*CALL C2#3
00000230
EQUIVALENCE( WK(1),YY1(3))
00000260
INTEGER*2 WK(2)
00000270
FUNC = 0.1
00000280
RETURN
00000290
E N D
00000300
*DECK BLOCKD
C----- TEST BLOCK DATA -----00000310
BLOCK DATA
00000320
*CALL C1#3
00000330
*CALL BLANKC#3
00000330
*CALL C2
00000330
DATA (YD3(J),J=1,80)/1.0,79*0.0/
00000350
E N D
00000360

```

(c) EOS ソース・プログラムのプログラム・ブロック

6. ま と め

TRITON計画の一環として可変配列サイズのフォートラン・プログラムの処理用プリプロセッサ・システム EOS (EOS, 逆EOS プリプロセッサ) の開発を行った。このプリプロセッサを利用することにより、必要最少限の記憶容量のもとで大規模計算コードの開発および利用が能率よく実行されるようになった。この報告書は EOS システムを利用する際の使用手引書となるものである。

この報告書に記した EOS システムは、現在、核融合計算コードの開発とこれらによる解析のために積極的に利用されているが、まだいくつかの改良点が存在する。例えば、ジョブ制御文や、TSS コマンドの一層の整備、EOS システム全体の簡易化等があげられる。また、EOS, 逆EOS プリプロセッサの機能は、本来ならば、少なくともフォートラン・コンパイラが持っていることが望ましいものである。実際、フォートラン 77 では INCLUDE 文を使うことによって EOS プリプロセッサの一部の機能を利用することが可能である。今後、一層機能が充実されることが望ましい。

謝 辞

EOS システムの開発に際しては松下 樹氏 (数理技研) の献身的努力に負う所が大きい。ここに記して感謝に変えます。安積 正史, 滝塚 知典, 徳田 伸二, 田中 幸夫, 奈良岡 賢逸各氏との有益な議論は、本システムの完成に大きな力となった事を記し感謝します。TRITON 計画全体を通じての、核融合研究部小幡 行雄部長, 田中 正俊次長の御激励に深く感謝します。

6. ま と め

TRITON計画の一環として可変配列サイズのフォートラン・プログラムの処理用プリプロセッサ・システム EOS (EOS, 逆EOSプリプロセッサ) の開発を行った。このプリプロセッサを利用することにより、必要最少限の記憶容量のもとで大規模計算コードの開発および利用が能率よく実行されるようになった。この報告書はEOSシステムを利用する際の使用手引書となるものである。

この報告書に記したEOSシステムは、現在、核融合計算コードの開発とこれらによる解析のために積極的に利用されているが、まだいくつかの改良点が存在する。例えば、ジョブ制御文や、TSSコマンドの一層の整備、EOSシステム全体の簡易化等があげられる。また、EOS, 逆EOSプリプロセッサの機能は、本来ならば、少くともフォートラン・コンパイラが持っていることが望ましいものである。実際、フォートラン77ではINCLUDE文を使うことによってEOSプリプロセッサの一部の機能を利用することが可能である。今後、一層機能が充実されることが望ましい。

謝 辞

EOSシステムの開発に際しては松下 樹氏(数理技研)の献身的努力に負う所が大きい。ここに記して感謝に変えます。安積 正史, 滝塚 知典, 徳田 伸二, 田中 幸夫, 奈良岡 賢逸各氏との有益な議論は、本システムの完成に大きな力となった事を記し感謝します。TRITON計画全体を通じての、核融合研究部小幡 行雄部長, 田中 正俊次長の御激励に深く感謝します。

文 献

- 1) M. Azumi, G. Kurita, T. Matsuura, T. Takeda, Y. Tanaka, and T. Tsunematsu, "A Fluid Model Numerical Code System for Tokamak Fusion Research", Proc. 4th International Symposium on Computing Methods in Applied Sciences and Engineering, Versailles, 1979.
- 2) K.V. Roberts, Computer Phys. Commun. 7 (1974) 237.
- 3) 岡本 正雄, 竹田 辰興, 田中 正俊, 浅井 清, 中野 鴻, 川上 一郎, "OLYMPUS システムの導入とプリ・プロセッサの開発", JAERI-M 7228 (1977).
- 4) 滝塚 知典, 岡本 正雄, "COMMON文を活用するためのプリ・プロセッサの作成およびその使用法", 未公開資料 (1977).

付録A データ・セット一覧

プリプロセッサ・システム EOS に関連したデータ・セットは、HARMONIAコードの管理下に TRITON システムのコードとして登録されている。第A.1表にこれらデータ・セットの一覧表、第A.2表に GEMファイル EOS 200, PREP の MODULE - MODID 対応表を示す。

第A.1表 EOS プリプロセッサ・システム関連データ・セット・リスト

	データ・セット名	内 容
1	EOS 200, PREP	・EOS, 逆EOSプリプロセッサのEOSソース・プログラム
2	EOS 200, FORT	・EOS, 逆EOSプリプロセッサのフォートラン・ソース・プログラム
3	EOS 200, LOAD	・EOS, 逆EOSプリプロセッサのロード・モジュール
4	EOS 200, CNTL	・JCLプロシジャ
5	EOS 200, CLIST	・TSSコマンド
6	EOS 200, DATA	・EOS, 逆EOSプリプロセッサのためのテスト入力
7	EOSUTY, FORT	・ユーティリティ・ルーチンのEOSソース・プログラム
8	EOSUTY, LOAD	・ユーティリティ・ルーチンのロード・モジュール

第 A.2 表 ファイル EOS 200. PREP の MODULE - MODID 対照表 E1: EOS プリプロセッサ,
E2: 逆 EOS プリプロセッサ, ESHARE: 共通サブルーチン *印 メイン・プログラム

NO.	MODULE NAME	MODID	NO.	MODULE NAME	MODID
1	##SESH	ESHARE	61	INSERT	ESHARE
2	##SE1	E1	62	INZELM	E2
3	##SE2	E2	63	ITOC	ESHARE
4	#CESH	ESHARE	64	IVSET	ESHARE
5	#CE1	E1	65	LABEL	E2
6	#CE2	E2	66	LBNUM	E2
7	ACCNT1	E1	67	LOOKUP	ESHARE
8	ACCNT2	E2	68	MEMBER	ESHARE
9	ACOMM	E1	69	OTHER	E1
10	ADIME	E1	70	PACKA4	ESHARE
11	AEQUI	E1	71	PRINTL	ESHARE
12	AJLINE	ESHARE	72	PRMVAL	ESHARE
13	ALPHA	ESHARE	73	PROG	E1
14	ATYPE	E1	74	PRSTR	ESHARE
15	BACKCH	ESHARE	75	PUTST	ESHARE
16	BCOMM	E2	76	PUTTX	E1
17	BDIME	E2	77	PUTWK	E2
18	BEQUI	E2	78	READDT	ESHARE
19	BLOCKD	ESHARE	79	READL	ESHARE
20	BTYPE	E2	80	REGDL	E2
21	CCOMM	E1	81	REGDM	E2
22	CDIME	E1	82	REGLB	E2
23	CEQUI	E1	83	REGVR	E2
24	CMVAR	E2	84	REPNAM	E1
25	CNUM	ESHARE	85	RESERV	E1
26	COMGEN	E2	86	RNMAP	E1
27	COMPAR	E2	87	RNTAB	E1
28	COMTAB	E2	88	SBMAIN	E1
29	COMWRT	E2	89	SEARCR	E1
30	COPY	ESHARE	90	SEARCS	E1
31	CTOI	ESHARE	91	SEARCT	E1
32	CTYPE	E1	92	SETP	ESHARE
33	DARRAY	ESHARE	93	SEVAL	E1
34	DEVELP	ESHARE	94	SINIT	ESHARE
35	DPBUF	ESHARE	95	SIZE	E1
36	DPTAB	ESHARE	96	SKIPEL	ESHARE
37	EBINIT	E2	97	SKIPS	ESHARE
38	ELEMNT	E2	98	SKIPST	ESHARE
* 39	EOSMAIN	E1	99	SORT	E2
40	ERROR	ESHARE	100	SREST	E1
* 41	ERVMAIN	E2	101	STARTL	E2
42	EXECUT	E2	102	STCALL	E2
43	EXPRES	E1	103	STORSB	E1
44	FTOW1	E2	104	STRWK	ESHARE
45	FTOTP	E2	105	SUBSCR	E2
46	GETCH	ESHARE	106	SZMAP	E1
47	GETCMD	ESHARE	107	TBCOPY	ESHARE
48	GETDST	ESHARE	108	TEXT	E1
49	GETEXP	E1	109	TPTOE	E2
50	GETKEY	ESHARE	110	TXGEN	E2
51	GETNUM	ESHARE	111	TXMAP	E1
52	GETST	ESHARE	112	UNIF	E1
53	GETSYL	E1	113	VALUE	E1
54	GETSYM	ESHARE	114	WCOMDK	E2
55	GETTX	E1	115	WCOMON	E2
56	HEAD	ESHARE	116	WTYPE	E2
57	INFBUF	ESHARE	117	WVAR	E2
58	INFST	E2	118	W1TOW2	E2
59	INIT1	E1	119	W2TOTP	E2
60	INIT2	E2			

付録B ユーティリティ一覧

EOSプリプロセッサ・システムの利用を容易にするためのユーティリティ・プログラム (EOSUTY. PREP, EOSUTY. LOAD) が用意されている。

また, JCLプロシジャによってEOSプリプロセッサ, 逆EOSプリプロセッサを用いる時, GEM, PS, POファイル間の変換を行うためのプロシジャ, GEMPO, GEMPS, POPS, PSEPO, PSFPO が用意されている (第B.1表)。これらのプロシジャの内, POGEMのみは, 出力データセット用にあらかじめGEMファイルを確認しておかなければならない。これは, センター・プロシジャ GEMINIT を使えばよい。

第B.1表 ユーティリティ・ルーチン利用のためのJCLプロシジャ

プロシジャ	機能	パラメータ (デフォルト値)
GEMPO	GEM→PO	DSN=, ODSN=, UNIT=TSSWK, SYSOUT='*', OLD=NEW, CATLG=CATLG, SPC='30, 10, 5'
POPS	PO→PS	DSN=, ODSN=, UNIT=TSSWK, SYSOUT='*', OLD=NEW, CATLG=CATLG, SPC='30, 10'
GEMPS	GEM→PS	DSN=, ODSN=, UNIT=TSSWK, SYSOUT='*', OLD=NEW, CATLG=CATLG, SPC='30, 10'
PSEPO	EOS-PS→PO	DSN=, ODSN=, UNIT=TSSWK, SYSOUT='*', OLD=NEW, CATLG=CATLG, SPC='30, 10, 5'
PSFPO	フォートラン-PS→PO	DSN=, ODSN=, UNIT=TSSWK, SYSOUT='*', OLD=NEW, CATLG=CATLG, SPC='30, 10, 5'
POGEM	PO→GEM	DSN=, ODSN=, SYSOUT='*'

付録C ロード・モジュールの作成

EOSプリプロセッサ、逆EOSプリプロセッサは、テキスト文の格納等のためいくつかのオンコア・バッファ（コモン領域）を使用している。必要とされる領域の大きさは処理対象プログラムによって異なるので既存のEOS、逆EOSロード・モジュールによって処理できない時は領域を広げる必要がある。このような場合の便宜のため、EOS、逆EOSプリプロセッサは共にEOSソース・プログラムの形で保管されている。現在原研で利用している計算機システム（FACOM M200計算機システム）と異なる計算機システム上で、EOSを利用しようとする場合は、EOSおよび逆EOSプリプロセッサを一度EOSプリプロセッサによってEOSソース・プログラムからフォートラン・ソース・プログラムに変換して移動すればよい。

EOSおよび逆EOSプリプロセッサ専用部及び共通部のサイズ・ブロックは、それぞれ##SE1, ##SE2, および##SESHの名称を持っている（第C.1図）。EOSプリプロセッサのサイズ・ブロックで定義されているサイズ変数は第C.1表に、逆EOSプリプロセッサのサイズ・ブロックで定義されているサイズ変数は第C.2表に示すとおりである。また、領域不足等が生じた時に出力されるエラー・メッセージおよびそれに関連したサイズ変数の名称を、EOS、逆EOSについてそれぞれ、第C.3表、第C.4表に示す。これらの表にあるエラー・メッセージが出力された場合には、対応するサイズ変数の値を大きくすればよい。この作業のためのJCLが、EOSおよび逆EOSについて用意されており、J2636, EOS200, CNTL (EOSUPDT)およびJ2636, EOS200, CNTL (ERVUPDT)である。サイズ変数の変更はJ2636, EOS200, DATA (SIZE)にカード形式の入力データを追加すればよい。EOSUPDTおよびERVUPDTの実行により、J2636, EOS200, LOADの中に、それぞれ、E1TESTおよびE2TESTというメンバが登録される。これらの動作を確認の後、古いE1, E2を消去してE1TEST, E2TESTの名称を、E1, E2に変更すればよい。EOSUPDT, ERVUPDTのJCLリストを、それぞれ、第C.2図、第C.3図に示す。

第C.1表 EOSプリプロセッサのサイズ変数の意味、デフォルト値及び拡張性

サイズ変数名	意 味	デフォルト	拡張性
MAXLM	EOSソースリスティング行数	55	×
STMAX	フォートラン文バッファ長	1600	×
EBMAX	編集用バッファ長	1600	×
MXW	ワークエリアサイズ	40	×
MXW2	"	200	×
SZMAX	サイズ変数テーブル最大値	200	○
SZPMAX	サイズ変数プール最大値	1000	○
TXMAX	COMDECK名テーブル最大値	200	○
TXPMAX2	COMDECK別名テーブル最大値	500	○
TXPMAX	COMDECKテキストプール最大値	20000	○
REMAX	変数名変更テーブル最大値	30	○
REPMAX	新変数名用バッファ最大値	500	○

第C.2表 逆EOSプリプロセッサのサイズ変数の意味、デフォルト値及び拡張性

サイズ変数名	意味	デフォルト	拡張
MAXP	テキスト格納用バッファ長	5 0 0 0	○
MXL	1サブプログラム内のCOMMONラベル数	1 0 0	○
MXV	" コモン変数の数	5 0 0	○
MXD	" コモン配列用次元バッファ	1 0 0 0	○
MX1	全プログラム中のCOMMONラベル数	2 0 0	○
MX2	" コモンラベル別名数	3 0 0	○
MX3	" コモン変数の数	1 0 0 0	○
MX4	" " の次元バッファ	2 0 0 0	○
MSPEC	指定ラベル数最大値	3 0	○
MAXDL	COMMON文以外の宣言文に含まれるCOMMON変数の数	1 0 0	○
MAXCM	1COMMON文中のラベル数最大値	1 0	○
STMAX	フォートラン文バッファ長	1 6 0 0	×
EBMAX	編集用バッファ長	1 6 0 0	×
MXW	ワークエリアサイズ	4 0	×
MXW 2	"	2 0 0	×
MAXLN	1頁当りのリスティング行数	5 5	×

```

-----1-----2-----3-----4-----5-----6-----7-R-----8
NO.1  MODULE NAME  ##SESH  BLOCKS  2
      LEVEL      1      DATE      82.03.25  TIME      15.45.22

      ***** APPOINTED MODULE INFORMATION *****
      LEVEL      1      DATE      82.03.25  TIME      15.45.22

C-----SIZE BLOCK START ----- 00000100
*SIZE
C === STATMT === 00000200
      STMAX=1600 00000300
C === EDITBF === 00000400
      EBMAX=1600 00000500
C === WORK === 00000600
      MXW = 40 00000700
      MXW2 = 200 00000800
C === PHEAD === 00000900
      MAXLN = 55 00001000
C-----SIZE BLOCK END ----- 00001100

```

(a) ESHARE

第C.1図 プリプロセッサのサイズ・ブロック

```

-----1-----2-----3-----4-----5-----6-----7-R-----8
NO.2  MODULE NAME  ##SE1  BLOCKS  2
      LEVEL        1      DATE    82.03.25  TIME    15.45.22

      ***** APPOINTED MODULE INFORMATION *****
      LEVEL        1      DATE    82.03.25  TIME    15.45.22

C ----- SIZE SECTION ----- 00000100
*SIZE
C=== PHEAD === 00000200
      MAXLN = 55 00000300
C=== STATMT === 00000400
      STMAX = 1600 00000500
C=== EDITBF === 00000600
      EBMAX = 1600 00000700
C=== WORK === 00000800
      MXW = 40 00000900
      MXW2 = 200 00001000
C=== SIZETB === 00001100
      SZMAX = 200 00001200
      SZPMAX = 1000 00001300
C=== TEXTTB === 00001400
      TXMAX = 200 00001500
      TXMAX2 = 500 00001600
      TXPMAX = 20000 00001700
C=== RENMTB === 00001800
      REMAX = 30 00001900
      REPMAX = 500 00002000
C ----- SIZE SECTION END ----- 00002100

```

(b) E1

```

-----1-----2-----3-----4-----5-----6-----7-R-----8
NO.3  MODULE NAME  ##SE2  BLOCKS  2
      LEVEL        1      DATE    82.03.25  TIME    15.45.22

      ***** APPOINTED MODULE INFORMATION *****
      LEVEL        1      DATE    82.03.25  TIME    15.45.22

*SIZE
C=== TEXTBF === 00000100
      MAXP=5000 00000200
C=== TABLE === 00000300
      MXL=100 00000400
      MXV=500 00000500
      MXD=1000 00000600
C=== COM === 00000700
      MX1=200 00000800
      MX2=300 00000900
      MX3=1000 00001000
      MX4=2000 00001100
C=== LIDENT === 00001200
      MSPEC=30 00001300
C=== EINF === 00001400
      MAXDL=100 00001500
C=== CMBUF === 00001600
      MAXCM=10 00001700
C=== STATMT === 00001800
      STMAX=1600 00001900
C=== EDITBF === 00002000
      EBMAX=1600 00002100
C=== WORK === 00002200
      MXW = 40 00002300
      MXW2 = 200 00002400
C=== PHEAD === 00002500
      MAXLN = 55 00002600

```

(c) E2

☒ C.1 (続き)

第C. 3表 EOS プリプロセッサのサイズに関連した
エラーメッセージと対策 (変更すべきサイズ変数)

No.	メッセージ	サイズ変数
1	PROG -- RENAME TABLE OVER FLOW	REMAX
2	RESERV -- SZNAM OVER FLOW	SZMAX
3	RESERV -- SZPOOL OVER FLOW	SZPMAX
4	SEARCR -- RENAME INDEX TABLE PANK	REMAX
5	SEARCR -- RENAME POOL PANK	REPMAX
6	SEARCS -- SIZE TABLE OVER FLOW	SZMAX
7	SEARCS -- SZPOOL PANK	SZPMAX
8	SEVAL -- SZNAM/SZPOOL PANK	SZMAX, SZPMAX
9	STORSB -- TXPOOL OVER FLOW	TXPMAX
10	TEXT -- TXNAM OVER FLOW	TXMAX, TXMAX2

第C. 4表 逆EOS プリプロセッサのサイズに関する
エラーメッセージと対策 (変更すべきサイズ変数)

No.	メッセージ	サイズ変数
1	BCOMM -- CM OVER FLOW	MAXCM
2	BCOMM -- LBL OVER FLOW	MXL
3	BDIME -- VAR OVER FLOW	MXV
4	BEDUI -- VAR OVER FLOW	MXV
5	BTYPE -- LBL OVER FLOW	MXL
6	COMWRT -- VRF EXCEEDED MX3	MX3
7	COMWRT -- SZF EXCEEDED MX4	MX4
8	PUTWK -- POOL NO.1 OVER FLOW	MAXP
9	PUTWK -- POOL NO.2 OVER FLOW	MAXP
10	PUTWK -- POOL NO.1 NOT ENOUGH SPACE	MAXP
11	PUTWK -- POOL NO.2 NOT ENOUGH SPACE	MAXP
12	PEGDL -- DELVAR OVER FLOW	MAXDL
13	REGDM -- DF EXCEEDED MXD	MXD
14	REGLB -- LBL OVER FLOW	MXL
15	REGVR -- VAR OVER FLOW	MXV

```

//JCLG JOB                                00000010
//      EXEC JCLG                          00000020
//SYSIN DD DATA,DLM='++'                 00000030
// JUSER 72502636,TA.TAKEDA,0021.100      00000040
// W.0,T.2,C.2,I.4,P.0,OPN                00000050
// OPTP PASSWORD=TRITON,MSGLEVEL=(1,1),NOTIFY=J2855 00000060
//*****                                00000070
//G1 EXEC GEM,DSN='J2636.EOS200',Q='.PREP' 00000080
// - GROUP A,MODID=ESHARE                  00000090
// - PUT A,OUT=OUT1                        00000100
// - END                                    00000110
// EXPAND DISKPSN,DDN=OUT1,DSN=EOS1,      00000120
//      DCB='RECFM=FB,LRECL=80,BLKSIZE=3120,DSORG=PS' 00000121
//*****                                00000130
//EOSG01 EXEC PGM=E1,PARM='SLIST=0,LINECM=0' 00000140
//STEPLIB DD DSN=J2636.EOS200.LOAD,DISP=SHR 00000141
// EXPAND DISKTO,DDN=FT05F001,DSN='J2636.EOS200',Q='.DATA(SIZE)'' 00000150
// EXPAND PRTFD                             00000160
// EXPAND DISKPSO,DDN=FT09F001,DSN=EOS1   00000170
// EXPAND DISKPSN,DDN=FT10F001,DSN=FORT,  00000180
//      DCB='RECFM=FB,LRECL=80,BLKSIZE=3120,DSORG=PS' 00000181
//FT11F001 DD DUMMY                         00000190
//FT30F001 DD DUMMY                         00000200
//*****                                00000210
//G2 EXEC GEM,DSN='J2636.EOS200',Q='.PREP' 00000220
// - GROUP A,MODID=E1                      00000230
// - PUT A,OUT=OUT2                        00000240
// - END                                    00000250
// EXPAND DISKPSN,DDN=OUT2,DSN=EOS2,      00000260
//      DCB='RECFM=FB,LRECL=80,BLKSIZE=3120,DSORG=PS' 00000261
//*****                                00000270
//EOSG02 EXEC PGM=E1,PARM='SLIST=0,LINECM=0' 00000271
//STEPLIB DD DSN=J2636.EOS200.LOAD,DISP=SHR 00000272
// EXPAND DISKTO,DDN=FT05F001,DSN='J2636.EOS200',Q='.DATA(SIZE)'' 00000290
// EXPAND PRTFD                             00000300
// EXPAND DISKPSO,DDN=FT09F001,DSN=EOS2   00000310
//FT10F001 DD DSN=&&FORT,DISP=(MOD,PASS,DELETE) 00000320
//FT11F001 DD DUMMY                         00000330
//FT30F001 DD DUMMY                         00000340
//*****                                00000350
//COMP EXEC FORTGE,SO='&&FORT',Q=,        00000360
//      A='NOSOURCE,NOISN,NOMAP'          00000361
//*****                                00000370
//ASM EXEC ASM,SO='J2636.EOS200',DISP='(MOD,PASS)'' 00000380
//*****                                00000390
//LINK EXEC LKEDUP,LM='J2636.EOS200',B=NOLIST,CNTL=NO,CUNIT=TSSWK 00000400
// ENTRY MAIN                               00000410
// NAME E1TEST(R)                           00000420
//++                                         00000430
//                                           00000440

```

第C.2図 EOS プリプロセッサ・ロード・モジュール更新用
JCLリスト: J2636.EOS200.CNTL (EOSUPDT)

```

//JCLG JOB                                00000010
// EXEC JCLG                                00000020
//SYSIN DD DATA,DLM='++'                  00000030
// JUSER 72502636,TA.TAKEDA,0021.100      00000040
W.O,T.2,C.2,I.4,P.O,OPN                    00000050
OPTP PASSWD=TRITON,MSGLEVEL=(1,1),NOTIFY=J2636 00000060
//*****                                00000070
//G1 EXEC GEM,DSN='J2636.EOS200',Q=''.PREP' 00000080
- GROUP A,MODID=ESHARE                      00000090
- PUT A,OUT=OUT1                            00000100
- END                                         00000110
// EXPAND DISKPSN,DDN=OUT1,DSN=EOS1,        00000120
// DCB='RECFM=FB,LRECL=80,BLKSIZE=3120,DSORG=PS' 00000121
//*****                                00000130
//EOSG01 EXEC PGM=E1,PARM='SLIST=0,LINECM=0' 00000140
//STEPLIB DD DSN=J2636.EOS200.LOAD,DISP=SHR 00000141
// EXPAND DISKTO,DDN=FT05F001,DSN='J2636.EOS200',Q=''.DATA(SIZE)' 00000150
// EXPAND PRTFD                               00000160
// EXPAND DISKPSO,DDN=FT09F001,DSN=EOS1     00000170
// EXPAND DISKPSN,DDN=FT10F001,DSN=FORT,    00000180
// DCB='RECFM=FB,LRECL=80,BLKSIZE=3120,DSORG=PS' 00000181
//FT11F001 DD DUMMY                          00000190
//FT30F001 DD DUMMY                          00000200
//*****                                00000210
//G2 EXEC GEM,DSN='J2636.EOS200',Q=''.PREP' 00000220
- GROUP A,MODID=E2                          00000230
- PUT A,OUT=OUT2                            00000240
- END                                         00000250
// EXPAND DISKPSN,DDN=OUT2,DSN=EOS2,        00000260
// DCB='RECFM=FB,LRECL=80,BLKSIZE=3120,DSORG=PS' 00000261
//*****                                00000270
//EDSG02 EXEC PGM=E1,PARM='SLIST=0,LINECM=0' 00000271
//STEPLIB DD DSN=J2636.EOS200.LOAD,DISP=SHR 00000272
// EXPAND DISKTO,DDN=FT05F001,DSN='J2636.EOS200',Q=''.DATA(SIZE)' 00000290
// EXPAND PRTFD                               00000300
// EXPAND DISKPSO,DDN=FT09F001,DSN=EOS2     00000310
//FT10F001 DD DSN=*&FORT,DISP=(MOD,PASS,DELETE) 00000320
//FT11F001 DD DUMMY                          00000330
//FT30F001 DD DUMMY                          00000340
//*****                                00000350
//COMP EXEC FORTGE,SO='*&FORT',Q=,         00000360
// A='NOSOURCE,NOISN,NOMAP'                 00000361
//*****                                00000370
//ASM EXEC ASM,SO='J2636.EOS200',DISP='(MOD,PASS)' 00000380
//*****                                00000390
//LINK EXEC LKEDUP,LM='J2636.EOS200',B=NOLIST,CNTL=NO,CUNIT=TSSWK 00000400
ENTRY MAIN                                  00000410
NAME E2TEST(R)                              00000420
++                                           00000430
//                                           00000440

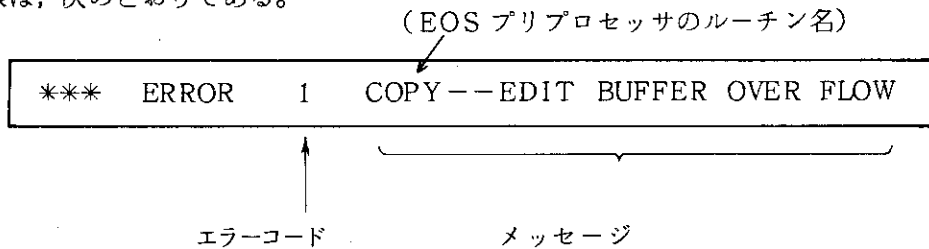
```

第C.3図 逆EOS プリプロセッサ・ロード・モジュール更新用
JCLリスト: J2636.EOS200.CNTL (ERVUPDT)

付録D EOSプリプロセッサのエラー・メッセージ

EOSソース・プログラム上のEOS文法エラー，EOS自身の内部バッファ領域不足の際に出力されるエラー・メッセージをまとめておく（第D.1表）。

エラー・メッセージの出力機番はFT06F001である。SLIST=1が指定されていれば，エラー・メッセージはEOSソース・リストのエラーの起った個所に出力される。SLIST=0ならば，エラー・メッセージの次にその時点で処理されている文が出力される。エラー・メッセージの様子は，次のとおりである。



ここで，エラー・コード1は，軽度のエラーを示し，この場合処理は続行される。エラー・コード2は重度のエラーを示し，この場合処理は停止する。

第D.1表 EOSプリプロセッサ・エラー・メッセージ一覧表

エラー・メッセージの欄の1，2，3は，それぞれエラーの説明，対策，関連コモン・ブロック名等，を示す。

CODE	ERROR MESSAGE
1	ACOMM-- COMMON LABEL SYNTAX ERROR 1. COMDECKのCOMMON文のラベル名の文法エラー
1	ACOMM-- VARIABLE NAME ERROR 1. COMDECKのCOMMON文の変数名に誤りがある。
1	ACOMM-- SIZE EQUATION ERROR 1. COMDECKのCOMMON文中のサイズ式の誤り
1	ACOMM-- DECLARATOR SUBSCRIPT ERROR 1. COMDECKのCOMMON文の配列宣言子の文法エラー
1	ACOMM-- STATEMENT END SYNTAX ERROR 1. COMDECKのCOMMON文が正しく終わっていない。

CODE	ERROR MESSAGE
1	ADIME -- VARIABLE NAME ERROR
	1. COMDECK の DIMENSION 文の変数名の誤り
1	ADIME -- SIZE EQUATION ERROR
	1. COMDECK の DIMENSION 文のサイズ式の誤り
1	ADIME -- DECLARATOR SUBSCRIPT ERROR
	1. COMDECK の DIMENSION 文の配列宣言子の文法エラー
1	ADIME -- STATEMENT END SYNTAX ERROR
	1. COMDECK の DIMENSION 文が正しく終わっていない
1	AEQUI -- EQUIVALENCE GROUP LEFT PAREN ERROR
	1. COMDECK の EQUIVALENCE 文の左カッコがない
1	AEQUI -- VARIABLE NAME ERROR
	1. COMDECK の EQUIVALENCE 文の変数名エラー
1	AEQUI -- SIZE EQUATION ERROR
	1. COMDECK の EQUIVALENCE 文のサイズ式の誤り
1	AEQUI -- DECLARATOR SUBSCRIPT ERROR
	1. COMDECK の EQUIVALENCE 文の配列宣言子の文法エラー
1	AEQUI -- EQUIVALENCE GROUP RIGHT PAREN ERROR
	1. COMDECK の EQUIVALENCE 文の右カッコがない
1	AEQUI -- STATEMENT END SYNTAX ERROR
	1. COMDECK の EQUIVALENCE 文が正しく終わっていない
1	ATYPE -- VARIABLE NAME ERROR
	1. COMDECK の型宣言文の変数名の誤り
1	ATYPE -- VARIABLE LENGTH ERROR
	1. COMDECK の型宣言文の長さ指定に誤りがある

CODE	ERROR MESSAGE
1	ATYPE -- SIZE EQUATION ERROR 1. COMDECK の型宣言文のサイズ式の誤り
1	ATYPE -- DECLARATOR SUBSCRIPT ERROR 1. COMDECK の型宣言文の配列宣言子の文法エラー
1	ATYPE -- INITIAL VALUE PART ERROR 1. COMDECK の型宣言文の初期値設定部に誤りがある
1	ATYPE -- STATEMENT END SYNTAX ERROR 1. COMDECK の型宣言文が正しく終わっていない
1	CCOMM -- LABEL SYNTAX ERROR 1. COMDECK のCOMMON 文のラベル名の誤り
1	CCOMM -- NAME REPLACE ERROR 1. COMDECK のCOMMON 文の変数名の変更時にエラーが生じた
1	CCOMM -- STATEMENT END SYNTAX ERROR 1. COMDECK のCOMMON 文が正しく終わっていない
1	CDIME -- NAME REPLACE ERROR 1. COMDECK のDIMENSION 文の変数名の変更時にエラーが生じた
1	CDIME -- STATEMENT END SYNTAX ERROR 1. COMDECK のDIMENSION 文が正しく終わっていない
1	CEQUI -- EQUIVALENCE GROUP LEFT PAREN ERROR 1. COMDECK のEQUIVALENCE 文の左カッコがない
1	CEQUI -- NAME REPLACE ERROR 1. COMDECK のEQUIVALENCE 文で、変数名の変更時にエラーが生じた
1	CEQUI -- STATEMENT END SYNTAX ERROR 1. COMDECK のEQUIVALENCE 文が正しく終わっていない

CODE	ERROR MESSAGE
2	COPY -- EDIT BUFFER OVER FLOW
	1. 文解析 BUFFER (SBUF) から編集用 BUFFER (EBUF) へのコピー時に EBUF の最大値を越えた。 2. COMDECK テキストが編集後に、文の長さの制限を越えないように修正する。 3. /STATMT/
1	CTYPE -- NAME REPLACE ERROR
	1. COMDECK の型宣言文の変数名の変更時にエラーが生じた。 2. 変数名変更リストに誤りがないか調べる
1	CTYPE -- INITIAL VALUE PART ERROR
	1. COMDECK の型宣言文の初期値設定部のエラー
1	CTYPE -- STATEMENT END SYNTAX ERROR
	1. COMDECK の型宣言文が正しく終わっていない。
1	EXPRES -- INVALID EXPRESSION
	1. サイズ式の文法が誤っている。
2	EXPRES -- PROGRAM ERROR
	1. サイズ式の解析の誤り 2. プログラムエラー。担当者に連絡する。
2	GETDST -- SBUF OVER FLOW
	1. サイズブロックの文の長さが長過ぎる。 2. *COMDECK, *DECK 等の KEYWORD を書忘れていないか調べる。
2	GETDST -- MORE THAN 20 LINES
	1. サイズブロックの継続行が 20 行を越えた。
1	GETST -- START LINE HAS A CONT. MARK
	1. 文の開始行の第 6 カラムが空白でも 0 でもない 2. 文の継続の途中でのコメント行は EOS では許されないので注意すること。
1	GETST -- BLANK OR ONLY LABEL LINE
	1. 空白行又は、ラベル名のみが書かれた行である。

CODE	ERROR MESSAGE
2	GETST -- MORE THAN 20 LINES
	1. 継続行が19行を越えた。
2	GETST -- SBUF OVER FLOW
	1. 文の長さが1600文字を越えている。
1	GETSYL -- SYMBOL NOT FOUND
	1. サイズ式中のサイズ変数が定義されていない。
1	NSKIP -- INVALID N
	1. NSKIP の引数の値が負か、大き過ぎる。 2. プログラムエラー。担当者に連絡する。
1	OTHER -- ¥VARIABLE ERROR
	1. COMDECK 中の、E 宣言文以外の文に表れる ' ¥ ' 変数の使い方の誤り
1	PRMVAL -- SYNTAX ERROR
	1. プログラム実行時パラメータの書き方の誤り 2. TSS の時はCALL文の後の引用符の中、BATCH処理の時はEXEC文の ' PARM ' パラメータの中を修正する
1	PROG -- *DECK COMMAND ERROR
	1. *DECK 文の文法エラー 2. *DECK文を正しく書きなおす。このエラーが起ると、次の*コマンドまで空読みされる。
1	PROG -- *DECK NAME ERROR
	1. *DECKの後に、DECK名が正しく書かれていない
1	PROG -- TEXT NAME SYNTAX ERROR
	1. *CALL文のCOMDECK名に文法エラーがある
1	PROG -- TEXT NAME NOT FOUND
	1. COMDECK名が登録されていない

CODE	ERROR MESSAGE
2	PROG -- RENAME TABLE OVER FLOW 1. *CALL文の後の変数名変更リストが多過ぎる。 2. 変更できる変数名は、1つの文に対して30個までである
1	PUTST -- STATEMENT HAS MORE THAN 20 LINES 1. 出力時に文が20行を越えた 2. 編集を受けた後に、テキストが20行を越えていないか調べる。
1	READL -- NO EFFECTIVE CHARACTER 1. 1行の中に有効な文字がない 2. ' ; ' (行コメントセパレータ) に注意して修正する
1	REFFERT -- TEXT NAME NOT FOUND 1. *CALL文のCOMDECK名が登録されていない
1	REPNAM -- VARIABLE NAME ERROR 1. 変数名を変更す可き文の旧変数名の誤り
1	REPNAM -- VARIABLE LENGTH ERROR 1. 変数名を変更す可き文の変数の長さ指定の誤り
1	REPNAM -- DECLARATOR SUBSCRIPT ERROR 1. 変数名を変更す可き文の配列宣言子の文法エラー
2	RESERV -- SZNAM OVER FLOW 1. サイズブロック中で定義された変数名の数が多過ぎる 2. サイズ変数は100個まで使用できる
2	RESERV -- SZPOOL OVER FLOW 1. サイズ式を収めるBUFFERがOVER FLOWした 2. プログラムエラー。担当者に連絡する 3. SZPOOL (SZPMAX) で SZPMAX = 500 /SIZETB/
1	RNMAP -- INVALID RENAME POINTER 1. 変数名変更リストのポインターの値の誤り 2. プログラムエラー。担当者に連絡する

CODE	ERROR MESSAGE
1	RNTAB -- INVALID SYMBOL NAME = OLD
	1. *CALL文の変数名変更リストの旧変数名がおかしい
1	RNTAB --RENAME LIST SYNTAX ERROR
	1. 変数名変更リストの文法エラー
1	RNTAB --INVALID SYMBOL NAME =NEW
	1. *CALL文の変更リストの中の新変数名の誤り
1	SBMAIN-- *COMMAND LACKED
	1. *COMDECK 又は *DECK が欠落している 2. *COMDECK 文又は *DECK 文に文法エラーがないか調べる
1	SBMAIN -- SIZE INPUT ERROR
	1. サイズブロックの文法エラー 2. サイズブロックにエラーがあると、プログラムは停止し、それ以後の処理は行われないので注意すること
1	SERCR -- DUPLICATED SYMBOL
	1. *CALL文の変更リストの中に同一の旧変数名がある
2	SEARCHR -- RENAMA INDEX TABLE PANK
	1. *CALL文の変更リストの旧変数名の数が30個を越えた。 2. 変更リストが30を越える時はEOSの修正が必要である 担当者に連絡する 3. /RENMTB/ REMAX =30
2	SEARCHR -- RENAME POOL PANK
	1. *CALL文の変更リストの新変数名がバッファに入りきらない 2. EOSの修正が必要, 担当者に連絡する 3. /RENMTB/ REPMAX = 100
2	SEARCHR -- SIZE TABLE OVER FLOW
	1. サイズ変数が100個を越えている 2. EOSの修正が必要, 担当者に連絡する 3. /SIZETB/ SZMAX = 100

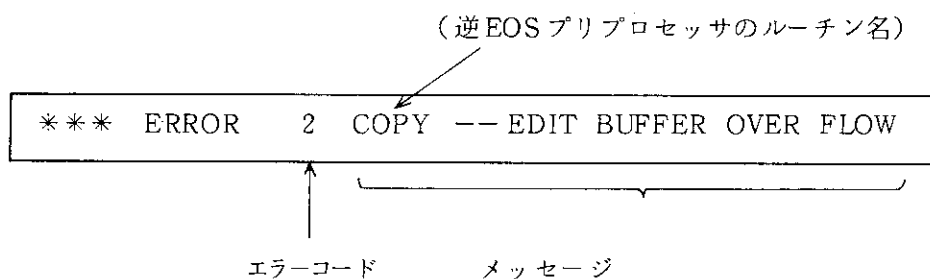
CODE	ERROR MESSAGE
2	SEARCS -- SZPOOL PANK
	1. サイズ変数の値の為のバッファが足りない 2. EOS の修正が必要, 担当者に連絡する 3. /SIZE TB / SZPMAX = 500
1	SEARCS -- ERROR IN LOOKUP
	1. "LOOKUP" のエラー 2. プログラムエラー
1	SEARCT -- DUPLICATED SYMBOL
	1. *COMDECK に COMDECK 名の同一のものが既にある
1	SEARCT -- ERROR IN LOOKUP
	1. "LOOKUP" のエラー 2. プログラムエラー
1	SEVAL -- SIZE SYNTAX ERROR
	1. サイズブロックに文法エラーがある
2	SEVAL -- SZNAM/SZPOOL PANK
	1. サイズ変数の為のバッファの OVER FLOW 2. EOS の修正が必要, 担当者に連絡する
1	SIZE -- ONLY SIZE SECTION
	1. ソースプログラム中にサイズブロックのみしか無い
1	SREST -- SYNTAX ERROR
	1. サイズブロックの文法エラー
2	STORSB -- TXPOOL OVER FLOW
	1. COMDECK テキストの為のバッファが OVER FLOW した 2. *DECK 文が欠けていないか調べる。正常なら EOS の修正が必要
1	SZMAP -- INVALID SIZE POINTER
	1. サイズテーブルダンプ時のポインターの値がおかしい 2. プログラムエラー

CODE	ERROR MESSAGE
1	TEXT -- *COMDECK LINE LACKED
	1. COMDECK ブロックの先頭が *COMDECK でない
1	TEXT -- *COMDECK LINE SYNTAX ERROR
	1. *COMDECK の行の文法エラー
1	TEXT -- DUPLICATED COMDECK NAME
	1. *COMDECK の中に同一の名前を持ったものがある
2	TEXT -- TXNAM OVER FLOW
	1. COMDECK 名の数がバッファを越えた 2. COMDECK 名は 200 個まで定義できる
1	TXMAP -- CURRENT ADDRESS IS INCORRECT
	1. テキストテーブルダンプ時のエラー 2. プログラムエラー
1	UNIF -- SYMBOL NOT FOUND
	1. COMDECK テキスト中のサイズ式で単項式の中に未定義サイズ変数があら われている

付録E 逆EOSプリプロセッサのエラー・メッセージ

逆EOSプリプロセッサは処理中に文法エラーを発見したり、内部バッファ領域が不足した場合には第E.1表に示すエラー・メッセージを出力する。

エラー・メッセージの出力機番はFT 06F 001であり、取出し処理情報の途中に出力される。エラー・メッセージの仕様は次のとおりである。



ここで、エラー・コード1は、軽度のエラーを示し、この場合処理は続行される。エラー・コード2は重度のエラーを示し、この場合処理は停止する。

第E.1表 逆EOSプリプロセッサ・エラー・メッセージ一覧表

エラー・メッセージの欄の1, 2, 3は、それぞれエラーの説明、対策、関連するコモン・ラベル名等、を示す

CODE	ERROR MESSAGE
1	BCOMM -- CM OVER FLOW
	1. 1つのCOMMON文で、ラベル名を10ヶ以上含んでいるものがある 2. EOSREVの修正が必要
1	BCOMM -- COMMON LABEL SYNTAX ERROR
	1. COMMON文のラベル名の文法エラー
1	BCOMM -- VARIABLE NAME ERROR
	1. COMMON文の変数名のエラー
1	BCOMM -- SUBSCRIPT SYNTAX ERROR
	1. COMMON文の配列宣言子に誤りがある
1	BCOMM -- STATEMENT END SYNTAX ERROR
	1. COMMON文が正しく終わっていない

CODE	ERROR MESSAGE
1	BCOMM -- LBL OVER FLOW
	1. 現在のルーチンが, 100 個以上の COMMON 文を含んでいる 2. EOSREV の修正が必要
1	BDIME -- VARIABLE NAME ERROR
	1. DIMENSION 文の変数名のエラー
1	BDIME -- VAR OVER FLOW
	1. 現在のルーチンが, 500 個以上の COMMON 変数を含んでいる 2. EOSREV の修正が必要
1	BDIME -- SUBSCRIPT NOT FOUND
	1. DIMENSION 文の配列宣言がない
1	BDIME -- ERROR IN REGDM
	1. DIMENSION 中の COMMON 変数の登録エラー
1	BDIME -- ERROR IN SUBSCR
	1. DIMENSION 文中の配列宣言子の誤り
1	BDIME -- STATEMENT END SYNTAX ERROR
	1. DIMENSION 文が正しく終わっていない
1	BEQUI -- EQUIVALENCE GROUP LEFT PAREN ERROR
	1. EQUIVALENCE 文の左カッコのエラー
1	BEQUI -- VARIABLE NAME ERROR
	1. EQUIVALENCE 文の変数名のエラー
1	BEQUI -- VAR OVER FLOW
	1. 現在のルーチンが 500 個以上の COMMON 変数を含んでいる 2. EOSREV の修正が必要
1	BEQUI -- ARRAY SUBSCRIPT ERROR
	1. EQUIVALENCE 文の配列宣言のエラー

CODE	ERROR MESSAGE
1	BEQUI -- EQUIVALENCE GROUP RIGHT PAREN ERROR
	1. EQUIVALENCE 文の右カッコエラー
1	BEQUI -- STATEMENT END SYNTAX ERROR
	1. EQUIVALENCE 文が正しく終わっていない
1	BTYPE -- INVALID VARIABLE NAME
	1. 型宣言文の変数名のエラー
1	BTYPE -- LBL OVER FLOW
	1. 現在のルーチンが 100 個以上のCOMMON文を含んでいる 2. EOSREV の修正が必要
1	BTYPE -- VARIABLE LENGHT ERROR =FOUND
	1. 型宣言文中の COMMON 変数の長さ指定の誤り
1	BTYPE -- SUBSCRIPT SYNTAX ERROR =FOUND
	1. 型宣言文中の COMMON 配列の配列宣言子の文法エラー
1	BTYPE -- INITIAL VALUE SYNTAX ERROR =FOUND
	1. 型宣言文中の COMMON 配列の初期値セット部の文法エラー
1	BTYPE -- VARIABLE LENGTH ERROR =NOT FOUND
	1. 型宣言文中の変数の長さ指定の誤り
1	BTYPE -- ARRAY SUBSCRIPT ERROR
	1. 型宣言文中の配列宣言子の誤り
1	BTYPE INITIAL VALUE SYNTAX ERROR =NOT FOUND
	1. 型宣言文の初期値設定部の文法エラー
1	BTYPE -- STATEMENT END SYNTAX ERROR
	1. 型宣言文が正しく終わっていない
1	CMVAR -- COMMON LABEL SYNTAX ERROR
	1. COMMON 文ラベルの文法エラー

CODE	ERROR MESSAGE
1	CMVAR --- VARIABLE NAME ERROR
	1. COMMON 文の変数名エラー
1	CMVAR -- STATEMENT END SYNTAX ERROR
	1. COMMON 文が正しく終わっていない
1	CMVAR -- ERROR IN = REGDM =
	1. COMMON 変数の登録エラー
2	COMWRT -- VRF EXCEEDED MX 3
	1. プログラム内の COMMON 変数が 1000 個を越えた。 2. EOSREV の修正が必要 3. /COM/
2	COMWRT -- SZF EXCEEDED MX4
	1. プログラム内の COMMON 配列の為の Buffer が足りない。 2. EOSREV の修正が必要 3. /COM/
2	COPY -- EDIT BUFFER OVER FLOW
	1. 編集用バッファオーバーフロー 2. 現在の文が編集後も 20 行を越えない様に修正する
2	EBINIT -- EBUF OVER FLOW, EBNL OR EBEOL
	1. 編集用バッファオーバーフロー 2. 生成される COMDECK 文の行数が 20 行を越えないように変更する。
1	FTOW 1 -- END LINE LACKED
	1. END 行が無い
1	FTOTP -- END LINE MISSING
	1. END 行が無い
2	GETDST -- SBUF OVER FLOW
	1. ラベル名指定の入力が継続し過ぎている。

CODE	ERROR MESSAGE
2	GETDST -- MORE THAN 20 LINES 1. ラベル名指定の入力が20行以上継続している
1	GETDST -- STARTING LINE HAS A CONTMARK 1. 文の開始行の第6カラムが空白でも0でもない
1	GETST -- BLANK OR ONLY LABEL LINE 1. 空白行又はラベル名だけの行がある
2	GETST -- MORE THAN 20 LINES 1. 文が20行以上継続している。
2	GETST -- SBUF OVER FLOW 1. 文の解析用バッファがOVER FLOWした。
1	LABEL -- FIRST WORD WAS NOT EXTRACT NOR HOLD 1. ラベル指定入力カードの先頭が,EXTRACTでも,HOLDでも無い 2. ラベル指定入力カードはすべて先頭に,EXTRACT又は,HOLDと書かなければいけない
1	LABEL -- LABEL NAME SYNTAX ERROR 1. ラベル指定入力の文法エラー
1	LBNUM -- COMDECK NAME NOT FOUND IN LBL 1. COMDECK名がLBLに登録されていない 2. プログラムエラー
1	NSKIP -- INVALID N 1. NSKIPに渡されるNの値の誤り 2. プログラムエラー
1	PRMVAL -- SYNTAX ERROR 1. 実行時パラメータの記述の文法エラー
1	PUTST -- STATEMENT HAS MORE THAN 20 LINES 1. 文の出力時に1文が20行を越えた 2. 現在の文が編集されても20行を越えない様に変更する

CODE	ERROR MESSAGE
1	PUTWK -- INVALID POOL NO
	1. PUTWK への引数 P N の値が 1 でも 2 でもない 2. プログラムエラー
2	PUTWK -- POOL No 1 OVER FLOW
	1. 中間のテキストバッファがオーバーフローした 2. EOSREV を修正する必要がある。バッファは 10KW 3. /TEXTBF /
2	PUTWK -- POOL No 2 OVER FLOW
	同 上
2	PUTWK -- POOL 1 NOT ENOUGH SPACE
	同 上
2	PUTWK -- POOL 2 NOT ENOUGH SPACE
	同 上
1	READDT -- INVALID DATA
	1. データ入力の文法エラー
1	READDT -- NO EFFECTIVE CHARACTER
	1. 入力データカード上に有効文字列がない 2. 入力データを CHECK する
2	REGDL -- DELVAR OVER FLOW
	1. 編集を受ける COMMON 変数の数が 100 個を越えた。 2. COMMON 変数を含む、他の宣言文の数を減らす。できなければ EOSREV を修正する必要がある
1	REGDM -- INVALED CVR VALUE
	1. 変数の登録時に CVR の値が正しくない 2. プログラムエラー
1	REGDM -- ERROR IN NUMBER
	1. 数字列の認識ルーチンの誤り 2. プログラムエラー

CODE	ERROR MESSAGE
1	REGDM -- DECLARATOR SUBSCRIPT END SYNTAX ERROR
	1. 配列宣言子が正しく終わっていない
2	REGDM -- DF EXCEEDED MXD
	1. 登録する変数の数 (COMMON変数の数) が500個を越えた。 2. EOSREVを修正する必要がある
2	REGLB -- LBL OVER FLOW
	1. COMMONラベルの数が100個を越えた。(1つのルーチン) 2. EOSREVを修正する必要がある。
2	REGVR -- VAR OVER FLOW
	1. 1つのルーチン内のコモン変数の数が500個を越えた。 2. EOSREVを修正する。
1	REGVR -- COMMON VARIABLE DUPLICATED
	1. COMMON変数が2重に宣言されている
1	REGVR -- INVALID CLB VALUE
	1. 変数登録時のCLBの値が正しくない 2. プログラムエラー
1	STARTL -- ROUTINE NAME ERROR
	1. ルーチンの開始行にルーチン名が正しく記されていない
1	STCALL -- LABEL NAME NOT FOUND
	1. ラベル名が正しく登録されていない 2. プログラムエラー
1	TPTOE -- LABEL NAME NOT FOUND
	1. Temporary ファイル中のラベル名が登録されていない 2. プログラムエラー
1	TPTOE -- OLD #NO. NOT FOUND
	1. Temporary ファイル中の# No. が登録されていない
1	WCOMON -- TEXT TOO LONG (OVER 20 LINES)
	1. 生成するCOMMON文が長過ぎる。