

DCPAM Reference Manual

GFD Dennou Club

平成 17 年 2 月 18 日

目次

1	Module varinfo_mod	2
1.1	Overview	2
1.2	Error Handling	2
1.3	Known Bugs	2
1.4	Note	2
1.5	Future Plans	2
1.6	Dependency	2
1.7	Public Interface	2
1.8	Generic Procedure	2
1.9	Derived Types	3
1.10	Procedure Interface	3
1.10.1	Initialize module and acquire NAMELIST	3
1.10.2	Return Variable Information about varkey.	6
1.10.3	Terminate module	6
1.10.4	Copy VAR_INFO array.	6
1.10.5	Copy GT_HISTORY_ATTR array.	7
1.10.6	Initialize GT_HISTORY_ATTR variable.	7
1.10.7	Initialize GT_HISTORY_ATTR variable.	8

1 Module varinfo_mod

- Developers: Morikawa Yasuhiro
- Version: \$Id: varinfo.f90,v 1.8 2005/01/19 08:52:45 morikawa Exp \$
- Tag Name: \$Name: \$
- Change History:

1.1 Overview

This module store Variable Infomation from NAMELIST and Provide. These Information is expected to be used by io_gt4_out_mod .

NAMELIST から出力する変数に関する情報を取得する。これらの情報は io_gt4_out_mod にて用いられることが想定されている。

1.2 Error Handling

1.3 Known Bugs

1.4 Note

1.5 Future Plans

もしかすると、io_gt4_out_mod モジュール内に格納されるべきかも知れませんが、

1.6 Dependency

```
use type_mod,      only : INTKIND, STRING, TOKEN
use gt4_history,   only : GT_HISTORY_VARINFO, GT_HISTORY_ATTR
```

1.7 Public Interface

```
private
public :: VAR_INFO                                ! derived types
public :: varinfo_init, varinfo_inquire, varinfo_end ! subroutines
```

1.8 Generic Procedure

```
interface varinfo_attrs_init
```

```

    module procedure varinfo_attrs_init0, varinfo_attrs_init1
end interface

```

1.9 Derived Types

変数に関する各情報を格納 (データ本体は含まない)

```

type VAR_INFO
  character(STRING)      :: varkey = '' ! 変数キー
  character(STRING)      :: file   = '' ! 出力するファイル
  integer(INTKIND)       :: StepInterval = 0 ! 出力ステップ間隔

  integer(INTKIND)       :: OutputStep = 0 ! 出力回数
  type(GT_HISTORY_VARINFO):: varinfo   ! gt4 変数情報
  type(GT_HISTORY_ATTR), allocatable &
    &                       :: attrs(:) ! 属性情報群
end type VAR_INFO

```

1.10 Procedure Interface

1.10.1 Initialize module and acquire NAMELIST

モジュールを初期化し、NAMELIST から値を取得する。NAMELIST から値が取得できないものに関しては上記のデフォルト値が用いられる。

NAMELIST ファイルは、メインプログラムにて nmlfile_mod の nmlfile_init で指定されることが想定されているが、もしもこの初期化ルーチンより以前に指定されていなければ、nmlfile_init のデフォルトで指定される NAMELIST ファイルを読む。

```

subroutine varinfo_init

```

Dependency

```

use type_mod, only: STRING, TOKEN, INTKIND, REKIND, DBKIND, &
  &                NMLARRAY
use nmlfile_mod, only: nmlfile_init, nmlfile_open, nmlfile_close
use dc_types, only: GT_TOKEN => TOKEN, GT_STRING => STRING
use dc_trace, only: DbgMessage, BeginSub, EndSub
use dc_message, only: MessageNotify

```

NAMELIST varinfo_nml

変数に関する基本情報を設定するための NAMELIST で、複数の varinfo_nml を用意する事で、複数の変数の設定が可能である。変数 varkey はモデル毎に内部で設定される物理量のマーカで、具体的には io_gt4_out_mod モジュールの io_gt4_out_SetVars サブルーチンでマーカを設定し、io_gt4_out_Put サブルーチンでデータを file に指定されたファイルに出力する。もしも file を与えない、または空文字を与えた場合には、io_gt4_out_nml で与えた default_output 変数で指定されたファイルに出力する。

varname, dimnum, dimnames, longname, units, xtype は出力される変数に付加される情報である。

StepInterval, OutputStep を与えない、またはゼロ以下の値を与えた場合には time_mod の time_nml で与えた StepInterval, OutputStep が用いられる。

```
character (STRING)    :: varkey   = '' ! 変数キー
character (STRING)    :: file     = '' ! 出力するファイル

character (GT_TOKEN)  :: varname  = '' ! 変数名
integer (INTKIND)     :: dimnum   = 0 ! 依存する次元
character (GT_TOKEN)  :: dimnames (NMLARRAY) = '' ! 依存する次元
character (GT_STRING) :: longname = '' ! 変数の記述的名称
character (GT_STRING) :: units    = '' ! 変数の単位
character (GT_TOKEN)  :: xtype    = '' ! 変数の型

integer (INTKIND)     :: StepInterval = 0 ! 出力ステップ間隔
integer (INTKIND)     :: OutputStep   = 0 ! 出力回数

namelist /varinfo_nml/ &
  & varkey      , & ! 変数キー
  & file       , & ! 出力するファイル
  &
  & varname    , & ! 変数名
  & dimnum     , & ! 依存する次元の数
  & dimnames   , & ! 依存する次元変数名
  & longname   , & ! 変数の記述的名称
  & units      , & ! 変数の単位
  & xtype     , & ! 変数の型
  &
  & StepInterval , & ! 出力回数
  & OutputStep   ! 出力ステップ間隔
```

NAMELIST varinfo_attr_nml

変数 varattr の属性情報を与える。NAMELIST に複数の varinfo_attr_nml を用意しておく事で複数の変数に対し、複数の情報を与える事が可能である。与えない場合には属性情報は付加されない。

attrtype には与える属性値の種類を設定する。<URL:http://www.gfd-dennou.org/arch/gtool4/gt4f90io-current/doc/gt_history.htm#derived_gthistoryattr> を参照せよ。なお、arraysize に 1 以上の値を設定すると、配列データが優先されて属性値に設定される。

```
character(GT_STRING) :: varattr = '' ! 属性を付加する変数名
character(GT_TOKEN)  :: attrname = '' ! 属性名
character(GT_TOKEN)  :: attrtype = '' ! 属性値の型
character(GT_STRING) :: cvalue   = '' ! 属性の値 (文字)
integer(INTKIND)     :: ivalue   = 0    ! 属性の値 (整数)
real(REKIND)         :: rvalue   = 0.0  ! 属性の値 (単精度実数)
real(DBKIND)         :: dvalue   = 0.0d0 ! 属性の値 (倍精度実数)
logical              :: lvalue   = .false. ! 属性の値 (論理)
integer(INTKIND)     :: arraysize= 0    ! 配列のサイズ
integer(INTKIND) :: iarray(NMLARRAY) = 0    ! 属性の値 (整数)
real(REKIND)      :: rarray(NMLARRAY) = 0.0 ! 属性の値 (単精度実数)
real(DBKIND)      :: darray(NMLARRAY) = 0.0d0 ! 属性の値 (倍精度実数)
```

```
namelist /varinfo_attr_nml/ &
  & varattr      , & ! 属性を付加する変数名
  & attrname     , & ! 属性名
  & attrtype     , & ! 属性値の型
  & cvalue       , & ! 属性の値 (文字)
  & ivalue       , & ! 属性の値 (整数)
  & rvalue       , & ! 属性の値 (単精度実数)
  & dvalue       , & ! 属性の値 (倍精度実数)
  & lvalue       , & ! 属性の値 (論理)
  & arraysize    , & ! 配列のサイズ
  & iarray       , & ! 属性の値 (整数)
  & rarray       , & ! 属性の値 (単精度実数)
  & darray       , & ! 属性の値 (倍精度実数)
```

1.10.2 Return Variable Information about varkey.

変数キー varkey に対応する変数情報 info を返す。stat にはステータスを返す。正常に変数情報が返れば 0, varkey に対応する情報が無ければ 1, varinfo_init による初期化が行なわれていない場合は -1 を返す。

```
subroutine varinfo_inquire(varkey, info, stat)
```

Dependency

```
use type_mod,    only: STRING, TOKEN, INTKIND, REKIND, DBKIND
use dc_trace,    only: DbgMessage, BeginSub, EndSub
use dc_error,    only: StoreError, USR_ECHAR
```

Input

```
character(*),    intent(in)  :: varkey ! 変数キー
```

Output

```
type(VAR_INFO),  intent(out) :: info   ! VAR_INFO 変数
integer(INTKIND), intent(out) :: stat   ! ステータス
```

1.10.3 Terminate module

varinfo_init で設定された値を破棄し、デフォルトに戻す。

```
subroutine varinfo_end
```

Dependency

```
use dc_trace,    only: DbgMessage, BeginSub, EndSub
```

1.10.4 Copy VAR_INFO array.

varinfo_mod の内部サブルーチン。構造体 VAR_INFO の 1 次元配列のコピーを行なう。

```
subroutine varinfo_copy(from, to)
```

Dependency

```
use type_mod     , only: STRING, INTKIND
use dc_trace     , only: BeginSub, EndSub, DbgMessage
```

Input

```
type(VAR_INFO), intent(in) :: from(:)
```

Output

```
type(VAR_INFO), intent(out) :: to(:)
```

1.10.5 Copy GT_HISTORY_ATTR array.

varinfo_mod の内部サブルーチン。構造体 GT_HISTORY_ATTR の 1 次元配列のコピーを行なう。

```
subroutine varinfo_attrs_copy(from, to)
```

Dependency

```
use type_mod, only: STRING, INTKIND
use gt4_history, only: GT_HISTORY_ATTR
use dc_trace, only: BeginSub, EndSub, DbgMessage
```

Input

```
type(GT_HISTORY_ATTR), intent(in) :: from(:)
```

Output

```
type(GT_HISTORY_ATTR), intent(out) :: to(:)
```

1.10.6 Initialize GT_HISTORY_ATTR variable.

varinfo_mod の内部サブルーチン。構造体 GT_HISTORY_ATTR 変数の初期化を行なう。

```
subroutine varinfo_attrs_init0(attrs)
```

Dependency

```
use type_mod, only: STRING, INTKIND
use gt4_history, only: GT_HISTORY_ATTR
use dc_trace, only: BeginSub, EndSub, DbgMessage
```

In/Out

```
type(GT_HISTORY_ATTR), intent(inout):: attrs
```

1.10.7 Initialize GT_HISTORY_ATTR variable.

varinfo_mod の内部サブルーチン。構造体 GT_HISTORY_ATTR の1次元配列の初期化を行なう。

```
subroutine varinfo_attrs_init1(attrs)
```

Dependency

```
use type_mod, only: STRING, INTKIND  
use gt4_history, only: GT_HISTORY_ATTR  
use dc_trace, only: BeginSub, EndSub, DbgMessage
```

In/Out

```
type(GT_HISTORY_ATTR), intent(inout):: attrs(:)
```