

DCPAM Reference Manual

GFD Dennou Club

平成 17 年 2 月 18 日

目次

1	Module io_gt4_out_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	3
1.9	Derived Types	3
1.10	Procedure Interface	3
1.10.1	Initialize module and acquire NAMELIST	3
1.10.2	Set Dimension	4
1.10.3	Set Variables	5
1.10.4	Put 3-Dimensional Single Precision Data to netCDF file	5
1.10.5	Put 2-Dimensional Single Precision Data to netCDF file	6
1.10.6	Put 0-Dimensional Single Precision Data to netCDF file	7
1.10.7	Put 3-Dimensional Double Precision Data to netCDF file	7
1.10.8	Put 2-Dimensional Double Precision Data to netCDF file	8
1.10.9	Put 2-Dimensional Double Precision Data to netCDF file	8
1.10.10	Terminate module	9

1 Module io_gt4_out_mod

- Developers: Morikawa Yasuhiro
- Version: \$Id: io_gt4_out.f90,v 1.10 2005/01/19 08:52:36 morikawa Exp \$
- Tag Name: \$Name: \$
- Change History:

1.1 Overview

This module output data with gtool4 netCDF conventions.
gtool4 netCDF 規約に基づくデータを出力する。

1.2 Error Handling

1.3 Known Bugs

- 1つのファイルに異なる StepInterval が設定される場合、HistoryCreate には先に設定されたものの StepInterval で時間間隔を決めるため、後で設定した変数の StepInterval が前のものよりも小さい場合には整合的でなくなる。
 - 回避方法
 - * 異なる StepInterval を設定する場合にはそれらは別個のファイルに分ける。

1.4 Note

1.5 Future Plans

1.6 Dependency

```
use type_mod,      only : REKIND, DBKIND, INTKIND, TOKEN, STRING
use axis_type_mod, only : AXISINFO
use gt4_history,   only : GT_HISTORY
```

1.7 Public Interface

```
private
```

```

public :: io_gt4_out_init, io_gt4_out_end      ! subroutines
public :: io_gt4_out_SetDims                  ! subroutines
public :: io_gt4_out_SetVars, io_gt4_out_Put ! subroutines

```

1.8 Generic Procedure

```

interface io_gt4_out_Put
  module procedure io_gt4_out_Put3Real
  module procedure io_gt4_out_Put2Real
  module procedure io_gt4_out_Put0Real
  module procedure io_gt4_out_Put3Double
  module procedure io_gt4_out_Put2Double
  module procedure io_gt4_out_Put0Double
end interface

```

1.9 Derived Types

出力変数データ格納用構造体。 io_gt4_out_SetVars で設定。

```

type IO_GT4_OUT_VARS
  character(STRING), allocatable :: varkeys(:) ! 変数キー
  character(STRING)              :: file      ! 出力ファイル
  type(GT_HISTORY)               :: gt_history ! GT_HISTORY 変
数
  logical                        :: created   ! HistoryCreate 済
かどうか
  type(IO_GT4_OUT_VARS), pointer :: next
end type IO_GT4_OUT_VARS

```

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 io_gt4_out_init

```

Dependency

```
use type_mod,      only : REKIND, DBKIND, INTKIND, TOKEN, STRING
use nmlfile_mod,  only : nmlfile_init, nmlfile_open, nmlfile_close
use time_mod,     only : time_init
use varinfo_mod,  only : varinfo_init
use dc_trace,     only : SetDebug, BeginSub, EndSub, DbgMessage
use dc_message,   only : MessageNotify
```

NAMELIST

出力ファイル設定。file に与えたものがデフォルトの出力ファイルとなる。その他の情報は出力する gtool4 netCDF データの大域データとして与えられる。

```
character (STRING) ::
  & file           = 'result.nc'      , & ! 出力ファイル名 (デ
フォルト)
  & title          = 'GCM Test'       , & ! タイトル
  & source         = 'DCPAM'          , & ! モデル名 (作成手段)
  & institution    = 'GFD Dennou Club' ! 実行者名 (作成者)

namelist /io_gt4_out_nml/ &
  & file           , & ! 出力ファイル名 (デフォルト)
  & title          , & ! タイトル
  & source         , & ! モデル名 (作成手段)
  & institution    ! 実行者名 (作成者)
```

1.10.2 Set Dimension

出力する gtool4 netCDF データの座標情報を設定する。複数回呼ぶ事で複数の座標を設定する。現在の所、設定した座標は出力する全ての netCDF ファイルに出力される。

```
subroutine io_gt4_out_SetDims(axis)
```

Dependency

```
use type_mod,      only : REKIND, DBKIND, INTKIND, TOKEN, STRING
use axis_type_mod, only : AXISINFO, axis_type_copy
use dc_trace,     only : SetDebug, BeginSub, EndSub, DbgMessage
```

Input

```
type(AxisInfo), intent(in) :: axis
```

1.10.3 Set Variables

出力する変数を設定する。複数回呼ぶ事で複数の変数を設定できる。このサブルーチンで設定するのは変数キー `varkey` のみであり、具体的な情報は `varinfo_mod` の `varinfo_init` にて `NAMELIST varinfo_nml` で設定される。(本来は、デフォルトの値はプログラム無いにハードコードすべきかも知れない)。

```
subroutine io_gt4_out_SetVars(varkey)
```

Dependency

```
use type_mod,      only : REKIND, DBKIND, INTKIND, TOKEN, STRING
use time_mod,      only : InitTime, DelTime, StepInterval,      &
                        &          tvar, ttype, tname, tunit
use varinfo_mod,   only : varinfo_inquire, VAR_INFO
use gt4_history,   only : GT_HISTORY,   GT_HISTORY_AXIS,      &
                        &          HistoryCreate, HistoryAddVariable, &
                        &          HistoryCopyVariable,          &
                        &          HistoryAddAttr, HistoryPut
use dc_string,     only : JoinChar
use dc_trace,      only : SetDebug, BeginSub, EndSub, DbgMessage
```

Input

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

1.10.4 Put 3-Dimensional Single Precision Data to netCDF file

変数キー `varkey` にデータ `xyz_Var` を出力する。`varinfo_mod` の `varinfo_init` の `NAMELIST varinfo_nml` で対応する `varkey` が与えられていない場合、データは出力されない。

各 `varkey` に対応する `varinfo_mod` の `StepInterval` および `OutputStep` と、`time_mod` の `CurrentLoop` から、出力するタイミングが正しいかどうかをチェックして出力する。出力するタイミングでないと判定された場合は何もせず終了する。なお、`varinfo_mod` の `StepInterval` および `OutputStep` が無効な値 (ゼロ以下) の場合には `time_mod` の `StepInterval` と `OutputStep` が用いられる。具体的な判定方法は以下の通りである。

- CurrentLoop を StepInterval で割り、余りが 0 の場合には出力。
- CurrentLoop が StepInterval * OutputStep よりも大きくなってしまったら以降出力は行なわない。

```
subroutine io_gt4_out_Put3Real(varkey, xyz_Var)
```

Dependency

```
use type_mod,    only : REKIND, DBKIND, INTKIND, TOKEN, STRING
use time_mod,    only : StepInterval, OutputStep, CurrentLoop
use varinfo_mod, only : varinfo_inquire, VAR_INFO
use gt4_history, only : HistoryPut
use dc_trace,    only : SetDebug, BeginSub, EndSub, DbgMessage
use dc_message,  only : MessageNotify
```

Input

```
character(*),    intent(in):: varkey          ! 変数名
real(REKIND),    intent(in):: xyz_Var(:, :, :) ! 出力データ
```

1.10.5 Put 2-Dimensional Single Precision Data to netCDF file

機能は io_gt4_out_Put3Real と基本的に同じ。ただしこちらは単精度実数 2 次元のデータを出力する。

```
subroutine io_gt4_out_Put2Real(varkey, xy_Var)
```

Dependency

```
use type_mod,    only : REKIND, DBKIND, INTKIND, TOKEN, STRING
use time_mod,    only : StepInterval, OutputStep, CurrentLoop
use varinfo_mod, only : varinfo_inquire, VAR_INFO
use gt4_history, only : HistoryPut
use dc_trace,    only : SetDebug, BeginSub, EndSub, DbgMessage
use dc_message,  only : MessageNotify
```

Input

```
character(*),    intent(in):: varkey          ! 変数名
real(REKIND),    intent(in):: xy_Var(:, :) ! 出力データ
```

1.10.6 Put 0-Dimensional Single Precision Data to netCDF file

機能は io_gt4_out_Put3Real と基本的に同じ。ただしこちらは単精度実数 0 次元のデータを出力する。

```
subroutine io_gt4_out_Put0Real(varkey, Var)
```

Dependency

```
use type_mod,    only : REKIND, DBKIND, INTKIND, TOKEN, STRING
use time_mod,    only : StepInterval, OutputStep, CurrentLoop
use varinfo_mod, only : varinfo_inquire, VAR_INFO
use gt4_history, only : HistoryPut
use dc_trace,    only : SetDebug, BeginSub, EndSub, DbgMessage
use dc_message,  only : MessageNotify
```

Input

```
character(*),    intent(in):: varkey      ! 変数名
real(REKIND),    intent(in):: Var        ! 出力データ
```

1.10.7 Put 3-Dimensional Double Precision Data to netCDF file

機能は io_gt4_out_Put3Real と基本的に同じ。ただしこちらは倍精度実数 3 次元のデータを出力する。

```
subroutine io_gt4_out_Put3Double(varkey, xyz_Var)
```

Dependency

```
use type_mod,    only : REKIND, DBKIND, INTKIND, TOKEN, STRING
use time_mod,    only : StepInterval, OutputStep, CurrentLoop
use varinfo_mod, only : varinfo_inquire, VAR_INFO
use gt4_history, only : HistoryPut
use dc_trace,    only : SetDebug, BeginSub, EndSub, DbgMessage
use dc_message,  only : MessageNotify
```

Input

```
character(*),    intent(in):: varkey      ! 変数名
real(DBKIND),    intent(in):: xyz_Var(:, :, :) ! 出力データ
```


1.10.8 Put 2-Dimensional Double Precision Data to netCDF file

機能は io_gt4_out_Put3Real と基本的に同じ。ただしこちらは倍精度実数 2 次元のデータを出力する。

```
subroutine io_gt4_out_Put2Double(varkey, xy_Var)
```

Dependency

```
use type_mod,    only : REKIND, DBKIND, INTKIND, TOKEN, STRING
use time_mod,    only : StepInterval, OutputStep, CurrentLoop
use varinfo_mod, only : varinfo_inquire, VAR_INFO
use gt4_history, only : HistoryPut
use dc_trace,    only : SetDebug, BeginSub, EndSub, DbgMessage
use dc_message,  only : MessageNotify
```

Input

```
character(*),    intent(in):: varkey      ! 変数名
real(DBKIND),    intent(in):: xy_Var(:,) ! 出力データ
```

1.10.9 Put 2-Dimensional Double Precision Data to netCDF file

機能は io_gt4_out_Put3Real と基本的に同じ。ただしこちらは倍精度実数 2 次元のデータを出力する。

```
subroutine io_gt4_out_Put0Double(varkey, Var)
```

Dependency

```
use type_mod,    only : REKIND, DBKIND, INTKIND, TOKEN, STRING
use time_mod,    only : StepInterval, OutputStep, CurrentLoop
use varinfo_mod, only : varinfo_inquire, VAR_INFO
use gt4_history, only : HistoryPut
use dc_trace,    only : SetDebug, BeginSub, EndSub, DbgMessage
use dc_message,  only : MessageNotify
```

Input

```
character(*),    intent(in):: varkey      ! 変数名
real(DBKIND),    intent(in):: Var        ! 出力データ
```

1.10.10 Terminate module

io_gt4_out_init で設定された値を破棄し、デフォルトの値に戻す。また、HistoryClose によって、HistoryCreate 等に対応する終了処理を行なう。

```
subroutine io_gt4_out_end
```

Dependency

```
use type_mod,    only : REKIND, DBKIND, INTKIND, TOKEN, STRING
use gt4_history, only : HistoryClose
use dc_trace,    only : SetDebug, BeginSub, EndSub, DbgMessage
```