

Fortran ソースプログラム静的解析ツール
S&E CodeScope®/Fortran マニュアル

エス・アンド・イー株式会社

はじめに

エス・アンド・イー株式会社は科学技術計算分野を主とするソフトウェア会社です。エス（S）は science、イー（E）は engineering の頭文字をとったものです。科学計算分野の研究者・技術者のご満足いただけるようなソフトウェア技術の提供を目指して努力しております。本書は、弊社で開発・利用している Fortran プログラムの静的解析ツールの使用手引書です。

科学計算用のプログラムは、いまだに Fortran 言語によって作成されるものが多くあります。機能の少ないプログラムは修正・改良等のプログラム保守も簡単ですが、よく利用されるプログラムほど機能が追加され、プログラム規模が大規模になっていき、それにつれて修正・改良等のプログラム保守に大変な労力がかかるようになるものです。特に、大規模プログラムで当初のプログラム作成者が担当を離れてしまったような場合には、プログラム保守が事実上できなくなることもあります。その理由は、いろいろあると思いますが、その主な原因の一つに、プログラムの構造、変数の使用状況等の把握が困難になることがあげられます。

それが原因で保守が困難になってきた大規模プログラムでも、プログラムの構造、変数の使用状況等をソースプログラムから自動的に取り出してくれるツールがあれば、プログラムの保守が容易になります。

S&E CodeScope®/Fortran は、Fortran 言語で書かれたソースプログラムを静的に解析し、共通領域、変数、サブルーチンの使用状況や、プログラムの論理構造等プログラムの保守に必要な情報を自動的に取り出すツールで、大規模 Fortran プログラムの修正・改良作業の労力を軽減することを可能にするツールです。CodeScope とはプログラム（コード）を見る道具という意味合いでつけた名前で「コードスコープ」と呼んでおります。

弊社の S&E CodeScope®/Fortran がユーザーの方々の Fortran プログラムに役立つことを願っております。

1 . 利用環境

(1) 計算機システム

Linux、Windows98/Me/2000/xp

UNIX 環境でも動作するものがありますが、弊社が他者に提供する実行モジュールは上記の計算機システムに限っております。

(2) 使用メモリ

3 万枚程度のプログラムで 4MB のメモリを使用します。使用するメモリは制御コマンドで変更できます。ただし、試用版では使用可能なメモリは約 2MB に固定されています。

(3) 処理時間

計算機の機種および解析オプションに依存しますが、8000 枚のプログラムの解析は 5 秒程度で終了します。

(4) 解析対象プログラム

Fortran77 言語で書かれており、コンパイルエラーのないプログラム。Fortran77 からの拡張構文でよく使用されるものは正常に解析できます。タブがあると解析結果が正常でないことがあります。

2 . 出力情報および制御コマンド

(1) 共通領域に関する情報

制御コマンドは `common`、`com` または `com(d)` になります。

共通領域が各ルーチンでどのように宣言されているかを要約して出力します。宣言パターンが同じがどうかを簡単に調べることができます。また、`com(d)`を指定すると、共通領域の変数(共通変数)が各ルーチンでどのように使用されているか(定義、参照、代入、値の入出力)の一覧表を出力します。

(2) プログラムの概略情報

制御コマンドは `program` または `pgm` になります。

各ルーチンの次の情報を出力します。

- ・ 仮引数の数と名前
- ・ 定義してある文関数名
- ・ 仮手続きの有無
- ・ 外部参照手続き名
- ・ 呼び出し元の手続き名
- ・ 参照している共通領域名
- ・ `include` されたファイル名

(3) 外部参照手続き名の情報

制御コマンドは `library` または `lib` になります。

ソースプログラム内で未解決となる外部参照手続き名とそれを参照しているルーチンの一覧表を出力します。

(4) プログラムのツリー図

制御コマンドは `tree` または `tree(e=xxxx)` になります。

`xxxx` はソースプログラムにあるプログラム名またはサブルーチン名です。

プログラムのツリー図を出力します。単に `tree` を指定した場合には、どこからも呼ばれていないルーチン(メインルーチンを含む)からのツリー図を出力します。 `tree(e=xxxx)` の形式で指定した場合には、`xxxx` からのツリー図を出力します。

(5) サブルーチン内の論理構造情報

制御コマンドは `nest` になります。

各サブルーチンの論理構造をソースプログラムとともに、線または矢印等で表現して出力します。DO 文、ブロック IF 文の入れ子の状態や、算術 IF 文や GOTO 文による処理流れの変更を確認することができます。

(6) Fortran 文の出現頻度一覧表

制御コマンドは `class` になります。

プログラムで使用されている Fortran 文を分類し、その集計情報を出力します。

(7) 総索引(マスターインデックス)

制御コマンドは `index` になります。

プログラムに現れた名前(Fortran のキーワードは含みません)を手続き名、変数名、配列名、共通領域名等に分類し、出現したルーチン名とそこでどのように使用されているか(定義、参照、代入、値の入出力)の一覧表を出力します。

(8) 未定義変数名の情報

制御コマンドは `undef` になります。

各ルーチンに現れる局所的な変数名または配列名で値の定義、代入が無いにもかかわらず参照があるものについてその名前を出力します。

(9) 定数名の情報

制御コマンドは `parameter` または `prm` になります。

`parameter` 文で定義された定数名が出現したルーチンと定義文字列の一覧表を出力します。

標準では、`common`、`library`、`class`、`tree` および `undef` が制御コマンドとして指定された状態になっています。これらの情報を出力したくない場合には、それぞれ `com(off)`、`lib(off)`、`class(off)`、`tree(off)` または `undef(off)` を 制御コマンドで指定します。

3 . インストールおよび実行方法

(1) インストール

ファイル `csfort.exe`(Windows 版)または `csfort`(Intel Linux 版)が S&E CodeScope®/Fortran の実行ファイルです。このファイルをパスの通った適切なディレクトリにコピーして下さい。さらに、Linux 版の場合には `chmod` で実行権を与えてください。これでインストールは終了です。

(2) 実行方法

S&E CodeScope®/Fortran コンソールアプリケーションです。Windows の場合は MS-DOS 画面を開いてコンソール上から次の形式でコマンドを入れます。

```
csfort [-option [-option ..]] [-foption-file]
      [-Idirectry] source- file [ [-Idirectry] source- file ..]
```

-option :解析の制御コマンドを記述します。省略可能です。 .

-foption-file :解析用の制御コマンドをファイルから入力したい場合に指定します。
`option-file` がそのファイル名です。 .

-Idirectry :インクルードファイル検索の起点ディレクトリを指定します。
それ以後に指定されるソースファイルにインクルードされるインクルードファイルの検索起点ディレクトリになります。
このオプションは省略可能です。
このオプションを指定しない場合には、インクルードファイル検索の起点ディレクトリはソースプログラムファイルのあるディレクトリになります。
指定したディレクトリをプログラムの標準値 (ソースプログラムファイルのあるディレクトリと同じディレクトリ)に戻すには、単に `-I` を指定します。

インクルードファイル名がソースファイルディレクトリからの相対パス名でソースプログラム中に指定されている場合にはこのオプションを使用する必要はありません。

source- file :解析対象のソースプログラムファイル名を指定します。
ファイル名に相対パス名を指定することができます。また、ファイル名にはワイルドカード文字 `*` または `?`を含めることができます。ここで、`*` は任意の文字列、`?` は任意の 1文字を意味します。ただし、相対パス名のディレクトリ名部分にワイルドカード文字を指定することはできません。ソースプログラムファイル名は省略できません。

また、単に `csfort` のみをタイプして実行すると簡単なガイドメッセージがでますのでご参考にして下さい。

S&E CodeScope®/Fortran の実行例

```
csfort -index -com(d) -nest *.f >result.txt
```

解析オプションとして、総索引、共通領域の変数使用状況、プログラムのネスト構造を出力するように指定し、解析結果が出力される標準出力を `result.txt` にリダイレクトしています。プログラムのツリー構造、ライブラリルーチンの参照状況は標準で出力されます。

3 . 制御コマンドの記述方法および説明

(1) 制御コマンドの記述方法

ソースプログラム解析を制御するコマンドは、実行する際のコマンドラインまたはオプションで指定された制御コマンド入力ファイルより入力します。制御コマンドは次の形式で記述します。

コマンドラインより入力するときは制御コマンドの前にハイフン'-'を付けます。

`-option(sub-option,sub-option,.....)`

制御コマンド入力ファイルより入力するときは制御コマンドの前にハイフン'-'を付けません。

`option(sub-option,sub-option,.....)`

`sub-option` の記述方法は次のいずれかになります。どの形式で記述するかは各 `option` によって異なります。`sub-option` を指定しない場合には `keyword` として `on` が指定されたものとみなされます。

`keyword`

整数値

`keyword=整数値`

`keyword(整数値)`

`keyword` の後の括弧は続けて記述します。

記述規則には次の制限があります。

- 1) `option` に `sub-option` をつけるときは `sub-option` の全体を括弧'()'で囲み `option` の後の'('は空白なしで記述しなければなりません。
- 2) `option` または `sub-option` を複数指定するときは、区切り記号','または空白を使用して記述します。
- 3) 制御コマンドを制御コマンド入力ファイルより入力するときは、1 から 7 2 カラムまでの間に記述し、何ラインでも記述できます。
- 4) 制御コマンドを制御コマンド入力ファイルより入力するとき 1 カラムの文字が'*'の場合にはコメントとして扱われます。

制御コマンドの評価規則は次のようになります。

- 1) 同じ制御コマンドを二回以上指定した場合には最後に記述したものが優先されます。
- 2) 未定義の制御コマンドは無視され、エラー出力はしません。
- 3) 未定義または正しくない `sub-option` はエラーとして検出されメッセージが出力されます。その `sub-option` は無効としてプログラムは実行されます。
- 4) コマンドラインと制御コマンド入力ファイルで同じ制御コマンドを入力した場合には制御コマンド入力ファイルで指定されたものが優先されます。

(2) 制御コマンド説明

各制御コマンドの説明を次の形式で行います。

- ・ **option**<option alias> 概要説明。
- sub-option**<sub-option alias> 説明。

制御コマンドの指定は、**option** または **option(sub-option)**の形式であり、**sub-option** を指定しない場合には **on** が指定されたものとみなされます。

- ・ **class** FORTRAN の文の出現頻度一覧表。
 - on** 出力する。
 - off** 出力しない。

- ・ **common**<com> 共通領域に関する情報。
 - on** 宣言様式を出力する。
 - off** 出力しない。
 - brief** **on** と同じ。
 - detail**<d> 宣言様式に加えて共通領域の変数の使用状況を出力する。

- ・ **header** 出力のページコントロール。
 - on** ページコントロールを行う。一定の出力行数で改ページを行い、ヘッダーを出力する。
 - off** ページコントロールをしない。解析結果の出力において各出力の最初にヘッダーが出力されるだけになる。

- ・ **index** 出現した名前の使用状況を示す総索引。
 - on** すべての名前について出力する。
 - off** 出力しない。
 - lower**<l> 出力する名前の下限。**lower=c** とすると英文字で **c** より前の名前については出力しない。
 - upper**<u> 出力する名前の上限。**upper=x** とすると英文字で **x** より後の名前については出力しない。

- ・ **library**<lib> ライブラリーサブルーチンの参照状況。
 - on** 出力する。
 - off** 出力しない。

- ・ **lines**
 整数値

1 ページあたりの出力行数を制御する。
 1 ページあたりの出力行数を指定する。
 指定した行数ごとに改ページが行われヘッダーが出力される。
 9 以下の行数を指定した場合は、ページコントロールがなくなり
header(off)を指定したことと同じになる。

- ・ **nest<nes>**

プログラムの論理構造情報。do...enddo、if...then...else...endif 等の
 入れ子状態の図示。

on 出力する。
off 出力しない。

- ・ **parameter<prm>**

定数名の一覧表。

on 出力する。
off 出力しない。

- ・ **pgname**

表紙および各出力ページのヘッダーに出力されるプログラム名を指
 定する。この制御コマンドを指定しないときはプログラム名は
PROGRAM となる。sub-option は必ず指定しなければならない。
 文字列 プログラム名を表す 8 文字以内の文字列。空白 'または特殊記号があ
 る場合には引用符号()で囲む。

- ・ **program<pgm>**

サブルーチンの概略情報。

on 出力する。
off 出力しない。

- ・ **size**

解析に必要な表の大きさを指定する。
sub-option は必ず指定する必要がある。また、複数の sub-option を
 指定できる。標準では 20 万枚程度のプログラムの解析ができるよう
 に表の大きさが設定されている。

small<s> 表の大きさを 1 万枚以内のプログラムを解析する場合に設定。
micld<m> 表の大きさを 3 万枚以内のプログラムを解析する場合に設定。
large<l> 表の大きさを 5 万枚以内のプログラムを解析する場合に設定。
mxxxx=整数値 表の大きさを示す名前 mxxxx の値を整数値に設定する。
mxxxx の種類とその意味およびプログラムでの標準値は表 1 および
 表 2 の通りである。

- ・ **tree**

プログラムのツリー図。

on 出力する。どこからも呼ばれていないツーチンからのツリー図を出力

| | |
|-------------|-----------------------------------|
| | する。 |
| off | 出力しない。 |
| entry<e>=名前 | 指定した名前のルーチンからのツリー図を出力する。 |
| • undef | 未定義変数名の情報。 |
| on | 出力する。ただし、equivalence 文の考慮はされていない。 |
| | する。 |
| off | 出力しない。 |

表 1 size オプションで指定する名前の意味

| 名前 | 意味 |
|--------|--|
| MXARG | サブルーチンの仮引数名を記憶しておくための表の大きさ。 |
| MXCBF | 共通領域宣言の文字列、配列宣言の文字列、実引数の文字列および定数名の文字列を記憶しておくための表の大きさ。 |
| MXCBUF | 一つのサブルーチン内の共通領域宣言の文字列、配列宣言の文字列、実引数の文字列および定数名の文字列を記憶しておくための表の大きさ。 |
| MXCDBF | NESTまたはSTATEMENTで出力されるカードを記憶しておくための表の大きさ。 |
| MXCOM | プログラムで使用した共通領域名を記憶しておくための表の大きさ。 |
| MXCOM1 | プログラムで使用した共通領域についてその宣言文字列を管理するための表の大きさ。 |
| MXCOM2 | プログラムで使用した共通領域についてその中の変数名を記憶しておくための表の大きさ。 |
| MXCOM3 | プログラムで使用した共通領域についてその中の変数名の使用状況を記憶しておくための表の大きさ。 |
| MXDXV | 総索引で出力される名前を記憶しておくための表の大きさ。名前の数はサブルーチンが異なれば別の名前として数える。 |
| MXINC | プログラムでインクルードされたファイルの名前とその参照関係を管理するための表の大きさ。 |
| MXLBDR | 一つのサブルーチン内の文番号の定義・参照を管理するための表の大きさ。 |
| MXMBDR | インクルードされるファイルの参照関係を管理するための表の大きさ。 |
| MXPCNT | 一つのサブルーチン内の定数名の定義文字列とその値を管理するための表の大きさ。 |
| MXPRM | プログラム内で定義した定数名とその定義文字列を管理するための表の大きさ。 |
| MXPRO | プログラム名およびサブルーチン名を記憶し管理するための表の大きさ。 |
| MXRCOM | サブルーチンで使用している共通領域名を記憶しておくための表の大きさ。 |
| MXRENT | サブルーチンにある別エントリー名を記憶しておくための表の大きさ。 |
| MXRINC | サブルーチンでインクルードしたファイル名を記憶しておくための表の大きさ。 |
| MXRPRM | サブルーチンで定義した定数名を記憶しておくための表の大きさ。 |
| MXRXAD | 名前を参照しているサブルーチン名とその参照形式を記憶しておくための表の大きさ。 |
| MXRXBF | 名前を参照しているサブルーチン名とその参照形式を記憶しておくための表の大きさ。 |
| MXSFX | サブルーチンで参照したサブルーチンまたは関数の名前を管理するための表の大きさ。 |
| MXSQBF | 一つのサブルーチン内の仮引数名、共通変数名の出現順を記憶しておくための表の大きさ。 |
| MXVCNT | 一つのサブルーチン内の仮引数名、共通変数名、共通領域宣言の文字列、実引数の管理を行うための表の大きさ。 |
| MXVNAM | 一つのサブルーチン内に出てくる名前を記憶しておくための表の大きさ。 |
| MXXCNT | 実引数の文字列を管理するための表の大きさ。 |

表2 size オプションで指定する名前の値

| 名前 | 標準値 | small | middle | large |
|--------|--------|-------|--------|-------|
| MXARG | 20000 | 1500 | 3000 | 10000 |
| MXCBF | 80000 | 10000 | 15000 | 80000 |
| MXCBUF | 20000 | 5000 | 5000 | 20000 |
| MXCDBF | 10000 | 2500 | 2500 | 10000 |
| MXCOM | 200 | 50 | 100 | 200 |
| MXCOM1 | 300 | 100 | 150 | 300 |
| MXCOM2 | 4000 | 1000 | 2000 | 4000 |
| MXCOM3 | 20000 | 5000 | 10000 | 20000 |
| MXDXV | 200000 | 10000 | 10000 | 10000 |
| MXINC | 500 | 100 | 100 | 500 |
| MXLBDR | 500 | 500 | 500 | 500 |
| MXMBDR | 10000 | 500 | 500 | 10000 |
| MXPCNT | 1000 | 100 | 100 | 1000 |
| MXPRM | 500 | 100 | 250 | 500 |
| MXPRO | 2000 | 250 | 300 | 1000 |
| MXRCOM | 6000 | 1000 | 1500 | 6000 |
| MXRENT | 500 | 20 | 50 | 100 |
| MXRINC | 5000 | 500 | 1000 | 4000 |
| MXRPRM | 2000 | 500 | 1000 | 2000 |
| MXRXAD | 2000 | 250 | 1000 | 2000 |
| MXRXBF | 20000 | 500 | 2000 | 20000 |
| MXSFX | 10000 | 500 | 1000 | 5000 |
| MXSQBF | 2000 | 1000 | 1500 | 2000 |
| MXVCNT | 2000 | 500 | 500 | 2000 |
| MXVNAM | 2000 | 1000 | 1500 | 2000 |
| MXXCNT | 1000 | 50 | 50 | 1000 |

4 . お問い合わせ先

S&E CodeScope®/Fortran の詳細については以下にお問い合わせ下さい。

275-0026

千葉県習志野市谷津 5 - 4 0 - 2 5 - 2 0 1

エス・アンド・イー株式会社

CodeScope 担当

TEL +81-47-470-3281

FAX +81-47-470-3282

E-mail codescope@s-e.co.jp

- (1) 本書の内容は将来予告無しに変更することがあります。
- (2) S&E CodeScope®/Fortran を運用した結果については責任を負いかねますので御了承下さい。
- (3) S&E CodeScope®/Fortran には使用期限があります。
- (4) S&E および CodeScope は、エス・アンド・イー株式会社の登録商標です。

Copyright © 2000,2001 S&E,LTD. All rights reserved.