

Cartesian Data Management Library  
0.9.5

作成 : Doxygen 1.8.5

Mon Oct 31 2016 10:26:02



# Contents

<b>1</b>	<b>ネームスペース索引</b>	<b>1</b>
1.1	ネームスペース一覧	1
<b>2</b>	<b>階層索引</b>	<b>3</b>
2.1	クラス階層	3
<b>3</b>	<b>構成索引</b>	<b>5</b>
3.1	構成	5
<b>4</b>	<b>ファイル索引</b>	<b>7</b>
4.1	ファイル一覧	7
<b>5</b>	<b>ネームスペース</b>	<b>9</b>
5.1	ネームスペース CDM	9
5.1.1	説明	11
5.1.2	列挙型	11
5.1.2.1	E_CDM_ARRAYSHAPE	11
5.1.2.2	E_CDM_DFITYPE	11
5.1.2.3	E_CDM_DTYPE	11
5.1.2.4	E_CDM_ENDIANATYPE	12
5.1.2.5	E_CDM_ERRORCODE	12
5.1.2.6	E_CDM_FILE_TYPE	15
5.1.2.7	E_CDM_FORMAT	15
5.1.2.8	E_CDM_ONOFF	15
5.1.2.9	E_CDM_OUTPUT_FNAME	16
5.1.2.10	E_CDM_READTYPE	16
5.1.3	関数	16
5.1.3.1	cdmPath_ConnectPath	16
5.1.3.2	cdmPath_DirName	16
5.1.3.3	cdmPath_FileName	16
5.1.3.4	cdmPath_getDelimChar	16
5.1.3.5	cdmPath_getDelimString	17
5.1.3.6	cdmPath_hasDrive	17

5.1.3.7	<code>cdmPath_isAbsolute</code>	17
5.1.3.8	<code>ExtractPathWithoutExt</code>	17
5.1.3.9	<code>vfvPath_emitDrive</code>	17
5.1.4	変数	17
5.1.4.1	<code>C_CDM_RANKNOPREFIX</code>	17
<b>6</b>	<b>クラス</b>	<b>19</b>
6.1	クラス <code>cdm_ActiveSubDomain</code>	19
6.1.1	説明	19
6.1.2	コンストラクタとデストラクタ	19
6.1.2.1	<code>cdm_ActiveSubDomain</code>	19
6.1.2.2	<code>cdm_ActiveSubDomain</code>	19
6.1.2.3	<code>~cdm_ActiveSubDomain</code>	20
6.1.3	関数	20
6.1.3.1	<code>clear</code>	20
6.1.3.2	<code>GetPos</code>	20
6.1.3.3	<code>operator!=</code>	20
6.1.3.4	<code>operator==</code>	20
6.1.3.5	<code>SetPos</code>	20
6.1.4	変数	21
6.1.4.1	<code>m_pos</code>	21
6.2	クラス <code>cdm_Array</code>	21
6.2.1	説明	24
6.2.2	コンストラクタとデストラクタ	24
6.2.2.1	<code>~cdm_Array</code>	24
6.2.2.2	<code>cdm_Array</code>	24
6.2.2.3	<code>cdm_Array</code>	24
6.2.3	関数	24
6.2.3.1	<code>_getArraySize</code>	24
6.2.3.2	<code>_getArraySizeInt</code>	24
6.2.3.3	<code>copyArray</code>	24
6.2.3.4	<code>copyArray</code>	24
6.2.3.5	<code>copyArrayNvari</code>	25
6.2.3.6	<code>copyArrayNvari</code>	25
6.2.3.7	<code>copyArrayNvari_to_ijk</code>	25
6.2.3.8	<code>copyArrayNvari_to_ijk</code>	25
6.2.3.9	<code>getArrayLength</code>	25
6.2.3.10	<code>getArrayShape</code>	25
6.2.3.11	<code>getArrayShapeString</code>	25
6.2.3.12	<code>getArraySize</code>	26

6.2.3.13	<a href="#">getArraySizeInt</a>	26
6.2.3.14	<a href="#">getData</a>	26
6.2.3.15	<a href="#">getDataType</a>	26
6.2.3.16	<a href="#">getDataTypeString</a>	26
6.2.3.17	<a href="#">getGc</a>	26
6.2.3.18	<a href="#">getGcInt</a>	26
6.2.3.19	<a href="#">getHeadIndex</a>	27
6.2.3.20	<a href="#">getNvari</a>	27
6.2.3.21	<a href="#">getNvariInt</a>	27
6.2.3.22	<a href="#">getTailIndex</a>	27
6.2.3.23	<a href="#">instanceArray</a>	27
6.2.3.24	<a href="#">instanceArray</a>	28
6.2.3.25	<a href="#">instanceArray</a>	28
6.2.3.26	<a href="#">instanceArray</a>	28
6.2.3.27	<a href="#">instanceArray</a>	28
6.2.3.28	<a href="#">instanceArray</a>	28
6.2.3.29	<a href="#">instanceArray</a>	28
6.2.3.30	<a href="#">instanceArray</a>	28
6.2.3.31	<a href="#">instanceArray</a>	28
6.2.3.32	<a href="#">instanceArray</a>	28
6.2.3.33	<a href="#">instanceArray</a>	29
6.2.3.34	<a href="#">instanceArray</a>	29
6.2.3.35	<a href="#">interp_coarse</a>	29
6.2.3.36	<a href="#">readBinary</a>	29
6.2.3.37	<a href="#">setHeadIndex</a>	29
6.2.3.38	<a href="#">writeAscii</a>	29
6.2.3.39	<a href="#">writeBinary</a>	29
6.2.4	<a href="#">変数</a>	29
6.2.4.1	<a href="#">m_dtype</a>	29
6.2.4.2	<a href="#">m_gc</a>	30
6.2.4.3	<a href="#">m_gcl</a>	30
6.2.4.4	<a href="#">m_gcl</a>	30
6.2.4.5	<a href="#">m_headIndex</a>	30
6.2.4.6	<a href="#">m_nvari</a>	30
6.2.4.7	<a href="#">m_nvaril</a>	30
6.2.4.8	<a href="#">m_shape</a>	30
6.2.4.9	<a href="#">m_sz</a>	30
6.2.4.10	<a href="#">m_Sz</a>	31
6.2.4.11	<a href="#">m_szl</a>	31
6.2.4.12	<a href="#">m_Szl</a>	31

6.2.4.13	m_tailIndex	31
6.3	クラス cdm_DFI	31
6.3.1	説明	39
6.3.2	コンストラクタとデストラクタ	39
6.3.2.1	cdm_DFI	39
6.3.2.2	~cdm_DFI	39
6.3.3	関数	39
6.3.3.1	AddTimeSlice	39
6.3.3.2	AddTimeSlice	39
6.3.3.3	AddUnit	39
6.3.3.4	cdm_Create_dfiProcessInfo	40
6.3.3.5	CheckAddWriteMode	40
6.3.3.6	CheckReadRank	40
6.3.3.7	CheckReadType	40
6.3.3.8	ConvDatatypeE2S	41
6.3.3.9	ConvDatatypeS2E	41
6.3.3.10	CreateReadStartEnd	41
6.3.3.11	Generate_DFI_Name	42
6.3.3.12	Generate_Directory_Path	42
6.3.3.13	Generate_FieldFileName	42
6.3.3.14	Generate_FileName	42
6.3.3.15	get_cdm_Datasize	43
6.3.3.16	get_dfi_fname	43
6.3.3.17	GetArrayShape	43
6.3.3.18	GetArrayShapeString	43
6.3.3.19	getBufSize	44
6.3.3.20	GetcdmDomain	44
6.3.3.21	GetcdmFieldFileNameFormat	44
6.3.3.22	GetcdmFileInfo	44
6.3.3.23	GetcdmFilePath	44
6.3.3.24	GetcdmMPI	44
6.3.3.25	GetcdmProcess	45
6.3.3.26	GetcdmRankNoPrefix	45
6.3.3.27	GetcdmTimeSlice	45
6.3.3.28	GetcdmUnit	45
6.3.3.29	GetcdmVisIt	45
6.3.3.30	GetDataType	45
6.3.3.31	GetDataTypeString	46
6.3.3.32	GetDFIGlobalDivision	46
6.3.3.33	GetDFIGlobalVoxel	46

6.3.3.34	GetDFIType	46
6.3.3.35	GetDFITypeString	46
6.3.3.36	GetFileFormat	46
6.3.3.37	GetFileFormatString	47
6.3.3.38	getFileNameFromFileList	47
6.3.3.39	getMinMax	47
6.3.3.40	GetNumGuideCell	47
6.3.3.41	GetNumVariables	47
6.3.3.42	GetUnit	47
6.3.3.43	GetUnitElem	48
6.3.3.44	getVariableName	48
6.3.3.45	getVectorMinMax	48
6.3.3.46	getVersionInfo	49
6.3.3.47	MakeDirectory	49
6.3.3.48	MakeDirectoryPath	49
6.3.3.49	MakeDirectorySub	49
6.3.3.50	normalizeBaseTime	49
6.3.3.51	normalizeDeltT	49
6.3.3.52	normalizeIntervalTime	50
6.3.3.53	normalizeLastTime	50
6.3.3.54	normalizeStartTime	50
6.3.3.55	normalizeTime	50
6.3.3.56	read_averaged	50
6.3.3.57	read_Datarecord	51
6.3.3.58	read_HeaderRecord	52
6.3.3.59	ReadCoordinateData	52
6.3.3.60	ReadCoordinateData	52
6.3.3.61	ReadCoordinateData	53
6.3.3.62	ReadCoordinateData	53
6.3.3.63	ReadCoordinateData	53
6.3.3.64	ReadData	55
6.3.3.65	ReadData	55
6.3.3.66	ReadData	55
6.3.3.67	ReadData	56
6.3.3.68	ReadData	56
6.3.3.69	ReadFieldData	57
6.3.3.70	ReadInit	57
6.3.3.71	set_input_type	58
6.3.3.72	set_interp_flag	58
6.3.3.73	set_output_fname	58

6.3.3.74	<a href="#">set_output_type</a>	58
6.3.3.75	<a href="#">set_output_type_coord</a>	58
6.3.3.76	<a href="#">set_RankID</a>	59
6.3.3.77	<a href="#">SetcdmDomain</a>	59
6.3.3.78	<a href="#">SetcdmFieldFileNameFormat</a>	59
6.3.3.79	<a href="#">SetcdmFileInfo</a>	59
6.3.3.80	<a href="#">SetcdmFilePath</a>	59
6.3.3.81	<a href="#">SetcdmMPI</a>	59
6.3.3.82	<a href="#">SetcdmProcess</a>	59
6.3.3.83	<a href="#">SetcdmRankNoPrefix</a>	59
6.3.3.84	<a href="#">SetcdmTimeSlice</a>	60
6.3.3.85	<a href="#">SetcdmUnit</a>	60
6.3.3.86	<a href="#">SetcdmVisIt</a>	60
6.3.3.87	<a href="#">setGridData</a>	60
6.3.3.88	<a href="#">setGridData</a>	60
6.3.3.89	<a href="#">setIntervalStep</a>	61
6.3.3.90	<a href="#">setIntervalTime</a>	61
6.3.3.91	<a href="#">SetTimeSliceFlag</a>	61
6.3.3.92	<a href="#">setVariableName</a>	61
6.3.3.93	<a href="#">VolumeDataDivide</a>	61
6.3.3.94	<a href="#">VolumeDataDivide</a>	62
6.3.3.95	<a href="#">write_ascii_header</a>	63
6.3.3.96	<a href="#">write_averaged</a>	63
6.3.3.97	<a href="#">write_DataRecord</a>	63
6.3.3.98	<a href="#">write_GridData</a>	63
6.3.3.99	<a href="#">write_HeaderRecord</a>	64
6.3.3.100	<a href="#">WriteCoordinateData</a>	64
6.3.3.101	<a href="#">WriteCoordinateData</a>	64
6.3.3.102	<a href="#">WriteCoordinateData</a>	64
6.3.3.103	<a href="#">WriteData</a>	65
6.3.3.104	<a href="#">WriteData</a>	65
6.3.3.105	<a href="#">WriteData</a>	65
6.3.3.106	<a href="#">WriteFieldData</a>	66
6.3.3.107	<a href="#">WriteFieldDataFile</a>	66
6.3.3.108	<a href="#">WriteFieldDataFile</a>	66
6.3.3.109	<a href="#">WriteFieldDataFile</a>	67
6.3.3.110	<a href="#">WriteGridFile</a>	67
6.3.3.111	<a href="#">WriteIndexDfiFile</a>	67
6.3.3.112	<a href="#">WriteIndexDfiFile</a>	67
6.3.3.113	<a href="#">WriteInit</a>	67



6.3.3.114	WriteInit	68
6.3.3.115	WriteInit	69
6.3.3.116	WriteInit	69
6.3.3.117	WriteInit	70
6.3.3.118	WriteInit	70
6.3.3.119	WriteProcDfiFile	70
6.3.4	変数	70
6.3.4.1	DFI_Domain	71
6.3.4.2	DFI_Finfo	71
6.3.4.3	DFI_Fpath	71
6.3.4.4	DFI_MPI	71
6.3.4.5	DFI_Process	71
6.3.4.6	DFI_TimeSlice	71
6.3.4.7	DFI_Unit	72
6.3.4.8	DFI_VisIt	72
6.3.4.9	m_bgrid_interp_flag	72
6.3.4.10	m_comm	72
6.3.4.11	m_directoryPath	72
6.3.4.12	m_indexDfiName	72
6.3.4.13	m_input_type	72
6.3.4.14	m_output_fname	73
6.3.4.15	m_output_type	73
6.3.4.16	m_output_type_coord	73
6.3.4.17	m_RankID	73
6.3.4.18	m_read_type	73
6.3.4.19	m_readRankList	73
6.4	クラス cdm_DFI_AVS	73
6.4.1	説明	74
6.4.2	コンストラクタとデストラクタ	74
6.4.2.1	cdm_DFI_AVS	74
6.4.2.2	cdm_DFI_AVS	75
6.4.2.3	~cdm_DFI_AVS	76
6.4.3	関数	76
6.4.3.1	read_averaged	76
6.4.3.2	read_Datarecord	76
6.4.3.3	read_HeaderRecord	77
6.4.3.4	write_ascii_header	77
6.4.3.5	write_averaged	77
6.4.3.6	write_avs_cord	78
6.4.3.7	write_avs_header	78

6.4.3.8	write_DataRecord	78
6.4.3.9	write_HeaderRecord	78
6.5	クラス cdm_DFI_BOV	79
6.5.1	説明	80
6.5.2	コンストラクタとデストラクタ	80
6.5.2.1	cdm_DFI_BOV	80
6.5.2.2	cdm_DFI_BOV	80
6.5.2.3	~cdm_DFI_BOV	80
6.5.3	関数	80
6.5.3.1	read_averaged	80
6.5.3.2	read_Datarecord	81
6.5.3.3	read_HeaderRecord	81
6.5.3.4	write_ascii_header	81
6.5.3.5	write_averaged	82
6.5.3.6	write_DataRecord	82
6.5.3.7	write_HeaderRecord	82
6.6	クラス cdm_DFI_FUB	83
6.6.1	説明	84
6.6.2	コンストラクタとデストラクタ	84
6.6.2.1	cdm_DFI_FUB	84
6.6.2.2	cdm_DFI_FUB	84
6.6.2.3	~cdm_DFI_FUB	84
6.6.3	関数	85
6.6.3.1	GetcdmFieldFileNameFormat	85
6.6.3.2	getCoordinateFileName	86
6.6.3.3	read_averaged	86
6.6.3.4	read_Datarecord	87
6.6.3.5	read_HeaderRecord	87
6.6.3.6	ReadCoordinateData	88
6.6.3.7	SetcdmFieldFileNameFormat	88
6.6.3.8	write_averaged	88
6.6.3.9	write_DataRecord	88
6.6.3.10	write_HeaderRecord	89
6.6.3.11	WriteCoordinateData	89
6.6.4	変数	89
6.6.4.1	DFI_FieldFileNameFormat	89
6.7	クラス cdm_DFI_NETCDF	89
6.7.1	説明	92
6.7.2	コンストラクタとデストラクタ	92
6.7.2.1	cdm_DFI_NETCDF	92

6.7.2.2	cdm_DFI_NETCDF	92
6.7.2.3	~cdm_DFI_NETCDF	92
6.7.3	関数	92
6.7.3.1	AddNcUnit	92
6.7.3.2	CheckAddWriteMode	93
6.7.3.3	CheckArrayVarInfo	93
6.7.3.4	GetCdmType	93
6.7.3.5	GetDimName	93
6.7.3.6	GetNcType	94
6.7.3.7	GetNcUnit	94
6.7.3.8	GetVarInfo	94
6.7.3.9	read_averaged	94
6.7.3.10	read_Datarecord	95
6.7.3.11	read_Datarecord	95
6.7.3.12	read_HeaderRecord	96
6.7.3.13	read_HeaderRecord	97
6.7.3.14	ReadAdditionalTP	97
6.7.3.15	SetDimName	97
6.7.3.16	SetWriteFlag	98
6.7.3.17	write_AttUnits	98
6.7.3.18	write_averaged	98
6.7.3.19	write_DataRecord	98
6.7.3.20	write_DataRecord	99
6.7.3.21	write_HeaderRecord	99
6.7.3.22	write_HeaderRecord	99
6.7.3.23	WriteAdditionalTP	100
6.7.4	変数	100
6.7.4.1	m_nameT	100
6.7.4.2	m_nameX	100
6.7.4.3	m_nameY	100
6.7.4.4	m_nameZ	100
6.7.4.5	m_NC_Unit	101
6.7.4.6	m_varInfoT	101
6.7.4.7	m_varInfoX	101
6.7.4.8	m_varInfoY	101
6.7.4.9	m_varInfoZ	101
6.7.4.10	m_vecVarInfo	101
6.7.4.11	m_writeFlag	101
6.8	クラス cdm_DFI_PLOT3D	101
6.8.1	説明	102

6.8.2	コンストラクタとデストラクタ	103
6.8.2.1	cdm_DFI_PLOT3D	103
6.8.2.2	cdm_DFI_PLOT3D	103
6.8.2.3	~cdm_DFI_PLOT3D	103
6.8.3	関数	103
6.8.3.1	read_averaged	103
6.8.3.2	read_Datarecord	104
6.8.3.3	read_Func	105
6.8.3.4	read_Func	105
6.8.3.5	read_HeaderRecord	105
6.8.3.6	write_averaged	106
6.8.3.7	write_DataRecord	106
6.8.3.8	write_Func	106
6.8.3.9	write_Func	107
6.8.3.10	write_GridData	107
6.8.3.11	write_HeaderRecord	107
6.8.3.12	write_XYZ	107
6.8.3.13	write_XYZ	107
6.9	クラス cdm_DFI_SPH	108
6.9.1	説明	109
6.9.2	列挙型	109
6.9.2.1	DataDims	109
6.9.2.2	RealType	109
6.9.3	コンストラクタとデストラクタ	109
6.9.3.1	cdm_DFI_SPH	109
6.9.3.2	cdm_DFI_SPH	109
6.9.3.3	~cdm_DFI_SPH	110
6.9.4	関数	110
6.9.4.1	read_averaged	110
6.9.4.2	read_Datarecord	110
6.9.4.3	read_HeaderRecord	111
6.9.4.4	write_averaged	112
6.9.4.5	write_DataRecord	112
6.9.4.6	write_HeaderRecord	112
6.10	クラス cdm_DFI_VTK	113
6.10.1	説明	114
6.10.2	コンストラクタとデストラクタ	114
6.10.2.1	cdm_DFI_VTK	114
6.10.2.2	cdm_DFI_VTK	114
6.10.2.3	~cdm_DFI_VTK	114

6.10.3	関数	114
6.10.3.1	read_averaged	114
6.10.3.2	read_Datarecord	115
6.10.3.3	read_HeaderRecord	115
6.10.3.4	write_averaged	116
6.10.3.5	write_DataRecord	116
6.10.3.6	write_HeaderRecord	116
6.11	クラス cdm_Domain	117
6.11.1	説明	118
6.11.2	コンストラクタとデストラクタ	118
6.11.2.1	cdm_Domain	118
6.11.2.2	cdm_Domain	118
6.11.2.3	cdm_Domain	118
6.11.2.4	~cdm_Domain	118
6.11.3	関数	118
6.11.3.1	CellX	118
6.11.3.2	CellY	119
6.11.3.3	CellZ	119
6.11.3.4	Clear	119
6.11.3.5	GetCoordinateFile	119
6.11.3.6	GetCoordinateFileEndian	119
6.11.3.7	GetCoordinateFilePrecision	120
6.11.3.8	GetCoordinateFileType	120
6.11.3.9	NodeX	120
6.11.3.10	NodeY	120
6.11.3.11	NodeZ	120
6.11.3.12	Read	121
6.11.3.13	Write	121
6.11.4	変数	121
6.11.4.1	ActiveSubdomainFile	121
6.11.4.2	GlobalDivision	121
6.11.4.3	GlobalOrigin	122
6.11.4.4	GlobalRegion	122
6.11.4.5	GlobalVoxel	122
6.11.4.6	Pitch	122
6.12	クラス cdm_FieldFileNameFormat	122
6.12.1	説明	123
6.12.2	コンストラクタとデストラクタ	123
6.12.2.1	cdm_FieldFileNameFormat	123
6.12.2.2	~cdm_FieldFileNameFormat	123

6.12.3	関数	123
6.12.3.1	AddFieldFileNameFormatElem	123
6.12.3.2	FileExist	123
6.12.3.3	GenerateFileName	123
6.12.3.4	GetFieldFileNameFormatElem	123
6.12.3.5	GetLabelList	123
6.12.3.6	Print	123
6.12.3.7	Read	123
6.12.3.8	Write	124
6.12.4	変数	124
6.12.4.1	LabelList	124
6.12.4.2	mapElem	124
6.13	クラス cdm_FieldFileNameFormatElem	124
6.13.1	説明	125
6.13.2	コンストラクタとデストラクタ	125
6.13.2.1	cdm_FieldFileNameFormatElem	125
6.13.2.2	~cdm_FieldFileNameFormatElem	125
6.13.3	関数	125
6.13.3.1	FileExist	125
6.13.3.2	GenerateFileName	125
6.13.3.3	GetDisitNo	125
6.13.3.4	PrintParse	125
6.13.3.5	Read	125
6.13.3.6	SetDisitNoFormat	125
6.13.3.7	SetFnameFormat	126
6.13.3.8	Write	126
6.13.4	変数	126
6.13.4.1	FileName	126
6.13.4.2	FileNameFormat	126
6.13.4.3	FnameFormat	126
6.13.4.4	FnameLabel	126
6.13.4.5	RankIdDisit	126
6.13.4.6	RankIdKey	126
6.13.4.7	StepNoDisit	126
6.13.4.8	StepNoKey	126
6.14	クラス cdm_FILE	127
6.14.1	説明	127
6.14.2	コンストラクタとデストラクタ	128
6.14.2.1	cdm_FILE	128
6.14.2.2	~cdm_FILE	128

6.14.3	関数	128
6.14.3.1	CloseFile	128
6.14.3.2	OpenReadBinary	128
6.14.3.3	OpenWriteAscii	128
6.14.3.4	OpenWriteBinary	129
6.14.4	変数	129
6.14.4.1	m_addMode	129
6.14.4.2	m_fname	129
6.14.4.3	m_format	129
6.14.4.4	m_fp	130
6.14.4.5	m_ncid	130
6.15	クラス cdm_FileInfo	130
6.15.1	説明	131
6.15.2	コンストラクタとデストラクタ	131
6.15.2.1	cdm_FileInfo	131
6.15.2.2	cdm_FileInfo	131
6.15.2.3	~cdm_FileInfo	132
6.15.3	関数	132
6.15.3.1	getVariableName	132
6.15.3.2	Read	132
6.15.3.3	setVariableName	132
6.15.3.4	Write	132
6.15.4	変数	133
6.15.4.1	ArrayShape	133
6.15.4.2	DataType	133
6.15.4.3	DFIType	133
6.15.4.4	DirectoryPath	133
6.15.4.5	Endian	133
6.15.4.6	FieldFilenameFormat	133
6.15.4.7	FileFormat	134
6.15.4.8	GuideCell	134
6.15.4.9	NumVariables	134
6.15.4.10	Prefix	134
6.15.4.11	RankNoPrefix	134
6.15.4.12	TimeSliceDirFlag	134
6.15.4.13	VariableName	134
6.16	クラス cdm_FilePath	135
6.16.1	説明	135
6.16.2	コンストラクタとデストラクタ	135
6.16.2.1	cdm_FilePath	135

6.16.2.2	cdm_FilePath	135
6.16.2.3	~cdm_FilePath	135
6.16.3	関数	135
6.16.3.1	Read	135
6.16.3.2	Write	136
6.16.4	変数	136
6.16.4.1	ProcDFIFile	136
6.17	クラス cdm_MPI	136
6.17.1	説明	137
6.17.2	コンストラクタとデストラクタ	137
6.17.2.1	cdm_MPI	137
6.17.2.2	cdm_MPI	137
6.17.2.3	~cdm_MPI	137
6.17.3	関数	137
6.17.3.1	Read	137
6.17.3.2	Write	137
6.17.4	変数	138
6.17.4.1	NumberOfGroup	138
6.17.4.2	NumberOfRank	138
6.18	クラス テンプレート cdm_NonUniformDomain< T >	138
6.18.1	説明	140
6.18.2	コンストラクタとデストラクタ	140
6.18.2.1	cdm_NonUniformDomain	140
6.18.2.2	cdm_NonUniformDomain	140
6.18.2.3	cdm_NonUniformDomain	141
6.18.2.4	~cdm_NonUniformDomain	141
6.18.3	関数	141
6.18.3.1	CellX	141
6.18.3.2	CellY	142
6.18.3.3	CellZ	142
6.18.3.4	Clear	142
6.18.3.5	GetCoordinateFile	143
6.18.3.6	GetCoordinateFileEndian	143
6.18.3.7	GetCoordinateFilePrecision	143
6.18.3.8	GetCoordinateFileType	143
6.18.3.9	NodeX	144
6.18.3.10	NodeY	145
6.18.3.11	NodeZ	145
6.18.3.12	Read	145
6.18.3.13	Read_CoordinateFile	146



6.18.3.14	readCoordData	146
6.18.3.15	readCoordDataCount	147
6.18.3.16	Write	147
6.18.4	変数	147
6.18.4.1	CoordinateFile	147
6.18.4.2	CoordinateFileEndian	148
6.18.4.3	CoordinateFilePrecision	148
6.18.4.4	CoordinateFileType	148
6.18.4.5	pit_gcXend	148
6.18.4.6	pit_gcXsta	148
6.18.4.7	pit_gcYend	148
6.18.4.8	pit_gcYsta	149
6.18.4.9	pit_gcZend	149
6.18.4.10	pit_gcZsta	149
6.18.4.11	XCoordinates	149
6.18.4.12	YCoordinates	149
6.18.4.13	ZCoordinates	149
6.19	クラス cdm_Process	150
6.19.1	説明	151
6.19.2	型定義	151
6.19.2.1	headT	151
6.19.3	コンストラクタとデストラクタ	151
6.19.3.1	cdm_Process	151
6.19.3.2	~cdm_Process	151
6.19.4	関数	151
6.19.4.1	CheckReadRank	151
6.19.4.2	CheckStartEnd	151
6.19.4.3	CreateHeadMap	152
6.19.4.4	CreateHeadMap	152
6.19.4.5	CreateRankList	152
6.19.4.6	CreateRankList	152
6.19.4.7	CreateRankMap	153
6.19.4.8	CreateRankMap	153
6.19.4.9	CreateSubDomainInfo	153
6.19.4.10	isMatchEndianSbdmMagick	154
6.19.4.11	Read	154
6.19.4.12	ReadActiveSubdomainFile	154
6.19.4.13	Write	154
6.19.5	変数	155
6.19.5.1	m_rankMap	155

6.19.5.2	RankList	155
6.20	クラス cdm_Rank	155
6.20.1	説明	155
6.20.2	コンストラクタとデストラクタ	156
6.20.2.1	cdm_Rank	156
6.20.2.2	~cdm_Rank	156
6.20.3	関数	156
6.20.3.1	Read	156
6.20.3.2	Write	156
6.20.4	変数	156
6.20.4.1	bc_id	156
6.20.4.2	c_id	156
6.20.4.3	HeadIndex	157
6.20.4.4	HostName	157
6.20.4.5	RankID	157
6.20.4.6	TailIndex	157
6.20.4.7	VoxelSize	157
6.21	クラス cdm_Slice	157
6.21.1	説明	158
6.21.2	コンストラクタとデストラクタ	158
6.21.2.1	cdm_Slice	158
6.21.2.2	~cdm_Slice	158
6.21.3	関数	158
6.21.3.1	Read	158
6.21.3.2	Write	158
6.21.4	変数	159
6.21.4.1	AveragedStep	159
6.21.4.2	AveragedTime	159
6.21.4.3	avr_mode	159
6.21.4.4	Max	159
6.21.4.5	Min	159
6.21.4.6	step	159
6.21.4.7	time	159
6.21.4.8	VectorMax	160
6.21.4.9	VectorMin	160
6.22	クラス cdm_TextParser	160
6.22.1	説明	161
6.22.2	コンストラクタとデストラクタ	161
6.22.2.1	cdm_TextParser	161
6.22.2.2	~cdm_TextParser	161

6.22.3	関数	161
6.22.3.1	chkLabel	161
6.22.3.2	chkNode	161
6.22.3.3	countLabels	161
6.22.3.4	GetNodeStr	162
6.22.3.5	getTPInstance	163
6.22.3.6	getTPPtr	163
6.22.3.7	GetValue	163
6.22.3.8	GetValue	163
6.22.3.9	GetValue	163
6.22.3.10	GetVector	163
6.22.3.11	GetVector	164
6.22.3.12	GetVector	164
6.22.3.13	readTPfile	164
6.22.3.14	remove	164
6.22.4	変数	164
6.22.4.1	tp	164
6.23	クラス cdm_TimeSlice	165
6.23.1	説明	165
6.23.2	コンストラクタとデストラクタ	165
6.23.2.1	cdm_TimeSlice	165
6.23.2.2	~cdm_TimeSlice	165
6.23.3	関数	165
6.23.3.1	AddSlice	165
6.23.3.2	getMinMax	166
6.23.3.3	getVectorMinMax	166
6.23.3.4	Read	166
6.23.3.5	Write	167
6.23.4	変数	168
6.23.4.1	SliceList	168
6.24	クラス テンプレート cdm_TypeArray< T >	168
6.24.1	説明	169
6.24.2	コンストラクタとデストラクタ	169
6.24.2.1	cdm_TypeArray	169
6.24.2.2	cdm_TypeArray	169
6.24.2.3	~cdm_TypeArray	170
6.24.2.4	cdm_TypeArray	170
6.24.3	関数	170
6.24.3.1	_val	170
6.24.3.2	_val	170

6.24.3.3	copyArray	170
6.24.3.4	copyArray	170
6.24.3.5	copyArrayNvari	170
6.24.3.6	copyArrayNvari	171
6.24.3.7	copyArrayNvari_to_ijk	171
6.24.3.8	copyArrayNvari_to_ijk	171
6.24.3.9	getData	171
6.24.3.10	hval	171
6.24.3.11	hval	172
6.24.3.12	readBinary	172
6.24.3.13	val	172
6.24.3.14	val	172
6.24.3.15	writeAscii	172
6.24.3.16	writeBinary	172
6.24.4	変数	172
6.24.4.1	m_data	172
6.24.4.2	m_outptr	172
6.25	クラス cdm_Unit	173
6.25.1	説明	173
6.25.2	コンストラクタとデストラクタ	173
6.25.2.1	cdm_Unit	173
6.25.2.2	~cdm_Unit	173
6.25.3	関数	174
6.25.3.1	AddUnit	174
6.25.3.2	GetUnit	175
6.25.3.3	GetUnitElem	175
6.25.3.4	Read	175
6.25.3.5	Write	175
6.25.4	変数	176
6.25.4.1	UnitList	176
6.26	クラス cdm_UnitElem	176
6.26.1	説明	176
6.26.2	コンストラクタとデストラクタ	177
6.26.2.1	cdm_UnitElem	177
6.26.2.2	cdm_UnitElem	177
6.26.2.3	~cdm_UnitElem	177
6.26.3	関数	177
6.26.3.1	Read	177
6.26.3.2	Write	177
6.26.4	変数	177

6.26.4.1	BsetDiff	177
6.26.4.2	difference	177
6.26.4.3	Name	178
6.26.4.4	reference	178
6.26.4.5	Unit	178
6.27	クラス cdm_VisIt	178
6.27.1	説明	178
6.27.2	コンストラクタとデストラクタ	178
6.27.2.1	cdm_VisIt	178
6.27.2.2	cdm_VisIt	179
6.27.2.3	~cdm_VisIt	180
6.27.3	関数	180
6.27.3.1	Read	180
6.27.3.2	Write	180
6.27.4	変数	180
6.27.4.1	PlotGC	180
6.28	構造体 cdm_DFI_NETCDF::stDimInfo	180
6.28.1	説明	181
6.28.2	変数	181
6.28.2.1	id	181
6.28.2.2	len	181
6.28.2.3	name	181
6.29	構造体 cdm_DFI_NETCDF::stVarInfo	181
6.29.1	説明	182
6.29.2	コンストラクタとデストラクタ	182
6.29.2.1	stVarInfo	182
6.29.2.2	~stVarInfo	182
6.29.3	関数	182
6.29.3.1	clear	182
6.29.4	変数	182
6.29.4.1	data	182
6.29.4.2	dims	182
6.29.4.3	endian	183
6.29.4.4	id	183
6.29.4.5	name	183
6.29.4.6	type	183
6.29.4.7	unit	183
7	ファイル	185
7.1	cdm_ActiveSubDomain.h	185

7.2	cdm_Array.h	185
7.2.1	関数	185
7.2.1.1	cdm_interp_ijkn_r4_	185
7.2.1.2	cdm_interp_ijkn_r8_	186
7.2.1.3	cdm_interp_nijk_r4_	186
7.2.1.4	cdm_interp_nijk_r8_	186
7.3	cdm_Array_inline.h	186
7.3.1	マクロ定義	186
7.3.1.1	CDM_INLINE	186
7.3.1.2	CDM_MEMFUN	186
7.4	cdm_Define.h	186
7.4.1	説明	189
7.4.2	マクロ定義	189
7.4.2.1	_CDM_IDX_IJ	189
7.4.2.2	_CDM_IDX_IJK	190
7.4.2.3	_CDM_IDX_IJKN	190
7.4.2.4	_CDM_IDX_NIJ	191
7.4.2.5	_CDM_IDX_NIJK	191
7.4.2.6	_CDM_TAB_STR	192
7.4.2.7	_CDM_WRITE_TAB	192
7.4.2.8	D_CDM_BIG	192
7.4.2.9	D_CDM_BYTE	192
7.4.2.10	D_CDM_DFITYPE_CARTESIAN	192
7.4.2.11	D_CDM_DFITYPE_NON_UNIFORM_CARTESIAN	192
7.4.2.12	D_CDM_DOUBLE	193
7.4.2.13	D_CDM_EXT_BOV	193
7.4.2.14	D_CDM_EXT_BOV_DATAFILE	193
7.4.2.15	D_CDM_EXT_FUB	193
7.4.2.16	D_CDM_EXT_FUNC	193
7.4.2.17	D_CDM_EXT_NC	193
7.4.2.18	D_CDM_EXT_SPH	193
7.4.2.19	D_CDM_EXT_VTK	193
7.4.2.20	D_CDM_EXT_XYZ	193
7.4.2.21	D_CDM_FLOAT	193
7.4.2.22	D_CDM_FLOAT32	193
7.4.2.23	D_CDM_FLOAT64	193
7.4.2.24	D_CDM_IJNK	194
7.4.2.25	D_CDM_INT	194
7.4.2.26	D_CDM_INT16	194
7.4.2.27	D_CDM_INT32	194

7.4.2.28	D_CDM_INT64	194
7.4.2.29	D_CDM_INT8	194
7.4.2.30	D_CDM_LITTLE	194
7.4.2.31	D_CDM_NIJK	194
7.4.2.32	D_CDM_OFF	194
7.4.2.33	D_CDM_ON	194
7.4.2.34	D_CDM_UINT16	194
7.4.2.35	D_CDM_UINT32	194
7.4.2.36	D_CDM_UINT64	195
7.4.2.37	D_CDM_UINT8	195
7.4.2.38	stmpd_printf	195
7.5	cdm_DFI.h	195
7.5.1	説明	195
7.6	cdm_DFI_AVS.h	196
7.6.1	説明	196
7.7	cdm_DFI_BOV.h	196
7.7.1	説明	196
7.8	cdm_DFI_FUB.h	197
7.8.1	説明	197
7.9	cdm_DFI_inline.h	197
7.9.1	マクロ定義	197
7.9.1.1	CDM_INLINE	197
7.10	cdm_DFI_NETCDF.h	197
7.10.1	説明	198
7.11	cdm_DFI_PLOT3D.h	198
7.11.1	説明	198
7.12	cdm_DFI_SPH.h	198
7.12.1	説明	199
7.13	cdm_DFI_VTK.h	199
7.13.1	説明	199
7.14	cdm_Domain.h	199
7.14.1	説明	199
7.15	cdm_endianUtil.h	200
7.15.1	説明	200
7.15.2	マクロ定義	200
7.15.2.1	BSWAP16	200
7.15.2.2	BSWAP32	201
7.15.2.3	BSWAP64	201
7.15.2.4	BSWAP_X_16	201
7.15.2.5	BSWAP_X_32	201

7.15.2.6	BSWAP_X_64	201
7.15.2.7	BSWAPVEC	201
7.15.2.8	CDM_INLINE	202
7.15.2.9	DBSWAPVEC	202
7.15.2.10	SBSWAPVEC	202
7.16	cdm_FieldFileNameFormat.h	202
7.16.1	マクロ定義	202
7.16.1.1	_CDM_FIELDFILENAMEFORMAT_H_	202
7.17	cdm_FieldFileNameFormatElem.h	203
7.17.1	マクロ定義	203
7.17.1.1	_FUB_TAB_STR	203
7.17.1.2	_FUB_WRITE_TAB	203
7.17.2	列挙型	203
7.17.2.1	FUB_FNAME_TYPE	203
7.18	cdm_FILE.h	204
7.18.1	説明	204
7.19	cdm_FileInfo.h	204
7.19.1	説明	204
7.20	cdm_FilePath.h	204
7.20.1	説明	205
7.21	cdm_interp_ijkn.h	205
7.21.1	関数	205
7.21.1.1	!Copyright	205
7.22	cdm_interp_nijk.h	205
7.22.1	関数	205
7.22.1.1	!Copyright	205
7.23	cdm_MPI.h	205
7.23.1	説明	205
7.24	cdm_NonUniformDomain.h	206
7.24.1	説明	206
7.25	cdm_NonUniformDomain_inline.h	206
7.25.1	説明	206
7.25.2	マクロ定義	207
7.25.2.1	CDM_INLINE	207
7.26	cdm_PathUtil.h	207
7.26.1	マクロ定義	207
7.26.1.1	MAXPATHLEN	207
7.27	cdm_Plot3d_inline.h	207
7.27.1	マクロ定義	208
7.27.1.1	CDM_INLINE	208



7.28	cdm_Process.h	208
7.28.1	説明	208
7.29	cdm_TextParser.h	208
7.29.1	説明	208
7.30	cdm_TimeSlice.h	209
7.30.1	説明	209
7.31	cdm_TypeArray.h	209
7.32	cdm_Unit.h	209
7.32.1	説明	210
7.33	cdm_Version.h	210
7.33.1	説明	210
7.33.2	マクロ定義	210
7.33.2.1	CDM_REVISION	210
7.33.2.2	CDM_VERSION_NO	210
7.34	cdm_VisIt.h	210
7.34.1	説明	211
7.35	mpi_stubs.h	211
7.35.1	マクロ定義	211
7.35.1.1	MPI_CHAR	211
7.35.1.2	MPI_COMM_WORLD	211
7.35.1.3	MPI_INT	211
7.35.1.4	MPI_SUCCESS	211
7.35.2	型定義	212
7.35.2.1	MPI_Comm	212
7.35.2.2	MPI_Datatype	212
7.35.3	関数	212
7.35.3.1	MPI_Allgather	212
7.35.3.2	MPI_Comm_rank	212
7.35.3.3	MPI_Comm_size	212
7.35.3.4	MPI_Gather	212
7.35.3.5	MPI_Init	212



## Chapter 1

# ネームスペース索引

### 1.1 ネームスペース一覧

ネームスペースの一覧です。

CDM	.....	??
-----	-------	----



## Chapter 2

# 階層索引

### 2.1 クラス階層

この継承一覧はおおまかにはソートされていますが、完全にアルファベット順でソートされてはいません。

cdm_ActiveSubDomain . . . . .	??
cdm_Array . . . . .	??
cdm_TypeArray< T > . . . . .	??
cdm_DFI . . . . .	??
cdm_DFI_AVS . . . . .	??
cdm_DFI_BOV . . . . .	??
cdm_DFI_FUB . . . . .	??
cdm_DFI_NETCDF . . . . .	??
cdm_DFI_PLOT3D . . . . .	??
cdm_DFI_SPH . . . . .	??
cdm_DFI_VTK . . . . .	??
cdm_Domain . . . . .	??
cdm_NonUniformDomain< T > . . . . .	??
cdm_FieldFileNameFormat . . . . .	??
cdm_FieldFileNameFormatElem . . . . .	??
cdm_FILE . . . . .	??
cdm_FileInfo . . . . .	??
cdm_FilePath . . . . .	??
cdm_MPI . . . . .	??
cdm_Process . . . . .	??
cdm_Rank . . . . .	??
cdm_Slice . . . . .	??
cdm_TextParser . . . . .	??
cdm_TimeSlice . . . . .	??
cdm_Unit . . . . .	??
cdm_UnitElem . . . . .	??
cdm_VisIt . . . . .	??
cdm_DFI_NETCDF::stDimInfo . . . . .	??
cdm_DFI_NETCDF::stVarInfo . . . . .	??



## Chapter 3

# 構成索引

### 3.1 構成

クラス、構造体、共用体、インタフェースの説明です。

cdm_ActiveSubDomain	??
cdm_Array	??
cdm_DFI	??
cdm_DFI_AVS	??
cdm_DFI_BOV	??
cdm_DFI_FUB	??
cdm_DFI_NETCDF	??
cdm_DFI_PLOT3D	??
cdm_DFI_SPH	??
cdm_DFI_VTK	??
cdm_Domain	??
cdm_FieldFileNameFormat	??
cdm_FieldFileNameFormatElem	??
cdm_FILE	??
cdm_FileInfo	??
cdm_FilePath	??
cdm_MPI	??
cdm_NonUniformDomain< T >	??
cdm_Process	??
cdm_Rank	??
cdm_Slice	??
cdm_TextParser	??
cdm_TimeSlice	??
cdm_TypeArray< T >	??
cdm_Unit	??
cdm_UnitElem	??
cdm_VisIt	??
cdm_DFI_NETCDF::stDimInfo	??
cdm_DFI_NETCDF::stVarInfo	??





## Chapter 4

# ファイル索引

### 4.1 ファイル一覧

これはファイル一覧です。

cdm_ActiveSubDomain.h	??
cdm_Array.h	??
cdm_Array_inline.h	??
cdm_Define.h	
CDM の定義マクロ記述ヘッダーファイル	??
cdm_DFI.h	
Cdm_DFI Class Header	??
cdm_DFI_AVS.h	
Cdm_DFI_AVS Class Header	??
cdm_DFI_BOV.h	
Cdm_DFI_BOV Class Header	??
cdm_DFI_FUB.h	
Cdm_DFI_FUB Class Header	??
cdm_DFI_inline.h	??
cdm_DFI_NETCDF.h	
Cdm_DFI_NETCDF Class Header	??
cdm_DFI_PLOT3D.h	
Cdm_DFI_PLOT3D Class Header	??
cdm_DFI_SPH.h	
Cdm_DFI_SPH Class Header	??
cdm_DFI_VTK.h	
Cdm_DFI_VTK Class Header	??
cdm_Domain.h	
Cdm_Domain Class Header	??
cdm_endianUtil.h	
エンディアンユーティリティマクロ・関数ファイル	??
cdm_FieldFileNameFormat.h	??
cdm_FieldFileNameFormatElem.h	??
cdm_FILE.h	
Cdm_FILE Class Header	??
cdm_FileInfo.h	
Cdm_FileInfo Class Header	??
cdm_FilePath.h	
Cdm_FilePath Class Header	??
cdm_interp_ijkn.h	??
cdm_interp_nijk.h	??
cdm_MPI.h	
Cdm_MPI Class Header	??

<a href="#">cdm_NonUniformDomain.h</a>	
Cdm_NonUniformDomain Class Header . . . . .	??
<a href="#">cdm_NonUniformDomain_inline.h</a>	
Cdm_NonUniformDomain template Header . . . . .	??
<a href="#">cdm_PathUtil.h</a> . . . . .	??
<a href="#">cdm_Plot3d_inline.h</a> . . . . .	??
<a href="#">cdm_Process.h</a>	
Cdm_RANK & <a href="#">cdm_Process</a> Class Header . . . . .	??
<a href="#">cdm_TextParser.h</a>	
TextParser Control class Header . . . . .	??
<a href="#">cdm_TimeSlice.h</a>	
Cdm_Slice & cdm_TimeSliceClass Header . . . . .	??
<a href="#">cdm_TypeArray.h</a> . . . . .	??
<a href="#">cdm_Unit.h</a>	
Cdm_UnitElem & <a href="#">cdm_Unit</a> Class Header . . . . .	??
<a href="#">cdm_Version.h</a> . . . . .	??
<a href="#">cdm_VisIt.h</a>	
Cdm_VisIt Class Header . . . . .	??
<a href="#">mpi_stubs.h</a> . . . . .	??

## Chapter 5

# ネームスペース

### 5.1 ネームスペース CDM

#### 列挙型

- enum `E_CDM_DFITYPE` { `E_CDM_DFITYPE_UNKNOWN` = -1, `E_CDM_DFITYPE_CARTESIAN`, `E_CDM_DFITYPE_NON_UNIFORM_CARTESIAN` }
- enum `E_CDM_FORMAT` { `E_CDM_FMT_UNKNOWN` = -1, `E_CDM_FMT_SPH`, `E_CDM_FMT_BOV`, `E_CDM_FMT_AVIS`, `E_CDM_FMT_PLOT3D`, `E_CDM_FMT_VTK`, `E_CDM_FMT_NETCDF4`, `E_CDM_FMT_FUB`, `E_CDM_FMT_FUB_COD` }
- enum `E_CDM_ONOFF` { `E_CDM_OFF` = 0, `E_CDM_ON` }
- enum `E_CDM_DTYPE` { `E_CDM_DTYPE_UNKNOWN` = 0, `E_CDM_INT8`, `E_CDM_INT16`, `E_CDM_INT32`, `E_CDM_INT64`, `E_CDM_UINT8`, `E_CDM_UINT16`, `E_CDM_UINT32`, `E_CDM_UINT64`, `E_CDM_FLOAT32`, `E_CDM_FLOAT64` }
- enum `E_CDM_ARRAYSHAPE` { `E_CDM_ARRAYSHAPE_UNKNOWN` = -1, `E_CDM_IJKN` = 0, `E_CDM_NIJK` }
- enum `E_CDM_ENDIANTYPE` { `E_CDM_ENDIANTYPE_UNKNOWN` = -1, `E_CDM_LITTLE` = 0, `E_CDM_BIG` }
- enum `E_CDM_READTYPE` { `E_CDM_SAMEDIV_SAMERES` = 1, `E_CDM_SAMEDIV_REFINEMENT`, `E_CDM_DIFFDIV_SAMERES`, `E_CDM_DIFFDIV_REFINEMENT`, `E_CDM_READTYPE_UNKNOWN` }
- enum `E_CDM_FILE_TYPE` { `E_CDM_FILE_TYPE_DEFAULT` = -1, `E_CDM_FILE_TYPE_ASCII` = 0, `E_CDM_FILE_TYPE_BINARY`, `E_CDM_FILE_TYPE_FBINAR` }
- enum `E_CDM_OUTPUT_FNAME` { `E_CDM_FNAME_DEFAULT` = -1, `E_CDM_FNAME_STEP_RANK` = 0, `E_CDM_FNAME_RANK_STEP`, `E_CDM_FNAME_RANK`, `E_CDM_FNAME_CUSTOM` }
- enum `E_CDM_ERRORCODE` { `E_CDM_SUCCESS` = 1, `E_CDM_ERROR` = -1, `E_CDM_ERROR_READ_DFI_GLOBALORIGIN` = 1000, `E_CDM_ERROR_READ_DFI_GLOBALREGION` = 1001, `E_CDM_ERROR_READ_DFI_GLOBALVOXEL` = 1002, `E_CDM_ERROR_READ_DFI_GLOBALDIVISION` = 1003, `E_CDM_ERROR_READ_DFI_DIRECTORYPATH` = 1004, `E_CDM_ERROR_READ_DFI_TIMESLICEDIRECTORY` = 1005, `E_CDM_ERROR_READ_DFI_PREFIX` = 1006, `E_CDM_ERROR_READ_DFI_FILEFORMAT` = 1007, `E_CDM_ERROR_READ_DFI_GUIDECCELL` = 1008, `E_CDM_ERROR_READ_DFI_DATATYPE` = 1009, `E_CDM_ERROR_READ_DFI_ENDIAN` = 1010, `E_CDM_ERROR_READ_DFI_NUMVARIABLES` = 1012, `E_CDM_ERROR_READ_DFI_FILEPATH_PROCESS` = 1013, `E_CDM_ERROR_READ_DFI_NO_RANK` = 1014, `E_CDM_ERROR_READ_DFI_ID` = 1015, `E_CDM_ERROR_READ_DFI_HOSTNAME` = 1016, `E_CDM_ERROR_READ_DFI_V`

```

= 1017, E_CDM_ERROR_READ_DFI_HEADINDEX = 1018,
E_CDM_ERROR_READ_DFI_TAILINDEX = 1019, E_CDM_ERROR_READ_DFI_CELLID = 1020,
E_CDM_ERROR_READ_DFI_BCFLAGID = 1021, E_CDM_ERROR_READ_DFI_NO_SLICE = 1022,
E_CDM_ERROR_READ_DFI_STEP = 1023, E_CDM_ERROR_READ_DFI_TIME = 1024, E_CDM_ERROR_READ_DFI_NO_
= 1025, E_CDM_ERROR_READ_DFI_MIN = 1026,
E_CDM_ERROR_READ_DFI_MAX = 1027, E_CDM_ERROR_READ_DFI_DFITYPE = 1028, E_CDM_ERROR_READ_DFI_F
= 1029, E_CDM_ERROR_READ_DFI_COORDINATEFILE = 1030,
E_CDM_ERROR_READ_DFI_COORDINATEFILETYPE = 1031, E_CDM_ERROR_READ_DFI_COORDINATEFILEPRECISIO
= 1032, E_CDM_ERROR_READ_DFI_COORDINATEFILEENDIAN = 1033, E_CDM_ERROR_OPEN_COORDINATEFILE
= 1034,
E_CDM_ERROR_READ_COORDINATEFILE = 1035, E_CDM_ERROR_READ_INDEXFILE_OPENERROR
= 1050, E_CDM_ERROR_TEXTPARSER = 1051, E_CDM_ERROR_READ_FILEINFO = 1052,
E_CDM_ERROR_READ_FILEPATH = 1053, E_CDM_ERROR_READ_UNIT = 1054, E_CDM_ERROR_READ_TIMESLICE
= 1055, E_CDM_ERROR_READ_PROCFILE_OPENERROR = 1056,
E_CDM_ERROR_READ_DOMAIN = 1057, E_CDM_ERROR_READ_MPI = 1058, E_CDM_ERROR_READ_PROCESS
= 1059, E_CDM_ERROR_READ_DFI_NETCDF = 1060,
E_CDM_ERROR_READ_NETCDF_MISMATCH_TYPE = 1061, E_CDM_ERROR_READ_FILELIST_ID =
1062, E_CDM_ERROR_READ_FILELIST_CCORDINATEFILENAME = 1063, E_CDM_ERROR_READ_FILELIST_FIELDDAT
= 1064,
E_CDM_ERROR_UNDEFINED_FIELDFILENAMEFORMAT = 1065, E_CDM_ERROR_READ_FIELDDATA_FILE
= 1900, E_CDM_ERROR_READ_SPH_FILE = 2000, E_CDM_ERROR_READ_SPH_REC1 = 2001,
E_CDM_ERROR_READ_SPH_REC2 = 2002, E_CDM_ERROR_READ_SPH_REC3 = 2003, E_CDM_ERROR_READ_SPH_F
= 2004, E_CDM_ERROR_READ_SPH_REC5 = 2005,
E_CDM_ERROR_READ_SPH_REC6 = 2006, E_CDM_ERROR_READ_SPH_REC7 = 2007, E_CDM_ERROR_UNMATCH_V
= 2050, E_CDM_ERROR_NOMATCH_ENDIAN = 2051,
E_CDM_ERROR_UNMATCH_NUM_OF_VARIABLES = 2052, E_CDM_ERROR_NUM_OF_GUIDECCELLS
= 2053, E_CDM_ERROR_READ_BOV_FILE = 2100, E_CDM_ERROR_READ_FIELD_HEADER_RECORD
= 2102,
E_CDM_ERROR_READ_FIELD_DATA_RECORD = 2103, E_CDM_ERROR_READ_FIELD_AVERAGED_RECORD
= 2104, E_CDM_ERROR_READ_DFI_FILELIST = 2105, E_CDM_ERROR_READ_NETCDF_FUNC = 2200,
E_CDM_ERROR_READ_NETCDF_VAR_1D = 2201, E_CDM_ERROR_READ_FUB_REC1 = 2210,
E_CDM_ERROR_READ_FUB_REC2 = 2211, E_CDM_ERROR_MISMATCH_NP_SUBDOMAIN = 3003,
E_CDM_ERROR_INVALID_DIVNUM = 3011, E_CDM_ERROR_OPEN_SBDM = 3012, E_CDM_ERROR_READ_SBDM_HEA
= 3013, E_CDM_ERROR_READ_SBDM_FORMAT = 3014,
E_CDM_ERROR_READ_SBDM_DIV = 3015, E_CDM_ERROR_READ_SBDM_CONTENTS = 3016,
E_CDM_ERROR_SBDM_NUMDOMAIN_ZERO = 3017, E_CDM_ERROR_MAKEDIRECTORY = 3100,
E_CDM_ERROR_OPEN_FIELDDATA = 3101, E_CDM_ERROR_WRITE_FIELD_HEADER_RECORD =
3102, E_CDM_ERROR_WRITE_FIELD_DATA_RECORD = 3103, E_CDM_ERROR_WRITE_FIELD_AVERAGED_RECORD
= 3104,
E_CDM_ERROR_WRITE_SPH_REC1 = 3201, E_CDM_ERROR_WRITE_SPH_REC2 = 3202, E_CDM_ERROR_WRITE_SPH
= 3203, E_CDM_ERROR_WRITE_SPH_REC4 = 3204,
E_CDM_ERROR_WRITE_SPH_REC5 = 3205, E_CDM_ERROR_WRITE_SPH_REC6 = 3206, E_CDM_ERROR_WRITE_SPH
= 3207, E_CDM_ERROR_WRITE_PROCFILENAME_EMPTY = 3500,
E_CDM_ERROR_WRITE_PROCFILE_OPENERROR = 3501, E_CDM_ERROR_WRITE_DOMAIN = 3502,
E_CDM_ERROR_WRITE_MPI = 3503, E_CDM_ERROR_WRITE_PROCESS = 3504,
E_CDM_ERROR_WRITE_RANKID = 3505, E_CDM_ERROR_WRITE_INDEXFILENAME_EMPTY = 3510,
E_CDM_ERROR_WRITE_PREFIX_EMPTY = 3511, E_CDM_ERROR_WRITE_INDEXFILE_OPENERROR
= 3512,
E_CDM_ERROR_WRITE_FILEINFO = 3513, E_CDM_ERROR_WRITE_UNIT = 3514, E_CDM_ERROR_WRITE_TIMESLICE
= 3515, E_CDM_ERROR_WRITE_FILEPATH = 3516,
E_CDM_ERROR_WRITE_VISIT = 3517, E_CDM_ERROR_WRITE_GRIDFILE = 3518, E_CDM_ERROR_WRITE_DFI_NETC
= 3600, E_CDM_WARN_GETUNIT = 4000 }

```

## 関数

- char `cdmPath_getDelimChar` ()
- std::string `cdmPath_getDelimString` ()
- bool `cdmPath_hasDrive` (const std::string &path)

- std::string [vfvPath\\_emitDrive](#) (std::string &path)
- bool [cdmPath\\_isAbsolute](#) (const std::string &path)
- std::string [cdmPath\\_DirName](#) (const std::string &path, const char dc=[cdmPath\\_getDelimChar](#)())
- std::string [cdmPath\\_FileName](#) (const std::string &path, const std::string &addext=std::string(""), const char dc=[cdmPath\\_getDelimChar](#)())
- std::string [cdmPath\\_ConnectPath](#) (std::string dirName, std::string fName)
- std::string [ExtractPathWithoutExt](#) (const std::string &fn)

## 変数

- static const char \* [C\\_CDM\\_RANKNOPREFIX](#) = "\_id"  
ファイル名のランク番号前の文字列

### 5.1.1 説明

namespace の設定

### 5.1.2 列挙型

#### 5.1.2.1 enum CDM::E\_CDM\_ARRAYSHAPE

配列形式

列挙型の値

- E\_CDM\_ARRAYSHAPE\_UNKNOWN*** 未定
- E\_CDM\_IJKN*** ijkn
- E\_CDM\_NIJK*** nijk

cdm\_Define.h の 126 行で定義されています。

#### 5.1.2.2 enum CDM::E\_CDM\_DFITYPE

列挙型の値

- E\_CDM\_DFITYPE\_UNKNOWN*** 未定
- E\_CDM\_DFITYPE\_CARTESIAN*** Cartesian.
- E\_CDM\_DFITYPE\_NON\_UNIFORM\_CARTESIAN*** Non Uniform Cartesian.

cdm\_Define.h の 76 行で定義されています。

#### 5.1.2.3 enum CDM::E\_CDM\_DTYPE

データ形式

列挙型の値

- E\_CDM\_DTYPE\_UNKNOWN*** 未定
- E\_CDM\_INT8*** char
- E\_CDM\_INT16*** short
- E\_CDM\_INT32*** int
- E\_CDM\_INT64*** long long

***E\_CDM\_UINT8*** unsigned char  
***E\_CDM\_UINT16*** unsigned short  
***E\_CDM\_UINT32*** unsigned int  
***E\_CDM\_UINT64*** unsigned long long  
***E\_CDM\_FLOAT32*** float  
***E\_CDM\_FLOAT64*** double

cdm\_Define.h の 110 行で定義されています。

#### 5.1.2.4 enum CDM::E\_CDM\_ENDIANTYPE

Endian 形式

列挙型の値

***E\_CDM\_ENDIANTYPE\_UNKNOWN***  
***E\_CDM\_LITTLE***  
***E\_CDM\_BIG***

cdm\_Define.h の 134 行で定義されています。

#### 5.1.2.5 enum CDM::E\_CDM\_ERRORCODE

CDM のエラーコード

列挙型の値

***E\_CDM\_SUCCESS*** 正常終了  
***E\_CDM\_ERROR*** エラー終了  
***E\_CDM\_ERROR\_READ\_DFI\_GLOBALORIGIN*** DFI GlobalOrigin 読み込みエラー  
***E\_CDM\_ERROR\_READ\_DFI\_GLOBALREGION*** DFI GlobalRegion 読み込みエラー  
***E\_CDM\_ERROR\_READ\_DFI\_GLOBALVOXEL*** DFI GlobalVoxel 読み込みエラー  
***E\_CDM\_ERROR\_READ\_DFI\_GLOBALDIVISION*** DFI GlobalDivision 読み込みエラー  
***E\_CDM\_ERROR\_READ\_DFI\_DIRECTORYPATH*** DFI DirectoryPath 読み込みエラー  
***E\_CDM\_ERROR\_READ\_DFI\_TIMESLICEDIRECTORY*** DFI TimeSliceDirectoryPath 読み込みエラー  
***E\_CDM\_ERROR\_READ\_DFI\_PREFIX*** DFI Prefix 読み込みエラー  
***E\_CDM\_ERROR\_READ\_DFI\_FILEFORMAT*** DFI FileFormat 読み込みエラー  
***E\_CDM\_ERROR\_READ\_DFI\_GUIDECELL*** DFI GuideCell 読み込みエラー  
***E\_CDM\_ERROR\_READ\_DFI\_DATATYPE*** DFI DataType 読み込みエラー  
***E\_CDM\_ERROR\_READ\_DFI\_ENDIAN*** DFI Endian 読み込みエラー  
***E\_CDM\_ERROR\_READ\_DFI\_NUMVARIABLES*** DFI NumVariables 読み込みエラー  
***E\_CDM\_ERROR\_READ\_DFI\_FILEPATH\_PROCESS*** DFI FilePath/Process 読み込みエラー  
***E\_CDM\_ERROR\_READ\_DFI\_NO\_RANK*** DFI Rank 要素なし  
***E\_CDM\_ERROR\_READ\_DFI\_ID*** DFI ID 読み込みエラー  
***E\_CDM\_ERROR\_READ\_DFI\_HOSTNAME*** DFI HostName 読み込みエラー  
***E\_CDM\_ERROR\_READ\_DFI\_VOXELSIZE*** DFI VoxelSize 読み込みエラー  
***E\_CDM\_ERROR\_READ\_DFI\_HEADINDEX*** DFI HeadIndex 読み込みエラー  
***E\_CDM\_ERROR\_READ\_DFI\_TAILINDEX*** DFI TailIndex 読み込みエラー  
***E\_CDM\_ERROR\_READ\_DFI\_CELLID*** DFI CellID 読み込みエラー

***E\_CDM\_ERROR\_READ\_DFI\_BCFLAGID*** DFI BCflagID 読み込みエラー  
***E\_CDM\_ERROR\_READ\_DFI\_NO\_SLICE*** DFI TimeSlice 要素なし  
***E\_CDM\_ERROR\_READ\_DFI\_STEP*** DFI Step 読み込みエラー  
***E\_CDM\_ERROR\_READ\_DFI\_TIME*** DFI Time 読み込みエラー  
***E\_CDM\_ERROR\_READ\_DFI\_NO\_MINMAX*** DFI MinMax 要素なし  
***E\_CDM\_ERROR\_READ\_DFI\_MIN*** DFI Min 読み込みエラー  
***E\_CDM\_ERROR\_READ\_DFI\_MAX*** DFI Max 読み込みエラー  
***E\_CDM\_ERROR\_READ\_DFI\_DFITYPE*** DFI DFIType 読み込みエラー  
***E\_CDM\_ERROR\_READ\_DFI\_FIELDFILENAMEFORMAT*** DFI FieldfilenameFormat 読み込みエラー  
***E\_CDM\_ERROR\_READ\_DFI\_COORDINATEFILE*** DFI Coordinate File 読み込みエラー  
***E\_CDM\_ERROR\_READ\_DFI\_COORDINATEFILETYPE*** DFI Coordinate File Type 読み込みエラー  
***E\_CDM\_ERROR\_READ\_DFI\_COORDINATEFILEPRECISION*** DFI Coordinate File Precision 読み込みエラー  
  
***E\_CDM\_ERROR\_READ\_DFI\_COORDINATEFILEENDIAN*** DFI Coordinate File Endian 読み込みエラー  
***E\_CDM\_ERROR\_OPEN\_COORDINATEFILE*** Coordinate File オープンに失敗  
***E\_CDM\_ERROR\_READ\_COORDINATEFILE*** Coordinate File 読み込みエラー  
***E\_CDM\_ERROR\_READ\_INDEXFILE\_OPENERROR*** Index ファイルオープンエラー  
***E\_CDM\_ERROR\_TEXTPARSER*** TextParser エラー  
***E\_CDM\_ERROR\_READ\_FILEINFO*** FileInfo 読み込みエラー  
***E\_CDM\_ERROR\_READ\_FILEPATH*** FilePath 読み込みエラー  
***E\_CDM\_ERROR\_READ\_UNIT*** UNIT 読み込みエラー  
***E\_CDM\_ERROR\_READ\_TIMESLICE*** TimeSlice 読み込みエラー  
***E\_CDM\_ERROR\_READ\_PROCFILE\_OPENERROR*** Proc ファイルオープンエラー  
***E\_CDM\_ERROR\_READ\_DOMAIN*** Domain 読み込みエラー  
***E\_CDM\_ERROR\_READ\_MPI*** MPI 読み込みエラー  
***E\_CDM\_ERROR\_READ\_PROCESS*** Process 読み込みエラー  
***E\_CDM\_ERROR\_READ\_DFI\_NETCDF*** NetCDF 読み込みエラー  
***E\_CDM\_ERROR\_READ\_NETCDF\_MISMATCH\_TYPE*** DFI とNetCDF のデータ型の不一致エラー  
***E\_CDM\_ERROR\_READ\_FILELIST\_ID*** FileList のID 読み込みエラー  
***E\_CDM\_ERROR\_READ\_FILELIST\_CCORDINATEFILENAME*** FileList のCoordinateFileName 読み込みエラー  
***E\_CDM\_ERROR\_READ\_FILELIST\_FIELDDATAFILENAME*** FileList のFieldDataFileName 読み込みエラー  
  
***E\_CDM\_ERROR\_UNDEFINED\_FIELDFILENAMEFORMAT*** FieldFileNameFormat がない  
***E\_CDM\_ERROR\_READ\_FIELDDATA\_FILE*** フィールドデータファイル読み込みエラー  
***E\_CDM\_ERROR\_READ\_SPH\_FILE*** SPH ファイル読み込みエラー  
***E\_CDM\_ERROR\_READ\_SPH\_REC1*** SPH ファイルレコード 1 読み込みエラー  
***E\_CDM\_ERROR\_READ\_SPH\_REC2*** SPH ファイルレコード 2 読み込みエラー  
***E\_CDM\_ERROR\_READ\_SPH\_REC3*** SPH ファイルレコード 3 読み込みエラー  
***E\_CDM\_ERROR\_READ\_SPH\_REC4*** SPH ファイルレコード 4 読み込みエラー  
***E\_CDM\_ERROR\_READ\_SPH\_REC5*** SPH ファイルレコード 5 読み込みエラー  
***E\_CDM\_ERROR\_READ\_SPH\_REC6*** SPH ファイルレコード 6 読み込みエラー  
***E\_CDM\_ERROR\_READ\_SPH\_REC7*** SPH ファイルレコード 7 読み込みエラー  
***E\_CDM\_ERROR\_UNMATCH\_VOXELSIZE*** SPH のボクセルサイズとDFI のボクセルサイズが合致しない  
  
***E\_CDM\_ERROR\_NOMATCH\_ENDIAN*** 出力Format が合致しない (Endian 形式がBig,Little 以外)

**E\_CDM\_ERROR\_UNMATCH\_NUM\_OF\_VARIABLES** フィールドデータの変数の個数と登録された変数名の個数が合致しない

**E\_CDM\_ERROR\_NUM\_OF\_GUIDECCELLS** フィールドデータのガイドセル値エラー

**E\_CDM\_ERROR\_READ\_BOV\_FILE** BOV ファイル読み込みエラー

**E\_CDM\_ERROR\_READ\_FIELD\_HEADER\_RECORD** フィールドヘッダーレコード読み込み失敗

**E\_CDM\_ERROR\_READ\_FIELD\_DATA\_RECORD** フィールドデータレコード読み込み失敗

**E\_CDM\_ERROR\_READ\_FIELD\_AVERAGED\_RECORD** フィールドAverage 読み込み失敗

**E\_CDM\_ERROR\_READ\_DFI\_FILELIST** FileList 読み込み失敗

**E\_CDM\_ERROR\_READ\_NETCDF\_FUNC** NetCDF の nc 関数でエラー

**E\_CDM\_ERROR\_READ\_NETCDF\_VAR\_1D** NetCDF の 1 次元配列として読み込む variable が 1 次元で無い

**E\_CDM\_ERROR\_READ\_FUB\_REC1** fub の第 1 レコード読み込みエラー

**E\_CDM\_ERROR\_READ\_FUB\_REC2** fub の第 2 レコード読み込みエラー

**E\_CDM\_ERROR\_MISMATCH\_NP\_SUBDOMAIN** 並列数とサブドメイン数が一致していない

**E\_CDM\_ERROR\_INVALID\_DIVNUM** 領域分割数が不正

**E\_CDM\_ERROR\_OPEN\_SBDM** ActiveSubdomain ファイルのオープンに失敗

**E\_CDM\_ERROR\_READ\_SBDM\_HEADER** ActiveSubdomain ファイルのヘッダー読み込みに失敗

**E\_CDM\_ERROR\_READ\_SBDM\_FORMAT** ActiveSubdomain ファイルのフォーマットエラー

**E\_CDM\_ERROR\_READ\_SBDM\_DIV** ActiveSubdomain ファイルの領域分割数読み込みに失敗

**E\_CDM\_ERROR\_READ\_SBDM\_CONTENTS** ActiveSubdomain ファイルのContents 読み込みに失敗

**E\_CDM\_ERROR\_SBDM\_NUMDOMAIN\_ZERO** ActiveSubdomain ファイルの活性ドメイン数が 0.

**E\_CDM\_ERROR\_MAKEDIRECTORY** Directory 生成で失敗

**E\_CDM\_ERROR\_OPEN\_FIELDDATA** フィールドデータのオープンに失敗

**E\_CDM\_ERROR\_WRITE\_FIELD\_HEADER\_RECORD** フィールドヘッダーレコード出力失敗

**E\_CDM\_ERROR\_WRITE\_FIELD\_DATA\_RECORD** フィールドデータレコード出力失敗

**E\_CDM\_ERROR\_WRITE\_FIELD\_AVERAGED\_RECORD** フィールドAverage 出力失敗

**E\_CDM\_ERROR\_WRITE\_SPH\_REC1** SPH ファイルレコード 1 出力エラー

**E\_CDM\_ERROR\_WRITE\_SPH\_REC2** SPH ファイルレコード 2 出力エラー

**E\_CDM\_ERROR\_WRITE\_SPH\_REC3** SPH ファイルレコード 3 出力エラー

**E\_CDM\_ERROR\_WRITE\_SPH\_REC4** SPH ファイルレコード 4 出力エラー

**E\_CDM\_ERROR\_WRITE\_SPH\_REC5** SPH ファイルレコード 5 出力エラー

**E\_CDM\_ERROR\_WRITE\_SPH\_REC6** SPH ファイルレコード 6 出力エラー

**E\_CDM\_ERROR\_WRITE\_SPH\_REC7** SPH ファイルレコード 7 出力エラー

**E\_CDM\_ERROR\_WRITE\_PROCFILENAME\_EMPTY** proc dfi ファイル名が未定義

**E\_CDM\_ERROR\_WRITE\_PROCFILE\_OPENERERROR** proc dfi ファイルオープン失敗

**E\_CDM\_ERROR\_WRITE\_DOMAIN** Domain 出力失敗

**E\_CDM\_ERROR\_WRITE\_MPI** MPI 出力失敗

**E\_CDM\_ERROR\_WRITE\_PROCESS** Process 出力失敗

**E\_CDM\_ERROR\_WRITE\_RANKID** 出力ランク以外

**E\_CDM\_ERROR\_WRITE\_INDEXFILENAME\_EMPTY** index dfi ファイル名が未定義

**E\_CDM\_ERROR\_WRITE\_PREFIX\_EMPTY** Prefix が未定義

**E\_CDM\_ERROR\_WRITE\_INDEXFILE\_OPENERERROR** proc dfi ファイルオープン失敗

**E\_CDM\_ERROR\_WRITE\_FILEINFO** FileInfo 出力失敗

**E\_CDM\_ERROR\_WRITE\_UNIT** Unit 出力失敗

**E\_CDM\_ERROR\_WRITE\_TIMESLICE** TimeSlice 出力失敗



***E\_CDM\_ERROR\_WRITE\_FILEPATH*** FilePath 出力失敗  
***E\_CDM\_ERROR\_WRITE\_VISIT*** Visit 出力失敗  
***E\_CDM\_ERROR\_WRITE\_GRIDFILE*** grid ファイル出力失敗  
***E\_CDM\_ERROR\_WRITE\_DFI\_NETCDF*** NetCDF のDFI 出力エラー  
***E\_CDM\_WARN\_GETUNIT*** Unit の単位がない

cdm\_Define.h の 174 行で定義されています。

#### 5.1.2.6 enum CDM::E\_CDM\_FILE\_TYPE

ファイルタイプ

列挙型の値

***E\_CDM\_FILE\_TYPE\_DEFAULT*** デフォルト ( binary)  
***E\_CDM\_FILE\_TYPE\_ASCII*** ascii  
***E\_CDM\_FILE\_TYPE\_BINARY*** binary  
***E\_CDM\_FILE\_TYPE\_FBINAR*** Fortran Binary.

cdm\_Define.h の 152 行で定義されています。

#### 5.1.2.7 enum CDM::E\_CDM\_FORMAT

File 形式

列挙型の値

***E\_CDM\_FMT\_UNKNOWN*** 未定  
***E\_CDM\_FMT\_SPH*** sph format  
***E\_CDM\_FMT\_BOV*** bov format  
***E\_CDM\_FMT\_AVS*** avs format  
***E\_CDM\_FMT\_PLOT3D*** plot3d format  
***E\_CDM\_FMT\_VTK*** vtk format  
***E\_CDM\_FMT\_NETCDF4*** NetCDF4(/w HDF5) format.  
***E\_CDM\_FMT\_FUB*** upacs fub field data format  
***E\_CDM\_FMT\_FUB\_COD*** upacs fub coordinate data format

cdm\_Define.h の 84 行で定義されています。

#### 5.1.2.8 enum CDM::E\_CDM\_ONOFF

スイッチ on or off

列挙型の値

***E\_CDM\_OFF*** off  
***E\_CDM\_ON*** on

cdm\_Define.h の 103 行で定義されています。

## 5.1.2.9 enum CDM::E\_CDM\_OUTPUT\_FNAME

フィールドファイルの命名規約タイプ

列挙型の値

**E\_CDM\_FNAME\_DEFAULT** 出力ファイル命名規約デフォルト (step\_rank)  
**E\_CDM\_FNAME\_STEP\_RANK** step\_rank  
**E\_CDM\_FNAME\_RANK\_STEP** rank\_step  
**E\_CDM\_FNAME\_RANK** rank(NetCDF, fub のみ)  
**E\_CDM\_FNAME\_CUSTOM** FieldFileNameFormat に指示あり

cdm\_Define.h の 161 行で定義されています。

## 5.1.2.10 enum CDM::E\_CDM\_READTYPE

読み込みタイプコード

列挙型の値

**E\_CDM\_SAMEDIV\_SAMERES** 同一分割 & 同一密度  
**E\_CDM\_SAMEDIV\_REFINEMENT** 同一分割 & 粗密  
**E\_CDM\_DIFFDIV\_SAMERES** MxN & 同一密度  
**E\_CDM\_DIFFDIV\_REFINEMENT** MxN & 粗密  
**E\_CDM\_READTYPE\_UNKNOWN** error

cdm\_Define.h の 142 行で定義されています。

## 5.1.3 関数

## 5.1.3.1 std::string CDM::cdmPath\_ConnectPath ( std::string dirName, std::string fname ) [inline]

cdm\_PathUtil.h の 169 行で定義されています。

参照先 cdmPath\_getDelimChar(), と cdmPath\_getDelimString().

参照元 cdm\_NonUniformDomain< T >::Read().

## 5.1.3.2 std::string CDM::cdmPath\_DirName ( const std::string &amp; path, const char dc = cdmPath\_getDelimChar() ) [inline]

cdm\_PathUtil.h の 67 行で定義されています。

参照先 cdmPath\_isAbsolute().

参照元 cdm\_DFI::WriteInit().

## 5.1.3.3 std::string CDM::cdmPath\_FileName ( const std::string &amp; path, const std::string &amp; addext = std::string(""), const char dc = cdmPath\_getDelimChar() ) [inline]

cdm\_PathUtil.h の 122 行で定義されています。

## 5.1.3.4 char CDM::cdmPath\_getDelimChar ( ) [inline]

cdm\_PathUtil.h の 21 行で定義されています。

参照元 cdmPath\_ConnectPath(), cdmPath\_getDelimString(), と cdmPath\_isAbsolute().

#### 5.1.3.5 `std::string CDM::cdmPath_getDelimString ( ) [inline]`

cdm\_PathUtil.h の 30 行で定義されています。

参照先 `cdmPath_getDelimChar()`.

参照元 `cdmPath_ConnectPath()`.

#### 5.1.3.6 `bool CDM::cdmPath_hasDrive ( const std::string & path ) [inline]`

cdm\_PathUtil.h の 37 行で定義されています。

参照元 `vfvPath_emitDrive()`.

#### 5.1.3.7 `bool CDM::cdmPath_isAbsolute ( const std::string & path ) [inline]`

cdm\_PathUtil.h の 57 行で定義されています。

参照先 `cdmPath_getDelimChar()`, と `vfvPath_emitDrive()`.

参照元 `cdmPath_DirName()`, と `cdm_NonUniformDomain< T >::Read()`.

#### 5.1.3.8 `std::string CDM::ExtractPathWithoutExt ( const std::string & fn ) [inline]`

cdm\_PathUtil.h の 185 行で定義されています。

#### 5.1.3.9 `std::string CDM::vfvPath_emitDrive ( std::string & path ) [inline]`

cdm\_PathUtil.h の 46 行で定義されています。

参照先 `cdmPath_hasDrive()`.

参照元 `cdmPath_isAbsolute()`.

### 5.1.4 変数

#### 5.1.4.1 `const char* CDM::C_CDM_RANKNOPREFIX = "_id" [static]`

ファイル名のランク番号前の文字列

cdm\_Define.h の 74 行で定義されています。



## Chapter 6

# クラス

### 6.1 クラス `cdm_ActiveSubDomain`

```
#include <cdm_ActiveSubDomain.h>
```

#### Public メソッド

- `cdm_ActiveSubDomain ()`
- `cdm_ActiveSubDomain (int pos[3])`
- `virtual ~cdm_ActiveSubDomain ()`
- `virtual void clear ()`
- `void SetPos (int pos[3])`
- `const int * GetPos () const`
- `bool operator== (cdm_ActiveSubDomain dom)`
- `bool operator!= (cdm_ActiveSubDomain dom)`

#### Private 変数

- `int m_pos [3]`  
領域分割内での位置

#### 6.1.1 説明

ActiveSubDomian class

`cdm_ActiveSubDomain.h` の 19 行で定義されています。

#### 6.1.2 コンストラクタとデストラクタ

##### 6.1.2.1 `cdm_ActiveSubDomain::cdm_ActiveSubDomain ( )`

デフォルトコンストラクタ

##### 6.1.2.2 `cdm_ActiveSubDomain::cdm_ActiveSubDomain ( int pos[3] )`

コンストラクタ

引数

<i>in</i>	<i>pos</i>	領域分割内での位置
-----------	------------	-----------

6.1.2.3 `virtual cdm_ActiveSubDomain::~cdm_ActiveSubDomain ( ) [virtual]`

デストラクタ

## 6.1.3 関数

6.1.3.1 `virtual void cdm_ActiveSubDomain::clear ( ) [virtual]`

情報のクリア

6.1.3.2 `const int* cdm_ActiveSubDomain::GetPos ( ) const`

位置の取得

戻り値

位置情報整数配列のポインタ

6.1.3.3 `bool cdm_ActiveSubDomain::operator!= ( cdm_ActiveSubDomain dom )`

比較演算子

引数

<i>in</i>	<i>dom</i>	比較対象の活性サブドメイン情報
-----------	------------	-----------------

戻り値

<i>true</i>	違う位置情報を持つ
<i>false</i>	同じ位置情報を持つ

6.1.3.4 `bool cdm_ActiveSubDomain::operator== ( cdm_ActiveSubDomain dom )`

比較演算子

引数

<i>in</i>	<i>dom</i>	比較対象の活性サブドメイン情報
-----------	------------	-----------------

戻り値

<i>true</i>	同じ位置情報を持つ
<i>false</i>	違う位置情報を持つ

6.1.3.5 `void cdm_ActiveSubDomain::SetPos ( int pos[3] )`

位置のセット

引数

<code>in</code>	<code>pos</code>	領域分割内での位置
-----------------	------------------	-----------

### 6.1.4 変数

#### 6.1.4.1 `int cdm_ActiveSubDomain::m_pos[3] [private]`

領域分割内での位置

`cdm_ActiveSubDomain.h` の 63 行で定義されています。

このクラスの説明は次のファイルから生成されました:

- [cdm\\_ActiveSubDomain.h](#)

## 6.2 クラス `cdm_Array`

```
#include <cdm_Array.h>
```

`cdm_Array` に対する継承グラフ

### Public メソッド

- `virtual ~cdm_Array ()`  
デストラクタ
- `void * getData (bool extract=false)`  
データポインタを取得
- `CDM::E_CDM_DTYPE getDataType () const`  
データタイプの取得
- `const char * getDataTypeString () const`  
データタイプ文字列の取得
- `CDM::E_CDM_ARRAYSHAPE getArrayShape () const`  
配列形状の取得
- `const char * getArrayShapeString () const`  
配列形状文字列の取得
- `size_t getGc () const`  
ガイドセル数を取得
- `int getGcInt () const`  
ガイドセル数を取得 (*int* 版)
- `size_t getNvari () const`  
変数の個数を取得
- `int getNvariInt () const`  
変数の個数を取得 (*int* 版)
- `const size_t * getArraySize ()`  
格子数を取得
- `const int * getArraySizeInt ()`  
格子数を取得 (*int* 版)
- `const int * getHeadIndex ()`  
*head* インデクスを取得
- `const int * getTailIndex ()`  
*tail* インデクスを取得

- `const size_t * _getArraySize ()`  
ガイドセルを含んだ格子数を取得
- `const int * _getArraySizeInt ()`  
ガイドセルを含んだ格子数を取得 (*int* 版)
- `size_t getArrayLength () const`  
配列長を取得
- `void setHeadIndex (int head[3])`  
*head/tail* をセット
- `virtual int copyArray (cdm_Array *dst, bool ignoreGc=false)=0`  
配列コピー (自信を *dst* にコピー。 *head/tail* を考慮した重複範囲をコピー)
- `virtual int copyArray (int sta[3], int end[3], cdm_Array *dst)=0`  
範囲指定での配列コピー (自信を *dst* にコピー。 *head/tail* を考慮した重複範囲をコピー)
- `virtual int copyArrayNvari (cdm_Array *dst, int vari, bool ignoreGc=false)=0`  
指定変数の配列コピー (自信を *dst* にコピー。 *head/tail* を考慮した重複範囲をコピー)
- `virtual int copyArrayNvari (int sta[3], int end[3], cdm_Array *dst, int vari)=0`  
指定変数の範囲指定での配列コピー (自信を *dst* にコピー。 *head/tail* を考慮した重複範囲をコピー)
- `virtual int copyArrayNvari_to_ijk (cdm_Array *dst, int vari, bool ignoreGc=false)=0`  
指定した変数の配列のみ取得し、*IJK* 配列で *dst* にコピー
- `virtual int copyArrayNvari_to_ijk (int sta[3], int end[3], cdm_Array *dst, int vari)=0`  
指定した変数の配列のみ範囲指定で取得し、*IJK* 配列で *dst* にコピー
- `virtual size_t readBinary (FILE *fp, bool bMatchEndian)=0`  
配列サイズ分のバイナリデータを読み込み (戻り値は読み込んだ要素数)
- `virtual size_t writeBinary (FILE *fp)=0`  
配列サイズ分のバイナリデータを書き出す (戻り値は読み込んだ要素数)
- `virtual size_t writeAscii (FILE *fp)=0`  
配列サイズ分の *ascii* データを書き出す (戻り値は読み込んだ要素数)
- `template<class T >`  
`instanceArray (T *data, CDM::E_CDM_ARRAYSHAPE shape, size_t ix, size_t jx, size_t kx, size_t gc, size_t nvari)`
- `template<class T >`  
`instanceArray (T *data, CDM::E_CDM_ARRAYSHAPE shape, size_t sz[3], size_t gc, size_t nvari)`
- `template<class T >`  
`instanceArray (T *data, CDM::E_CDM_ARRAYSHAPE shape, int ix, int jx, int kx, int gc, int nvari)`
- `template<class T >`  
`instanceArray (T *data, CDM::E_CDM_ARRAYSHAPE shape, int sz[3], int gc, int nvari)`

## Static Public メソッド

- `static cdm_Array * instanceArray (CDM::E_CDM_DTYPE dtype, CDM::E_CDM_ARRAYSHAPE shape, size_t ix, size_t jx, size_t kx, size_t gc, size_t nvari=1)`  
インスタンス
- `static cdm_Array * instanceArray (CDM::E_CDM_DTYPE dtype, CDM::E_CDM_ARRAYSHAPE shape, size_t sz[3], size_t gc, size_t nvari=1)`  
インスタンス
- `static cdm_Array * instanceArray (CDM::E_CDM_DTYPE dtype, CDM::E_CDM_ARRAYSHAPE shape, int ix, int jx, int kx, int gc, int nvari=1)`  
インスタンス
- `static cdm_Array * instanceArray (CDM::E_CDM_DTYPE dtype, CDM::E_CDM_ARRAYSHAPE shape, int sz[3], int gc, int nvari=1)`  
インスタンス
- `template<class T >`  
`static cdm_Array * instanceArray (T *data, CDM::E_CDM_ARRAYSHAPE shape, size_t ix, size_t jx, size_t kx, size_t gc, size_t nvari=1)`



## インスタンス

- `template<class T >`  
static `cdm_Array * instanceArray` (`T *data`, `CDM::E_CDM_ARRAYSHAPE` `shape`, `size_t sz[3]`, `size_t gc`, `size_t nvari=1`)

## インスタンス

- `template<class T >`  
static `cdm_Array * instanceArray` (`T *data`, `CDM::E_CDM_ARRAYSHAPE` `shape`, `int ix`, `int jx`, `int kx`, `int gc`, `int nvari=1`)

## インスタンス

- `template<class T >`  
static `cdm_Array * instanceArray` (`T *data`, `CDM::E_CDM_ARRAYSHAPE` `shape`, `int sz[3]`, `int gc`, `int nvari=1`)

## インスタンス

- static `cdm_Array * interp_coarse` (`cdm_Array *src`, `int &err`, `bool head0start=true`)

粗密データの補間処理を行う

## Protected メソッド

- `cdm_Array` ()  
デフォルトコンストラクタ
- `cdm_Array` (`CDM::E_CDM_DTYPE` `dtype`, `CDM::E_CDM_ARRAYSHAPE` `shape`, `size_t ix`, `size_t jx`, `size_t kx`, `size_t gc`, `size_t nvari=1`)  
コンストラクタ

## Protected 変数

- `CDM::E_CDM_DTYPE m_dtype`  
データタイプ
- `CDM::E_CDM_ARRAYSHAPE m_shape`  
配列形状
- `size_t m_gc`  
ガイドセル数
- `size_t m_sz` [3]  
格子数
- `size_t m_Sz` [4]  
ガイドセルを含んだ格子数
- `size_t m_gcl` [4]  
ガイドセル数 (インデクス毎)
- `size_t m_nvari`  
変数の個数
- `int m_gcl`  
ガイドセル数 (*int*)
- `int m_szl` [3]  
格子数 (*int*)
- `int m_Szl` [4]  
ガイドセルを含んだ格子数 (*int*)
- `int m_nvaril`  
変数の個数 (*int*)
- `int m_headIndex` [4]  
*head* インデックス
- `int m_tailIndex` [4]  
*tail* インデックス

### 6.2.1 説明

cdm\_Array.h の 22 行で定義されています。

### 6.2.2 コンストラクタとデストラクタ

6.2.2.1 `virtual cdm_Array::~cdm_Array( ) [inline],[virtual]`

デストラクタ

cdm\_Array.h の 59 行で定義されています。

6.2.2.2 `cdm_Array::cdm_Array( ) [inline],[protected]`

デフォルトコンストラクタ

cdm\_Array.h の 378 行で定義されています。

参照先 CDM::E\_CDM\_ARRAYSHAPE\_UNKNOWN, CDM::E\_CDM\_DTYPE\_UNKNOWN, m\_dtype, m\_gc, m\_gcl, m\_headIndex, m\_nvari, m\_shape, m\_sz, m\_Sz, と m\_tailIndex.

6.2.2.3 `cdm_Array::cdm_Array( CDM::E_CDM_DTYPE dtype, CDM::E_CDM_ARRAYSHAPE shape, size_t ix, size_t jx, size_t kx, size_t gc, size_t nvari = 1 ) [inline],[protected]`

コンストラクタ

cdm\_Array.h の 392 行で定義されています。

参照先 CDM::E\_CDM\_IJKN, CDM::E\_CDM\_NIJK, m\_dtype, m\_gc, m\_gcl, m\_gcl, m\_nvari, m\_nvaril, m\_shape, m\_sz, m\_Sz, m\_szl, m\_Szl, と setHeadIndex().

### 6.2.3 関数

6.2.3.1 `const size_t* cdm_Array::_getArraySize( ) [inline]`

ガイドセルを含んだ格子数を取得

cdm\_Array.h の 275 行で定義されています。

参照先 CDM::E\_CDM\_IJKN, CDM::E\_CDM\_NIJK, m\_shape, と m\_Sz.

6.2.3.2 `const int* cdm_Array::_getArraySizeInt( ) [inline]`

ガイドセルを含んだ格子数を取得 (int 版)

cdm\_Array.h の 290 行で定義されています。

参照先 CDM::E\_CDM\_IJKN, CDM::E\_CDM\_NIJK, m\_shape, と m\_Szl.

6.2.3.3 `virtual int cdm_Array::copyArray( cdm_Array * dst, bool ignoreGc = false ) [pure virtual]`

配列コピー (自信を dst にコピー。head/tail を考慮した重複範囲をコピー)

[cdm\\_TypeArray< T >](#) で実装されています。

6.2.3.4 `virtual int cdm_Array::copyArray( int sta[3], int end[3], cdm_Array * dst ) [pure virtual]`

範囲指定での配列コピー (自信を dst にコピー。head/tail を考慮した重複範囲をコピー)

`cdm_TypeArray< T >` で実装されています。

6.2.3.5 `virtual int cdm_Array::copyArrayNvari ( cdm_Array * dst, int vari, bool ignoreGc = false ) [pure virtual]`

指定変数の配列コピー (自信を `dst` にコピー。head/tail を考慮した重複範囲をコピー)

`cdm_TypeArray< T >` で実装されています。

6.2.3.6 `virtual int cdm_Array::copyArrayNvari ( int sta[3], int end[3], cdm_Array * dst, int vari ) [pure virtual]`

指定変数の範囲指定での配列コピー (自信を `dst` にコピー。head/tail を考慮した重複範囲をコピー)

`cdm_TypeArray< T >` で実装されています。

6.2.3.7 `virtual int cdm_Array::copyArrayNvari_to_ijk ( cdm_Array * dst, int vari, bool ignoreGc = false ) [pure virtual]`

指定した変数の配列のみ取得し、IJK 配列で `dst` にコピー

`cdm_TypeArray< T >` で実装されています。

6.2.3.8 `virtual int cdm_Array::copyArrayNvari_to_ijk ( int sta[3], int end[3], cdm_Array * dst, int vari ) [pure virtual]`

指定した変数の配列のみ範囲指定で取得し、IJK 配列で `dst` にコピー

`cdm_TypeArray< T >` で実装されています。

6.2.3.9 `size_t cdm_Array::getArrayLength ( ) const [inline]`

配列長を取得

`cdm_Array.h` の 305 行で定義されています。

参照先 `m_Sz`.

参照元 `cdm_DFI_PLOT3D::read_Func()`.

6.2.3.10 `CDM::E_CDM_ARRAYSHAPE cdm_Array::getArrayShape ( ) const [inline]`

配列形状の取得

`cdm_Array.h` の 188 行で定義されています。

参照先 `m_shape`.

参照元 `cdm_TypeArray< T >::copyArray()`, `cdm_TypeArray< T >::copyArrayNvari()`, `cdm_TypeArray< T >::copyArrayNvari_to_ijk()`, `cdm_DFI_PLOT3D::read_Func()`, `cdm_DFI::setGridData()`, `cdm_DFI::VolumeDataDivide()`, と `cdm_DFI_PLOT3D::write_Func()`.

6.2.3.11 `const char* cdm_Array::getArrayShapeString ( ) const [inline]`

配列形状文字列の取得

`cdm_Array.h` の 194 行で定義されています。

参照先 `CDM::E_CDM_IJKN`, `CDM::E_CDM_NIJK`, と `m_shape`.

#### 6.2.3.12 `const size_t* cdm_Array::getArraySize ( ) [inline]`

格子数を取得

cdm\_Array.h の 233 行で定義されています。

参照先 m\_sz.

#### 6.2.3.13 `const int* cdm_Array::getArraySizeInt ( ) [inline]`

格子数を取得 (int 版)

cdm\_Array.h の 239 行で定義されています。

参照先 m\_szl.

参照元 cdm\_DFI\_PLOT3D::read\_Func(), cdm\_DFI::setGridData(), と cdm\_DFI::VolumeDataDivide().

#### 6.2.3.14 `cdm_Array::getData ( bool extract = false )`

データポインタを取得

cdm\_Array\_inline.h の 244 行で定義されています。

参照先 CDM::E\_CDM\_FLOAT32, CDM::E\_CDM\_FLOAT64, CDM::E\_CDM\_INT16, CDM::E\_CDM\_INT32, CDM::E\_CDM\_INT64, CDM::E\_CDM\_INT8, CDM::E\_CDM\_UINT16, CDM::E\_CDM\_UINT32, CDM::E\_CDM\_UINT64, CDM::E\_CDM\_UINT8, と cdm\_TypeArray< T >::getData().

参照元 interp\_coarse(), cdm\_DFI::ReadCoordinateData(), と cdm\_DFI::ReadData().

#### 6.2.3.15 `CDM::E_CDM_DTYPE cdm_Array::getDataType ( ) const [inline]`

データタイプの取得

cdm\_Array.h の 143 行で定義されています。

参照先 m\_dtype.

参照元 cdm\_TypeArray< T >::copyArray(), cdm\_TypeArray< T >::copyArrayNvari(), cdm\_TypeArray< T >::copyArrayNvari\_to\_ijk(), と interp\_coarse().

#### 6.2.3.16 `const char* cdm_Array::getDataTypeString ( ) const [inline]`

データタイプ文字列の取得

cdm\_Array.h の 149 行で定義されています。

参照先 CDM::E\_CDM\_FLOAT32, CDM::E\_CDM\_FLOAT64, CDM::E\_CDM\_INT16, CDM::E\_CDM\_INT32, CDM::E\_CDM\_INT64, CDM::E\_CDM\_INT8, CDM::E\_CDM\_UINT16, CDM::E\_CDM\_UINT32, CDM::E\_CDM\_UINT64, CDM::E\_CDM\_UINT8, と m\_dtype.

#### 6.2.3.17 `size_t cdm_Array::getGc ( ) const [inline]`

ガイドセル数を取得

cdm\_Array.h の 209 行で定義されています。

参照先 m\_gc.

#### 6.2.3.18 `int cdm_Array::getGcInt ( ) const [inline]`

ガイドセル数を取得 (int 版)

`cdm_Array.h` の 215 行で定義されています。

参照先 `m_gcl`.

参照元 `cdm_TypeArray< T >::copyArray()`, `cdm_TypeArray< T >::copyArrayNvari()`, と `cdm_TypeArray< T >::copyArrayNvari_to_ijk()`.

#### 6.2.3.19 `const int* cdm_Array::getHeadIndex ( ) [inline]`

head インデクスを取得

`cdm_Array.h` の 245 行で定義されています。

参照先 `CDM::E_CDM_IJKN`, `CDM::E_CDM_NIJK`, `m_headIndex`, と `m_shape`.

参照元 `cdm_TypeArray< T >::copyArray()`, `cdm_TypeArray< T >::copyArrayNvari()`, と `cdm_TypeArray< T >::copyArrayNvari_to_ijk()`.

#### 6.2.3.20 `size_t cdm_Array::getNvari ( ) const [inline]`

変数の個数を取得

`cdm_Array.h` の 221 行で定義されています。

参照先 `m_nvari`.

参照元 `cdm_TypeArray< T >::copyArray()`, `cdm_TypeArray< T >::copyArrayNvari()`, `cdm_TypeArray< T >::copyArrayNvari_to_ijk()`, と `cdm_DFI_PLOT3D::read_Func()`.

#### 6.2.3.21 `int cdm_Array::getNvariInt ( ) const [inline]`

変数の個数を取得 (int 版)

`cdm_Array.h` の 227 行で定義されています。

参照先 `m_nvaril`.

参照元 `cdm_DFI::setGridData()`, と `cdm_DFI::VolumeDataDivide()`.

#### 6.2.3.22 `const int* cdm_Array::getTailIndex ( ) [inline]`

tail インデクスを取得

`cdm_Array.h` の 260 行で定義されています。

参照先 `CDM::E_CDM_IJKN`, `CDM::E_CDM_NIJK`, `m_shape`, と `m_tailIndex`.

参照元 `cdm_TypeArray< T >::copyArray()`, `cdm_TypeArray< T >::copyArrayNvari()`, と `cdm_TypeArray< T >::copyArrayNvari_to_ijk()`.

#### 6.2.3.23 `cdm_Array::instanceArray ( CDM::E_CDM_DTYPE dtype, CDM::E_CDM_ARRAYSHAPE shape, size_t ix, size_t jx, size_t kx, size_t gx, size_t nvari = 1 ) [static]`

インスタンス

`cdm_Array_inline.h` の 30 行で定義されています。

参照先 `CDM::E_CDM_FLOAT32`, `CDM::E_CDM_FLOAT64`, `CDM::E_CDM_INT16`, `CDM::E_CDM_INT32`, `CDM::E_CDM_INT64`, `CDM::E_CDM_INT8`, `CDM::E_CDM_UINT16`, `CDM::E_CDM_UINT32`, `CDM::E_CDM_UINT64`, `CDM::E_CDM_UINT8`, `m_gcl`, と `m_Sz`.

参照元 `interp_coarse()`, `cdm_DFI::ReadCoordinateData()`, `cdm_DFI::ReadData()`, `cdm_DFI::WriteCoordinateData()`, `cdm_DFI::WriteData()`, と `cdm_DFI::WriteFieldDataFile()`.

6.2.3.24 `cdm_Array::instanceArray ( CDM::E_CDM_DTYPE dtype, CDM::E_CDM_ARRAYSHAPE shape, size_t sz[3], size_t gc, size_t nvari = 1 ) [static]`

インスタンス

`cdm_Array_inline.h` の 93 行で定義されています。

6.2.3.25 `cdm_Array::instanceArray ( CDM::E_CDM_DTYPE dtype, CDM::E_CDM_ARRAYSHAPE shape, int ix, int jx, int kx, int gc, int nvari = 1 ) [static]`

インスタンス

`cdm_Array_inline.h` の 104 行で定義されています。

6.2.3.26 `cdm_Array::instanceArray ( CDM::E_CDM_DTYPE dtype, CDM::E_CDM_ARRAYSHAPE shape, int sz[3], int gc, int nvari = 1 ) [static]`

インスタンス

`cdm_Array_inline.h` の 117 行で定義されています。

6.2.3.27 `template<class T> static cdm_Array* cdm_Array::instanceArray ( T* data, CDM::E_CDM_ARRAYSHAPE shape, size_t ix, size_t jx, size_t kx, size_t gc, size_t nvari = 1 ) [static]`

インスタンス

6.2.3.28 `template<class T> static cdm_Array* cdm_Array::instanceArray ( T* data, CDM::E_CDM_ARRAYSHAPE shape, size_t sz[3], size_t gc, size_t nvari = 1 ) [static]`

インスタンス

6.2.3.29 `template<class T> static cdm_Array* cdm_Array::instanceArray ( T* data, CDM::E_CDM_ARRAYSHAPE shape, int ix, int jx, int kx, int gc, int nvari = 1 ) [static]`

インスタンス

6.2.3.30 `template<class T> cdm_Array::instanceArray ( T* data, CDM::E_CDM_ARRAYSHAPE shape, size_t ix, size_t jx, size_t kx, size_t gc, size_t nvari )`

`cdm_Array_inline.h` の 129 行で定義されています。

参照先 `CDM::E_CDM_DTYPE_UNKNOWN`, `CDM::E_CDM_FLOAT32`, `CDM::E_CDM_FLOAT64`, `CDM::E_CDM_INT16`, `CDM::E_CDM_INT32`, `CDM::E_CDM_INT64`, `CDM::E_CDM_INT8`, `CDM::E_CDM_UINT16`, `CDM::E_CDM_UINT32`, `CDM::E_CDM_UINT64`, `CDM::E_CDM_UINT8`, `m_gcl`, と `m_Sz`.

6.2.3.31 `template<class T> static cdm_Array* cdm_Array::instanceArray ( T* data, CDM::E_CDM_ARRAYSHAPE shape, int sz[3], int gc, int nvari = 1 ) [static]`

インスタンス

6.2.3.32 `template<class T> cdm_Array::instanceArray ( T* data, CDM::E_CDM_ARRAYSHAPE shape, size_t sz[3], size_t gc, size_t nvari )`

`cdm_Array_inline.h` の 207 行で定義されています。

6.2.3.33 `template<class T> cdm_Array::instanceArray ( T * data, CDM::E_CDM_ARRAYSHAPE shape, int ix, int jx, int kx, int gc, int nvari )`

`cdm_Array_inline.h` の 219 行で定義されています。

6.2.3.34 `template<class T> cdm_Array::instanceArray ( T * data, CDM::E_CDM_ARRAYSHAPE shape, int sz[3], int gc, int nvari )`

`cdm_Array_inline.h` の 233 行で定義されています。

6.2.3.35 `cdm_Array::interp_coarse ( cdm_Array * src, int & err, bool head0start=true ) [static]`

粗密データの補間処理を行う

`cdm_Array_inline.h` の 683 行で定義されています。

参照先 `cdm_interp_ijkn_r4_()`, `cdm_interp_ijkn_r8_()`, `cdm_interp_nijk_r4_()`, `cdm_interp_nijk_r8_()`, `CDM::E_CDM_FLOAT32`, `CDM::E_CDM_FLOAT64`, `CDM::E_CDM_IJKN`, `getData()`, `getDataType()`, `instanceArray()`, と `setHeadIndex()`.

6.2.3.36 `virtual size_t cdm_Array::readBinary ( FILE * fp, bool bMatchEndian ) [pure virtual]`

配列サイズ分のバイナリデータを読み込み (戻り値は読み込んだ要素数)

`cdm_TypeArray< T >` で実装されています。

6.2.3.37 `void cdm_Array::setHeadIndex ( int head[3] ) [inline]`

head/tail をセット

`cdm_Array.h` の 316 行で定義されています。

参照先 `CDM::E_CDM_IJKN`, `CDM::E_CDM_NIJK`, `m_headIndex`, `m_shape`, `m_sz`, と `m_tailIndex`.

参照元 `cdm_Array()`, `interp_coarse()`, と `cdm_DFI_PLOT3D::read_Func()`.

6.2.3.38 `virtual size_t cdm_Array::writeAscii ( FILE * fp ) [pure virtual]`

配列サイズ分の ascii データを書き出す (戻り値は読み込んだ要素数)

`cdm_TypeArray< T >` で実装されています。

6.2.3.39 `virtual size_t cdm_Array::writeBinary ( FILE * fp ) [pure virtual]`

配列サイズ分のバイナリデータを書き出す (戻り値は読み込んだ要素数)

`cdm_TypeArray< T >` で実装されています。

## 6.2.4 変数

6.2.4.1 `CDM::E_CDM_DTYPE cdm_Array::m_dtype [protected]`

データタイプ

`cdm_Array.h` の 444 行で定義されています。

参照元 `cdm_Array()`, `getDataType()`, と `getDataTypeString()`.

#### 6.2.4.2 `size_t cdm_Array::m_gc` [protected]

##### ガイドセル数

cdm\_Array.h の 450 行で定義されています。

参照元 `cdm_Array()`, `cdm_TypeArray< T >::cdm_TypeArray()`, と `getGc()`.

#### 6.2.4.3 `int cdm_Array::m_gcl` [protected]

##### ガイドセル数 (int)

cdm\_Array.h の 466 行で定義されています。

参照元 `cdm_Array()`, と `getGclnt()`.

#### 6.2.4.4 `size_t cdm_Array::m_gcl[4]` [protected]

##### ガイドセル数 (インデクス毎)

cdm\_Array.h の 459 行で定義されています。

参照元 `cdm_Array()`, と `instanceArray()`.

#### 6.2.4.5 `int cdm_Array::m_headIndex[4]` [protected]

##### head インデックス

cdm\_Array.h の 479 行で定義されています。

参照元 `cdm_Array()`, `getHeadIndex()`, と `setHeadIndex()`.

#### 6.2.4.6 `size_t cdm_Array::m_nvari` [protected]

##### 変数の個数

cdm\_Array.h の 462 行で定義されています。

参照元 `cdm_Array()`, `cdm_TypeArray< T >::cdm_TypeArray()`, と `getNvari()`.

#### 6.2.4.7 `int cdm_Array::m_nvaril` [protected]

##### 変数の個数 (int)

cdm\_Array.h の 475 行で定義されています。

参照元 `cdm_Array()`, と `getNvarilnt()`.

#### 6.2.4.8 `CDM::E_CDM_ARRAYSHAPE cdm_Array::m_shape` [protected]

##### 配列形状

cdm\_Array.h の 447 行で定義されています。

参照元 `_getArraySize()`, `_getArraySizeInt()`, `cdm_Array()`, `getArrayShape()`, `getArrayShapeString()`, `getHeadIndex()`, `getTailIndex()`, と `setHeadIndex()`.

#### 6.2.4.9 `size_t cdm_Array::m_sz[3]` [protected]

##### 格子数



cdm\_Array.h の 453 行で定義されています。

参照元 cdm\_Array(), cdm\_TypeArray< T >::cdm\_TypeArray(), getArraySize(), と setHeadIndex().

6.2.4.10 `size_t cdm_Array::m_Sz[4]` [protected]

ガイドセルを含んだ格子数

cdm\_Array.h の 456 行で定義されています。

参照元 `_getArraySize()`, `cdm_Array()`, `getArrayLength()`, と `instanceArray()`.

6.2.4.11 `int cdm_Array::m_szl[3]` [protected]

格子数 (int)

cdm\_Array.h の 469 行で定義されています。

参照元 `cdm_Array()`, と `getArraySizeInt()`.

6.2.4.12 `int cdm_Array::m_Szl[4]` [protected]

ガイドセルを含んだ格子数 (int)

cdm\_Array.h の 472 行で定義されています。

参照元 `_getArraySizeInt()`, と `cdm_Array()`.

6.2.4.13 `int cdm_Array::m_tailIndex[4]` [protected]

tail インデックス

cdm\_Array.h の 482 行で定義されています。

参照元 `cdm_Array()`, `getTailIndex()`, と `setHeadIndex()`.

このクラスの説明は次のファイルから生成されました:

- [cdm\\_Array.h](#)
- [cdm\\_Array\\_inline.h](#)

## 6.3 クラス cdm\_DFI

```
#include <cdm_DFI.h>
```

cdm\_DFI に対する継承グラフ

cdm\_DFI のコラボレーション図

### Public メソッド

- [cdm\\_DFI](#) ()
- `virtual ~cdm_DFI` ()
- `const cdm_FileInfo * GetcdmFileInfo` ()  
`cdmFileInfo` クラスのポインタを取得
- `void SetcdmFileInfo (cdm_FileInfo FInfo)`  
`cdm_FileInfo` クラスのセット
- `const cdm_FilePath * GetcdmFilePath` ()

- cdm\_FilePath* クラスのポインタを取得
- void [SetcdmFilePath](#) (*cdm\_FilePath* FPath)
- cdm\_FilePath* クラスのセット
- const [cdm\\_VisIt](#) \* [GetcdmVisIt](#) ()
- cdm\_VisIt* クラスのポインタを取得
- void [SetcdmVisIt](#) (*cdm\_VisIt* Visit)
- cdm\_VisIt* クラスのセット
- const [cdm\\_Unit](#) \* [GetcdmUnit](#) ()
- cdm\_Unit* クラスのポインタを取得
- void [SetcdmUnit](#) (*cdm\_Unit* unit)
- cdm\_Unit* クラスのセット
- const [cdm\\_Domain](#) \* [GetcdmDomain](#) ()
- cdm\_Domain* クラスのポインタ取得
- void [SetcdmDomain](#) (*cdm\_Domain* \*domain)
- cdm\_Domain* クラスのセット
- const [cdm\\_MPI](#) \* [GetcdmMPI](#) ()
- cdm\_MPI* クラスのポインタ取得
- void [SetcdmMPI](#) (*cdm\_MPI* mpi)
- cdm\_MPI* クラスセット
- const [cdm\\_TimeSlice](#) \* [GetcdmTimeSlice](#) ()
- cdm\_TimeSlice* クラスのポインタ取得
- void [SetcdmTimeSlice](#) (*cdm\_TimeSlice* TSlice)
- cdm\_TimeSlice* クラスセット
- const [cdm\\_Process](#) \* [GetcdmProcess](#) ()
- cdm\_Process* クラスのポインタ取得
- void [SetcdmProcess](#) (*cdm\_Process* Process)
- cdm\_Process* クラスセット
- virtual const [cdm\\_FieldFileNameFormat](#) \* [GetcdmFieldFileNameFormat](#) ()
- cdm\_FieldFileNameFormat* クラスのポインタ取得
- virtual void [SetcdmFieldFileNameFormat](#) (*cdm\_FieldFileNameFormat* FieldFileNameFormat)
- cdm\_FieldFileNameFormat* クラスセット
- void [SetcdmRankNoPrefix](#) (std::string prefix)
- cdm\_FileInfo* の *RankNoPrefix* をセット フィールドファイル名のランク番号前文字列を変更する
- std::string [GetcdmRankNoPrefix](#) ()
- cdm\_FileInfo* の *RankNoPrefix* を取得
- std::string [Generate\\_FieldFileName](#) (int RankID, int step, const bool mio)
- フィールドデータ ( *SPH,BOV* ) ファイル名の作成 (ディレクトリパスが付加されている)
- void [set\\_RankID](#) (const int rankID)
- RankID* をセットする
- void [set\\_interp\\_flag](#) (bool interp\_flag)
- 節点への補間フラグをセット
- void [set\\_input\\_type](#) (CDM::E\_CDM\_FILE\_TYPE input\_type)
- 入力形式 (*ascii,binary,FortranBinary*) をセット
- void [set\\_output\\_type](#) (CDM::E\_CDM\_FILE\_TYPE output\_type)
- 出力形式 (*ascii,binary,FortranBinary*) をセット
- void [set\\_output\\_type\\_coord](#) (CDM::E\_CDM\_FILE\_TYPE output\_type\_coord)
- 座標データの出力形式 (*ascii,binary*) をセット
- void [set\\_output\\_fname](#) (CDM::E\_CDM\_OUTPUT\_FNAME output\_fname)
- 出力ファイル命名規約 (*step\_rank,rank\_step*) をセット
- std::string [get\\_dfi\\_fname](#) ()

DFI ファイル名の取り出し

- `template<class TimeT, class TimeAvrT >`  
`void * ReadData (CDM::E_CDM_ERRORCODE &ret, const unsigned step, const int gc, const int Gvoxel[3],`  
`const int Gdivision[3], const int head[3], const int tail[3], TimeT &time, const bool mode, unsigned &step_avr,`  
`TimeAvrT &time_avr)`  
*read field data record (template function)*
- `template<class T, class TimeT, class TimeAvrT >`  
`CDM::E_CDM_ERRORCODE ReadData (T *val, const unsigned step, const int gc, const int Gvoxel[3], const`  
`int Gdivision[3], const int head[3], const int tail[3], TimeT &time, const bool mode, unsigned &step_avr, Time-`  
`AvrT &time_avr)`  
*read field data record (template function)*
- `CDM::E_CDM_ERRORCODE ReadData (cdm_Array *val, const unsigned step, const int gc, const int`  
`Gvoxel[3], const int Gdivision[3], const int head[3], const int tail[3], double &time, const bool mode, unsigned`  
`&step_avr, double &time_avr)`  
*read field data record*
- `template<class TimeT, class TimeAvrT >`  
`void * ReadCoordinateData (CDM::E_CDM_ERRORCODE &ret, const unsigned step, const int gc, const int`  
`Gvoxel[3], const int Gdivision[3], const int head[3], const int tail[3], TimeT &time, const bool mode, unsigned`  
`&step_avr, TimeAvrT &time_avr)`  
*read xyz(fub) data record (template function)*
- `template<class T, class TimeT, class TimeAvrT >`  
`CDM::E_CDM_ERRORCODE ReadCoordinateData (T *val, const unsigned step, const int gc, const int`  
`Gvoxel[3], const int Gdivision[3], const int head[3], const int tail[3], TimeT &time, const bool mode, unsigned`  
`&step_avr, TimeAvrT &time_avr)`  
*read xyz(fub) data record (template function)*
- `virtual CDM::E_CDM_ERRORCODE ReadCoordinateData (cdm_Array *val, const unsigned step, const int`  
`gc, const int Gvoxel[3], const int Gdivision[3], const int head[3], const int tail[3], double &time, const bool`  
`mode, unsigned &step_avr, double &time_avr)`  
*read xyz(fub) data record*
- `template<class T, class TimeT, class TimeAvrT >`  
`CDM::E_CDM_ERRORCODE WriteData (const unsigned step, TimeT time, const int sz[3], const int nVari,`  
`const int gc, T *val, T *minmax=NULL, bool avr_mode=true, unsigned step_avr=0, TimeAvrT time_avr=0.0)`  
*write field data record (template function)*
- `template<class T, class TimeT, class TimeAvrT >`  
`CDM::E_CDM_ERRORCODE WriteFieldDataFile (const unsigned step, TimeT time, const int sz[3], const int`  
`nVari, const int gc, T *val, bool avr_mode=true, unsigned step_avr=0, TimeAvrT time_avr=0.0)`  
*write field data record (template function)*
- `template<class T, class TimeT >`  
`CDM::E_CDM_ERRORCODE WriteCoordinateData (const unsigned step, TimeT time, const int sz[3], const`  
`int nVari, const int gc, T *val)`  
*write coordinate data record (template function)*
- `CDM::E_CDM_ERRORCODE WriteData (const unsigned step, const int gc, double time, cdm_Array *val,`  
`double *minmax, const bool avr_mode, const unsigned step_avr, double time_avr)`  
*write field data record*
- `CDM::E_CDM_ERRORCODE WriteFieldDataFile (const unsigned step, const int gc, double time, cdm_Array`  
`*val, const bool avr_mode, const unsigned step_avr, double time_avr)`  
*write field data record (not output dfi file)*
- `virtual CDM::E_CDM_ERRORCODE WriteCoordinateData (const unsigned step, const int gc, double time,`  
`cdm_Array *val)`  
*write coordinate data record*
- `CDM::E_CDM_ERRORCODE WriteProcDfiFile (const MPI_Comm comm, const bool out_host, const int cell-`  
`_id, const int bcf_id)`  
*proc DFI ファイル出力コントロール (float)*
- `CDM::E_CDM_ERRORCODE WriteIndexDfiFile ()`

- index DFI ファイル出力 (API 関数)*
- `CDM::E_CDM_ERRORCODE WriteGridFile` (const int \*iblack=NULL)  
*grid ファイル出力コントロール*
- `std::string GetArrayShapeString` ()  
*配列形状を文字列で返す*
- `CDM::E_CDM_ARRAYSHAPE GetArrayShape` ()  
*配列形状を返す*
- `std::string GetDataTimeString` ()  
*get DataType (データタイプの取り出し関数)*
- `CDM::E_CDM_DTYPE GetDataType` ()  
*get DataType (データタイプの取り出し関数)*
- `std::string GetDFITypeString` ()  
*get DFIType (dfi 種別の取り出し関数)*
- `CDM::E_CDM_DFITYPE GetDFIType` ()  
*get DFIType (dfi 種別の取り出し関数)*
- `std::string GetFileFormatString` ()  
*get FileFormat (FileFormat の取り出し関数)*
- `CDM::E_CDM_FORMAT GetFileFormat` ()  
*get FileFormat (FileFormat の取り出し関数)*
- `int GetNumVariables` ()  
*get Number of Variables (変数の個数の取り出し関数)*
- `int GetNumGuideCell` ()
- `const int * GetDFIGlobalVoxel` ()  
*DFI Domain の GlobalVoxel の取り出し*
- `const int * GetDFIGlobalDivision` ()  
*DFI Domain の GlobalDivision の取り出し*
- `void AddUnit` (const std::string Name, const std::string Unit, const double reference, const double difference=0.0, const bool BsetDiff=false)  
*Unit をセットする*
- `template<class T, class TimeT, class TimeAvrT>`  
`void AddTimeSlice` (const unsigned step, TimeT time, T \*minmax=NULL, bool avr\_mode=true, unsigned step\_avr=0, TimeAvrT time\_avr=0.0)  
*TimeSlice をセットする*
- `CDM::E_CDM_ERRORCODE GetUnitElem` (const std::string Name, `cdm_UnitElem` &unit)  
*UnitElem を取得する*
- `CDM::E_CDM_ERRORCODE GetUnit` (const std::string Name, std::string &unit, double &ref, double &diff, bool &bSetDiff)  
*UnitElem のメンバ変数毎に取得する*
- `void SetTimeSliceFlag` (const `CDM::E_CDM_ONOFF` ONOFF)  
*TimeSlice OnOff フラグをセットする*
- `void setVariableName` (int pvari, std::string variName)  
*FileInfo の変数名を登録する*
- `std::string getVariableName` (int pvari)  
*FileInfo の変数名を取得する*
- `CDM::E_CDM_ERRORCODE getVectorMinMax` (const unsigned step, double &vec\_min, double &vec\_max)  
*DFI に出力されている minmax の合成値を取得*
- `CDM::E_CDM_ERRORCODE getMinMax` (const unsigned step, const int variNo, double &min\_value, double &max\_value)
- `CDM::E_CDM_ERRORCODE CheckReadRank` (const `cdm_Domain` \*dfi\_domain, const int head[3], const int tail[3], `CDM::E_CDM_READTYPE` readflag, vector< int > &readRankList)  
*読み込みランクリストの作成*
- `void setIntervalStep` (int interval\_step, int base\_step=0, int start\_step=0, int last\_step=-1)

- 出力インターバルステップの登録
- void `setIntervalTime` (double interval\_time, double dt, double base\_time=0.0, double start\_time=0.0, double last\_time=-1.0)
  - インターバルタイムの登録
- bool `normalizeTime` (const double scale)
  - インターバルの計算に使われる全ての時間をスケールで無次元化する
- void `normalizeBaseTime` (const double scale)
  - インターバルの *base\_time* をスケールで無次元化する
- void `normalizeIntervalTime` (const double scale)
  - インターバルの *interval* をスケールで無次元化する
- void `normalizeStartTime` (const double scale)
  - インターバルの *start\_time* をスケールで無次元化する
- void `normalizeLastTime` (const double scale)
  - インターバルの *last\_time* をスケールで無次元化する
- void `normalizeDeltaT` (const double scale)
  - インターバルの *DeltaT* をスケールで無次元化する
- virtual std::string `getFileNameFromFileList` (const int ID)
  - field data file name* の取得
- virtual `cdm_Array` \* `ReadFieldData` (std::string fname, const unsigned step, double &time, const int sta[3], const int end[3], const int DFI\_head[3], const int DFI\_tail[3], bool avr\_mode, unsigned &avr\_step, double &avr\_time, `CDM::E_CDM_ERRORCODE` &ret)
  - read field data record*
- virtual `CDM::E_CDM_ERRORCODE` `read_HeaderRecord` (`cdm_FILE` \*pFile, bool matchEndian, unsigned step, const int head[3], const int tail[3], int gc, int voxsize[3], double &time)=0
  - フィールドデータファイルのヘッダーレコード読み込み
- virtual `CDM::E_CDM_ERRORCODE` `read_Datarecord` (`cdm_FILE` \*pFile, bool matchEndian, unsigned step, `cdm_Array` \*buf, int head[3], int nz, `cdm_Array` \*&src)=0
  - フィールドデータファイルのデータレコード読み込み
- virtual `CDM::E_CDM_ERRORCODE` `read_averaged` (`cdm_FILE` \*pFile, bool matchEndian, unsigned step, unsigned &avr\_step, double &avr\_time)=0
  - フィールドデータの *Average* データレコードの読み込み
- template<class T1, class T2>
  - bool `setGridData` (`cdm_TypeArray`< T1 > \*P, `cdm_TypeArray`< T2 > \*S)
    - セル中心データを格子点に値をセット
- template<class T>
  - void `VolumeDataDivide` (`cdm_TypeArray`< T > \*P)
    - 内部の格子点のデータを重み付けで割る
- int `MakeDirectory` (const std::string path)
  - ディレクトリパスの作成 (*MakeDirectorySub* を呼出して作成)
- int `MakeDirectoryPath` ()
  - ディレクトリパスの作成 (*MakeDirectory* 関数を呼出して作成)
- std::string `Generate_Directory_Path` ()
  - dfi* のパスと *DirectoryPath* を連結する関数
- int `getBufSize` ()
  - コンパイルオプションで与えたバッファサイズを取得
- virtual bool `CheckAddWriteMode` ()
  - 出力処理を追記モードにするかどうかをチェック (*NetCDF* 用)
- template<class TimeT, class TimeAvrT>
  - `CDM_INLINE` void \* `ReadData` (`CDM::E_CDM_ERRORCODE` &ret, const unsigned step, const int gc, const int Gvoxel[3], const int Gdivision[3], const int head[3], const int tail[3], TimeT &time, const bool mode, unsigned &step\_avr, TimeAvrT &time\_avr)

- `template<class T, class TimeT, class TimeAvrT >`  
`CDM_INLINE CDM::E_CDM_ERRORCODE ReadData` (T \*val, const unsigned step, const int gc, const int Gvoxel[3], const int Gdivision[3], const int head[3], const int tail[3], TimeT &time, const bool mode, unsigned &step\_avr, TimeAvrT &time\_avr)
- `template<class TimeT, class TimeAvrT >`  
`CDM_INLINE void * ReadCoordinateData` (CDM::E\_CDM\_ERRORCODE &ret, const unsigned step, const int gc, const int Gvoxel[3], const int Gdivision[3], const int head[3], const int tail[3], TimeT &time, const bool mode, unsigned &step\_avr, TimeAvrT &time\_avr)
- `template<class T, class TimeT, class TimeAvrT >`  
`CDM_INLINE CDM::E_CDM_ERRORCODE ReadCoordinateData` (T \*val, const unsigned step, const int gc, const int Gvoxel[3], const int Gdivision[3], const int head[3], const int tail[3], TimeT &time, const bool mode, unsigned &step\_avr, TimeAvrT &time\_avr)
- `template<class T, class TimeT, class TimeAvrT >`  
`CDM_INLINE CDM::E_CDM_ERRORCODE WriteData` (const unsigned step, TimeT time, const int sz[3], const int nVari, const int gc, T \*val, T \*minmax, const bool avr\_mode, const unsigned step\_avr, TimeAvrT time\_avr)
- `template<class T, class TimeT, class TimeAvrT >`  
`CDM_INLINE CDM::E_CDM_ERRORCODE WriteFieldDataFile` (const unsigned step, TimeT time, const int sz[3], const int nVari, const int gc, T \*val, const bool avr\_mode, const unsigned step\_avr, TimeAvrT time\_avr)
- `template<class T, class TimeT >`  
`CDM_INLINE CDM::E_CDM_ERRORCODE WriteCoordinateData` (const unsigned step, TimeT time, const int sz[3], const int nVari, const int gc, T \*val)
- `template<class T, class TimeT, class TimeAvrT >`  
`CDM_INLINE void AddTimeSlice` (const unsigned step, TimeT time, T \*minmax, bool avr\_mode, unsigned step\_avr, TimeAvrT time\_avr)
- `template<class T1, class T2 >`  
`CDM_INLINE bool setGridData` (cdm\_TypeArray< T1 > \*P, cdm\_TypeArray< T2 > \*S)
- `template<class T >`  
`CDM_INLINE void VolumeDataDivide` (cdm\_TypeArray< T > \*P)
- `template<typename T >`  
`CDM_INLINE cdm_DFI * Writelnit` (const MPI\_Comm comm, const std::string DfiName, const std::string Path, const std::string prefix, const CDM::E\_CDM\_FORMAT format, const int GCell, const CDM::E\_CDM\_DTYPE DataType, const int nVari, const std::string proc\_fname, const int G\_size[3], const T pitch[3], const T L\_origin[3], const int division[3], const int head[3], const int tail[3], const std::string hostname, const CDM::E\_CDM\_ONOFF TSliceOnOff)
- `template<typename T >`  
`CDM_INLINE cdm_DFI * Writelnit` (const MPI\_Comm comm, const std::string DfiName, const std::string Path, const std::string prefix, const CDM::E\_CDM\_FORMAT format, const int GCell, const CDM::E\_CDM\_DTYPE DataType, const int nVari, const std::string proc\_fname, const int G\_size[3], const T \*coord\_X, const T \*coord\_Y, const T \*coord\_Z, const std::string coord\_file, const CDM::E\_CDM\_FILE\_TYPE coord\_filetype, const CDM::E\_CDM\_ENDIANTYPE coord\_fileEndian, const int division[3], const int head[3], const int tail[3], const std::string hostname, const CDM::E\_CDM\_ONOFF TSliceOnOff)
- `template<typename T >`  
`CDM_INLINE cdm_DFI * Writelnit` (const MPI\_Comm comm, const std::string DfiName, const std::string Path, const std::string prefix, const CDM::E\_CDM\_FORMAT format, const int GCell, const CDM::E\_CDM\_DTYPE DataType, const int nVari, const std::string proc\_fname, const cdm\_Domain \*out\_domain, const int head[3], const int tail[3], const std::string hostname, const CDM::E\_CDM\_ONOFF TSliceOnOff)

## Static Public メソッド

- static `cdm_DFI * ReadInit` (const MPI\_Comm comm, const std::string dfifile, const int G\_Voxel[3], const int G\_Div[3], CDM::E\_CDM\_ERRORCODE &ret)  
*read インスタンス (BOV もしくは PLOT3D の場合にインスタンス生成)*
- static `std::string Generate_DFI_Name` (const std::string prefix)  
*出力 DFI ファイル名を作成する*

- static std::string `Generate_FileName` (std::string prefix, int RankID, int step, std::string ext, `CDM::E_CDM_OUTPUT_FNAME` output\_fname, bool mio, `CDM::E_CDM_ONOFF` TimeSliceDirFlag, std::string RankNoPrefix=std::string(`CDM::C_CDM_RANKNOPREFIX`))  
ファイル名生成
- template<typename T >  
static `cdm_DFI` \* `Writelnit` (const `MPI_Comm` comm, const std::string DfiName, const std::string Path, const std::string prefix, const `CDM::E_CDM_FORMAT` format, const int GCell, const `CDM::E_CDM_DTYPE` DataType, const int nVari, const std::string proc\_fname, const int G\_size[3], const T pitch[3], const T L\_origin[3], const int division[3], const int head[3], const int tail[3], const std::string hostname, const `CDM::E_CDM_ONOFF` TSliceOnOff)  
write インスタンス *template function* (等間隔格子用)
- template<typename T >  
static `cdm_DFI` \* `Writelnit` (const `MPI_Comm` comm, const std::string DfiName, const std::string Path, const std::string prefix, const `CDM::E_CDM_FORMAT` format, const int GCell, const `CDM::E_CDM_DTYPE` DataType, const int nVari, const std::string proc\_fname, const int G\_size[3], const T \*coord\_X, const T \*coord\_Y, const T \*coord\_Z, const std::string coord\_file, const `CDM::E_CDM_FILE_TYPE` coord\_filetype, const `CDM::E_CDM_ENDIANTYPE` coord\_fileEndian, const int division[3], const int head[3], const int tail[3], const std::string hostname, const `CDM::E_CDM_ONOFF` TSliceOnOff)  
write インスタンス *template function* (不等間隔格子用)
- template<typename T >  
static `cdm_DFI` \* `Writelnit` (const `MPI_Comm` comm, const std::string DfiName, const std::string Path, const std::string prefix, const `CDM::E_CDM_FORMAT` format, const int GCell, const `CDM::E_CDM_DTYPE` DataType, const int nVari, const std::string proc\_fname, const `cdm_Domain` \*out\_domain, const int head[3], const int tail[3], const std::string hostname, const `CDM::E_CDM_ONOFF` TSliceOnOff)  
write インスタンス *template function* (等間隔格子・不等間隔格子の共通処理部分)
- static `CDM::E_CDM_DTYPE` `ConvDatatypeS2E` (const std::string datatype)  
データタイプを文字列から `e_num` 番号に変換
- static std::string `ConvDatatypeE2S` (const `CDM::E_CDM_DTYPE` Dtype)  
データタイプを `e_num` 番号から文字列に変換
- static int `MakeDirectorySub` (std::string path)  
ディレクトリパスの作成 (*system* 関数 `mkdir` で作成)
- static std::string `getVersionInfo` ()

## Protected メソッド

- virtual `CDM::E_CDM_ERRORCODE` `WriteFieldData` (std::string fname, const unsigned step, double time, `cdm_Array` \*val, const bool mode, const unsigned step\_avr, const double time\_avr)  
write field data record (double)
- virtual `CDM::E_CDM_ERRORCODE` `write_HeaderRecord` (`cdm_FILE` \*pFile, const unsigned step, const double time, const int RankID)=0  
フィールドデータのヘッダレコードの出力 (純粋仮想関数)
- virtual `CDM::E_CDM_ERRORCODE` `write_DataRecord` (`cdm_FILE` \*pFile, `cdm_Array` \*val, const int gc, const int RankID)=0  
フィールドデータのデータレコードの出力 (純粋仮想関数)
- virtual `CDM::E_CDM_ERRORCODE` `write_averaged` (`cdm_FILE` \*pFile, const unsigned step\_avr, const double time\_avr)=0  
Average レコードの出力 (純粋仮想関数)
- virtual bool `write_GridData` (const int \*iblack)  
Grid data file 出力 コントロール
- virtual bool `write_ascii_header` (const unsigned step, const double time)  
ascii ヘッダレコード出力 (*bov,avs*)
- void `cdm_Create_dfiProcessInfo` (const `MPI_Comm` comm, `cdm_Process` &G\_Process)  
Create Process.



- [CDM::E\\_CDM\\_READTYPE CheckReadType](#) (const int G\_voxel[3], const int DFI\_GlobalVoxel[3], const int G\_Div[3], const int DFI\_GlobalDivision[3])  
読み判定判定
- void [CreateReadStartEnd](#) (bool isSame, const int head[3], const int tail[3], const int gc, const int DFI\_head[3], const int DFI\_tail[3], const int DFI\_gc, const [CDM::E\\_CDM\\_READTYPE](#) readflag, int copy\_sta[3], int copy\_end[3], int read\_sta[3], int read\_end[3])  
フィールドデータの読み範囲を求める
- [CDM::E\\_CDM\\_ERRORCODE WriteIndexDfiFile](#) (const std::string dfi\_name)  
*index DFI ファイル出力*

## Static Protected メソッド

- static int [get\\_cdm\\_Datasize](#) ([CDM::E\\_CDM\\_DTYPE](#) Dtype)  
データタイプ毎のサイズを取得

## Protected 変数

- [MPI\\_Comm m\\_comm](#)  
*MPI コミュニケータ*
- std::string [m\\_directoryPath](#)  
*index dfi ファイルのディレクトリパス*
- std::string [m\\_indexDfiName](#)  
*index dfi ファイル名*
- [CDM::E\\_CDM\\_READTYPE m\\_read\\_type](#)  
読みタイプ
- int [m\\_RankID](#)  
ランク番号
- [cdm\\_FileInfo DFI\\_Finfo](#)  
*FileInfo class.*
- [cdm\\_FilePath DFI\\_Fpath](#)  
*FilePath class.*
- [cdm\\_VisIt DFI\\_VisIt](#)  
*VisIt class.*
- [cdm\\_Unit DFI\\_Unit](#)  
*Unit class.*
- const [cdm\\_Domain](#) \* [DFI\\_Domain](#)  
*Domain class.*
- [cdm\\_MPI DFI\\_MPI](#)  
*MPI class.*
- [cdm\\_TimeSlice DFI\\_TimeSlice](#)  
*TimeSlice class.*
- [cdm\\_Process DFI\\_Process](#)  
*Process class.*
- vector< int > [m\\_readRankList](#)  
読みランクリスト
- bool [m\\_bgrid\\_interp\\_flag](#)  
節点への補間フラグ
- [CDM::E\\_CDM\\_FILE\\_TYPE m\\_input\\_type](#)  
入力形式 (*ascii, binary, FortranBinary*)
- [CDM::E\\_CDM\\_FILE\\_TYPE m\\_output\\_type](#)



- 出力形式 (*ascii, binary, FortranBinary*)
- CDM::E\_CDM\_FILE\_TYPE *m\_output\_type\_coord*  
座標データの出力形式 (*ascii, binary*)
- CDM::E\_CDM\_OUTPUT\_FNAME *m\_output\_fname*  
出力ファイル命名規約 (*step\_rank, rank\_step*)

### 6.3.1 説明

CDM main class

cdm\_DFI.h の 51 行で定義されています。

### 6.3.2 コンストラクタとデストラクタ

#### 6.3.2.1 cdm\_DFI::cdm\_DFI ( )

コンストラクタ

#### 6.3.2.2 virtual cdm\_DFI::~cdm\_DFI ( ) [virtual]

デストラクタ

### 6.3.3 関数

#### 6.3.3.1 template<class T, class TimeT, class TimeAvrT> CDM\_INLINE void cdm\_DFI::AddTimeSlice ( const unsigned *step*, TimeT *time*, T \* *minmax*, bool *avr\_mode*, unsigned *step\_avr*, TimeAvrT *time\_avr* )

cdm\_DFI\_inline.h の 369 行で定義されています。

参照先 cdm\_TimeSlice::AddSlice(), DFI\_Finfo, DFI\_TimeSlice, CDM::E\_CDM\_FMT\_SPH, cdm\_FileInfo::File-Format, と cdm\_FileInfo::NumVariables.

#### 6.3.3.2 template<class T, class TimeT, class TimeAvrT> void cdm\_DFI::AddTimeSlice ( const unsigned *step*, TimeT *time*, T \* *minmax* = NULL, bool *avr\_mode* = true, unsigned *step\_avr* = 0, TimeAvrT *time\_avr* = 0.0 )

TimeSlice をセットする

引数

in	<i>step</i>	出力ステップ番号
in	<i>time</i>	出力時刻
in	<i>minmax</i>	フィールドデータのMinMax
in	<i>avr_mode</i>	平均ステップ&時間出力 false : 出力 true : 出力しない
in	<i>step_avr</i>	平均ステップ
in	<i>time_avr</i>	平均時間

#### 6.3.3.3 void cdm\_DFI::AddUnit ( const std::string *Name*, const std::string *Unit*, const double *reference*, const double *difference* = 0.0, const bool *BsetDiff* = false )

Unit をセットする

## 引数

in	<i>Name</i>	追加する単位系 ("Length","Velocity",...)
in	<i>Unit</i>	単位ラベル ("M","CM","MM","M/S",...)
in	<i>reference</i>	規格化したスケール値
in	<i>difference</i>	差の値
in	<i>BsetDiff</i>	difference の有無

6.3.3.4 void cdm\_DFI::cdm\_Create\_dfiProcessInfo ( const MPI\_Comm comm, cdm\_Process & G\_Process )  
[protected]

Create Process.

## 引数

in	<i>comm</i>	MPI コミュニケータ
out	<i>G_Process</i>	Process class

6.3.3.5 virtual bool cdm\_DFI::CheckAddWriteMode ( ) [inline],[virtual]

出力処理を追記モードにするかどうかをチェック (NetCDF 用)

## 戻り値

モード (true:追記モード、false:新規作成モード)

[cdm\\_DFI\\_NETCDF](#)で再定義されています。

cdm\_DFI.h の 1434 行で定義されています。

6.3.3.6 CDM::E\_CDM\_ERRORCODE cdm\_DFI::CheckReadRank ( const cdm\_Domain \* dfi\_domain, const int head[3],  
const int tail[3], CDM::E\_CDM\_READTYPE readflag, vector< int > & readRankList )

## 読み込みランクリストの作成

RankList があるかないか判定しないときは新規にRankList を生成し それをもとにランクマップの生成、読み込みランクリスト readRankList を生成する

## 引数

in	<i>dfi_domain</i>	DFI の domain 情報
in	<i>head</i>	ソルバーのHeadIndex
in	<i>tail</i>	ソルバーのTailIndex
in	<i>readflag</i>	読み込み方法
out	<i>readRankList</i>	読み込みランクリスト

## 戻り値

error code

6.3.3.7 CDM::E\_CDM\_READTYPE cdm\_DFI::CheckReadType ( const int G\_voxel[3], const int DFI\_GlobalVoxel[3], const  
int G\_Div[3], const int DFI\_GlobalDivision[3] ) [protected]

## 読み込み判定判定

引数

in	<i>G_voxel</i>	計算空間全体のボクセルサイズ (自)
in	<i>DFI_GlobalVoxel</i>	計算空間全体のボクセルサイズ (DFI)
in	<i>G_Div</i>	分割数 (自)
in	<i>DFI_Global-Division</i>	分割数 (DFI)

戻り値

読み込みタイプコード

**6.3.3.8** `static std::string cdm_DFI::ConvDatatypeE2S ( const CDM::E_CDM_DTYPE Dtype ) [static]`

データタイプを `e_num` 番号から文字列に変換

引数

in	<i>Dtype</i>	データタイプ
----	--------------	--------

戻り値

データタイプ (string)

参照元 `cdm_NonUniformDomain< T >::Write()`.

**6.3.3.9** `static CDM::E_CDM_DTYPE cdm_DFI::ConvDatatypeS2E ( const std::string datatype ) [static]`

データタイプを文字列から `e_num` 番号に変換

引数

in	<i>datatype</i>	dfi から取得したデータタイプ
----	-----------------	------------------

戻り値

データタイプ (E\_CDM\_DTYPE)

**6.3.3.10** `void cdm_DFI::CreateReadStartEnd ( bool isSame, const int head[3], const int tail[3], const int gc, const int DFI_head[3], const int DFI_tail[3], const int DFI_gc, const CDM::E_CDM_READTYPE readflag, int copy_sta[3], int copy_end[3], int read_sta[3], int read_end[3] ) [protected]`

フィールドデータの読み込み範囲を求める

引数

in	<i>isSame</i>	粗密フラグ true:密、false:粗
in	<i>head</i>	計算領域の開始位置 (自)
in	<i>tail</i>	計算領域の終了位置 (自)
in	<i>gc</i>	仮想セル数 (自)
in	<i>DFI_head</i>	計算領域の開始位置 (DFI)

in	<i>DFI_tail</i>	計算領域の終了位置 (DFI)
in	<i>DFI_gc</i>	仮想セル数 (DFI)
in	<i>readflag</i>	読み込み方法
out	<i>copy_sta</i>	コピー開始位置
out	<i>copy_end</i>	コピー終了位置
out	<i>read_sta</i>	読み込み開始位置
out	<i>read_end</i>	読み込み終了位置

#### 6.3.3.11 static std::string cdm\_DFI::Generate\_DFI\_Name ( const std::string *prefix* ) [static]

出力DFI ファイル名を作成する

引数

in	<i>prefix</i>	ファイル接頭文字
----	---------------	----------

戻り値

DFI ファイル名

#### 6.3.3.12 std::string cdm\_DFI::Generate\_Directory\_Path ( )

dfi のパスとDirectoryPath を連結する関数

戻り値

パス名

#### 6.3.3.13 std::string cdm\_DFI::Generate\_FieldFileName ( int *RankID*, int *step*, const bool *mio* )

フィールドデータ (SPH,BOV) ファイル名の作成 (ディレクトリパスが付加されている)

引数

in	<i>RankID</i>	ランク番号
in	<i>step</i>	読み込みステップ番号
in	<i>mio</i>	並列判定フラグ (逐次 or 並列の判定用)

戻り値

生成されたファイル名

#### 6.3.3.14 static std::string cdm\_DFI::Generate\_FileName ( std::string *prefix*, int *RankID*, int *step*, std::string *ext*, CDM::E\_CDM\_OUTPUT\_FNAME *output\_fname*, bool *mio*, CDM::E\_CDM\_ONOFF *TimeSliceDirFlag*, std::string *RankNoPrefix* = std::string(CDM::C\_CDM\_RANKNOPREFIX) ) [static]

ファイル名生成

引数

in	<i>prefix</i>	ベースファイル名
in	<i>RankID</i>	ランク番号
in	<i>step</i>	出力ステップ番号 (負のとき、ステップ番号が付加されない)
in	<i>ext</i>	拡張子
in	<i>output_fname</i>	step_rank,rank_step 指示
in	<i>mio</i>	並列判定フラグ
in	<i>TimeSliceDir-Flag</i>	Time Slice 毎の出力指示
in	<i>RankNoPrefix</i>	ファイル名内のランク番号前の文字列 ("_id")

戻り値

生成されたファイル名

**6.3.3.15** static int cdm\_DFI::get\_cdm\_Datasize ( CDM::E\_CDM\_DTYPE Dtype ) [static],[protected]

データタイプ毎のサイズを取得

引数

in	<i>Dtype</i>	データタイプ (Int8,Int16,...etc)
----	--------------	----------------------------

戻り値

データサイズ  
0 エラー

**6.3.3.16** std::string cdm\_DFI::get\_dfi\_fname ( ) [inline]

DFI ファイル名の取り出し

戻り値

dfi ファイル名

cdm\_DFI.h の 454 行で定義されています。

参照先 m\_indexDfiName.

**6.3.3.17** CDM::E\_CDM\_ARRAYSHAPE cdm\_DFI::GetArrayShape ( )

配列形状を返す

戻り値

配列形状 ( e\_num 番号)

**6.3.3.18** std::string cdm\_DFI::GetArrayShapeString ( )

配列形状を文字列で返す

戻り値

配列形状 ( 文字列)

#### 6.3.3.19 CDM\_INLINE int cdm\_DFI::getBufSize ( )

コンパイルオプションで与えたバッファサイズを取得

戻り値

バッファサイズ

cdm\_DFI\_inline.h の 839 行で定義されています。

#### 6.3.3.20 const cdm\_Domain\* cdm\_DFI::GetcdmDomain ( )

cdm\_Domain クラスのポインタ取得

戻り値

cdm\_Domain クラスポインタ

#### 6.3.3.21 virtual const cdm\_FieldFileNameFormat\* cdm\_DFI::GetcdmFieldFileNameFormat ( ) [inline], [virtual]

cdm\_FieldFileNameFormat クラスのポインタ取得

戻り値

cdm\_FieldFileNameFormat クラスのポインタ

[cdm\\_DFI\\_FUB](#)で再定義されています。

cdm\_DFI.h の 205 行で定義されています。

#### 6.3.3.22 const cdm\_FileInfo\* cdm\_DFI::GetcdmFileInfo ( )

cdmFileInfo クラスのポインタを取得

戻り値

cdm\_FileInfo クラスポインタ

#### 6.3.3.23 const cdm\_FilePath\* cdm\_DFI::GetcdmFilePath ( )

cdm\_FilePath クラスのポインタを取得

戻り値

cdm\_FilePath クラスポインタ

#### 6.3.3.24 const cdm\_MPI\* cdm\_DFI::GetcdmMPI ( )

cdm\_MPI クラスのポインタ取得

戻り値

cdm\_MPI クラスポインタ

**6.3.3.25** `const cdm_Process* cdm_DFI::GetcdmProcess ( )`

cdm\_Process クラスのポインタ取得

戻り値

cdm\_Process クラスポインタ

**6.3.3.26** `std::string cdm_DFI::GetcdmRankNoPrefix ( )`

cdm\_FileInfo のRankNoPrefix を取得

戻り値

ランク番号前文字列

**6.3.3.27** `const cdm_TimeSlice* cdm_DFI::GetcdmTimeSlice ( )`

cdm\_TimeSlice クラスのポインタ取得

戻り値

cdm\_TimeSlice クラスポインタ

**6.3.3.28** `const cdm_Unit* cdm_DFI::GetcdmUnit ( )`

cdm\_Unit クラスのポインタを取得

戻り値

cdm\_Unit クラスポインタ

**6.3.3.29** `const cdm_VisIt* cdm_DFI::GetcdmVisIt ( )`

cdm\_VisIt クラスのポインタを取得

戻り値

cdm\_VisIt クラスポインタ

**6.3.3.30** `CDM::E_CDM_DTYPE cdm_DFI::GetDataType ( )`

get DataType (データタイプの取り出し関数)

戻り値

データタイプ (e\_num 番号)

**6.3.3.31** `std::string cdm_DFI::GetDataTimeString ( )`

get DataType ( データタイプの取り出し関数 )

戻り値

データタイプ ( 文字列 )

**6.3.3.32** `const int* cdm_DFI::GetDFIGlobalDivision ( )`

DFI Domain のGlobalDivision の取り出し

戻り値

GlobalDivision のポインタ

**6.3.3.33** `const int* cdm_DFI::GetDFIGlobalVoxel ( )`

DFI Domain のGlobalVoxel の取り出し

戻り値

GlobalVoxel のポインタ

**6.3.3.34** `CDM::E_CDM_DFITYPE cdm_DFI::GetDFIType ( )`

get DFIType ( dfi 種別の取り出し関数 )

戻り値

dfi 種別 (e\_num 番号)

**6.3.3.35** `std::string cdm_DFI::GetDFITypeString ( )`

get DFIType ( dfi 種別の取り出し関数 )

戻り値

dfi 種別 ( 文字列 )

**6.3.3.36** `CDM::E_CDM_FORMAT cdm_DFI::GetFileFormat ( )`

get FileFormat (FileFormat の取り出し関数)

戻り値

FileFormat(e\_num 番号)



## 6.3.3.37 std::string cdm\_DFI::GetFileFormatString ( )

get FileFormat ( FileFormat の取り出し関数 )

戻り値

FileFormat(文字列)

6.3.3.38 virtual std::string cdm\_DFI::getFileNameFromFileList ( const int *ID* ) [inline],[virtual]

field data file name の取得

index dfi ファイルの/FileList/Rank のID と読み込み指示されたランクのID と一致したID のRank で定義された FieldDataFileName で指示されたファイル名もしくは CoordinateFileName で指示されたファイル名を取得 ( E\_CDM\_FMT\_FUB のときは FieldDataFileName、 E\_CDM\_FMT\_FUB\_COD のときはCoordinateFileName) index dfi ファイルにFileList がないファイルは空白が戻る.

引数

in	ID	RankID
----	----	--------

戻り値

field data file name

cdm\_DFI.h の 1118 行で定義されています。

6.3.3.39 CDM::E\_CDM\_ERRORCODE cdm\_DFI::getMinMax ( const unsigned *step*, const int *variNo*, double & *min\_value*, double & *max\_value* )

brief DFI に出力されている minmax を取得

引数

in	<i>step</i>	取得するステップ
in	<i>variNo</i>	変数No(0 ~ n)
out	<i>min_value</i>	取得した min
out	<i>max_value</i>	取得した max

戻り値

error code 取得出来たときは E\_CDM\_SUCCESS

## 6.3.3.40 int cdm\_DFI::GetNumGuideCell ( )

## 6.3.3.41 int cdm\_DFI::GetNumVariables ( )

get Number of Variables ( 変数の個数の取り出し関数 )

戻り値

変数の個数

6.3.3.42 CDM::E\_CDM\_ERRORCODE cdm\_DFI::GetUnit ( const std::string *Name*, std::string & *unit*, double & *ref*, double & *diff*, bool & *bSetDiff* )

UnitElem のメンバ変数毎に取得する

引数

in	<i>Name</i>	取得する単位系
out	<i>unit</i>	単位文字列
out	<i>ref</i>	reference
out	<i>diff</i>	difference
out	<i>bSetDiff</i>	difference の有無 ( true:あり false:なし )

戻り値

error code

#### 6.3.3.43 CDM::E\_CDM\_ERRORCODE cdm\_DFI::GetUnitElem ( const std::string *Name*, cdm\_UnitElem & *unit* )

UnitElem を取得する

引数

in	<i>Name</i>	取得する単位系
out	<i>unit</i>	取得した cdm_UnitElem

戻り値

error code

#### 6.3.3.44 std::string cdm\_DFI::getVariableName ( int *pvari* )

FileInfo の変数名を取得する

引数

in	<i>pvari</i>	変数位置 0:u, 1:v, 2:w
----	--------------	--------------------

戻り値

変数名

#### 6.3.3.45 CDM::E\_CDM\_ERRORCODE cdm\_DFI::getVectorMinMax ( const unsigned *step*, double & *vec\_min*, double & *vec\_max* )

DFI に出力されている minmax の合成値を取得

引数

in	<i>step</i>	取得するステップ
out	<i>vec_min</i>	取得した minmax の合成値
out	<i>vec_max</i>	取得した minmax の合成値

戻り値

error code 取得出来たときは E\_CDM\_SUCCESS

**6.3.3.46** `static std::string cdm_DFI::getVersionInfo ( ) [inline],[static]`

バージョンを出力する

cdm\_DFI.h の 1417 行で定義されています。

参照先 CDM\_VERSION\_NO.

**6.3.3.47** `int cdm_DFI::MakeDirectory ( const std::string path )`

ディレクトリパスの作成 (MakeDirectorySub を呼出して作成)

引数

<i>in</i>	<i>path</i>	パス
-----------	-------------	----

戻り値

error code

**6.3.3.48** `int cdm_DFI::MakeDirectoryPath ( )`

ディレクトリパスの作成 (MakeDirectory 関数を呼出して作成)

戻り値

error code

**6.3.3.49** `static int cdm_DFI::MakeDirectorySub ( std::string path ) [static]`

ディレクトリパスの作成 (system 関数 mkdir で作成)

引数

<i>in</i>	<i>path</i>	パス
-----------	-------------	----

戻り値

error code

**6.3.3.50** `void cdm_DFI::normalizeBaseTime ( const double scale )`

インターバルの base\_time をスケールで無次元化する

引数

<i>in</i>	<i>scale</i>	スケール
-----------	--------------	------

**6.3.3.51** `void cdm_DFI::normalizeDeltaT ( const double scale )`

インターバルの DeltaT をスケールで無次元化する

引数

<i>in</i>	<i>scale</i>	スケール
-----------	--------------	------

6.3.3.52 void *cdm\_DFI::normalizeIntervalTime* ( const double *scale* )

インターバルの *interval* をスケールで無次元化する

引数

<i>in</i>	<i>scale</i>	スケール
-----------	--------------	------

6.3.3.53 void *cdm\_DFI::normalizeLastTime* ( const double *scale* )

インターバルの *last\_time* をスケールで無次元化する

引数

<i>in</i>	<i>scale</i>	スケール
-----------	--------------	------

6.3.3.54 void *cdm\_DFI::normalizeStartTime* ( const double *scale* )

インターバルの *start\_time* をスケールで無次元化する

引数

<i>in</i>	<i>scale</i>	スケール
-----------	--------------	------

6.3.3.55 bool *cdm\_DFI::normalizeTime* ( const double *scale* )

インターバルの計算に使われる全ての時間をスケールで無次元化する

(*base\_time*, *interval\_time*, *start\_time*, *last\_time*)

引数

<i>in</i>	<i>scale</i>	スケール return mode がStep のときは false を返す、無次元化しない
-----------	--------------	---

6.3.3.56 virtual CDM::E\_CDM\_ERRORCODE *cdm\_DFI::read\_averaged* ( *cdm\_FILE* \* *pFile*, bool *matchEndian*, unsigned *step*, unsigned & *avr\_step*, double & *avr\_time* ) [pure virtual]

フィールドデータのAverage データレコードの読み込み

引数

<i>in</i>	<i>pFile</i>	ファイルポインタ
<i>in</i>	<i>matchEndian</i>	true:Endian 一致
<i>in</i>	<i>step</i>	読み込み step 番号
<i>out</i>	<i>avr_step</i>	平均ステップ
<i>out</i>	<i>avr_time</i>	平均タイム

[cdm\\_DFI\\_NETCDF](#), [cdm\\_DFI\\_FUB](#), [cdm\\_DFI\\_SPH](#), [cdm\\_DFI\\_AVS](#), [cdm\\_DFI\\_VTK](#), [cdm\\_DFI\\_PLOT3D](#), と [cdm\\_DFI\\_BOV](#) で実装されています。

```
6.3.3.57 virtual CDM::E_CDM_ERRORCODE cdm_DFI::read_Datarecord ( cdm_FILE * pFile, bool matchEndian,  
    unsigned step, cdm_Array * buf, int head[3], int nz, cdm_Array *&src ) [pure virtual]
```

フィールドデータファイルのデータレコード読み込み

## 引数

in	<i>pFile</i>	ファイルポインタ
in	<i>matchEndian</i>	true:Endian 一致
in	<i>step</i>	ステップ番号
in	<i>buf</i>	読み込み用バッファ
in	<i>head</i>	読み込みバッファHeadIndex
in	<i>nz</i>	z 方向のボクセルサイズ ( 実セル + ガイドセル * 2 )
out	<i>src</i>	読み込んだデータを格納した配列のポインタ

[cdm\\_DFI\\_NETCDF](#), [cdm\\_DFI\\_FUB](#), [cdm\\_DFI\\_SPH](#), [cdm\\_DFI\\_AVS](#), [cdm\\_DFI\\_PLOT3D](#), [cdm\\_DFI\\_VTK](#), と [cdm\\_DFI\\_BOV](#) で実装されています。

6.3.3.58 `virtual CDM::E_CDM_ERRORCODE cdm_DFI::read_HeaderRecord ( cdm_FILE * pFile, bool matchEndian, unsigned step, const int head[3], const int tail[3], int gc, int voysize[3], double & time ) [pure virtual]`

## フィールドデータファイルのヘッダーレコード読み込み

## 引数

in	<i>pFile</i>	ファイルポインタ
in	<i>matchEndian</i>	true:Endian 一致
in	<i>step</i>	ステップ番号
in	<i>head</i>	dfi のHeadIndex
in	<i>tail</i>	dfi のTailIndex
in	<i>gc</i>	dfi のガイドセル数
out	<i>voysize</i>	voysize
out	<i>time</i>	時刻

## 戻り値

true:出力成功 false:出力失敗

[cdm\\_DFI\\_NETCDF](#), [cdm\\_DFI\\_FUB](#), [cdm\\_DFI\\_SPH](#), [cdm\\_DFI\\_AVS](#), [cdm\\_DFI\\_PLOT3D](#), [cdm\\_DFI\\_VTK](#), と [cdm\\_DFI\\_BOV](#) で実装されています。

6.3.3.59 `template<class TimeT , class TimeAvrT > CDM_INLINE void* cdm_DFI::ReadCoordinateData ( CDM::E_CDM_ERRORCODE & ret, const unsigned step, const int gc, const int Gvoxel[3], const int Gdivision[3], const int head[3], const int tail[3], TimeT & time, const bool mode, unsigned & step_avr, TimeAvrT & time_avr )`

`cdm_DFI_inline.h` の 145 行で定義されています。

参照先 `cdm_FileInfo::ArrayShape`, `cdm_FileInfo::DataType`, `DFI_Finfo`, `CDM::E_CDM_SUCCESS`, `cdm_Array::getData()`, `cdm_Array::instanceArray()`, `cdm_FileInfo::NumVariables`, と `ReadCoordinateData()`.

6.3.3.60 `template<class T , class TimeT , class TimeAvrT > CDM_INLINE CDM::E_CDM_ERRORCODE cdm_DFI::ReadCoordinateData ( T * val, const unsigned step, const int gc, const int Gvoxel[3], const int Gdivision[3], const int head[3], const int tail[3], TimeT & time, const bool mode, unsigned & step_avr, TimeAvrT & time_avr )`

`cdm_DFI_inline.h` の 190 行で定義されています。

参照先 `cdm_FileInfo::ArrayShape`, `DFI_Finfo`, `CDM::E_CDM_SUCCESS`, `cdm_Array::instanceArray()`, `cdm_FileInfo::NumVariables`, と `ReadCoordinateData()`.

6.3.3.61 `template<class TimeT , class TimeAvrT > void* cdm_DFI::ReadCoordinateData ( CDM::E_CDM_ERRORCODE & ret, const unsigned step, const int gc, const int Gvoxel[3], const int Gdivision[3], const int head[3], const int tail[3], TimeT & time, const bool mode, unsigned & step_avr, TimeAvrT & time_avr )`

read xyz(fub) data record (template function)

読み込んだデータのポインタを戻り値として返す

引数

out	ret	終了コード 1:正常、1 以外 : エラー
in	step	入力ステップ番号
in	gc	仮想セル数
in	Gvoxel	グローバルボクセルサイズ
in	Gdivision	領域分割数
in	head	計算領域の開始位置
in	tail	計算領域の終了位置
out	time	読み込んだ時間
in	mode	平均ステップ & 時間読み込みフラグ false : 読み込み true : 読み込まない
out	step_avr	平均ステップ
out	time_avr	平均時間

戻り値

読み込んだフィールドデータのポインタ

参照元 ReadCoordinateData().

6.3.3.62 `template<class T , class TimeT , class TimeAvrT > CDM::E_CDM_ERRORCODE cdm_DFI::ReadCoordinateData ( T * val, const unsigned step, const int gc, const int Gvoxel[3], const int Gdivision[3], const int head[3], const int tail[3], TimeT & time, const bool mode, unsigned & step_avr, TimeAvrT & time_avr )`

read xyz(fub) data record (template function)

引数で渡された配列ポインタにデータを読み込む

引数

out	val	読み込んだデータポインタ
in	step	入力ステップ番号
in	gc	仮想セル数
in	Gvoxel	グローバルボクセルサイズ
in	Gdivision	領域分割数
in	head	計算領域の開始位置
in	tail	計算領域の終了位置
out	time	読み込んだ時間
in	mode	平均ステップ & 時間読み込みフラグ false : 読み込み true : 読み込まない
out	step_avr	平均ステップ
out	time_avr	平均時間

戻り値

終了コード 1:正常 1 以外:エラー

6.3.3.63 `virtual CDM::E_CDM_ERRORCODE cdm_DFI::ReadCoordinateData ( cdm_Array * val, const unsigned step, const int gc, const int Gvoxel[3], const int Gdivision[3], const int head[3], const int tail[3], double & time, const bool mode, unsigned & step_avr, double & time_avr ) [inline],[virtual]`

read xyz(fub) data record

template ReadData 関数で型に応じた配列を確保した後、呼び出される



引数

out	<i>val</i>	読み込み先の配列をポインタで渡す
in	<i>step</i>	読み込むステップ番号
in	<i>gc</i>	仮想セル数
in	<i>Gvoxel</i>	グローバルボクセルサイズ
in	<i>Gdivision</i>	領域分割数
in	<i>head</i>	計算領域の開始位置
in	<i>tail</i>	計算領域の終了位置
out	<i>time</i>	読み込んだ時間
in	<i>mode</i>	平均ステップ & 時間読み込みフラグ    false : 読み込み true : 読み込まない
out	<i>step_avr</i>	平均ステップ
out	<i>time_avr</i>	平均時間

戻り値

終了コード 1:正常 1 以外:エラー

cdm\_DFI\_FUBで再定義されています。

cdm\_DFI.h の 630 行で定義されています。

参照先 CDM::E\_CDM\_ERROR.

```
6.3.3.64  template<class TimeT , class TimeAvrT > CDM_INLINE void* cdm_DFI::ReadData (
           CDM::E_CDM_ERRORCODE & ret, const unsigned step, const int gc, const int Gvoxel[3], const int Gdivision[3],
           const int head[3], const int tail[3], TimeT & time, const bool mode, unsigned & step_avr, TimeAvrT & time_avr )
```

cdm\_DFI\_inline.h の 50 行で定義されています。

参照先 cdm\_FileInfo::ArrayShape, cdm\_FileInfo::DataType, DFI\_Finfo, CDM::E\_CDM\_SUCCESS, cdm\_Array::getData(), cdm\_Array::instanceArray(), cdm\_FileInfo::NumVariables, と ReadData().

```
6.3.3.65  template<class T , class TimeT , class TimeAvrT > CDM_INLINE CDM::E_CDM_ERRORCODE
           cdm_DFI::ReadData ( T * val, const unsigned step, const int gc, const int Gvoxel[3], const int Gdivision[3], const int
           head[3], const int tail[3], TimeT & time, const bool mode, unsigned & step_avr, TimeAvrT & time_avr )
```

cdm\_DFI\_inline.h の 97 行で定義されています。

参照先 cdm\_FileInfo::ArrayShape, DFI\_Finfo, CDM::E\_CDM\_SUCCESS, cdm\_Array::instanceArray(), cdm\_FileInfo::NumVariables, と ReadData().

```
6.3.3.66  template<class TimeT , class TimeAvrT > void* cdm_DFI::ReadData ( CDM::E_CDM_ERRORCODE & ret, const
           unsigned step, const int gc, const int Gvoxel[3], const int Gdivision[3], const int head[3], const int tail[3], TimeT &
           time, const bool mode, unsigned & step_avr, TimeAvrT & time_avr )
```

read field data record (template function)

読み込んだデータのポインタを戻り値として返す

引数

out	<i>ret</i>	終了コード 1:正常、1 以外 : エラー
in	<i>step</i>	入力ステップ番号
in	<i>gc</i>	仮想セル数

in	<i>Gvoxel</i>	グローバルボクセルサイズ
in	<i>Gdivision</i>	領域分割数
in	<i>head</i>	計算領域の開始位置
in	<i>tail</i>	計算領域の終了位置
out	<i>time</i>	読み込んだ時間
in	<i>mode</i>	平均ステップ & 時間読み込みフラグ    false : 読み込み true : 読み込まない
out	<i>step_avr</i>	平均ステップ
out	<i>time_avr</i>	平均時間

戻り値

読み込んだフィールドデータのポインタ

参照元 ReadData().

```
6.3.3.67  template<class T, class TimeT, class TimeAvrT > CDM::E_CDM_ERRORCODE cdm_DFI::ReadData ( T * val,
const unsigned step, const int gc, const int Gvoxel[3], const int Gdivision[3], const int head[3], const int tail[3],
TimeT & time, const bool mode, unsigned & step_avr, TimeAvrT & time_avr )
```

read field data record (template function)

引数で渡された配列ポインタにデータを読み込む

引数

out	<i>val</i>	読み込んだデータポインタ
in	<i>step</i>	入力ステップ番号
in	<i>gc</i>	仮想セル数
in	<i>Gvoxel</i>	グローバルボクセルサイズ
in	<i>Gdivision</i>	領域分割数
in	<i>head</i>	計算領域の開始位置
in	<i>tail</i>	計算領域の終了位置
out	<i>time</i>	読み込んだ時間
in	<i>mode</i>	平均ステップ & 時間読み込みフラグ    false : 読み込み true : 読み込まない
out	<i>step_avr</i>	平均ステップ
out	<i>time_avr</i>	平均時間

戻り値

終了コード 1:正常 1 以外:エラー

```
6.3.3.68  CDM::E_CDM_ERRORCODE cdm_DFI::ReadData ( cdm_Array * val, const unsigned step, const int gc, const
int Gvoxel[3], const int Gdivision[3], const int head[3], const int tail[3], double & time, const bool mode, unsigned &
step_avr, double & time_avr )
```

read field data record

template ReadData 関数で型に応じた配列を確保した後、呼び出される

引数

out	<i>val</i>	読み込み先の配列をポインタで渡す
in	<i>step</i>	読み込むステップ番号

in	<i>gc</i>	仮想セル数
in	<i>Gvoxel</i>	グローバルボクセルサイズ
in	<i>Gdivision</i>	領域分割数
in	<i>head</i>	計算領域の開始位置
in	<i>tail</i>	計算領域の終了位置
out	<i>time</i>	読み込んだ時間
in	<i>mode</i>	平均ステップ&時間読み込みフラグ    false : 読み込み true : 読み込まない
out	<i>step_avr</i>	平均ステップ
out	<i>time_avr</i>	平均時間

戻り値

終了コード 1:正常 1 以外:エラー

```
6.3.3.69 virtual cdm_Array* cdm_DFI::ReadFieldData ( std::string fname, const unsigned step, double & time, const int
sta[3], const int end[3], const int DFI_head[3], const int DFI_tail[3], bool avr_mode, unsigned & avr_step, double &
avr_time, CDM::E_CDM_ERRORCODE & ret ) [virtual]
```

read field data record

引数

in	<i>fname</i>	FieldData ファイル名
in	<i>step</i>	読み込みステップ番号
out	<i>time</i>	読み込んだ時間
in	<i>sta</i>	読み込みスタート位置
in	<i>end</i>	読み込みエンド位置
in	<i>DFI_head</i>	dfi のHeadIndex
in	<i>DFI_tail</i>	dfi のTailIndex
in	<i>avr_mode</i>	平均ステップ&時間読み込みフラグ    false : 読み込み

true : 読み込まない

引数

out	<i>avr_step</i>	平均ステップ
out	<i>avr_time</i>	平均時間
out	<i>ret</i>	終了コード

戻り値

読み込んだ配列のポインタ

```
6.3.3.70 static cdm_DFI* cdm_DFI::ReadInit ( const MPI_Comm comm, const std::string dfifile, const int G_Voxel[3],
const int G_Div[3], CDM::E_CDM_ERRORCODE & ret ) [static]
```

read インスタンス (BOV もしくはPLOT3D の場合にインスタンス生成)

引数

in	<i>comm</i>	MPI コミュニケーター
in	<i>dfifile</i>	DFI ファイル名
in	<i>G_Voxel</i>	計算空間全体のボクセルサイズ

in	<i>G_Div</i>	計算空間の領域分割数
out	<i>ret</i>	終了コード

戻り値

インスタンスされたクラスのポインタ

**6.3.3.71** void cdm\_DFI::set\_input\_type ( CDM::E\_CDM\_FILE\_TYPE *input\_type* ) [inline]

入力形式 (ascii,binary,FortranBinary) をセット

引数

in	<i>input_type</i>	出力形式
----	-------------------	------

cdm\_DFI.h の 417 行で定義されています。

参照先 m\_input\_type.

**6.3.3.72** void cdm\_DFI::set\_interp\_flag ( bool *interp\_flag* ) [inline]

節点への補間フラグをセット

引数

in	<i>bgrid_interp_flag</i>	節点への補間フラグ
----	--------------------------	-----------

cdm\_DFI.h の 410 行で定義されています。

参照先 m\_bgrid\_interp\_flag.

**6.3.3.73** void cdm\_DFI::set\_output\_fname ( CDM::E\_CDM\_OUTPUT\_FNAME *output\_fname* ) [inline]

出力ファイル命名規約 (step\_rank,rank\_step) をセット

引数

in	<i>output_fname</i>	出力ファイル命名規約
----	---------------------	------------

cdm\_DFI.h の 439 行で定義されています。

参照先 DFI\_Finfo, CDM::E\_CDM\_FMT\_NETCDF4, CDM::E\_CDM\_FNAME\_RANK, cdm\_FileInfo::FileFormat, と m\_output\_fname.

**6.3.3.74** void cdm\_DFI::set\_output\_type ( CDM::E\_CDM\_FILE\_TYPE *output\_type* ) [inline]

出力形式 (ascii,binary,FortranBinary) をセット

引数

in	<i>output_type</i>	出力形式
----	--------------------	------

cdm\_DFI.h の 424 行で定義されています。

参照先 m\_output\_type.

**6.3.3.75** void cdm\_DFI::set\_output\_type\_coord ( CDM::E\_CDM\_FILE\_TYPE *output\_type\_coord* ) [inline]

座標データの出力形式 (ascii,binary) をセット

AVS およびVTK 形式で利用

引数

in	<i>output_type_coord</i>	座標データの出力形式
----	--------------------------	------------

cdm\_DFI.h の 432 行で定義されています。

参照先 m\_output\_type\_coord.

**6.3.3.76** void cdm\_DFI::set\_RankID ( const int *rankID* ) [inline]

RankID をセットする

引数

in	<i>rankID</i>	RankID
----	---------------	--------

cdm\_DFI.h の 403 行で定義されています。

参照先 m\_RankID.

**6.3.3.77** void cdm\_DFI::SetcdmDomain ( cdm\_Domain \* *domain* )

cdm\_Domain クラスのセット

**6.3.3.78** virtual void cdm\_DFI::SetcdmFieldFileNameFormat ( cdm\_FieldFileNameFormat *FieldFileNameFormat* )  
[inline],[virtual]

cdm\_FieldFileNameFormat クラスセット

[cdm\\_DFI\\_FUB](#)で再定義されています。

cdm\_DFI.h の 212 行で定義されています。

**6.3.3.79** void cdm\_DFI::SetcdmFileInfo ( cdm\_FileInfo *FInfo* )

cdm\_FileInfo クラスのセット

**6.3.3.80** void cdm\_DFI::SetcdmFilePath ( cdm\_FilePath *FPath* )

cdm\_FilePath クラスのセット

**6.3.3.81** void cdm\_DFI::SetcdmMPI ( cdm\_MPI *mpi* )

cdm\_MPI クラスセット

**6.3.3.82** void cdm\_DFI::SetcdmProcess ( cdm\_Process *Process* )

cdm\_Process クラスセット

**6.3.3.83** void cdm\_DFI::SetcdmRankNoPrefix ( std::string *prefix* )

cdm\_FileInfo のRankNoPrefix をセット フィールドファイル名のランク番号前文字列を変更する

引数

<i>in</i>	<i>prefix</i>	ランク番号前文字列
-----------	---------------	-----------

6.3.3.84 void cdm\_DFI::SetcdmTimeSlice ( cdm\_TimeSlice *TSlice* )

cdm\_TimeSlice クラスセット

6.3.3.85 void cdm\_DFI::SetcdmUnit ( cdm\_Unit *unit* )

cdm\_Unit クラスのセット

6.3.3.86 void cdm\_DFI::SetcdmVislt ( cdm\_Vislt *Visit* )

cdm\_Vislt クラスのセット

6.3.3.87 template<class T1 , class T2 > CDM\_INLINE bool cdm\_DFI::setGridData ( cdm\_TypeArray< T1 > \* *P*,  
cdm\_TypeArray< T2 > \* *S* )

<0,0,0>

<1,0,0>

<1,0,1>

<0,0,1>

<0,1,0>

<1,1,0>

<1,1,1>

<0,1,1>

<0,0,0>

<1,0,0>

<1,0,1>

<0,0,1>

<0,1,0>

<1,1,0>

<1,1,1>

<0,1,1>

cdm\_DFI\_inline.h の 406 行で定義されています。

参照先 CDM::E\_CDM\_NIJK, cdm\_Array::getArrayShape(), cdm\_Array::getArraySizeInt(), cdm\_TypeArray< T >::getData(), cdm\_Array::getNvarInt(), cdm\_TypeArray< T >::val(), と VolumeDataDivide().

6.3.3.88 template<class T1 , class T2 > bool cdm\_DFI::setGridData ( cdm\_TypeArray< T1 > \* *P*, cdm\_TypeArray< T2 > \* *S* )

セル中心データを格子点に値をセット

引数

out	P	格子点データ
in	S	セル中心 data

6.3.3.89 void cdm\_DFI::setIntervalStep ( int *interval\_step*, int *base\_step* = 0, int *start\_step* = 0, int *last\_step* = -1 )

出力インターバルステップの登録

登録しない ( 本メソッドがコールされない ) 場合は CDM でのインターバル 制御は行わない

引数

in	<i>interval_step</i>	インターバルステップ
in	<i>base_step</i>	基準となるステップ ( デフォルト 0 ステップ )
in	<i>start_step</i>	セッション開始ステップ ( デフォルト 0 ステップ )
in	<i>last_step</i>	セッション最終ステップ ( デフォルト、-1 : 最終ステップで出力しない )

6.3.3.90 void cdm\_DFI::setIntervalTime ( double *interval\_time*, double *dt*, double *base\_time* = 0.0, double *start\_time* = 0.0, double *last\_time* = -1.0 )

インターバルタイムの登録

引数

in	<i>interval_time</i>	出力インターバルタイム
in	<i>dt</i>	計算の時間間隔
in	<i>base_time</i>	基準となるタイム ( デフォルト 0.0 タイム )
in	<i>start_time</i>	セッション開始タイム ( デフォルト 0.0 タイム )
in	<i>last_time</i>	セッション最終タイム ( デフォルト、-1.0 : 最終タイムで出力しない )

6.3.3.91 void cdm\_DFI::SetTimeSliceFlag ( const CDM::E\_CDM\_ONOFF *ONOFF* )

TimeSlice OnOff フラグをセットする

引数

in	<i>ONOFF</i>	
----	--------------	--

6.3.3.92 void cdm\_DFI::setVariableName ( int *pvari*, std::string *variName* )

FileInfo の変数名を登録する

引数

in	<i>pvari</i>	変数位置 0:u, 1:v, 2:w
in	<i>variName</i>	変数名 "u","v","w",,,

6.3.3.93 template<class T> CDM\_INLINE void cdm\_DFI::VolumeDataDivide ( cdm\_TypeArray< T > \* *P* )

cdm\_DFI\_inline.h の 479 行で定義されています。

参照先 CDM::E\_CDM\_NIJK, cdm\_Array::getArrayShape(), cdm\_Array::getArraySizeInt(), cdm\_Array::getNvarInt(), と cdm\_TypeArray< T >::val().

6.3.3.94 `template<class T > void cdm_DFI::VolumeDataDivide ( cdm_TypeArray< T > * P )`

内部の格子点のデータを重み付けで割る



引数

out	<i>P</i>	格子点 data
-----	----------	----------

参照元 setGridData().

6.3.3.95 `virtual bool cdm_DFI::write_ascii_header ( const unsigned step, const double time ) [inline],  
[protected],[virtual]`

ascii ヘッダーレコード出力 (bov,avs)

引数

in	<i>step</i>	step 番号
in	<i>time</i>	time

[cdm\\_DFI\\_AVS](#), と [cdm\\_DFI\\_BOV](#) で再定義されています。

cdm\_DFI.h の 1291 行で定義されています。

6.3.3.96 `virtual CDM::E_CDM_ERRORCODE cdm_DFI::write_averaged ( cdm_FILE * pFile, const unsigned step_avr,  
const double time_avr ) [protected],[pure virtual]`

Average レコードの出力 (純粋仮想関数)

引数

in	<i>pFile</i>	ファイルポインタ
in	<i>step_avr</i>	平均ステップ番号
in	<i>time_avr</i>	平均時刻

戻り値

error code

[cdm\\_DFI\\_NETCDF](#), [cdm\\_DFI\\_PLOT3D](#), [cdm\\_DFI\\_SPH](#), [cdm\\_DFI\\_FUB](#), [cdm\\_DFI\\_AVS](#), [cdm\\_DFI\\_VTK](#), と [cdm\\_DFI\\_BOV](#) で実装されています。

6.3.3.97 `virtual CDM::E_CDM_ERRORCODE cdm_DFI::write_DataRecord ( cdm_FILE * pFile, cdm_Array * val,  
const int gc, const int RankID ) [protected],[pure virtual]`

フィールドデータのデータレコードの出力 (純粋仮想関数)

引数

in	<i>pFile</i>	ファイルポインタ
in	<i>val</i>	データポインタ
in	<i>gc</i>	ガイドセル
in	<i>RankID</i>	ランク番号

戻り値

error code

[cdm\\_DFI\\_NETCDF](#), [cdm\\_DFI\\_PLOT3D](#), [cdm\\_DFI\\_FUB](#), [cdm\\_DFI\\_SPH](#), [cdm\\_DFI\\_AVS](#), [cdm\\_DFI\\_VTK](#), と [cdm\\_DFI\\_BOV](#) で実装されています。

6.3.3.98 `virtual bool cdm_DFI::write_GridData ( const int * iblack ) [inline],[protected],[virtual]`

Grid data file 出力 コントロール

引数

in	<i>iblack</i>	iblack データポインタ (PLOT3D の xyz ファイル用)
----	---------------	-------------------------------------

[cdm\\_DFI\\_PLOT3D](#)で再定義されています。

cdm\_DFI.h の 1281 行で定義されています。

6.3.3.99 `virtual CDM::E_CDM_ERRORCODE cdm_DFI::write_HeaderRecord ( cdm_FILE * pFile, const unsigned step, const double time, const int RankID )` [protected],[pure virtual]

フィールドデータのヘッダレコードの出力 (純粋仮想関数)

引数

in	<i>pFile</i>	ファイルポインタ
in	<i>step</i>	ステップ番号
in	<i>time</i>	時刻
in	<i>RankID</i>	ランク番号

戻り値

error code

[cdm\\_DFI\\_NETCDF](#), [cdm\\_DFI\\_PLOT3D](#), [cdm\\_DFI\\_FUB](#), [cdm\\_DFI\\_SPH](#), [cdm\\_DFI\\_AVS](#), [cdm\\_DFI\\_VTK](#), と [cdm\\_DFI\\_BOV](#)で実装されています。

6.3.3.100 `template<class T, class TimeT > CDM_INLINE CDM::E_CDM_ERRORCODE cdm_DFI::WriteCoordinateData ( const unsigned step, TimeT time, const int sz[3], const int nVari, const int gc, T * val )`

cdm\_DFI\_inline.h の 336 行で定義されています。

参照先 [cdm\\_FileInfo::ArrayShape](#), [DFI\\_Finfo](#), [DFI\\_Process](#), [cdm\\_Array::instanceArray\(\)](#), [m\\_RankID](#), [cdm\\_FileInfo::NumVariables](#), [cdm\\_Process::RankList](#), と [WriteCoordinateData\(\)](#).

6.3.3.101 `template<class T, class TimeT > CDM::E_CDM_ERRORCODE cdm_DFI::WriteCoordinateData ( const unsigned step, TimeT time, const int sz[3], const int nVari, const int gc, T * val )`

write coordinate data record (template function)

fub の座標値データファイル出力

引数

in	<i>step</i>	出力ステップ番号
in	<i>time</i>	出力時刻
in	<i>sz</i>	val の実ボクセルサイズ
in	<i>nVari</i>	val の変数の個数
in	<i>gc</i>	val の仮想セル数
in	<i>val</i>	出力データポインタ

参照元 [WriteCoordinateData\(\)](#).

6.3.3.102 `virtual CDM::E_CDM_ERRORCODE cdm_DFI::WriteCoordinateData ( const unsigned step, const int gc, double time, cdm_Array * val )` [inline],[virtual]

write coordinate data record

引数

in	<i>step</i>	出力ステップ番号
in	<i>gc</i>	仮想セル数
in	<i>time</i>	出力時刻
in	<i>val</i>	出力データポインタ

cdm\_DFI\_FUBで再定義されています。

cdm\_DFI.h の 773 行で定義されています。

参照先 CDM::E\_CDM\_ERROR.

**6.3.3.103** `template<class T, class TimeT, class TimeAvrT > CDM_INLINE CDM::E_CDM_ERRORCODE  
cdm_DFI::WriteData ( const unsigned step, TimeT time, const int sz[3], const int nVari, const int gc, T * val, T *  
minmax, const bool avr_mode, const unsigned step_avr, TimeAvrT time_avr )`

cdm\_DFI\_inline.h の 235 行で定義されています。

参照先 cdm\_FileInfo::ArrayShape, DFI\_Finfo, DFI\_Process, CDM::E\_CDM\_ERROR\_UNMATCH\_NUM\_OF\_VARIABLES, CDM::E\_CDM\_FMT\_SPH, cdm\_FileInfo::FileFormat, cdm\_Array::instanceArray(), m\_RankID, cdm\_FileInfo::NumVariables, cdm\_Process::RankList, cdm\_FileInfo::VariableName, と WriteData().

**6.3.3.104** `template<class T, class TimeT, class TimeAvrT > CDM::E_CDM_ERRORCODE cdm_DFI::WriteData ( const  
unsigned step, TimeT time, const int sz[3], const int nVari, const int gc, T * val, T * minmax = NULL, bool  
avr_mode = true, unsigned step_avr = 0, TimeAvrT time_avr = 0.0 )`

write field data record (template function)

minmax[0] =変数 1 の minX minmax[1] =変数 1 の maxX ... minmax[2n-2]=変数 n の minX minmax[2n-1]=変数 n の maxX SPH で nVari=3 のとき、 minmax[2n] =合成値の min minmax[2n+1]=合成値の max

引数

in	<i>step</i>	出力ステップ番号
in	<i>time</i>	出力時刻
in	<i>sz</i>	val の実ボクセルサイズ
in	<i>nVari</i>	val の変数の個数
in	<i>gc</i>	val の仮想セル数
in	<i>val</i>	出力データポインタ
in	<i>minmax</i>	フィールドデータのMinMax
in	<i>avr_mode</i>	平均ステップ&時間出力 false : 出力 true : 出力しない
in	<i>step_avr</i>	平均ステップ
in	<i>time_avr</i>	平均時間

参照元 WriteData().

**6.3.3.105** `CDM::E_CDM_ERRORCODE cdm_DFI::WriteData ( const unsigned step, const int gc, double time,  
cdm_Array * val, double * minmax, const bool avr_mode, const unsigned step_avr, double time_avr )`

write field data record

template WriteData 関数で方に応じた配列を確保した後、呼び出される

引数

in	<i>step</i>	出力ステップ番号
----	-------------	----------

in	<i>gc</i>	仮想セル数
in	<i>time</i>	出力時刻
in	<i>val</i>	出力データポインタ
in	<i>minmax</i>	フィールドデータのMinMax
in	<i>avr_mode</i>	平均ステップ&時間出力    false : 出力 true : 出力しない
in	<i>step_avr</i>	平均ステップ
in	<i>time_avr</i>	平均時間

6.3.3.106 `virtual CDM::E_CDM_ERRORCODE cdm_DFI::WriteFieldData ( std::string fname, const unsigned step, double time, cdm_Array * val, const bool mode, const unsigned step_avr, const double time_avr )` [protected], [virtual]

write field data record (double)

引数

in	<i>fname</i>	出力フィールドファイル名
in	<i>step</i>	出力ステップ番号
in	<i>time</i>	出力時刻
in	<i>val</i>	出力データポインタ
in	<i>mode</i>	平均ステップ&時間出力    false : 出力 true : 出力しない
in	<i>step_avr</i>	平均ステップ
in	<i>time_avr</i>	平均時間

戻り値

error code

6.3.3.107 `template<class T , class TimeT , class TimeAvrT > CDM_INLINE CDM::E_CDM_ERRORCODE cdm_DFI::WriteFieldDataFile ( const unsigned step, TimeT time, const int sz[3], const int nVari, const int gc, T * val, const bool avr_mode, const unsigned step_avr, TimeAvrT time_avr )`

cdm\_DFI\_inline.h の 295 行で定義されています。

参照先 cdm\_FileInfo::ArrayShape, DFI\_Finfo, DFI\_Process, CDM::E\_CDM\_ERROR\_UNMATCH\_NUM\_OF\_VARIABLES, cdm\_Array::instanceArray(), m\_RankID, cdm\_FileInfo::NumVariables, cdm\_Process::RankList, cdm\_FileInfo::VariableName, と WriteFieldDataFile().

6.3.3.108 `template<class T , class TimeT , class TimeAvrT > CDM::E_CDM_ERRORCODE cdm_DFI::WriteFieldDataFile ( const unsigned step, TimeT time, const int sz[3], const int nVari, const int gc, T * val, bool avr_mode = true, unsigned step_avr = 0, TimeAvrT time_avr = 0.0 )`

write field data record (template function)

フィールドデータのみ出力。dfi ファイルの出力はなし。

引数

in	<i>step</i>	出力ステップ番号
in	<i>time</i>	出力時刻
in	<i>sz</i>	val の実ボクセルサイズ
in	<i>nVari</i>	val の変数の個数

in	gc	val の仮想セル数
in	val	出力データポインタ
in	avr_mode	平均ステップ & 時間出力 false : 出力 true : 出力しない
in	step_avr	平均ステップ
in	time_avr	平均時間

参照元 WriteFieldDataFile().

**6.3.3.109 CDM::E\_CDM\_ERRORCODE cdm\_DFI::WriteFieldDataFile ( const unsigned step, const int gc, double time, cdm\_Array \* val, const bool avr\_mode, const unsigned step\_avr, double time\_avr )**

write field data record (not output dfi file)

template WriteFieldDataFile 関数で方に応じた配列を確保した後、呼び出される

引数

in	step	出力ステップ番号
in	gc	仮想セル数
in	time	出力時刻
in	val	出力データポインタ
in	avr_mode	平均ステップ & 時間出力 false : 出力 true : 出力しない
in	step_avr	平均ステップ
in	time_avr	平均時間

**6.3.3.110 CDM::E\_CDM\_ERRORCODE cdm\_DFI::WriteGridFile ( const int \* iblack = NULL )**

grid ファイル出力コントロール

引数

in	iblack	iblack データポインタ (PLOT3D の xyz ファイル用)
----	--------	-------------------------------------

**6.3.3.111 CDM::E\_CDM\_ERRORCODE cdm\_DFI::WriteIndexDfiFile ( )**

index DFI ファイル出力 (API 関数)

**6.3.3.112 CDM::E\_CDM\_ERRORCODE cdm\_DFI::WriteIndexDfiFile ( const std::string dfi\_name )** [protected]

index DFI ファイル出力

引数

in	dfi_name	DFI ファイル名
----	----------	-----------

戻り値

true:出力成功 false:出力失敗

**6.3.3.113** template<typename T> static cdm\_DFI\* cdm\_DFI::Writelnit ( const MPI\_Comm comm, const std::string DfiName, const std::string Path, const std::string prefix, const CDM::E\_CDM\_FORMAT format, const int GCell, const CDM::E\_CDM\_DTYPE DataType, const int nVari, const std::string proc\_fname, const int G\_size[3], const T pitch[3], const T L\_origin[3], const int division[3], const int head[3], const int tail[3], const std::string hostname, const CDM::E\_CDM\_ONOFF TSliceOnOff ) [static]

write インスタンス template function (等間隔格子用)

## 引数

in	<i>comm</i>	MPI コミュニケーター
in	<i>DfiName</i>	DFI ファイル名
in	<i>Path</i>	フィールドデータのディレクトリ
in	<i>prefix</i>	ベースファイル名
in	<i>format</i>	ファイルフォーマット
in	<i>GCell</i>	出力仮想セル数
in	<i>DataType</i>	データタイプ
in	<i>nVari</i>	変数の個数
in	<i>proc_fname</i>	proc.dfi ファイル名
in	<i>G_size</i>	グローバルボクセルサイズ
in	<i>pitch</i>	ピッチ
in	<i>L_origin</i>	各ランクの局所領域における原点座標値
in	<i>division</i>	領域分割数
in	<i>head</i>	計算領域の開始位置
in	<i>tail</i>	計算領域の終了位置
in	<i>hostname</i>	ホスト名
in	<i>TSliceOnOff</i>	TimeSlice フラグ

## 戻り値

インスタンスされたクラスのポインタ

```
6.3.3.114 template<typename T> static cdm_DFI* cdm_DFI::Writelnit ( const MPI_Comm comm, const std::string
DfiName, const std::string Path, const std::string prefix, const CDM::E_CDM_FORMAT format, const
int GCell, const CDM::E_CDM_DTYPE DataType, const int nVari, const std::string proc_fname, const
int G_size[3], const T * coord_X, const T * coord_Y, const T * coord_Z, const std::string coord_file, const
CDM::E_CDM_FILE_TYPE coord_filetype, const CDM::E_CDM_ENDIAN_TYPE coord_fileEndian, const
int division[3], const int head[3], const int tail[3], const std::string hostname, const CDM::E_CDM_ONOFF
TSliceOnOff ) [static]
```

write インスタンス template function (不等間隔格子用)

template の型より、座標ファイルのデータ精度を指定

## 引数

in	<i>comm</i>	MPI コミュニケーター
in	<i>DfiName</i>	DFI ファイル名
in	<i>Path</i>	フィールドデータのディレクトリ
in	<i>prefix</i>	ベースファイル名
in	<i>format</i>	ファイルフォーマット
in	<i>GCell</i>	出力仮想セル数
in	<i>DataType</i>	データタイプ
in	<i>nVari</i>	変数の個数
in	<i>proc_fname</i>	proc.dfi ファイル名
in	<i>G_size</i>	グローバルボクセルサイズ
in	<i>coord_X</i>	X 座標データポインタ
in	<i>coord_Y</i>	Y 座標データポインタ
in	<i>coord_Z</i>	Z 座標データポインタ

in	<i>coord_file</i>	座標ファイル名
in	<i>coord_filetype</i>	座標ファイルのファイルタイプ
in	<i>coord_fileEndian</i>	座標ファイルのエンディアンタイプ
in	<i>division</i>	領域分割数
in	<i>head</i>	計算領域の開始位置
in	<i>tail</i>	計算領域の終了位置
in	<i>hostname</i>	ホスト名
in	<i>TSliceOnOff</i>	TimeSlice フラグ

戻り値

インスタンスされたクラスのポインタ

```
6.3.3.115 template<typename T> static cdm_DFI* cdm_DFI::Writelnit ( const MPI_Comm comm, const std::string
DfiName, const std::string Path, const std::string prefix, const CDM::E_CDM_FORMAT format, const int GCell,
const CDM::E_CDM_DTYPE DataType, const int nVari, const std::string proc_fname, const cdm_Domain *
out_domain, const int head[3], const int tail[3], const std::string hostname, const CDM::E_CDM_ONOFF
TSliceOnOff ) [static]
```

write インスタンス template function (等間隔格子・不等間隔格子の共通処理部分)

引数

in	<i>comm</i>	MPI コミュニケータ
in	<i>DfiName</i>	DFI ファイル名
in	<i>Path</i>	フィールドデータのディレクトリ
in	<i>prefix</i>	ベースファイル名
in	<i>format</i>	ファイルフォーマット
in	<i>GCell</i>	出力仮想セル数
in	<i>DataType</i>	データタイプ
in	<i>nVari</i>	変数の個数
in	<i>proc_fname</i>	proc.dfi ファイル名
in	<i>out_domain</i>	domain インスタンス
in	<i>head</i>	計算領域の開始位置
in	<i>tail</i>	計算領域の終了位置
in	<i>hostname</i>	ホスト名
in	<i>TSliceOnOff</i>	TimeSlice フラグ

戻り値

インスタンスされたクラスのポインタ

```
6.3.3.116 template<typename T> CDM_INLINE cdm_DFI* cdm_DFI::Writelnit ( const MPI_Comm comm, const
std::string DfiName, const std::string Path, const std::string prefix, const CDM::E_CDM_FORMAT format,
const int GCell, const CDM::E_CDM_DTYPE DataType, const int nVari, const std::string proc_fname, const int
G_size[3], const T pitch[3], const T L_origin[3], const int division[3], const int head[3], const int tail[3], const
std::string hostname, const CDM::E_CDM_ONOFF TSliceOnOff )
```

cdm\_DFI\_inline.h の 550 行で定義されています。

参照先 CDM::E\_CDM\_FMT\_AVIS, CDM::E\_CDM\_FMT\_PLOT3D, と CDM::E\_CDM\_FMT\_VTK.

6.3.3.117 `template<typename T> CDM_INLINE cdm_DFI* cdm_DFI::Writelnit ( const MPI_Comm comm, const std::string DfiName, const std::string Path, const std::string prefix, const CDM::E_CDM_FORMAT format, const int GCell, const CDM::E_CDM_DTYPE DataType, const int nVari, const std::string proc_fname, const int G_size[3], const T * coord_X, const T * coord_Y, const T * coord_Z, const std::string coord_file, const CDM::E_CDM_FILE_TYPE coord_filetype, const CDM::E_CDM_ENDIAN_TYPE coord_fileEndian, const int division[3], const int head[3], const int tail[3], const std::string hostname, const CDM::E_CDM_ONOFF TSliceOnOff )`

cdm\_DFI\_inline.h の 607 行で定義されています。

参照先 CDM::E\_CDM\_FMT\_BOV, と CDM::E\_CDM\_FMT\_SPH.

6.3.3.118 `template<typename T> CDM_INLINE cdm_DFI* cdm_DFI::Writelnit ( const MPI_Comm comm, const std::string DfiName, const std::string Path, const std::string prefix, const CDM::E_CDM_FORMAT format, const int GCell, const CDM::E_CDM_DTYPE DataType, const int nVari, const std::string proc_fname, const cdm_Domain * out_domain, const int head[3], const int tail[3], const std::string hostname, const CDM::E_CDM_ONOFF TSliceOnOff )`

cdm\_DFI\_inline.h の 693 行で定義されています。

参照先 cdm\_FileInfo::ArrayShape, CDM::cdmPath\_DirName(), cdm\_FileInfo::DataType, cdm\_FileInfo::DFIType, cdm\_FileInfo::DirectoryPath, CDM::E\_CDM\_BIG, CDM::E\_CDM\_DFITYPE\_CARTESIAN, CDM::E\_CDM\_DFITYPE\_NON\_UNIFORM\_CARTESIAN, CDM::E\_CDM\_FMT\_AVS, CDM::E\_CDM\_FMT\_BOV, CDM::E\_CDM\_FMT\_FUB, CDM::E\_CDM\_FMT\_FUB\_COD, CDM::E\_CDM\_FMT\_NETCDF4, CDM::E\_CDM\_FMT\_PLOT3D, CDM::E\_CDM\_FMT\_SPH, CDM::E\_CDM\_FMT\_VTK, CDM::E\_CDM\_IJKN, CDM::E\_CDM\_LITTLE, CDM::E\_CDM\_NIJK, cdm\_FileInfo::Endian, cdm\_FileInfo::FileFormat, cdm\_FileInfo::GuideCell, m\_comm, m\_directoryPath, m\_indexDfiName, m\_RankID, MPI\_Comm\_rank(), MPI\_Comm\_size(), cdm\_MPI::NumberOfGroup, cdm\_MPI::NumberOfRank, cdm\_FileInfo::NumVariables, cdm\_VisIt::PlotGC, cdm\_FileInfo::Prefix, cdm\_FilePath::ProcDFIFile, cdm\_Process::RankList, と cdm\_FileInfo::TimeSliceDirFlag.

6.3.3.119 `CDM::E_CDM_ERRORCODE cdm_DFI::WriteProcDfiFile ( const MPI_Comm comm, const bool out_host, const int cell_id, const int bcf_id )`

proc DFI ファイル出力コントロール (float)

引数

in	comm	MPI コミュニケータ
in	out_host	ホスト名出力フラグ

戻り値

true:出力成功 false:出力失敗 proc DFI ファイル出力コントロール

引数

in	comm	MPI コミュニケータ
in	out_host	ホスト名出力フラグ
in	cell_id	cell id
in	bcf_id	境界ID

戻り値

終了コード 1:正常 1 以外:エラー

## 6.3.4 変数



**6.3.4.1 const cdm\_Domain\* cdm\_DFI::DFI\_Domain [protected]**

Domain class.

cdm\_DFI.h の 67 行で定義されています。

参照元 cdm\_DFI\_AVIS::cdm\_DFI\_AVIS(), cdm\_DFI\_BOV::cdm\_DFI\_BOV(), cdm\_DFI\_FUB::cdm\_DFI\_FUB(), cdm\_DFI\_NETCDF::cdm\_DFI\_NETCDF(), cdm\_DFI\_PLOT3D::cdm\_DFI\_PLOT3D(), cdm\_DFI\_SPH::cdm\_DFI\_SPH(), cdm\_DFI\_VTK::cdm\_DFI\_VTK(), と cdm\_DFI\_PLOT3D::write\_XYZ().

**6.3.4.2 cdm\_FileInfo cdm\_DFI::DFI\_Finfo [protected]**

FileInfo class.

cdm\_DFI.h の 63 行で定義されています。

参照元 AddTimeSlice(), cdm\_DFI\_AVIS::cdm\_DFI\_AVIS(), cdm\_DFI\_BOV::cdm\_DFI\_BOV(), cdm\_DFI\_FUB::cdm\_DFI\_FUB(), cdm\_DFI\_NETCDF::cdm\_DFI\_NETCDF(), cdm\_DFI\_PLOT3D::cdm\_DFI\_PLOT3D(), cdm\_DFI\_SPH::cdm\_DFI\_SPH(), cdm\_DFI\_VTK::cdm\_DFI\_VTK(), ReadCoordinateData(), ReadData(), set\_output\_fname(), cdm\_DFI\_PLOT3D::write\_XYZ(), WriteCoordinateData(), WriteData(), と WriteFieldDataFile().

**6.3.4.3 cdm\_FilePath cdm\_DFI::DFI\_Fpath [protected]**

FilePath class.

cdm\_DFI.h の 64 行で定義されています。

参照元 cdm\_DFI\_AVIS::cdm\_DFI\_AVIS(), cdm\_DFI\_BOV::cdm\_DFI\_BOV(), cdm\_DFI\_FUB::cdm\_DFI\_FUB(), cdm\_DFI\_NETCDF::cdm\_DFI\_NETCDF(), cdm\_DFI\_PLOT3D::cdm\_DFI\_PLOT3D(), cdm\_DFI\_SPH::cdm\_DFI\_SPH(), と cdm\_DFI\_VTK::cdm\_DFI\_VTK().

**6.3.4.4 cdm\_MPI cdm\_DFI::DFI\_MPI [protected]**

MPI class.

cdm\_DFI.h の 68 行で定義されています。

参照元 cdm\_DFI\_AVIS::cdm\_DFI\_AVIS(), cdm\_DFI\_BOV::cdm\_DFI\_BOV(), cdm\_DFI\_FUB::cdm\_DFI\_FUB(), cdm\_DFI\_NETCDF::cdm\_DFI\_NETCDF(), cdm\_DFI\_PLOT3D::cdm\_DFI\_PLOT3D(), cdm\_DFI\_SPH::cdm\_DFI\_SPH(), と cdm\_DFI\_VTK::cdm\_DFI\_VTK().

**6.3.4.5 cdm\_Process cdm\_DFI::DFI\_Process [protected]**

Process class.

cdm\_DFI.h の 70 行で定義されています。

参照元 cdm\_DFI\_AVIS::cdm\_DFI\_AVIS(), cdm\_DFI\_BOV::cdm\_DFI\_BOV(), cdm\_DFI\_FUB::cdm\_DFI\_FUB(), cdm\_DFI\_NETCDF::cdm\_DFI\_NETCDF(), cdm\_DFI\_PLOT3D::cdm\_DFI\_PLOT3D(), cdm\_DFI\_SPH::cdm\_DFI\_SPH(), cdm\_DFI\_VTK::cdm\_DFI\_VTK(), WriteCoordinateData(), WriteData(), と WriteFieldDataFile().

**6.3.4.6 cdm\_TimeSlice cdm\_DFI::DFI\_TimeSlice [protected]**

TimeSlice class.

cdm\_DFI.h の 69 行で定義されています。

参照元 AddTimeSlice(), cdm\_DFI\_AVIS::cdm\_DFI\_AVIS(), cdm\_DFI\_BOV::cdm\_DFI\_BOV(), cdm\_DFI\_FUB::cdm\_DFI\_FUB(), cdm\_DFI\_NETCDF::cdm\_DFI\_NETCDF(), cdm\_DFI\_PLOT3D::cdm\_DFI\_PLOT3D(), cdm\_DFI\_SPH::cdm\_DFI\_SPH(), と cdm\_DFI\_VTK::cdm\_DFI\_VTK().

#### 6.3.4.7 `cdm_Unit cdm_DFI::DFI_Unit` [protected]

Unit class.

`cdm_DFI.h` の 66 行で定義されています。

参照元 `cdm_DFI_AVIS::cdm_DFI_AVIS()`, `cdm_DFI_BOV::cdm_DFI_BOV()`, `cdm_DFI_FUB::cdm_DFI_FUB()`, `cdm_DFI_NETCDF::cdm_DFI_NETCDF()`, `cdm_DFI_PLOT3D::cdm_DFI_PLOT3D()`, `cdm_DFI_SPH::cdm_DFI_SPH()`, と `cdm_DFI_VTK::cdm_DFI_VTK()`.

#### 6.3.4.8 `cdm_VisIt cdm_DFI::DFI_VisIt` [protected]

VisIt class.

`cdm_DFI.h` の 65 行で定義されています。

参照元 `cdm_DFI_AVIS::cdm_DFI_AVIS()`, `cdm_DFI_BOV::cdm_DFI_BOV()`, `cdm_DFI_FUB::cdm_DFI_FUB()`, `cdm_DFI_NETCDF::cdm_DFI_NETCDF()`, `cdm_DFI_PLOT3D::cdm_DFI_PLOT3D()`, `cdm_DFI_SPH::cdm_DFI_SPH()`, と `cdm_DFI_VTK::cdm_DFI_VTK()`.

#### 6.3.4.9 `bool cdm_DFI::m_bgrid_interp_flag` [protected]

節点への補間フラグ

`cdm_DFI.h` の 78 行で定義されています。

参照元 `cdm_DFI_AVIS::cdm_DFI_AVIS()`, `cdm_DFI_BOV::cdm_DFI_BOV()`, `cdm_DFI_FUB::cdm_DFI_FUB()`, `cdm_DFI_NETCDF::cdm_DFI_NETCDF()`, `cdm_DFI_PLOT3D::cdm_DFI_PLOT3D()`, `cdm_DFI_SPH::cdm_DFI_SPH()`, `cdm_DFI_VTK::cdm_DFI_VTK()`, と `set_interp_flag()`.

#### 6.3.4.10 `MPI_Comm cdm_DFI::m_comm` [protected]

MPI コミュニケーター

`cdm_DFI.h` の 56 行で定義されています。

参照元 `Writelnit()`.

#### 6.3.4.11 `std::string cdm_DFI::m_directoryPath` [protected]

index dfi ファイルのディレクトリパス

`cdm_DFI.h` の 57 行で定義されています。

参照元 `Writelnit()`.

#### 6.3.4.12 `std::string cdm_DFI::m_indexDfiName` [protected]

index dfi ファイル名

`cdm_DFI.h` の 58 行で定義されています。

参照元 `get_dfi_fname()`, と `Writelnit()`.

#### 6.3.4.13 `CDM::E_CDM_FILE_TYPE cdm_DFI::m_input_type` [protected]

入力形式 (ascii,binary,FortranBinary)

`cdm_DFI.h` の 79 行で定義されています。

参照元 cdm\_DFI\_FUB::cdm\_DFI\_FUB(), cdm\_DFI\_PLOT3D::cdm\_DFI\_PLOT3D(), cdm\_DFI\_PLOT3D::read\_Func(), と set\_input\_type().

#### 6.3.4.14 CDM::E\_CDM\_OUTPUT\_FNAME cdm\_DFI::m\_output\_fname [protected]

出力ファイル命名規約 (step\_rank,rank\_step)

cdm\_DFI.h の 82 行で定義されています。

参照元 set\_output\_fname().

#### 6.3.4.15 CDM::E\_CDM\_FILE\_TYPE cdm\_DFI::m\_output\_type [protected]

出力形式 (ascii,binary,FortranBinary)

cdm\_DFI.h の 80 行で定義されています。

参照元 cdm\_DFI\_FUB::cdm\_DFI\_FUB(), cdm\_DFI\_PLOT3D::cdm\_DFI\_PLOT3D(), set\_output\_type(), cdm\_DFI\_PLOT3D::write\_Func(), と cdm\_DFI\_PLOT3D::write\_XYZ().

#### 6.3.4.16 CDM::E\_CDM\_FILE\_TYPE cdm\_DFI::m\_output\_type\_coord [protected]

座標データの出力形式 (ascii,binary)

cdm\_DFI.h の 81 行で定義されています。

参照元 set\_output\_type\_coord().

#### 6.3.4.17 int cdm\_DFI::m\_RankID [protected]

ランク番号

cdm\_DFI.h の 61 行で定義されています。

参照元 set\_RankID(), WriteCoordinateData(), WriteData(), WriteFieldDataFile(), と WriteInit().

#### 6.3.4.18 CDM::E\_CDM\_READTYPE cdm\_DFI::m\_read\_type [protected]

読み込みタイプ

cdm\_DFI.h の 59 行で定義されています。

#### 6.3.4.19 vector<int> cdm\_DFI::m\_readRankList [protected]

読み込みランクリスト

cdm\_DFI.h の 76 行で定義されています。

このクラスの説明は次のファイルから生成されました:

- [cdm\\_DFI.h](#)
- [cdm\\_DFI\\_inline.h](#)

## 6.4 クラス cdm\_DFI\_AVS

```
#include <cdm_DFI_AVS.h>
```

cdm\_DFI\_AVS に対する継承グラフ

cdm\_DFI\_AVS のコラボレーション図

## Public メソッド

- `cdm_DFI_AVS ()`
- `cdm_DFI_AVS (const cdm_FileInfo F_Info, const cdm_FilePath F_Path, const cdm_VisIt visit, const cdm_Unit unit, const cdm_Domain *domain, const cdm_MPI mpi, const cdm_TimeSlice TSlice, const cdm_Process process)`  
コンストラクタ
- `~cdm_DFI_AVS ()`

## Protected メソッド

- `CDM::E_CDM_ERRORCODE read_HeaderRecord (cdm_FILE *pFile, bool matchEndian, unsigned step, const int head[3], const int tail[3], int gc, int voxsize[3], double &time)`  
*avs ファイルのヘッダーレコード読み込み*
- `CDM::E_CDM_ERRORCODE read_Datarecord (cdm_FILE *pFile, bool matchEndian, unsigned step, cdm_Array *buf, int head[3], int nz, cdm_Array *&src)`  
*フィールドデータファイルのデータレコード読み込み*
- `CDM::E_CDM_ERRORCODE read_averaged (cdm_FILE *pFile, bool matchEndian, unsigned step, unsigned &avr_step, double &avr_time)`  
*avs ファイルのAverage データレコードの読み込み*
- `CDM::E_CDM_ERRORCODE write_HeaderRecord (cdm_FILE *pFile, const unsigned step, const double time, const int RankID)`  
*avs ヘッダファイルの出力*
- `CDM::E_CDM_ERRORCODE write_DataRecord (cdm_FILE *pFile, cdm_Array *val, const int gc, const int RankID)`  
*avs データレコードの出力*
- `CDM::E_CDM_ERRORCODE write_averaged (cdm_FILE *pFile, const unsigned step_avr, const double time_avr)`  
*Average レコードの出力*
- `bool write_ascii_header (const unsigned step, const double time)`  
*avs の座標値データ、ヘッダーの出力コントロール*
- `bool write_avs_cord (int dims[3], int gc)`  
*座標値データファイル出力*
- `bool write_avs_header (int dims[3])`  
*ヘッダーデータファイルの出力*

## Additional Inherited Members

### 6.4.1 説明

cdm\_DFI\_AVS.h の 20 行で定義されています。

### 6.4.2 コンストラクタとデストラクタ

#### 6.4.2.1 cdm\_DFI\_AVS::cdm\_DFI\_AVS ( )

#### コンストラクタ

```
6.4.2.2 cdm_DFI_AVS::cdm_DFI_AVS ( const cdm_FileInfo F_Info, const cdm_FilePath F_Path, const cdm_VisIt visit,  
    const cdm_Unit unit, const cdm_Domain * domain, const cdm_MPI mpi, const cdm_TimeSlice TSlice, const  
    cdm_Process process ) [inline]
```

コンストラクタ

## 引数

in	<i>F_Info</i>	FileInfo
in	<i>F_Path</i>	FilePath
in	<i>visit</i>	VisIt option
in	<i>unit</i>	Unit
in	<i>domain</i>	Domain
in	<i>mpi</i>	MPI
in	<i>TSlice</i>	TimeSlice
in	<i>process</i>	Process

cdm\_DFI\_AVS.h の 40 行で定義されています。

参照先 cdm\_DFI::DFI\_Domain, cdm\_DFI::DFI\_Finfo, cdm\_DFI::DFI\_Fpath, cdm\_DFI::DFI\_MPI, cdm\_DFI::DFI\_Process, cdm\_DFI::DFI\_TimeSlice, cdm\_DFI::DFI\_Unit, cdm\_DFI::DFI\_VisIt, と cdm\_DFI::m\_bgrid\_interp\_flag.

## 6.4.2.3 cdm\_DFI\_AVS::~cdm\_DFI\_AVS ( )

## デストラクタ

## 6.4.3 関数

6.4.3.1 **CDM::E\_CDM\_ERRORCODE** cdm\_DFI\_AVS::read\_averaged ( cdm\_FILE \* *pFile*, bool *matchEndian*, unsigned *step*, unsigned & *avr\_step*, double & *avr\_time* ) [inline], [protected], [virtual]

avs ファイルのAverage データレコードの読み込み

## 引数

in	<i>pFile</i>	ファイルポインタ
in	<i>matchEndian</i>	true:Endian 一致
in	<i>step</i>	読み込み step 番号
out	<i>avr_step</i>	平均ステップ
out	<i>avr_time</i>	平均タイム

## 戻り値

error code

cdm\_DFIを実装しています。

cdm\_DFI\_AVS.h の 124 行で定義されています。

参照先 CDM::E\_CDM\_SUCCESS.

6.4.3.2 **CDM::E\_CDM\_ERRORCODE** cdm\_DFI\_AVS::read\_Datarecord ( cdm\_FILE \* *pFile*, bool *matchEndian*, unsigned *step*, cdm\_Array \* *buf*, int *head*[3], int *nz*, cdm\_Array \* & *src* ) [inline], [protected], [virtual]

フィールドデータファイルのデータレコード読み込み

## 引数

in	<i>pFile</i>	ファイルポインタ
in	<i>matchEndian</i>	true:Endian 一致

in	<i>step</i>	ステップ番号
in	<i>buf</i>	読み込み用バッファ
in	<i>head</i>	読み込みバッファHeadIndex
in	<i>nz</i>	z 方向のボクセルサイズ ( 実セル + ガイドセル * 2 )
out	<i>src</i>	読み込んだデータを格納した配列のポインタ

戻り値

error code

cdm\_DFIを実装しています。

cdm\_DFI\_AVS.h の 104 行で定義されています。

参照先 CDM::E\_CDM\_SUCCESS.

```
6.4.3.3 CDM::E_CDM_ERRORCODE cdm_DFI_AVS::read_HeaderRecord ( cdm_FILE * pFile, bool matchEndian,
    unsigned step, const int head[3], const int tail[3], int gc, int voysize[3], double & time ) [inline],
    [protected], [virtual]
```

avs ファイルのヘッダーレコード読み

引数

in	<i>pFile</i>	ファイルポインタ
in	<i>matchEndian</i>	エンディアンチェックフラグ true:合致
in	<i>step</i>	ステップ番号
in	<i>head</i>	dfi のHeadIndex
in	<i>tail</i>	dfi のTailIndex
in	<i>gc</i>	dfi のガイドセル数
out	<i>voysize</i>	voysize
out	<i>time</i>	時刻

戻り値

error code

cdm\_DFIを実装しています。

cdm\_DFI\_AVS.h の 81 行で定義されています。

参照先 CDM::E\_CDM\_SUCCESS.

```
6.4.3.4 bool cdm_DFI_AVS::write_ascii_header ( const unsigned step, const double time ) [protected],
    [virtual]
```

avs の座標値データ、ヘッダーの出力コントロール

引数

in	<i>step</i>	step 番号
in	<i>time</i>	time

cdm\_DFIを再定義しています。

```
6.4.3.5 CDM::E_CDM_ERRORCODE cdm_DFI_AVS::write_averaged ( cdm_FILE * pFile, const unsigned step_avr,
    const double time_avr ) [inline], [protected], [virtual]
```

Average レコードの出力

引数

in	<i>pFile</i>	ファイルポインタ
in	<i>step_avr</i>	平均ステップ番号
in	<i>time_avr</i>	平均時刻

戻り値

error code

[cdm\\_DFI](#)を実装しています。

cdm\_DFI\_AVS.h の 170 行で定義されています。

参照先 CDM::E\_CDM\_SUCCESS.

6.4.3.6 `bool cdm_DFI_AVS::write_avs_cord ( int dims[3], int gc )` [protected]

座標値データファイル出力

引数

in	<i>dims</i>	計算領域のボクセル数
in	<i>gc</i>	ガイドセル数

6.4.3.7 `bool cdm_DFI_AVS::write_avs_header ( int dims[3] )` [protected]

ヘッダーデータファイルの出力

引数

in	<i>dims</i>	計算領域のボクセル数
----	-------------	------------

6.4.3.8 `CDM::E_CDM_ERRORCODE cdm_DFI_AVS::write_DataRecord ( cdm_FILE * pFile, cdm_Array * val, const int gc, const int RankID )` [protected], [virtual]

avs データレコードの出力

引数

in	<i>pFile</i>	ファイルポインタ
in	<i>val</i>	データポインタ
in	<i>gc</i>	ガイドセル
in	<i>RankID</i>	ランク番号

戻り値

error code

[cdm\\_DFI](#)を実装しています。

6.4.3.9 `CDM::E_CDM_ERRORCODE cdm_DFI_AVS::write_HeaderRecord ( cdm_FILE * pFile, const unsigned step, const double time, const int RankID )` [protected], [virtual]

avs ヘッダファイルの出力



引数

in	<i>pFile</i>	ファイルポインタ
in	<i>step</i>	ステップ番号
in	<i>time</i>	時刻
in	<i>RankID</i>	ランク番号

戻り値

error code

cdm\_DFIを実装しています。

このクラスの説明は次のファイルから生成されました:

- [cdm\\_DFI\\_AVS.h](#)

## 6.5 クラス cdm\_DFI\_BOV

```
#include <cdm_DFI_BOV.h>
```

cdm\_DFI\_BOV に対する継承グラフ

cdm\_DFI\_BOV のコラボレーション図

### Public メソッド

- [cdm\\_DFI\\_BOV \(\)](#)
- [cdm\\_DFI\\_BOV \(const \[cdm\\\_FileInfo\]\(#\) F\\_Info, const \[cdm\\\_FilePath\]\(#\) F\\_Path, const \[cdm\\\_Visit\]\(#\) visit, const \[cdm\\\_Unit\]\(#\) unit, const \[cdm\\\_Domain\]\(#\) \\*domain, const \[cdm\\\_MPI\]\(#\) mpi, const \[cdm\\\_TimeSlice\]\(#\) TSlice, const \[cdm\\\_Process\]\(#\) process\)](#)  
コンストラクタ
- [~cdm\\_DFI\\_BOV \(\)](#)

### Protected メソッド

- [CDM::E\\_CDM\\_ERRORCODE read\\_HeaderRecord \(cdm\\_FILE \\*pFile, bool matchEndian, unsigned step, const int head\[3\], const int tail\[3\], int gc, int voxsize\[3\], double &time\)](#)  
*bov* ファイルのヘッダーレコード読み込み
- [CDM::E\\_CDM\\_ERRORCODE read\\_Datarecord \(cdm\\_FILE \\*pFile, bool matchEndian, unsigned step, \[cdm\\\_Array\]\(#\) \\*buf, int head\[3\], int nz, \[cdm\\\_Array\]\(#\) \\*&src\)](#)  
フィールドデータファイルのデータレコード読み込み
- [CDM::E\\_CDM\\_ERRORCODE read\\_averaged \(cdm\\_FILE \\*pFile, bool matchEndian, unsigned step, unsigned &avr\\_step, double &avr\\_time\)](#)  
*bov* ファイルの *Average* データレコードの読み込み
- [CDM::E\\_CDM\\_ERRORCODE write\\_HeaderRecord \(cdm\\_FILE \\*pFile, const unsigned step, const double time, const int RankID\)](#)  
*avs* ヘッダファイルの出力
- [CDM::E\\_CDM\\_ERRORCODE write\\_DataRecord \(cdm\\_FILE \\*pFile, \[cdm\\\_Array\]\(#\) \\*val, const int gc, const int RankID\)](#)  
*avs* データレコードの出力
- [CDM::E\\_CDM\\_ERRORCODE write\\_averaged \(cdm\\_FILE \\*pFile, const unsigned step\\_avr, const double time\\_avr\)](#)  
*Average* レコードの出力
- [bool write\\_ascii\\_header \(const unsigned step, const double time\)](#)  
ヘッダーデータファイルの出力

## Additional Inherited Members

### 6.5.1 説明

cdm\_DFI\_BOV.h の 20 行で定義されています。

### 6.5.2 コンストラクタとデストラクタ

#### 6.5.2.1 cdm\_DFI\_BOV::cdm\_DFI\_BOV ( )

コンストラクタ

6.5.2.2 cdm\_DFI\_BOV::cdm\_DFI\_BOV ( const cdm\_FileInfo *F\_Info*, const cdm\_FilePath *F\_Path*, const cdm\_VisIt *visit*, const cdm\_Unit *unit*, const cdm\_Domain \* *domain*, const cdm\_MPI *mpi*, const cdm\_TimeSlice *TSlice*, const cdm\_Process *process* ) [inline]

コンストラクタ

引数

in	<i>F_Info</i>	FileInfo
in	<i>F_Path</i>	FilePath
in	<i>visit</i>	VisIt option
in	<i>unit</i>	Unit
in	<i>domain</i>	Domain
in	<i>mpi</i>	MPI
in	<i>TSlice</i>	TimeSlice
in	<i>process</i>	Process

cdm\_DFI\_BOV.h の 37 行で定義されています。

参照先 cdm\_DFI::DFI\_Domain, cdm\_DFI::DFI\_Finfo, cdm\_DFI::DFI\_Fpath, cdm\_DFI::DFI\_MPI, cdm\_DFI::DFI\_Process, cdm\_DFI::DFI\_TimeSlice, cdm\_DFI::DFI\_Unit, cdm\_DFI::DFI\_VisIt, と cdm\_DFI::m\_bgrid\_interp\_flag.

#### 6.5.2.3 cdm\_DFI\_BOV::~~cdm\_DFI\_BOV ( )

デストラクタ

### 6.5.3 関数

6.5.3.1 CDM::E\_CDM\_ERRORCODE cdm\_DFI\_BOV::read\_averaged ( cdm\_FILE \* *pFile*, bool *matchEndian*, unsigned *step*, unsigned & *avr\_step*, double & *avr\_time* ) [protected], [virtual]

bov ファイルのAverage データレコードの読み込み

引数

in	<i>pFile</i>	ファイルポインタ
in	<i>matchEndian</i>	true:Endian 一致
in	<i>step</i>	読み込み step 番号
out	<i>avr_step</i>	平均ステップ
out	<i>avr_time</i>	平均タイム

戻り値

errorcode

cdm\_DFIを実装しています。

**6.5.3.2 CDM::E\_CDM\_ERRORCODE** cdm\_DFI\_BOV::read\_Datarecord ( cdm\_FILE \* *pFile*, bool *matchEndian*, unsigned *step*, cdm\_Array \* *buf*, int *head*[3], int *nz*, cdm\_Array \*&*src* ) [protected],[virtual]

フィールドデータファイルのデータレコード読み

引数

in	<i>pFile</i>	ファイルポインタ
in	<i>matchEndian</i>	true:Endian 一致
in	<i>step</i>	ステップ番号
in	<i>buf</i>	読み込み用バッファ
in	<i>head</i>	読み込みバッファHeadIndex
in	<i>nz</i>	z 方向のボクセルサイズ ( 実セル + ガイドセル * 2 )
out	<i>src</i>	読み込んだデータを格納した配列のポインタ

戻り値

error code

cdm\_DFIを実装しています。

**6.5.3.3 CDM::E\_CDM\_ERRORCODE** cdm\_DFI\_BOV::read\_HeaderRecord ( cdm\_FILE \* *pFile*, bool *matchEndian*, unsigned *step*, const int *head*[3], const int *tail*[3], int *gc*, int *voysize*[3], double & *time* ) [protected],[virtual]

bov ファイルのヘッダーレコード読み

引数

in	<i>pFile</i>	ファイルポインタ
in	<i>matchEndian</i>	エンディアンチェックフラグ true:合致
in	<i>step</i>	ステップ番号
in	<i>head</i>	dfi のHeadIndex
in	<i>tail</i>	dfi のTailIndex
in	<i>gc</i>	dfi のガイドセル数
out	<i>voysize</i>	voysize
out	<i>time</i>	時刻

戻り値

error code

cdm\_DFIを実装しています。

**6.5.3.4 bool** cdm\_DFI\_BOV::write\_ascii\_header ( const unsigned *step*, const double *time* ) [protected],[virtual]

ヘッダーデータファイルの出力

引数

in	<i>step</i>	step 番号
in	<i>time</i>	time

[cdm\\_DFI](#)を再定義しています。

**6.5.3.5 CDM::E\_CDM\_ERRORCODE cdm\_DFI\_BOV::write\_averaged ( cdm\_FILE \* *pFile*, const unsigned *step\_avr*, const double *time\_avr* )** [protected],[virtual]

Average レコードの出力

引数

in	<i>pFile</i>	ファイルポインタ
in	<i>step_avr</i>	平均ステップ番号
in	<i>time_avr</i>	平均時刻

戻り値

error code

[cdm\\_DFI](#)を実装しています。

**6.5.3.6 CDM::E\_CDM\_ERRORCODE cdm\_DFI\_BOV::write\_DataRecord ( cdm\_FILE \* *pFile*, cdm\_Array \* *val*, const int *gc*, const int *RankID* )** [protected],[virtual]

avs データレコードの出力

引数

in	<i>pFile</i>	ファイルポインタ
in	<i>val</i>	データポインタ
in	<i>gc</i>	ガイドセル
in	<i>RankID</i>	ランク番号

戻り値

error code

[cdm\\_DFI](#)を実装しています。

**6.5.3.7 CDM::E\_CDM\_ERRORCODE cdm\_DFI\_BOV::write\_HeaderRecord ( cdm\_FILE \* *pFile*, const unsigned *step*, const double *time*, const int *RankID* )** [protected],[virtual]

avs ヘッダファイルの出力

引数

in	<i>pFile</i>	ファイルポインタ
in	<i>step</i>	ステップ番号
in	<i>time</i>	時刻
in	<i>RankID</i>	ランク番号

戻り値

error code

cdm\_DFIを実装しています。

このクラスの説明は次のファイルから生成されました:

- [cdm\\_DFI\\_BOV.h](#)

## 6.6 クラス cdm\_DFI\_FUB

```
#include <cdm_DFI_FUB.h>
```

cdm\_DFI\_FUB に対する継承グラフ

cdm\_DFI\_FUB のコラボレーション図

### Public メソッド

- [cdm\\_DFI\\_FUB \(\)](#)
- [cdm\\_DFI\\_FUB \(const \[cdm\\\_FileInfo\]\(#\) F\\_Info, const \[cdm\\\_FilePath\]\(#\) F\\_Path, const \[cdm\\\_VisIt\]\(#\) visit, const \[cdm\\\_Unit\]\(#\) unit, const \[cdm\\\_Domain\]\(#\) \\*domain, const \[cdm\\\_MPI\]\(#\) mpi, const \[cdm\\\_TimeSlice\]\(#\) TSlice, const \[cdm\\\_Process\]\(#\) process\)](#)  
 コンストラクタ
- [~cdm\\_DFI\\_FUB \(\)](#)
- [std::string getCoordinateFileName \(std::string FieldFileName\)](#)  
*fub 特有のFileList 要素を読み込む*
- [CDM::E\\_CDM\\_ERRORCODE ReadCoordinateData \(\[cdm\\\_Array\]\(#\) \\*val, const unsigned step, const int gc, const int Gvoxel\[3\], const int Gdivision\[3\], const int head\[3\], const int tail\[3\], double &time, const bool mode, unsigned &step\\_avr, double &time\\_avr\)](#)  
*read xyz(fub) data record*
- [CDM::E\\_CDM\\_ERRORCODE WriteCoordinateData \(const unsigned step, const int gc, double time, \[cdm\\\_Array\]\(#\) \\*val\)](#)  
*write coordinate data record*
- [const \[cdm\\\_FieldFileNameFormat\]\(#\) \\* GetcdmFieldFileNameFormat \(\)](#)  
*index.dfi にFileList があるとき、FileList を出力*
- [void SetcdmFieldFileNameFormat \(\[cdm\\\_FieldFileNameFormat\]\(#\) \\_FieldFileNameFormat\)](#)  
*cdm\_FieldFileNameFormat クラスセット*

### Public 変数

- [cdm\\_FieldFileNameFormat DFI\\_FieldFileNameFormat](#)  
*FieldFileNameFormat class.*

### Protected メソッド

- [CDM::E\\_CDM\\_ERRORCODE read\\_HeaderRecord \(\[cdm\\\_FILE\]\(#\) \\*pFile, bool matchEndian, unsigned step, const int head\[3\], const int tail\[3\], int gc, int voxsize\[3\], double &time\)](#)  
*sph ファイルのヘッダーレコード読み込み*
- [CDM::E\\_CDM\\_ERRORCODE read\\_Datarecord \(\[cdm\\\_FILE\]\(#\) \\*pFile, bool matchEndian, unsigned step, \[cdm\\\_Array\]\(#\) \\*buf, int head\[3\], int nz, \[cdm\\\_Array\]\(#\) \\*&src\)](#)  
*フィールドデータファイルのデータレコード読み込み*

- `CDM::E_CDM_ERRORCODE read_averaged` (`cdm_FILE *pFile`, `bool matchEndian`, `unsigned step`, `unsigned &avr_step`, `double &avr_time`)  
Average データレコードの読み込み
- `CDM::E_CDM_ERRORCODE write_HeaderRecord` (`cdm_FILE *pFile`, `const unsigned step`, `const double time`, `const int RankID`)  
ヘッダファイルの出力
- `CDM::E_CDM_ERRORCODE write_DataRecord` (`cdm_FILE *pFile`, `cdm_Array *val`, `const int gc`, `const int RankID`)  
データレコードの出力
- `CDM::E_CDM_ERRORCODE write_averaged` (`cdm_FILE *pFile`, `const unsigned step_avr`, `const double time_avr`)  
Average レコードの出力

## Additional Inherited Members

### 6.6.1 説明

`cdm_DFI_FUB.h` の 20 行で定義されています。

### 6.6.2 コンストラクタとデストラクタ

#### 6.6.2.1 `cdm_DFI_FUB::cdm_DFI_FUB ( )`

コンストラクタ

6.6.2.2 `cdm_DFI_FUB::cdm_DFI_FUB ( const cdm_FileInfo F_Info, const cdm_FilePath F_Path, const cdm_VisIt visit, const cdm_Unit unit, const cdm_Domain * domain, const cdm_MPI mpi, const cdm_TimeSlice TSlice, const cdm_Process process ) [inline]`

コンストラクタ

コンストラクタ

引数

<code>in</code>	<code><i>F_Info</i></code>	FileInfo
<code>in</code>	<code><i>F_Path</i></code>	FilePath
<code>in</code>	<code><i>visit</i></code>	VisIt option
<code>in</code>	<code><i>unit</i></code>	Unit
<code>in</code>	<code><i>domain</i></code>	Domain
<code>in</code>	<code><i>mpi</i></code>	MPI
<code>in</code>	<code><i>process</i></code>	Process

`cdm_DFI_FUB.h` の 54 行で定義されています。

参照先 `cdm_DFI::DFI_Domain`, `cdm_DFI::DFI_Finfo`, `cdm_DFI::DFI_Fpath`, `cdm_DFI::DFI_MPI`, `cdm_DFI::DFI_Process`, `cdm_DFI::DFI_TimeSlice`, `cdm_DFI::DFI_Unit`, `cdm_DFI::DFI_VisIt`, `CDM::E_CDM_FILE_TYPE_FBINARY`, `cdm_DFI::m_bgrid_interp_flag`, `cdm_DFI::m_input_type`, と `cdm_DFI::m_output_type`.

#### 6.6.2.3 `cdm_DFI_FUB::~~cdm_DFI_FUB ( )`

デストラクタ

### 6.6.3 関数

6.6.3.1 `const cdm_FieldFileNameFormat* cdm_DFI_FUB::GetcdmFieldFileNameFormat ( )` [virtual]

index.dfi にFileList があるとき、FileList を出力

引数

in	<i>fp</i>	ファイルポインタ
in	<i>tab</i>	インデント

戻り値

error code cdm\_FieldFileNameFormat クラスのポインタ取得  
cdm\_FieldFileNameFormat クラスのポインタ

cdm\_DFIを再定義しています。

#### 6.6.3.2 std::string cdm\_DFI\_FUB::getCoordinateFileName ( std::string *FieldFileName* )

fub 特有のFileList 要素を読み込む

引数

in	<i>tpCntl</i>	cdm_TextParser クラス
----	---------------	--------------------

戻り値

error code field data file name の取得

index dfi ファイルの/FileList/Rank のID と読み込み指示されたランクのID と一致したID のRank で定義された FieldDataFileName で指示されたファイル名もしくは CoordinateFileName で指示されたファイル名を取得 ( E\_CDM\_FMT\_FUB のときは FieldDataFileName、 E\_CDM\_FMT\_FUB\_COD のときはCoordinateFileName) index dfi ファイルにFileList がないファイルは空白が戻る.

引数

in	<i>ID</i>	RankID
----	-----------	--------

戻り値

field data file name coordinate file name の取得

index dfi のFileList に定義してある、フィールドファイルとペアになっている座標値ファイル名をフィールドファイル名で検索して取得、index dfi にFileList がないときはフィールドファイル名の拡張子を xyz にしたファイル名が戻る .

引数

in	<i>FieldFileName</i>	field data file name
----	----------------------	----------------------

戻り値

coordinate data file name

#### 6.6.3.3 CDM::E\_CDM\_ERRORCODE cdm\_DFI\_FUB::read\_averaged ( cdm\_FILE \* *pFile*, bool *matchEndian*, unsigned *step*, unsigned & *avr\_step*, double & *avr\_time* ) [protected],[virtual]

Average データレコードの読み込み



引数

in	<i>pFile</i>	ファイルポインタ
in	<i>matchEndian</i>	true:Endian 一致
in	<i>step</i>	読み込み step 番号
out	<i>avr_step</i>	平均ステップ
out	<i>avr_time</i>	平均タイム

戻り値

error code

cdm\_DFIを実装しています。

**6.6.3.4 CDM::E\_CDM\_ERRORCODE cdm\_DFI\_FUB::read\_Datarecord ( cdm\_FILE \* *pFile*, bool *matchEndian*, unsigned *step*, cdm\_Array \* *buf*, int *head*[3], int *nz*, cdm\_Array \* &*src* )** [protected],[virtual]

フィールドデータファイルのデータレコード読み

引数

in	<i>pFile</i>	ファイルポインタ
in	<i>matchEndian</i>	true:Endian 一致
in	<i>step</i>	読み込み step 番号
in	<i>buf</i>	読み込み用バッファ
in	<i>head</i>	読み込みバッファHeadIndex
in	<i>nz</i>	z 方向のボクセルサイズ ( 実セル + ガイドセル * 2 )
out	<i>src</i>	読み込んだデータを格納した配列のポインタ

戻り値

error code

cdm\_DFIを実装しています。

**6.6.3.5 CDM::E\_CDM\_ERRORCODE cdm\_DFI\_FUB::read\_HeaderRecord ( cdm\_FILE \* *pFile*, bool *matchEndian*, unsigned *step*, const int *head*[3], const int *tail*[3], int *gc*, int *voysize*[3], double & *time* )** [protected],[virtual]

sph ファイルのヘッダーレコード読み

引数

in	<i>pFile</i>	ファイルポインタ
in	<i>matchEndian</i>	エンディアンチェックフラグ true:合致
in	<i>step</i>	ステップ番号
in	<i>head</i>	dfi のHeadIndex
in	<i>tail</i>	dfi のTailIndex
in	<i>gc</i>	dfi のガイドセル数
out	<i>voysize</i>	voysize
out	<i>time</i>	時刻

戻り値

error code

cdm\_DFIを実装しています。

**6.6.3.6 CDM::E\_CDM\_ERRORCODE** `cdm_DFI_FUB::ReadCoordinateData ( cdm_Array * val, const unsigned step, const int gc, const int Gvoxel[3], const int Gdivision[3], const int head[3], const int tail[3], double & time, const bool mode, unsigned & step_avr, double & time_avr )` [virtual]

read xyz(fub) data record

template ReadData 関数で型に応じた配列を確保した後、呼び出される

引数

out	val	読み込み先の配列をポインタで渡す
in	step	読み込むステップ番号
in	gc	仮想セル数
in	Gvoxel	グローバルボクセルサイズ
in	Gdivision	領域分割数
in	head	計算領域の開始位置
in	tail	計算領域の終了位置
out	time	読み込んだ時間
in	mode	平均ステップ&時間読み込みフラグ false : 読み込み true : 読み込まない
out	step_avr	平均ステップ
out	time_avr	平均時間

戻り値

終了コード 1:正常 1 以外:エラー

[cdm\\_DFI](#)を再定義しています。

**6.6.3.7 void** `cdm_DFI_FUB::SetcdmFieldFileNameFormat ( cdm_FieldFileNameFormat FieldFileNameFormat )` [inline],[virtual]

cdm\_FieldFileNameFormat クラスセット

[cdm\\_DFI](#)を再定義しています。

cdm\_DFI\_FUB.h の 277 行で定義されています。

参照先 DFI\_FieldFileNameFormat.

**6.6.3.8 CDM::E\_CDM\_ERRORCODE** `cdm_DFI_FUB::write_averaged ( cdm_FILE * pFile, const unsigned step_avr, const double time_avr )` [protected],[virtual]

Average レコードの出力

[cdm\\_DFI](#)を実装しています。

**6.6.3.9 CDM::E\_CDM\_ERRORCODE** `cdm_DFI_FUB::write_DataRecord ( cdm_FILE * pFile, cdm_Array * val, const int gc, const int RankID )` [protected],[virtual]

データレコードの出力

引数

in	pFile	ファイルポインタ
in	val	データポインタ

in	gc	ガイドセル
in	RankID	ランク番号

戻り値

error code

[cdm\\_DFI](#)を実装しています。

**6.6.3.10 CDM::E\_CDM\_ERRORCODE cdm\_DFI\_FUB::write\_HeaderRecord ( cdm\_FILE \* pFile, const unsigned step, const double time, const int RankID )** [protected],[virtual]

ヘッダファイルの出力

引数

in	pFile	ファイルポインタ
in	step	ステップ番号
in	time	時刻
in	RankID	ランク番号

戻り値

error code

[cdm\\_DFI](#)を実装しています。

**6.6.3.11 CDM::E\_CDM\_ERRORCODE cdm\_DFI\_FUB::WriteCoordinateData ( const unsigned step, const int gc, double time, cdm\_Array \* val )** [virtual]

write coordinate data record

引数

in	step	出力ステップ番号
in	gc	仮想セル数
in	time	出力時刻
in	val	出力データポインタ

[cdm\\_DFI](#)を再定義しています。

## 6.6.4 変数

**6.6.4.1 cdm\_FieldFileNameFormat cdm\_DFI\_FUB::DFI\_FieldFileNameFormat**

FieldFileNameFormat class.

cdm\_DFI\_FUB.h の 24 行で定義されています。

参照元 SetcdmFieldFileNameFormat().

このクラスの説明は次のファイルから生成されました:

- [cdm\\_DFI\\_FUB.h](#)

## 6.7 クラス cdm\_DFI\_NETCDF

```
#include <cdm_DFI_NETCDF.h>
```

cdm\_DFI\_NETCDF に対する継承グラフ

cdm\_DFI\_NETCDF のコラボレーション図

## 構成

- struct [stDimInfo](#)
- struct [stVarInfo](#)

## Public メソッド

- [cdm\\_DFI\\_NETCDF](#) ()
- [cdm\\_DFI\\_NETCDF](#) (const [cdm\\_FileInfo](#) F\_Info, const [cdm\\_FilePath](#) F\_Path, const [cdm\\_VisIt](#) visit, const [cdm\\_Unit](#) unit, const [cdm\\_Domain](#) \*domain, const [cdm\\_MPI](#) mpi, const [cdm\\_TimeSlice](#) TSlice, const [cdm\\_Process](#) process)  
コンストラクタ
- [~cdm\\_DFI\\_NETCDF](#) ()
- void [SetDimName](#) (string nameX, string nameY, string nameZ, string nameT)  
*NetCDF の次元変数名のセット*
- void [GetDimName](#) (string &nameX, string &nameY, string &nameZ, string &nameT)  
*NetCDF の次元変数名の取得*
- [CDM::E\\_CDM\\_ERRORCODE](#) [ReadAdditionalTP](#) ([cdm\\_TextParser](#) tpCntl)  
*NetCDF 特有の dfi パラメータ読み込み*
- [CDM::E\\_CDM\\_ERRORCODE](#) [WriteAdditionalTP](#) (FILE \*fp, int tab)  
*NetCDF 特有の dfi パラメータ出力*
- void [AddNcUnit](#) (const std::string Name, const std::string Unit, const double reference, const double difference=0.0, const bool BsetDiff=false)  
*Unit をセットする*
- const [cdm\\_Unit](#) \* [GetNcUnit](#) ()  
*cdm\_Unit クラスのポインタを取得*
- virtual bool [CheckAddWriteMode](#) ()  
*出力処理を追記モードにするかどうかをチェック (NetCDF 用)*
- void [SetWriteFlag](#) (bool writeFlag)  
*書き込み済みフラグのセット (FCONV 用)*

## Static Public メソッド

- static [CDM::E\\_CDM\\_ERRORCODE](#) [read\\_HeaderRecord](#) (int ncid, std::string nameX, std::string nameY, std::string nameZ, std::string nameT, [stVarInfo](#) &varInfoX, [stVarInfo](#) &varInfoY, [stVarInfo](#) &varInfoZ, [stVarInfo](#) &varInfoT)  
*ファイルのヘッダーレコード読み込み (static 関数)*
- static [cdm\\_Array](#) \* [read\\_Datarecord](#) (int ncid, unsigned step\_index, std::vector< std::string > &VariableName, [CDM::E\\_CDM\\_DTYPE](#) DataType, [stVarInfo](#) &varInfoX, [stVarInfo](#) &varInfoY, [stVarInfo](#) &varInfoZ, [stVarInfo](#) &varInfoT, std::vector< [stVarInfo](#) > &vecVarInfo, [CDM::E\\_CDM\\_ERRORCODE](#) &ret)  
*nc データファイルのデータレコード読み込み (static 関数)*
- static [CDM::E\\_CDM\\_ERRORCODE](#) [write\\_HeaderRecord](#) ([cdm\\_FILE](#) \*pFile, const unsigned step, const double time, int VoxelSize[3], int GuideCell, [CDM::E\\_CDM\\_DTYPE](#) DataType, vector< string > &vecVariable, [cdm\\_Unit](#) &Unit, [stVarInfo](#) &varInfoX, [stVarInfo](#) &varInfoY, [stVarInfo](#) &varInfoZ, [stVarInfo](#) &varInfoT, vector< [stVarInfo](#) > &vecVarInfo)  
*nc ファイルのヘッダレコードの出力 (static 関数)*
- static [CDM::E\\_CDM\\_ERRORCODE](#) [write\\_DataRecord](#) ([cdm\\_FILE](#) \*pFile, [cdm\\_Array](#) \*val, const int gc, int VoxelSize[3], [stVarInfo](#) &varInfoT, vector< [stVarInfo](#) > &vecVarInfo)

- `nc` ファイルのデータレコードの出力 (*static* 関数)
- static `nc_type` `GetNcType` (`CDM::E_CDM_DTYPE` type)  
*CDM* データ型に対応する *nc* データ型を取得

## Protected メソッド

- `CDM::E_CDM_ERRORCODE` `CheckArrayVarInfo` (`stVarInfo` &varInfo)  
*nc* 変数配列の *dimension* をチェック
- `CDM::E_CDM_ERRORCODE` `read_HeaderRecord` (`cdm_FILE` \*pFile, bool matchEndian, unsigned step, const int head[3], const int tail[3], int gc, int voxsize[3], double &time)  
*nc* ファイルのヘッダーレコード読み込み
- `CDM::E_CDM_ERRORCODE` `read_Datarecord` (`cdm_FILE` \*pFile, bool matchEndian, unsigned step, `cdm_Array` \*buf, int head[3], int nz, `cdm_Array` \*&src)  
*nc* データファイルのデータレコード読み込み
- `CDM::E_CDM_ERRORCODE` `read_averaged` (`cdm_FILE` \*pFile, bool matchEndian, unsigned step, unsigned &avr\_step, double &avr\_time)  
*nc* ファイルの *Average* データレコードの読み込み
- `CDM::E_CDM_ERRORCODE` `write_HeaderRecord` (`cdm_FILE` \*pFile, const unsigned step, const double time, const int RankID)  
*nc* ファイルのヘッダーレコードの出力
- `CDM::E_CDM_ERRORCODE` `write_DataRecord` (`cdm_FILE` \*pFile, `cdm_Array` \*val, const int gc, const int RankID)  
*nc* ファイルのデータレコードの出力
- `CDM::E_CDM_ERRORCODE` `write_averaged` (`cdm_FILE` \*pFile, const unsigned step\_avr, const double time\_avr)  
*Average* レコードの出力

## Static Protected メソッド

- static `CDM::E_CDM_ERRORCODE` `GetVarInfo` (int ncid, string var\_name, `stVarInfo` &varInfo, int nDim-Check=0)  
*nc* 変数の情報を取得する
- static `CDM::E_CDM_DTYPE` `GetCdmType` (`nc_type` type)  
*nc* データ型に対応する *CDM* データ型を取得
- static void `write_AttUnits` (int ncid, int varid, string var\_name, `cdm_Unit` &unit)  
*UnitList* に変数名の *unit* が存在する場合、*var* 属性として出力する

## Protected 変数

- string `m_nameX`  
*nc* ファイルの *X* 座標値配列の配列名
- string `m_nameY`  
*nc* ファイルの *Y* 座標値配列の配列名
- string `m_nameZ`  
*nc* ファイルの *Z* 座標値配列の配列名
- string `m_nameT`  
*nc* ファイルの時刻値配列の配列名
- `stVarInfo` `m_varInfoX`  
*x* の変数情報
- `stVarInfo` `m_varInfoY`  
*y* の変数情報

- [stVarInfo m\\_varInfoZ](#)  
z の変数情報
- [stVarInfo m\\_varInfoT](#)  
t の変数情報
- [cdm\\_Unit m\\_NC\\_Unit](#)  
nc ファイル内の unit 情報
- [vector< stVarInfo > m\\_vecVarInfo](#)
- [bool m\\_writeFlag](#)  
書き込み済みフラグ

### 6.7.1 説明

cdm\_DFI\_NETCDF.h の 22 行で定義されています。

### 6.7.2 コンストラクタとデストラクタ

#### 6.7.2.1 cdm\_DFI\_NETCDF::cdm\_DFI\_NETCDF ( )

コンストラクタ

6.7.2.2 `cdm_DFI_NETCDF::cdm_DFI_NETCDF ( const cdm_FileInfo F_Info, const cdm_FilePath F_Path, const cdm_VisIt visit, const cdm_Unit unit, const cdm_Domain * domain, const cdm_MPI mpi, const cdm_TimeSlice TSlice, const cdm_Process process ) [inline]`

コンストラクタ

引数

in	<i>F_Info</i>	FileInfo
in	<i>F_Path</i>	FilePath
in	<i>visit</i>	VisIt option
in	<i>unit</i>	Unit
in	<i>domain</i>	Domain
in	<i>mpi</i>	MPI
in	<i>TSlice</i>	TimeSlice
in	<i>process</i>	Process

cdm\_DFI\_NETCDF.h の 104 行で定義されています。

参照先 `cdm_DFI::DFI_Domain`, `cdm_DFI::DFI_Finfo`, `cdm_DFI::DFI_Fpath`, `cdm_DFI::DFI_MPI`, `cdm_DFI::DFI_Process`, `cdm_DFI::DFI_TimeSlice`, `cdm_DFI::DFI_Unit`, `cdm_DFI::DFI_VisIt`, `cdm_DFI::m_bgrid_interp_flag`, `m_nameT`, `m_nameX`, `m_nameY`, `m_nameZ`, と `m_writeFlag`.

#### 6.7.2.3 cdm\_DFI\_NETCDF::~~cdm\_DFI\_NETCDF ( )

デストラクタ

### 6.7.3 関数

6.7.3.1 `void cdm_DFI_NETCDF::AddNcUnit ( const std::string Name, const std::string Unit, const double reference, const double difference = 0.0, const bool BsetDiff = false )`

Unit をセットする

引数

in	<i>Name</i>	追加する単位系 ("Length","Velocity",...)
in	<i>Unit</i>	単位ラベル ("M","CM","MM","M/S",...)
in	<i>reference</i>	規格化したスケール値
in	<i>difference</i>	差の値
in	<i>BsetDiff</i>	difference の有無

6.7.3.2 virtual bool cdm\_DFI\_NETCDF::CheckAddWriteMode ( ) [virtual]

出力処理を追記モードにするかどうかをチェック (NetCDF 用)

戻り値

モード (true:追記モード、false:新規作成モード)

cdm\_DFIを再定義しています。

6.7.3.3 CDM::E\_CDM\_ERRORCODE cdm\_DFI\_NETCDF::CheckArrayVarInfo ( stVarInfo & varInfo ) [protected]

nc 変数配列の dimension をチェック

- x,y,z,time と同じ dimension かどうか
- (z,y,x) もしくは (time,z,y,x) の形状かどうか

引数

in	<i>verInfo</i>	
----	----------------	--

戻り値

error code

6.7.3.4 static CDM::E\_CDM\_DTYPE cdm\_DFI\_NETCDF::GetCdmType ( nc\_type type ) [static],[protected]

nc データ型に対応するCDM データ型を取得

引数

in	<i>type</i>	nc データ型
----	-------------	---------

戻り値

CDM データ型

6.7.3.5 void cdm\_DFI\_NETCDF::GetDimName ( string & nameX, string & nameY, string & nameZ, string & nameT )

NetCDF の次元変数名の取得

引数

out	<i>nameX</i>	座標値X の次元名
out	<i>nameY</i>	座標値Y の次元名
out	<i>nameZ</i>	座標値Z の次元名
out	<i>nameT</i>	時刻の次元名

**6.7.3.6** static nc\_type cdm\_DFI\_NETCDF::GetNcType ( CDM::E\_CDM\_DTYPE *type* ) [static]

CDM データ型に対応する nc データ型を取得

引数

in	<i>type</i>	nc データ型
----	-------------	---------

戻り値

CDM データ型

**6.7.3.7** const cdm\_Unit\* cdm\_DFI\_NETCDF::GetNcUnit ( )

cdm\_Unit クラスのポインタを取得

戻り値

cdm\_Unit クラスポインタ

**6.7.3.8** static CDM::E\_CDM\_ERRORCODE cdm\_DFI\_NETCDF::GetVarInfo ( int *ncid*, string *var\_name*, stVarInfo & *varInfo*, int *nDimCheck* = 0 ) [static],[protected]

nc 変数の情報を取得する

引数

in	<i>ncid</i>	ファイルID(netcdf library)
in	<i>var_name</i>	情報取得対象の変数名
out	<i>varInfo</i>	変数の情報
in	<i>nDimCheck</i>	変数の配列次元数のチェック (0 以下:しない、1 以上:その数で無いときエラー)

戻り値

error code

**6.7.3.9** CDM::E\_CDM\_ERRORCODE cdm\_DFI\_NETCDF::read\_averaged ( cdm\_FILE \* *pFile*, bool *matchEndian*, unsigned *step*, unsigned & *avr\_step*, double & *avr\_time* ) [protected],[virtual]

nc ファイルのAverage データレコードの読み込み

引数

in	<i>pFile</i>	ファイルポインタ
----	--------------	----------



in	<i>matchEndian</i>	true:Endian 一致
in	<i>step</i>	読み込み step 番号
out	<i>avr_step</i>	平均ステップ
out	<i>avr_time</i>	平均タイム

戻り値

error code

cdm\_DFIを実装しています。

6.7.3.10 static cdm\_Array\* cdm\_DFI\_NETCDF::read\_Datarecord ( int *ncid*, unsigned *step\_index*, std::vector< std::string > & *VariableName*, CDM::E\_CDM\_DTYPE *DataType*, stVarInfo & *varInfoX*, stVarInfo & *varInfoY*, stVarInfo & *varInfoZ*, stVarInfo & *varInfoT*, std::vector< stVarInfo > & *vecVarInfo*, CDM::E\_CDM\_ERRORCODE & *ret* )  
[static]

nc データファイルのデータレコード読み込み (static 関数)

引数

in	<i>ncid</i>	ncid
in	<i>step_index</i>	読み込むステップのインデクス
in	<i>VariableName</i>	読み込む変数名リスト
in	<i>DataType</i>	読み込む変数のデータ型
in	<i>varInfoX</i>	x 座標情報
in	<i>varInfoY</i>	y 座標情報
in	<i>varInfoZ</i>	z 座標情報
in	<i>varInfoT</i>	time 情報
out	<i>vecVarInfo</i>	読み込んだ変数の varInfo
out	<i>ret</i>	error code

戻り値

cdm\_Array のポインタ

6.7.3.11 CDM::E\_CDM\_ERRORCODE cdm\_DFI\_NETCDF::read\_Datarecord ( cdm\_FILE \* *pFile*, bool *matchEndian*, unsigned *step*, cdm\_Array \* *buf*, int *head[3]*, int *nz*, cdm\_Array \*& *src* ) [protected],[virtual]

nc データファイルのデータレコード読み込み

引数

in	<i>pFile</i>	ファイルポインタ
in	<i>matchEndian</i>	true:Endian 一致
in	<i>step</i>	ステップ番号
in	<i>buf</i>	読み込み用バッファ
in	<i>head</i>	読み込みバッファHeadIndex
in	<i>nz</i>	z 方向のボクセルサイズ (実セル + ガイドセル * 2 )
out	<i>src</i>	読み込んだデータを格納した配列のポインタ

戻り値

error code

cdm\_DFIを実装しています。

```
6.7.3.12 static CDM::E_CDM_ERRORCODE cdm_DFI_NETCDF::read_HeaderRecord ( int ncid, std::string nameX,  
std::string nameY, std::string nameZ, std::string nameT, stVarInfo & varInfoX, stVarInfo & varInfoY, stVarInfo &  
varInfoZ, stVarInfo & varInfoT ) [static]
```

ファイルのヘッダーレコード読み込み (static 関数)

引数

in	<i>ncid</i>	ncid
in	<i>nameX</i>	x の次元, 変数名
in	<i>nameY</i>	y の次元, 変数名
in	<i>nameZ</i>	z の次元, 変数名
in	<i>nameT</i>	time の次元, 変数名
out	<i>varInfoX</i>	x 座標情報
out	<i>varInfoY</i>	y 座標情報
out	<i>varInfoZ</i>	z 座標情報
out	<i>varInfoT</i>	time 情報

戻り値

error code

**6.7.3.13 CDM::E\_CDM\_ERRORCODE** cdm\_DFI\_NETCDF::read\_HeaderRecord ( cdm\_FILE \* *pFile*, bool *matchEndian*, unsigned *step*, const int *head*[3], const int *tail*[3], int *gc*, int *voysize*[3], double & *time* ) [protected], [virtual]

nc ファイルのヘッダーレコード読み込み

引数

in	<i>pFile</i>	ファイルポインタ
in	<i>matchEndian</i>	エンディアンチェックフラグ true:合致
in	<i>step</i>	ステップ番号
in	<i>head</i>	dfi のHeadIndex
in	<i>tail</i>	dfi のTailIndex
in	<i>gc</i>	dfi のガイドセル数
out	<i>voysize</i>	voysize
out	<i>time</i>	時刻

戻り値

error code

[cdm\\_DFI](#)を実装しています。

**6.7.3.14 CDM::E\_CDM\_ERRORCODE** cdm\_DFI\_NETCDF::ReadAdditionalTP ( cdm\_TextParser *tpCntl* )

NetCDF 特有の dfi パラメータ読み込み

引数

in	<i>tpCntl</i>	cdm_TextParser クラス
----	---------------	--------------------

戻り値

error code

**6.7.3.15 void** cdm\_DFI\_NETCDF::SetDimName ( string *nameX*, string *nameY*, string *nameZ*, string *nameT* )

NetCDF の次元変数名のセット

引数

in	<i>nameX</i>	座標値Xの次元名
in	<i>nameY</i>	座標値Yの次元名
in	<i>nameZ</i>	座標値Zの次元名
in	<i>nameT</i>	時刻の次元名

**6.7.3.16** void cdm\_DFI\_NETCDF::SetWriteFlag ( bool *writeFlag* ) [inline]

書き込み済みフラグのセット (FCONV 用)

引数

in	<i>writeFlag</i>	書き込み済みフラグ
----	------------------	-----------

cdm\_DFI\_NETCDF.h の 206 行で定義されています。

参照先 m\_writeFlag.

**6.7.3.17** static void cdm\_DFI\_NETCDF::write\_AttUnits ( int *ncid*, int *varid*, string *var\_name*, cdm\_Unit & *unit* )  
[static], [protected]

UnitList に変数名の unit が存在する場合、var 属性として出力する

引数

in	<i>ncid</i>	ncid
in	<i>varid</i>	var ID
in	<i>var_name</i>	var name
in	<i>unit</i>	unitlist

**6.7.3.18** CDM::E\_CDM\_ERRORCODE cdm\_DFI\_NETCDF::write\_averaged ( cdm\_FILE \* *pFile*, const unsigned *step\_avr*, const double *time\_avr* ) [protected], [virtual]

Average レコードの出力

引数

in	<i>pFile</i>	ファイルポインタ
in	<i>step_avr</i>	平均ステップ番号
in	<i>time_avr</i>	平均時刻

戻り値

error code

[cdm\\_DFI](#)を実装しています。

**6.7.3.19** static CDM::E\_CDM\_ERRORCODE cdm\_DFI\_NETCDF::write\_DataRecord ( cdm\_FILE \* *pFile*, cdm\_Array \* *val*, const int *gc*, int *VoxelSize*[3], stVarInfo & *varInfoT*, vector< stVarInfo > & *vecVarInfo* ) [static]

nc ファイルのデータレコードの出力 (static 関数)

引数

in	<i>pFile</i>	ファイルポインタ
in	<i>val</i>	データポインタ
in	<i>gc</i>	ガイドセル
in	<i>RankID</i>	ランク番号

戻り値

error code

**6.7.3.20** `CDM::E_CDM_ERRORCODE cdm_DFI_NETCDF::write_DataRecord ( cdm_FILE * pFile, cdm_Array * val,  
const int gc, const int RankID )` [protected],[virtual]

nc ファイルのデータレコードの出力

引数

in	<i>pFile</i>	ファイルポインタ
in	<i>val</i>	データポインタ
in	<i>gc</i>	ガイドセル
in	<i>RankID</i>	ランク番号

戻り値

error code

[cdm\\_DFI](#)を実装しています。

**6.7.3.21** `static CDM::E_CDM_ERRORCODE cdm_DFI_NETCDF::write_HeaderRecord ( cdm_FILE * pFile, const  
unsigned step, const double time, int VoxelSize[3], int GuideCell, CDM::E_CDM_DTYPE DataType, vector<  
string > & vecVariable, cdm_Unit & Unit, stVarInfo & varInfoX, stVarInfo & varInfoY, stVarInfo & varInfoZ,  
stVarInfo & varInfoT, vector< stVarInfo > & vecVarInfo )` [static]

nc ファイルのヘッダレコードの出力 (static 関数)

引数

in	<i>pFile</i>	ファイルポインタ
in	<i>step</i>	ステップ番号
in	<i>time</i>	時刻
in	<i>RankID</i>	ランク番号

戻り値

error code

**6.7.3.22** `CDM::E_CDM_ERRORCODE cdm_DFI_NETCDF::write_HeaderRecord ( cdm_FILE * pFile, const unsigned  
step, const double time, const int RankID )` [protected],[virtual]

nc ファイルのヘッダレコードの出力

引数

in	<i>pFile</i>	ファイルポインタ
in	<i>step</i>	ステップ番号
in	<i>time</i>	時刻
in	<i>RankID</i>	ランク番号

戻り値

error code

cdm\_DFIを実装しています。

#### 6.7.3.23 CDM::E\_CDM\_ERRORCODE cdm\_DFI\_NETCDF::WriteAdditionalTP ( FILE \* *fp*, int *tab* )

NetCDF 特有の dfi パラメータ出力

引数

in	<i>fp</i>	ファイルポインタ
----	-----------	----------

戻り値

error code

### 6.7.4 変数

#### 6.7.4.1 string cdm\_DFI\_NETCDF::m\_nameT [protected]

nc ファイルの時刻値配列の配列名

cdm\_DFI\_NETCDF.h の 71 行で定義されています。

参照元 cdm\_DFI\_NETCDF().

#### 6.7.4.2 string cdm\_DFI\_NETCDF::m\_nameX [protected]

nc ファイルのX 座標値配列の配列名

cdm\_DFI\_NETCDF.h の 68 行で定義されています。

参照元 cdm\_DFI\_NETCDF().

#### 6.7.4.3 string cdm\_DFI\_NETCDF::m\_nameY [protected]

nc ファイルのY 座標値配列の配列名

cdm\_DFI\_NETCDF.h の 69 行で定義されています。

参照元 cdm\_DFI\_NETCDF().

#### 6.7.4.4 string cdm\_DFI\_NETCDF::m\_nameZ [protected]

nc ファイルのZ 座標値配列の配列名

cdm\_DFI\_NETCDF.h の 70 行で定義されています。

参照元 cdm\_DFI\_NETCDF().

6.7.4.5 `cdm_Unit cdm_DFI_NETCDF::m_NC_Unit` [protected]

nc ファイル内の unit 情報

cdm\_DFI\_NETCDF.h の 78 行で定義されています。

6.7.4.6 `stVarInfo cdm_DFI_NETCDF::m_varInfoT` [protected]

t の変数情報

cdm\_DFI\_NETCDF.h の 76 行で定義されています。

6.7.4.7 `stVarInfo cdm_DFI_NETCDF::m_varInfoX` [protected]

x の変数情報

cdm\_DFI\_NETCDF.h の 73 行で定義されています。

6.7.4.8 `stVarInfo cdm_DFI_NETCDF::m_varInfoY` [protected]

y の変数情報

cdm\_DFI\_NETCDF.h の 74 行で定義されています。

6.7.4.9 `stVarInfo cdm_DFI_NETCDF::m_varInfoZ` [protected]

z の変数情報

cdm\_DFI\_NETCDF.h の 75 行で定義されています。

6.7.4.10 `vector<stVarInfo> cdm_DFI_NETCDF::m_vecVarInfo` [protected]

データレコード出力用の var id(変数定義順に格納) write\_HeaderRecord で格納し、write\_DataRecord で配列出力時に使用

cdm\_DFI\_NETCDF.h の 83 行で定義されています。

6.7.4.11 `bool cdm_DFI_NETCDF::m_writeFlag` [protected]

書き込み済みフラグ

cdm\_DFI\_NETCDF.h の 85 行で定義されています。

参照元 `cdm_DFI_NETCDF()`, と `SetWriteFlag()`.

このクラスの説明は次のファイルから生成されました:

- [cdm\\_DFI\\_NETCDF.h](#)

## 6.8 クラス cdm\_DFI\_PLOT3D

```
#include <cdm_DFI_PLOT3D.h>
```

cdm\_DFI\_PLOT3D に対する継承グラフ

cdm\_DFI\_PLOT3D のコラボレーション図

## Public メソッド

- `cdm_DFI_PLOT3D ()`
- `cdm_DFI_PLOT3D (const cdm_FileInfo F_Info, const cdm_FilePath F_Path, const cdm_VisIt visit, const cdm_Unit unit, const cdm_Domain *domain, const cdm_MPI mpi, const cdm_TimeSlice TSlice, const cdm_Process process)`  
コンストラクタ
- `~cdm_DFI_PLOT3D ()`
- `template<class T >`  
`CDM_INLINE CDM::E_CDM_ERRORCODE read_Func (FILE *fp, cdm_TypeArray< T > *dataS, cdm_TypeArray< T > *dataB, int head[3], int nz, bool matchEndian)`
- `template<class T >`  
`CDM_INLINE void write_XYZ (FILE *fp, int sz[3], int head[3], const int *iblack)`
- `template<class T >`  
`CDM_INLINE void write_Func (FILE *fp, cdm_TypeArray< T > *data, const int sz[3], int nvari)`

## Protected メソッド

- `CDM::E_CDM_ERRORCODE read_HeaderRecord (cdm_FILE *pFile, bool matchEndian, unsigned step, const int head[3], const int tail[3], int gc, int voxsize[3], double &time)`  
*plot3d ファイルのヘッダーレコード読み込み*
- `CDM::E_CDM_ERRORCODE read_Datarecord (cdm_FILE *pFile, bool matchEndian, unsigned step, cdm_Array *buf, int head[3], int nz, cdm_Array *&src)`  
*フィールドデータファイルのデータレコード読み込み*
- `CDM::E_CDM_ERRORCODE read_averaged (cdm_FILE *pFile, bool matchEndian, unsigned step, unsigned &avr_step, double &avr_time)`  
*plot3d ファイルのAverage データレコードの読み込み*
- `template<class T >`  
`CDM::E_CDM_ERRORCODE read_Func (FILE *fp, cdm_TypeArray< T > *dataS, cdm_TypeArray< T > *dataB, int head[3], int nz, bool matchEndian)`  
*func data 読み込み*
- `CDM::E_CDM_ERRORCODE write_HeaderRecord (cdm_FILE *pFile, const unsigned step, const double time, const int RankID)`  
*avs ヘッダファイルの出力*
- `CDM::E_CDM_ERRORCODE write_DataRecord (cdm_FILE *pFile, cdm_Array *val, const int gc, const int RankID)`  
*avs データレコードの出力*
- `CDM::E_CDM_ERRORCODE write_averaged (cdm_FILE *pFile, const unsigned step_avr, const double time_avr)`  
*Average レコードの出力*
- `bool write_GridData (const int *iblack)`  
*Grid data file 出力 コントロール*
- `template<class T >`  
`void write_XYZ (FILE *fp, int sz[3], int head[3], const int *iblack)`  
*xyz を出力*
- `template<class T >`  
`void write_Func (FILE *fp, cdm_TypeArray< T > *data, const int sz[3], int nvari)`  
*func data 出力*

## Additional Inherited Members

### 6.8.1 説明

cdm\_DFI\_PLOT3D.h の 20 行で定義されています。



## 6.8.2 コンストラクタとデストラクタ

### 6.8.2.1 cdm\_DFI\_PLOT3D::cdm\_DFI\_PLOT3D ( )

コンストラクタ

```
6.8.2.2 cdm_DFI_PLOT3D::cdm_DFI_PLOT3D ( const cdm_FileInfo F_Info, const cdm_FilePath F_Path, const
        cdm_VisIt visit, const cdm_Unit unit, const cdm_Domain * domain, const cdm_MPI mpi, const
        cdm_TimeSlice TSlice, const cdm_Process process ) [inline]
```

コンストラクタ

引数

in	<i>F_Info</i>	FileInfo
in	<i>F_Path</i>	FilePath
in	<i>visit</i>	VisIt option
in	<i>unit</i>	Unit
in	<i>domain</i>	Domain
in	<i>mpi</i>	MPI
in	<i>TSlice</i>	TimeSlice
in	<i>process</i>	Process

cdm\_DFI\_PLOT3D.h の 38 行で定義されています。

参照先 cdm\_DFI::DFI\_Domain, cdm\_DFI::DFI\_Finfo, cdm\_DFI::DFI\_Fpath, cdm\_DFI::DFI\_MPI, cdm\_DFI::DFI\_Process, cdm\_DFI::DFI\_TimeSlice, cdm\_DFI::DFI\_Unit, cdm\_DFI::DFI\_VisIt, CDM::E\_CDM\_FILE\_TYPE\_FBINARY, cdm\_DFI::m\_bgrid\_interp\_flag, cdm\_DFI::m\_input\_type, と cdm\_DFI::m\_output\_type.

### 6.8.2.3 cdm\_DFI\_PLOT3D::~cdm\_DFI\_PLOT3D ( )

デストラクタ

## 6.8.3 関数

```
6.8.3.1 CDM::E_CDM_ERRORCODE cdm_DFI_PLOT3D::read_averaged ( cdm_FILE * pFile, bool matchEndian,
        unsigned step, unsigned & avr_step, double & avr_time ) [protected],[virtual]
```

plot3d ファイルのAverage データレコードの読み込み

引数

in	<i>pFile</i>	ファイルポインタ
in	<i>matchEndian</i>	true:Endian 一致
in	<i>step</i>	読み込み step 番号
out	<i>avr_step</i>	平均ステップ
out	<i>avr_time</i>	平均タイム

戻り値

error code

[cdm\\_DFI](#)を実装しています。

6.8.3.2 CDM::E\_CDM\_ERRORCODE cdm\_DFI\_PLOT3D::read\_Datarecord ( cdm\_FILE \* *pFile*, bool *matchEndian*, unsigned *step*, cdm\_Array \* *buf*, int *head*[3], int *nz*, cdm\_Array \*& *src* ) [protected], [virtual]

フィールドデータファイルのデータレコード読み込み

引数

in	<i>pFile</i>	ファイルポインタ
in	<i>matchEndian</i>	true:Endian 一致
in	<i>step</i>	読み込み step 番号
in	<i>buf</i>	読み込み用バッファ
in	<i>head</i>	読み込みバッファHeadIndex
in	<i>nz</i>	z 方向のボクセルサイズ (実セル + ガイドセル * 2)
out	<i>src</i>	読み込んだデータを格納した配列のポインタ

戻り値

error code

`cdm_DFI`を実装しています。

6.8.3.3 `template<class T> CDM_INLINE CDM::E_CDM_ERRORCODE cdm_DFI_PLOT3D::read_Func ( FILE * fp, cdm_TypeArray< T > * dataS, cdm_TypeArray< T > * dataB, int head[3], int nz, bool matchEndian )`

`cdm_Plot3d_inline.h` の 33 行で定義されています。

参照先 `cdm_TypeArray< T >::copyArrayNvari()`, `CDM::E_CDM_ERROR_READ_FIELD_DATA_RECORD`, `CDM::E_CDM_FILE_TYPE_ASCII`, `CDM::E_CDM_FILE_TYPE_FBINAR`, `CDM::E_CDM_IJKN`, `CDM::E_CDM_SUCCESS`, `cdm_Array::getArrayLength()`, `cdm_Array::getArrayShape()`, `cdm_Array::getArraySizeInt()`, `cdm_Array::getNvari()`, `cdm_DFI::m_input_type`, `cdm_TypeArray< T >::readBinary()`, `cdm_Array::setHeadIndex()`, と `cdm_TypeArray< T >::val()`.

6.8.3.4 `template<class T> CDM::E_CDM_ERRORCODE cdm_DFI_PLOT3D::read_Func ( FILE * fp, cdm_TypeArray< T > * dataS, cdm_TypeArray< T > * dataB, int head[3], int nz, bool matchEndian ) [protected]`

func data 読み込み

引数

in	<i>fp</i>	読み込みファイルポインタ
in	<i>dataS</i>	読み込みデータポインタ
in	<i>dataB</i>	読み込みバッファポインタ
in	<i>head</i>	読み込みバッファHeadIndex
in	<i>nz</i>	z 方向のボクセルサイズ (実セル + ガイドセル * 2)
in	<i>matchEndian</i>	true:Endian 一致

6.8.3.5 `CDM::E_CDM_ERRORCODE cdm_DFI_PLOT3D::read_HeaderRecord ( cdm_FILE * pFile, bool matchEndian, unsigned step, const int head[3], const int tail[3], int gc, int voxsize[3], double & time ) [protected], [virtual]`

plot3d ファイルのヘッダーレコード読み込み

引数

in	<i>pFile</i>	ファイルポインタ
in	<i>matchEndian</i>	エンディアンチェックフラグ true:合致
in	<i>step</i>	ステップ番号

in	<i>head</i>	dfi のHeadIndex
in	<i>tail</i>	dfi のTailIndex
in	<i>gc</i>	dfi のガイドセル数
out	<i>voysize</i>	voysize
out	<i>time</i>	時刻

戻り値

error code

[cdm\\_DFI](#)を実装しています。

**6.8.3.6** `CDM::E_CDM_ERRORCODE cdm_DFI_PLOT3D::write_averaged ( cdm_FILE * pFile, const unsigned step_avr, const double time_avr )` [protected],[virtual]

Average レコードの出力

引数

in	<i>pFile</i>	ファイルポインタ
in	<i>step_avr</i>	平均ステップ番号
in	<i>time_avr</i>	平均時刻

戻り値

error code

[cdm\\_DFI](#)を実装しています。

**6.8.3.7** `CDM::E_CDM_ERRORCODE cdm_DFI_PLOT3D::write_DataRecord ( cdm_FILE * pFile, cdm_Array * val, const int gc, const int RankID )` [protected],[virtual]

avs データレコードの出力

引数

in	<i>pFile</i>	ファイルポインタ
in	<i>val</i>	データポインタ
in	<i>gc</i>	ガイドセル
in	<i>RankID</i>	ランク番号

戻り値

error code

[cdm\\_DFI](#)を実装しています。

**6.8.3.8** `template<class T> void cdm_DFI_PLOT3D::write_Func ( FILE * fp, cdm_TypeArray< T > * data, const int sz[3], int nvari )` [protected]

func data 出力

引数

in	<i>fp</i>	出力ファイルポインタ
in	<i>data</i>	出力データポインタ
in	<i>sz</i>	出力データのサイズ
in	<i>nvari</i>	出力変数の個数

6.8.3.9 `template<class T > CDM_INLINE void cdm_DFI_PLOT3D::write_Func ( FILE * fp, cdm_TypeArray< T > * data, const int sz[3], int nvari )`

cdm\_Plot3d\_inline.h の 267 行で定義されています。

参照先 `cdm_TypeArray< T >::_val()`, `CDM::E_CDM_FILE_TYPE_ASCII`, `CDM::E_CDM_FILE_TYPE_FBINARY`, `CDM::E_CDM_IJKN`, `cdm_Array::getArrayShape()`, と `cdm_DFI::m_output_type`.

6.8.3.10 `bool cdm_DFI_PLOT3D::write_GridData ( const int * iblack ) [protected], [virtual]`

Grid data file 出力 コントロール

引数

in	<i>iblack</i>	iblack データポインタ
----	---------------	----------------

[cdm\\_DFI](#)を再定義しています。

6.8.3.11 `CDM::E_CDM_ERRORCODE cdm_DFI_PLOT3D::write_HeaderRecord ( cdm_FILE * pFile, const unsigned step, const double time, const int RankID ) [protected], [virtual]`

avs ヘッダファイルの出力

引数

in	<i>pFile</i>	ファイルポインタ
in	<i>step</i>	ステップ番号
in	<i>time</i>	時刻
in	<i>RankID</i>	ランク番号

戻り値

error code

[cdm\\_DFI](#)を実装しています。

6.8.3.12 `template<class T > CDM_INLINE void cdm_DFI_PLOT3D::write_XYZ ( FILE * fp, int sz[3], int head[3], const int * iblack )`

cdm\_Plot3d\_inline.h の 122 行で定義されています。

参照先 `cdm_Domain::CellX()`, `cdm_Domain::CellY()`, `cdm_Domain::CellZ()`, `cdm_DFI::DFI_Domain`, `cdm_DFI::D-Fl_Finfo`, `CDM::E_CDM_FILE_TYPE_ASCII`, `CDM::E_CDM_FILE_TYPE_FBINARY`, `cdm_FileInfo::GuideCell`, と `cdm_DFI::m_output_type`.

6.8.3.13 `template<class T > void cdm_DFI_PLOT3D::write_XYZ ( FILE * fp, int sz[3], int head[3], const int * iblack ) [protected]`

xyz を出力

引数

in	<i>fp</i>	出力ファイルポインタ
in	<i>sz</i>	サイズ
in	<i>head</i>	dfi のHeadIndex
in	<i>iblack</i>	iblack データポインタ

このクラスの説明は次のファイルから生成されました:

- [cdm\\_DFI\\_PLOT3D.h](#)
- [cdm\\_Plot3d\\_inline.h](#)

## 6.9 クラス cdm\_DFI\_SPH

```
#include <cdm_DFI_SPH.h>
```

cdm\_DFI\_SPH に対する継承グラフ

cdm\_DFI\_SPH のコラボレーション図

### Public メソッド

- [cdm\\_DFI\\_SPH](#) ()
- [cdm\\_DFI\\_SPH](#) (const [cdm\\_FileInfo](#) F\_Info, const [cdm\\_FilePath](#) F\_Path, const [cdm\\_Visit](#) visit, const [cdm\\_Unit](#) unit, const [cdm\\_Domain](#) \*domain, const [cdm\\_MPI](#) mpi, const [cdm\\_TimeSlice](#) TSlice, const [cdm\\_Process](#) process)  
コンストラクタ
- [~cdm\\_DFI\\_SPH](#) ()

### Protected 型

- enum [DataDims](#) { [\\_DATA\\_UNKNOWN](#) =0, [\\_SCALAR](#), [\\_VECTOR](#) }
- enum [RealType](#) { [\\_REAL\\_UNKNOWN](#) =0, [\\_FLOAT](#), [\\_DOUBLE](#) }

### Protected メソッド

- [CDM::E\\_CDM\\_ERRORCODE](#) [read\\_HeaderRecord](#) ([cdm\\_FILE](#) \*pFile, bool matchEndian, unsigned step, const int head[3], const int tail[3], int gc, int voxsize[3], double &time)  
*sph* ファイルのヘッダーレコード読み込み
- [CDM::E\\_CDM\\_ERRORCODE](#) [read\\_Datarecord](#) ([cdm\\_FILE](#) \*pFile, bool matchEndian, unsigned step, [cdm\\_Array](#) \*buf, int head[3], int nz, [cdm\\_Array](#) \*&src)  
フィールドデータファイルのデータレコード読み込み
- [CDM::E\\_CDM\\_ERRORCODE](#) [read\\_averaged](#) ([cdm\\_FILE](#) \*pFile, bool matchEndian, unsigned step, unsigned &avr\_step, double &avr\_time)  
*sph* ファイルのAverage データレコードの読み込み
- [CDM::E\\_CDM\\_ERRORCODE](#) [write\\_HeaderRecord](#) ([cdm\\_FILE](#) \*pFile, const unsigned step, const double time, const int RankID)  
*SPH* ヘッダレコードの出力
- [CDM::E\\_CDM\\_ERRORCODE](#) [write\\_DataRecord](#) ([cdm\\_FILE](#) \*pFile, [cdm\\_Array](#) \*val, const int gc, const int RankID)  
*SPH* データレコードの出力
- [CDM::E\\_CDM\\_ERRORCODE](#) [write\\_averaged](#) ([cdm\\_FILE](#) \*pFile, const unsigned step\_avr, const double time\_avr)  
Average レコードの出力

## Additional Inherited Members

### 6.9.1 説明

cdm\_DFI\_SPH.h の 20 行で定義されています。

### 6.9.2 列挙型

#### 6.9.2.1 enum cdm\_DFI\_SPH::DataDims [protected]

data dims(scalar or vector)

列挙型の値

**`_DATA_UNKNOWN`**  
**`_SCALAR`**  
**`_VECTOR`**

cdm\_DFI\_SPH.h の 25 行で定義されています。

#### 6.9.2.2 enum cdm\_DFI\_SPH::RealType [protected]

data type(float or double)

列挙型の値

**`_REAL_UNKNOWN`**  
**`_FLOAT`**  
**`_DOUBLE`**

cdm\_DFI\_SPH.h の 28 行で定義されています。

### 6.9.3 コンストラクタとデストラクタ

#### 6.9.3.1 cdm\_DFI\_SPH::cdm\_DFI\_SPH ( )

コンストラクタ

#### 6.9.3.2 cdm\_DFI\_SPH::cdm\_DFI\_SPH ( const cdm\_FileInfo *F\_Info*, const cdm\_FilePath *F\_Path*, const cdm\_VisIt *visit*, const cdm\_Unit *unit*, const cdm\_Domain \* *domain*, const cdm\_MPI *mpi*, const cdm\_TimeSlice *TSlice*, const cdm\_Process *process* ) [inline]

コンストラクタ

引数

in	<i>F_Info</i>	FileInfo
in	<i>F_Path</i>	FilePath
in	<i>visit</i>	VisIt option
in	<i>unit</i>	Unit

in	<i>domain</i>	Domain
in	<i>mpi</i>	MPI
in	<i>TSlice</i>	TimeSlice
in	<i>process</i>	Process

cdm\_DFI\_SPH.h の 46 行で定義されています。

参照先 cdm\_DFI::DFI\_Domain, cdm\_DFI::DFI\_Finfo, cdm\_DFI::DFI\_Fpath, cdm\_DFI::DFI\_MPI, cdm\_DFI::DFI\_Process, cdm\_DFI::DFI\_TimeSlice, cdm\_DFI::DFI\_Unit, cdm\_DFI::DFI\_VisIt, と cdm\_DFI::m\_bgrid\_interp\_flag.

### 6.9.3.3 cdm\_DFI\_SPH::~cdm\_DFI\_SPH ( )

デストラクタ

## 6.9.4 関数

### 6.9.4.1 CDM::E\_CDM\_ERRORCODE cdm\_DFI\_SPH::read\_averaged ( cdm\_FILE \* *pFile*, bool *matchEndian*, unsigned *step*, unsigned & *avr\_step*, double & *avr\_time* ) [protected], [virtual]

sph ファイルのAverage データレコードの読み込み

引数

in	<i>pFile</i>	ファイルポインタ
in	<i>matchEndian</i>	true:Endian 一致
in	<i>step</i>	読み込み step 番号
out	<i>avr_step</i>	平均ステップ
out	<i>avr_time</i>	平均タイム

戻り値

error code

cdm\_DFIを実装しています。

### 6.9.4.2 CDM::E\_CDM\_ERRORCODE cdm\_DFI\_SPH::read\_Datarecord ( cdm\_FILE \* *pFile*, bool *matchEndian*, unsigned *step*, cdm\_Array \* *buf*, int *head*[3], int *nz*, cdm\_Array \* & *src* ) [protected], [virtual]

フィールドデータファイルのデータレコード読み込み

引数

in	<i>pFile</i>	ファイルポインタ
in	<i>matchEndian</i>	true:Endian 一致
in	<i>step</i>	ステップ番号
in	<i>buf</i>	読み込み用バッファ
in	<i>head</i>	読み込みバッファHeadIndex
in	<i>nz</i>	z 方向のボクセルサイズ ( 実セル + ガイドセル * 2 )
out	<i>src</i>	読み込んだデータを格納した配列のポインタ

戻り値

error code

cdm\_DFIを実装しています。



6.9.4.3 CDM::E\_CDM\_ERRORCODE cdm\_DFI\_SPH::read\_HeaderRecord ( cdm\_FILE \* *pFile*, bool *matchEndian*, unsigned *step*, const int *head*[3], const int *tail*[3], int *gc*, int *voxsizes*[3], double & *time* ) [protected], [virtual]

sph ファイルのヘッダーレコード読み込み

## 引数

in	<i>pFile</i>	ファイルポインタ
in	<i>matchEndian</i>	エンディアンチェックフラグ true:合致
in	<i>step</i>	ステップ番号
in	<i>head</i>	dfi のHeadIndex
in	<i>tail</i>	dfi のTailIndex
in	<i>gc</i>	dfi のガイドセル数
out	<i>voysize</i>	voysize
out	<i>time</i>	時刻

## 戻り値

error code

[cdm\\_DFI](#)を実装しています。

**6.9.4.4 CDM::E\_CDM\_ERRORCODE cdm\_DFI\_SPH::write\_averaged ( cdm\_FILE \* *pFile*, const unsigned *step\_avr*, const double *time\_avr* )** [protected],[virtual]

## Average レコードの出力

## 引数

in	<i>pFile</i>	ファイルポインタ
in	<i>step_avr</i>	平均ステップ番号
in	<i>time_avr</i>	平均時刻

## 戻り値

error code

[cdm\\_DFI](#)を実装しています。

**6.9.4.5 CDM::E\_CDM\_ERRORCODE cdm\_DFI\_SPH::write\_DataRecord ( cdm\_FILE \* *pFile*, cdm\_Array \* *val*, const int *gc*, const int *RankID* )** [protected],[virtual]

## SPH データレコードの出力

## 引数

in	<i>pFile</i>	ファイルポインタ
in	<i>val</i>	データポインタ
in	<i>gc</i>	ガイドセル
in	<i>RankID</i>	ランク番号

## 戻り値

error code

[cdm\\_DFI](#)を実装しています。

**6.9.4.6 CDM::E\_CDM\_ERRORCODE cdm\_DFI\_SPH::write\_HeaderRecord ( cdm\_FILE \* *pFile*, const unsigned *step*, const double *time*, const int *RankID* )** [protected],[virtual]

## SPH ヘッダレコードの出力

引数

in	<i>pFile</i>	ファイルポインタ
in	<i>step</i>	ステップ番号
in	<i>time</i>	時刻
in	<i>RankID</i>	ランク番号

戻り値

error code

cdm\_DFIを実装しています。

このクラスの説明は次のファイルから生成されました:

- [cdm\\_DFI\\_SPH.h](#)

## 6.10 クラス cdm\_DFI\_VTK

```
#include <cdm_DFI_VTK.h>
```

cdm\_DFI\_VTK に対する継承グラフ

cdm\_DFI\_VTK のコラボレーション図

### Public メソッド

- [cdm\\_DFI\\_VTK \(\)](#)
- [cdm\\_DFI\\_VTK \(const cdm\\_FileInfo F\\_Info, const cdm\\_FilePath F\\_Path, const cdm\\_VisIt visit, const cdm\\_Unit unit, const cdm\\_Domain \\*domain, const cdm\\_MPI mpi, const cdm\\_TimeSlice TSlice, const cdm\\_Process process\)](#)  
コンストラクタ
- [~cdm\\_DFI\\_VTK \(\)](#)

### Protected メソッド

- [CDM::E\\_CDM\\_ERRORCODE read\\_HeaderRecord \(cdm\\_FILE \\*pFile, bool matchEndian, unsigned step, const int head\[3\], const int tail\[3\], int gc, int voxsize\[3\], double &time\)](#)  
*vtk* ファイルのヘッダーレコード読み込み
- [CDM::E\\_CDM\\_ERRORCODE read\\_Datarecord \(cdm\\_FILE \\*pFile, bool matchEndian, unsigned step, cdm\\_Array \\*buf, int head\[3\], int nz, cdm\\_Array \\*&src\)](#)  
フィールドデータファイルのデータレコード読み込み
- [CDM::E\\_CDM\\_ERRORCODE read\\_averaged \(cdm\\_FILE \\*pFile, bool matchEndian, unsigned step, unsigned &avr\\_step, double &avr\\_time\)](#)  
*vtk* ファイルのAverage データレコードの読み込み
- [CDM::E\\_CDM\\_ERRORCODE write\\_HeaderRecord \(cdm\\_FILE \\*pFile, const unsigned step, const double time, const int RankID\)](#)  
*avs* ヘッダファイルの出力
- [CDM::E\\_CDM\\_ERRORCODE write\\_DataRecord \(cdm\\_FILE \\*pFile, cdm\\_Array \\*val, const int gc, const int RankID\)](#)  
*avs* データレコードの出力
- [CDM::E\\_CDM\\_ERRORCODE write\\_averaged \(cdm\\_FILE \\*pFile, const unsigned step\\_avr, const double time\\_avr\)](#)  
Average レコードの出力

## Additional Inherited Members

### 6.10.1 説明

cdm\_DFI\_VTK.h の 20 行で定義されています。

### 6.10.2 コンストラクタとデストラクタ

#### 6.10.2.1 cdm\_DFI\_VTK::cdm\_DFI\_VTK ( )

##### コンストラクタ

6.10.2.2 cdm\_DFI\_VTK::cdm\_DFI\_VTK ( const cdm\_FileInfo *F\_Info*, const cdm\_FilePath *F\_Path*, const cdm\_VisIt *visit*, const cdm\_Unit *unit*, const cdm\_Domain \* *domain*, const cdm\_MPI *mpi*, const cdm\_TimeSlice *TSlice*, const cdm\_Process *process* ) [inline]

##### コンストラクタ

##### 引数

in	<i>F_Info</i>	FileInfo
in	<i>F_Path</i>	FilePath
in	<i>visit</i>	VisIt option
in	<i>unit</i>	Unit
in	<i>domain</i>	Domain
in	<i>mpi</i>	MPI
in	<i>TSlice</i>	TimeSlice
in	<i>process</i>	Process

cdm\_DFI\_VTK.h の 40 行で定義されています。

参照先 cdm\_DFI::DFI\_Domain, cdm\_DFI::DFI\_Finfo, cdm\_DFI::DFI\_Fpath, cdm\_DFI::DFI\_MPI, cdm\_DFI::DFI\_Process, cdm\_DFI::DFI\_TimeSlice, cdm\_DFI::DFI\_Unit, cdm\_DFI::DFI\_VisIt, と cdm\_DFI::m\_bgrid\_interp\_flag.

#### 6.10.2.3 cdm\_DFI\_VTK::~cdm\_DFI\_VTK ( )

##### デストラクタ

### 6.10.3 関数

6.10.3.1 CDM::E\_CDM\_ERRORCODE cdm\_DFI\_VTK::read\_averaged ( cdm\_FILE \* *pFile*, bool *matchEndian*, unsigned *step*, unsigned & *avr\_step*, double & *avr\_time* ) [inline],[protected],[virtual]

vtk ファイルのAverage データレコードの読み込み

##### 引数

in	<i>pFile</i>	ファイルポインタ
in	<i>matchEndian</i>	true:Endian 一致
in	<i>step</i>	読み込み step 番号
out	<i>avr_step</i>	平均ステップ
out	<i>avr_time</i>	平均タイム

戻り値

error code

cdm\_DFIを実装しています。

cdm\_DFI\_VTK.h の 123 行で定義されています。

参照先 CDM::E\_CDM\_SUCCESS.

**6.10.3.2 CDM::E\_CDM\_ERRORCODE cdm\_DFI\_VTK::read\_Datarecord ( cdm\_FILE \* pFile, bool matchEndian, unsigned step, cdm\_Array \* buf, int head[3], int nz, cdm\_Array \*& src )** [inline],[protected],[virtual]

フィールドデータファイルのデータレコード読み込み

引数

in	<i>pFile</i>	ファイルポインタ
in	<i>matchEndian</i>	true:Endian 一致
in	<i>step</i>	ステップ番号
in	<i>buf</i>	読み込み用バッファ
in	<i>head</i>	読み込みバッファHeadIndex
in	<i>nz</i>	z 方向のボクセルサイズ ( 実セル + ガイドセル * 2 )
out	<i>src</i>	読み込んだデータを格納した配列のポインタ

戻り値

error code

cdm\_DFIを実装しています。

cdm\_DFI\_VTK.h の 103 行で定義されています。

参照先 CDM::E\_CDM\_SUCCESS.

**6.10.3.3 CDM::E\_CDM\_ERRORCODE cdm\_DFI\_VTK::read\_HeaderRecord ( cdm\_FILE \* pFile, bool matchEndian, unsigned step, const int head[3], const int tail[3], int gc, int voxsize[3], double & time )** [inline],[protected],[virtual]

vtk ファイルのヘッダーレコード読み込み

引数

in	<i>pFile</i>	ファイルポインタ
in	<i>matchEndian</i>	エンディアンチェックフラグ true:合致
in	<i>step</i>	ステップ番号
in	<i>head</i>	dfi のHeadIndex
in	<i>tail</i>	dfi のTailIndex
in	<i>gc</i>	dfi のガイドセル数
out	<i>voxsize</i>	voxsize
out	<i>time</i>	時刻

戻り値

error code

cdm\_DFIを実装しています。

cdm\_DFI\_VTK.h の 81 行で定義されています。

参照先 CDM::E\_CDM\_SUCCESS.

**6.10.3.4 CDM::E\_CDM\_ERRORCODE cdm\_DFI\_VTK::write\_averaged ( cdm\_FILE \* *pFile*, const unsigned *step\_avr*, const double *time\_avr* )** [inline],[protected],[virtual]

Average レコードの出力

引数

in	<i>pFile</i>	ファイルポインタ
in	<i>step_avr</i>	平均ステップ番号
in	<i>time_avr</i>	平均時刻

戻り値

error code

[cdm\\_DFI](#)を実装しています。

cdm\_DFI\_VTK.h の 169 行で定義されています。

参照先 CDM::E\_CDM\_SUCCESS.

**6.10.3.5 CDM::E\_CDM\_ERRORCODE cdm\_DFI\_VTK::write\_DataRecord ( cdm\_FILE \* *pFile*, cdm\_Array \* *val*, const int *gc*, const int *RankID* )** [protected],[virtual]

avs データレコードの出力

引数

in	<i>pFile</i>	ファイルポインタ
in	<i>val</i>	データポインタ
in	<i>gc</i>	ガイドセル
in	<i>RankID</i>	ランク番号

戻り値

error code

[cdm\\_DFI](#)を実装しています。

**6.10.3.6 CDM::E\_CDM\_ERRORCODE cdm\_DFI\_VTK::write\_HeaderRecord ( cdm\_FILE \* *pFile*, const unsigned *step*, const double *time*, const int *RankID* )** [protected],[virtual]

avs ヘッダファイルの出力

引数

in	<i>pFile</i>	ファイルポインタ
in	<i>step</i>	ステップ番号
in	<i>time</i>	時刻
in	<i>RankID</i>	ランク番号

戻り値

error code

[cdm\\_DFI](#)を実装しています。

このクラスの説明は次のファイルから生成されました:

- [cdm\\_DFI\\_VTK.h](#)

## 6.11 クラス `cdm_Domain`

```
#include <cdm_Domain.h>
```

`cdm_Domain` に対する継承グラフ

### Public メソッド

- `cdm_Domain` ()
- `cdm_Domain` (const double \*\_GlobalOrigin, const double \*\_GlobalPitch, const int \*\_GlobalVoxel, const int \*\_GlobalDivision)  
コンストラクタ (`_GlobalOrigin`, `_GlobalPitch` は、`double` 型と `float` 型の両方あり)
- `cdm_Domain` (const float \*\_GlobalOrigin, const float \*\_GlobalPitch, const int \*\_GlobalVoxel, const int \*\_GlobalDivision)
- virtual `~cdm_Domain` ()
- virtual double `CellX` (int i) const  
セル中心の  $X$  座標を取得 (*AVS, PLOT3D, VTK* 形式)
- virtual double `CellY` (int j) const  
セル中心の  $Y$  座標を取得 (*AVS, PLOT3D, VTK* 形式)
- virtual double `CellZ` (int k) const  
セル中心の  $Z$  座標を取得 (*AVS, PLOT3D, VTK* 形式)
- virtual double `NodeX` (int i) const  
格子点の  $X$  座標を取得 (*AVS, PLOT3D, VTK* 形式)
- virtual double `NodeY` (int j) const  
格子点の  $Y$  座標を取得 (*AVS, PLOT3D, VTK* 形式)
- virtual double `NodeZ` (int k) const  
格子点の  $Z$  座標を取得 (*AVS, PLOT3D, VTK* 形式)
- virtual std::string `GetCoordinateFile` () const
- virtual `CDM::E_CDM_FILE_TYPE` `GetCoordinateFileType` () const
- virtual `CDM::E_CDM_DTYPE` `GetCoordinateFilePrecision` () const
- virtual `CDM::E_CDM_ENDIANTYPE` `GetCoordinateFileEndian` () const
- virtual `CDM::E_CDM_ERRORCODE` `Read` (`cdm_TextParser` tpCntl, std::string dirName)  
*read Domain(proc.dfi)*
- virtual `CDM::E_CDM_ERRORCODE` `Write` (FILE \*fp, const unsigned tab) const  
*DFI ファイル:Domain を出力する*

### Public 変数

- double `GlobalOrigin` [3]  
起点座標
- double `GlobalRegion` [3]  
計算空間の各軸方向の長さ
- int `GlobalVoxel` [3]  
計算領域全体のボクセル数
- int `GlobalDivision` [3]  
計算領域の分割数
- std::string `ActiveSubdomainFile`  
*ActiveSubdomain* ファイル名

### Protected メソッド

- virtual void `Clear` ()

## Private 変数

- double [Pitch](#) [3]  
計算空間のピッチ

### 6.11.1 説明

proc.dfi ファイルの Domain

cdm\_Domain.h の 19 行で定義されています。

### 6.11.2 コンストラクタとデストラクタ

#### 6.11.2.1 cdm\_Domain::cdm\_Domain ( )

コンストラクタ

#### 6.11.2.2 cdm\_Domain::cdm\_Domain ( const double \* *\_GlobalOrigin*, const double \* *\_GlobalPitch*, const int \* *\_GlobalVoxel*, const int \* *\_GlobalDivision* )

コンストラクタ (*\_GlobalOrigin*, *\_GlobalPitch* は、double 型と float 型の両方あり)

AVS,PLOT3D,VTK 形式では、*\_GlobalOrigin* に計算領域全体の原点座標値を与える。SPH,BOV 形式では、*\_GlobalOrigin* に各ランクの局所領域における原点座標値を与える。

引数

in	<i>_GlobalOrigin</i>	起点座標
in	<i>_GlobalPitch</i>	ボクセルの長さ
in	<i>_GlobalVoxel</i>	ボクセル数
in	<i>_GlobalDivision</i>	分割数

#### 6.11.2.3 cdm\_Domain::cdm\_Domain ( const float \* *\_GlobalOrigin*, const float \* *\_GlobalPitch*, const int \* *\_GlobalVoxel*, const int \* *\_GlobalDivision* )

#### 6.11.2.4 virtual cdm\_Domain::~~cdm\_Domain ( ) [virtual]

デストラクタ

### 6.11.3 関数

#### 6.11.3.1 virtual double cdm\_Domain::CellX ( int *i* ) const [inline],[virtual]

セル中心のX 座標を取得 (AVS,PLOT3D,VTK 形式)

引数

in	<i>i</i>	X 方向のセル番号
----	----------	-----------

戻り値

セル中心のX 座標

[cdm\\_NonUniformDomain< T >](#) で再定義されています。

cdm\_Domain.h の 75 行で定義されています。



参照先 `GlobalOrigin`, と `Pitch`.

参照元 `cdm_DFI_PLOT3D::write_XYZ()`.

6.11.3.2 `virtual double cdm_Domain::CellY ( int j ) const` `[inline],[virtual]`

セル中心のY 座標を取得 (AVS,PLOT3D,VTK 形式)

引数

<code>in</code>	<code><i>j</i></code>	Y 方向のセル番号
-----------------	-----------------------	-----------

戻り値

セル中心のY 座標

`cdm_NonUniformDomain< T >` で再定義されています。

`cdm_Domain.h` の 84 行で定義されています。

参照先 `GlobalOrigin`, と `Pitch`.

参照元 `cdm_DFI_PLOT3D::write_XYZ()`.

6.11.3.3 `virtual double cdm_Domain::CellZ ( int k ) const` `[inline],[virtual]`

セル中心のZ 座標を取得 (AVS,PLOT3D,VTK 形式)

引数

<code>in</code>	<code><i>k</i></code>	Z 方向のセル番号
-----------------	-----------------------	-----------

戻り値

セル中心のZ 座標

`cdm_NonUniformDomain< T >` で再定義されています。

`cdm_Domain.h` の 93 行で定義されています。

参照先 `GlobalOrigin`, と `Pitch`.

参照元 `cdm_DFI_PLOT3D::write_XYZ()`.

6.11.3.4 `virtual void cdm_Domain::Clear ( )` `[protected],[virtual]`

`cdm_NonUniformDomain< T >` で再定義されています。

参照元 `cdm_NonUniformDomain< T >::Clear()`.

6.11.3.5 `virtual std::string cdm_Domain::GetCoordinateFile ( ) const` `[inline],[virtual]`

座標ファイル名を取得

`cdm_NonUniformDomain< T >` で再定義されています。

`cdm_Domain.h` の 125 行で定義されています。

6.11.3.6 `virtual CDM::E_CDM_ENDIANTYPE cdm_Domain::GetCoordinateFileEndian ( ) const` `[inline],[virtual]`

座標ファイルのエンディアンタイプを取得

`cdm_NonUniformDomain< T >` で再定義されています。

`cdm_Domain.h` の 134 行で定義されています。

6.11.3.7 `virtual CDM::E_CDM_DTYPE cdm_Domain::GetCoordinateFilePrecision ( ) const [inline],[virtual]`

座標ファイルのデータ精度を取得

`cdm_NonUniformDomain< T >` で再定義されています。

`cdm_Domain.h` の 131 行で定義されています。

6.11.3.8 `virtual CDM::E_CDM_FILE_TYPE cdm_Domain::GetCoordinateFileType ( ) const [inline],[virtual]`

座標ファイルのファイルタイプを取得

`cdm_NonUniformDomain< T >` で再定義されています。

`cdm_Domain.h` の 128 行で定義されています。

6.11.3.9 `virtual double cdm_Domain::NodeX ( int i ) const [inline],[virtual]`

格子点のX 座標を取得 (AVS,PLOT3D,VTK 形式)

引数

in	<i>i</i>	X 方向の格子番号
----	----------	-----------

戻り値

格子点のX 座標

`cdm_NonUniformDomain< T >` で再定義されています。

`cdm_Domain.h` の 102 行で定義されています。

参照先 `GlobalOrigin`, と `Pitch`.

6.11.3.10 `virtual double cdm_Domain::NodeY ( int j ) const [inline],[virtual]`

格子点のY 座標を取得 (AVS,PLOT3D,VTK 形式)

引数

in	<i>j</i>	Y 方向の格子番号
----	----------	-----------

戻り値

格子点のY 座標

`cdm_NonUniformDomain< T >` で再定義されています。

`cdm_Domain.h` の 111 行で定義されています。

参照先 `GlobalOrigin`, と `Pitch`.

6.11.3.11 `virtual double cdm_Domain::NodeZ ( int k ) const [inline],[virtual]`

格子点のZ 座標を取得 (AVS,PLOT3D,VTK 形式)

引数

<code>in</code>	<code>k</code>	Z 方向の格子番号
-----------------	----------------	-----------

戻り値

格子点のZ 座標

`cdm_NonUniformDomain< T >` で再定義されています。

`cdm_Domain.h` の 120 行で定義されています。

参照先 `GlobalOrigin`, と `Pitch`.

6.11.3.12 `virtual CDM::E_CDM_ERRORCODE cdm_Domain::Read ( cdm_TextParser tpCntl, std::string dirName )`  
[virtual]

read Domain(proc.dfi)

引数

<code>in</code>	<code>tpCntl</code>	<code>cdm_TextParser</code> クラス
<code>in</code>	<code>dirName</code>	DFI のディレクトリパス

戻り値

error code

`cdm_NonUniformDomain< T >` で再定義されています。

6.11.3.13 `virtual CDM::E_CDM_ERRORCODE cdm_Domain::Write ( FILE * fp, const unsigned tab ) const`  
[virtual]

DFI ファイル:Domain を出力する

引数

<code>in</code>	<code>fp</code>	ファイルポインタ
<code>in</code>	<code>tab</code>	インデント

戻り値

error code

`cdm_NonUniformDomain< T >` で再定義されています。

## 6.11.4 変数

6.11.4.1 `std::string cdm_Domain::ActiveSubdomainFile`

ActiveSubdomain ファイル名

`cdm_Domain.h` の 38 行で定義されています。

6.11.4.2 `int cdm_Domain::GlobalDivision[3]`

計算領域の分割数

`cdm_Domain.h` の 37 行で定義されています。

参照元 `cdm_NonUniformDomain< T >::cdm_NonUniformDomain()`.

#### 6.11.4.3 double cdm\_Domain::GlobalOrigin[3]

起点座標

cdm\_Domain.h の 23 行で定義されています。

参照元 cdm\_NonUniformDomain< T >::cdm\_NonUniformDomain(), CellX(), CellY(), CellZ(), NodeX(), NodeY(), と NodeZ().

#### 6.11.4.4 double cdm\_Domain::GlobalRegion[3]

計算空間の各軸方向の長さ

cdm\_Domain.h の 35 行で定義されています。

参照元 cdm\_NonUniformDomain< T >::cdm\_NonUniformDomain().

#### 6.11.4.5 int cdm\_Domain::GlobalVoxel[3]

計算領域全体のボクセル数

cdm\_Domain.h の 36 行で定義されています。

参照元 cdm\_NonUniformDomain< T >::cdm\_NonUniformDomain(), cdm\_NonUniformDomain< T >::CellX(), cdm\_NonUniformDomain< T >::CellY(), と cdm\_NonUniformDomain< T >::CellZ().

#### 6.11.4.6 double cdm\_Domain::Pitch[3] [private]

計算空間のピッチ

cdm\_Domain.h の 41 行で定義されています。

参照元 CellX(), CellY(), CellZ(), NodeX(), NodeY(), と NodeZ().

このクラスの説明は次のファイルから生成されました:

- [cdm\\_Domain.h](#)

## 6.12 クラス cdm\_FieldFileNameFormat

```
#include <cdm_FieldFileNameFormat.h>
```

### Public メソッド

- [cdm\\_FieldFileNameFormat \(\)](#)
- [~cdm\\_FieldFileNameFormat \(\)](#)
- [CDM::E\\_CDM\\_ERRORCODE Read \(cdm\\_TextParser tpCntl\)](#)
- [void Print \(\)](#)
- [bool AddFieldFileNameFormatElem \(cdm\\_FieldFileNameFormatElem elem\)](#)
- [cdm\\_FieldFileNameFormatElem \\* GetFieldFileNameFormatElem \(const string label\)](#)
- [vector< string > GetLabelList \(\)](#)
- [bool FileExist \(string label, string DirPath, int nStep, int nld\)](#)
- [string GenerateFileName \(string label, string DirPath, int nStep, int nld\)](#)
- [void Write \(FILE \\*fp, const unsigned tab\)](#)

## Public 変数

- `vector< string > LabelList`
- `map< string, cdm_FieldFileNameFormatElem > mapElem`

### 6.12.1 説明

`cdm_FieldFileNameFormat.h` の 16 行で定義されています。

### 6.12.2 コンストラクタとデストラクタ

#### 6.12.2.1 `cdm_FieldFileNameFormat::cdm_FieldFileNameFormat ( )`

コンストラクタ

#### 6.12.2.2 `cdm_FieldFileNameFormat::~cdm_FieldFileNameFormat ( )`

デストラクタ

### 6.12.3 関数

#### 6.12.3.1 `bool cdm_FieldFileNameFormat::AddFieldFileNameFormatElem ( cdm_FieldFileNameFormatElem elem )`

`FieldFileNameFormatElem` クラスの追加

#### 6.12.3.2 `bool cdm_FieldFileNameFormat::FileExist ( string label, string DirPath, int nStep, int nld )`

File 有無判定

#### 6.12.3.3 `string cdm_FieldFileNameFormat::GenerateFileName ( string label, string DirPath, int nStep, int nld )`

File 名生成

#### 6.12.3.4 `cdm_FieldFileNameFormatElem* cdm_FieldFileNameFormat::GetFieldFileNameFormatElem ( const string label )`

`FieldFileNameFormatElem` クラスの取得

#### 6.12.3.5 `vector<string> cdm_FieldFileNameFormat::GetLabelList ( )`

label list の取得

#### 6.12.3.6 `void cdm_FieldFileNameFormat::Print ( )`

パラメータの出力

#### 6.12.3.7 `CDM::E_CDM_ERRORCODE cdm_FieldFileNameFormat::Read ( cdm_TextParser tpCntl )`

`TextParser`

6.12.3.8 void cdm\_FieldFileNameFormat::Write ( FILE \* *fp*, const unsigned *tab* )

index.dfi FieldFileNameFormat{} 出力

## 6.12.4 変数

6.12.4.1 vector<string> cdm\_FieldFileNameFormat::LabelList

cdm\_FieldFileNameFormat.h の 20 行で定義されています。

6.12.4.2 map<string,cdm\_FieldFileNameFormatElem> cdm\_FieldFileNameFormat::mapElem

cdm\_FieldFileNameFormat.h の 22 行で定義されています。

このクラスの説明は次のファイルから生成されました:

- [cdm\\_FieldFileNameFormat.h](#)

## 6.13 クラス cdm\_FieldFileNameFormatElem

```
#include <cdm_FieldFileNameFormatElem.h>
```

### Public メソッド

- [cdm\\_FieldFileNameFormatElem](#) (const string label)  
*FieldFileNameFormat* のファイル名 ( *FieldFile*, *CoordinateFile* )
- [~cdm\\_FieldFileNameFormatElem](#) ()
- bool [Read](#) (TextParser \*tp)
- void [PrintParse](#) ()
- int [GetDisitNo](#) (TextParser \*tp, const string label, int &err)
- bool [Write](#) (FILE \*fp, const unsigned tab)
- string [GenerateFileName](#) (const string DirPath, const int nStep, const int nld)
- bool [FileExist](#) (const string DirPath, const int nStep, const int nld)

### Public 変数

- string [FnameLabel](#)

### Protected メソッド

- void [SetFnameFormat](#) ()
- string [SetDisitNoFormat](#) (const int DisitNo)

### Protected 変数

- string [FileName](#)
- string [StepNoKey](#)  
ステップ番号が付与されるキー文字 ( 例 ) "STEPNO"
- string [RankIdKey](#)  
ランク番号が付与されるキー文字 ( 例 ) "FILENO"

- int `StepNoDisit`  
ステップ番号桁数 ( 正:桁数, 0:shorten, 負:ステップ番号なし)
- int `RankIdDisit`  
ランク番号桁数 ( 正:桁数, 0:shorten, 負:ランク番号なし)
- string `FileNameFormat`
- `FUB_FNAME_TYPE` `FnameFormat`  
`FieldFilenameFormat(enum FUB_FNAME_TYPE)`

### 6.13.1 説明

`cdm_FieldFileNameFormatElem.h` の 31 行で定義されています。

### 6.13.2 コンストラクタとデストラクタ

6.13.2.1 `cdm_FieldFileNameFormatElem::cdm_FieldFileNameFormatElem ( const string label )`

`FieldFileNameFormat` のファイル名 ( `FieldFile`, `CoordinateFile` )

コンストラクタ

6.13.2.2 `cdm_FieldFileNameFormatElem::~cdm_FieldFileNameFormatElem ( )`

デストラクタ

### 6.13.3 関数

6.13.3.1 `bool cdm_FieldFileNameFormatElem::FileExist ( const string DirPath, const int nStep, const int nld )`

ファイル有無判定

6.13.3.2 `string cdm_FieldFileNameFormatElem::GenerateFileName ( const string DirPath, const int nStep, const int nld )`

ファイル名の生成

6.13.3.3 `int cdm_FieldFileNameFormatElem::GetDisitNo ( TextParser * tp, const string label, int & err )`

`Disit` の取得 ( shorten=0, 整数値)

6.13.3.4 `void cdm_FieldFileNameFormatElem::PrintParse ( )`

`Parse` の出力

6.13.3.5 `bool cdm_FieldFileNameFormatElem::Read ( TextParser * tp )`

`TextParser` でのParse

6.13.3.6 `string cdm_FieldFileNameFormatElem::SetDisitNoFormat ( const int DisitNo )` `[protected]`

桁数を考慮した出力形式をセットする

6.13.3.7 `void cdm_FieldFileNameFormatElem::SetNameFormat ( ) [protected]`

FileNameFormat をセットする

6.13.3.8 `bool cdm_FieldFileNameFormatElem::Write ( FILE * fp, const unsigned tab )`

FieldFilenameFormat のXXXXFile 出力

## 6.13.4 変数

6.13.4.1 `string cdm_FieldFileNameFormatElem::FileName [protected]`

読み込まれるフィールドファイル名の規約名（例）"cellPhys\_STEPNO\_idFILENO.fub"  
cdm\_FieldFileNameFormatElem.h の 35 行で定義されています。

6.13.4.2 `string cdm_FieldFileNameFormatElem::FileNameFormat [protected]`

ファイル名生成用のFormat（例）"cellPhy\_%06d\_id%10d.fub"  
cdm\_FieldFileNameFormatElem.h の 44 行で定義されています。

6.13.4.3 `FUB_FNAME_TYPE cdm_FieldFileNameFormatElem::FnameFormat [protected]`

FieldFilenameFormat(enum FUB\_FNAME\_TYPE)  
cdm\_FieldFileNameFormatElem.h の 47 行で定義されています。

6.13.4.4 `string cdm_FieldFileNameFormatElem::FnameLabel`

cdm\_FieldFileNameFormatElem.h の 51 行で定義されています。

6.13.4.5 `int cdm_FieldFileNameFormatElem::RankIdDisit [protected]`

ランク番号桁数（正:桁数, 0:shorten, 負:ランク番号なし）  
cdm\_FieldFileNameFormatElem.h の 42 行で定義されています。

6.13.4.6 `string cdm_FieldFileNameFormatElem::RankIdKey [protected]`

ランク番号が付与されるキー文字（例）"FILENO"  
cdm\_FieldFileNameFormatElem.h の 39 行で定義されています。

6.13.4.7 `int cdm_FieldFileNameFormatElem::StepNoDisit [protected]`

ステップ番号桁数（正:桁数, 0:shorten, 負:ステップ番号なし）  
cdm\_FieldFileNameFormatElem.h の 41 行で定義されています。

6.13.4.8 `string cdm_FieldFileNameFormatElem::StepNoKey [protected]`

ステップ番号が付与されるキー文字（例）"STEPNO"  
cdm\_FieldFileNameFormatElem.h の 38 行で定義されています。



このクラスの説明は次のファイルから生成されました:

- [cdm\\_FieldFileNameFormatElem.h](#)

## 6.14 クラス cdm\_FILE

```
#include <cdm_FILE.h>
```

### Static Public メソッド

- static [cdm\\_FILE](#) \* [OpenReadBinary](#) (const std::string fname, [CDM::E\\_CDM\\_FORMAT](#) format)  
読み込みファイルをオープン
- static [cdm\\_FILE](#) \* [OpenWriteBinary](#) (const std::string fname, [CDM::E\\_CDM\\_FORMAT](#) format, bool add-Mode=false)  
出力ファイルをオープン (*binary* モード)
- static [cdm\\_FILE](#) \* [OpenWriteAscii](#) (const std::string fname, [CDM::E\\_CDM\\_FORMAT](#) format, bool add-Mode=false)  
出力ファイルをオープン (*ascii* モード)
- static void [CloseFile](#) ([cdm\\_FILE](#) \*pFile)  
ファイルをクローズ

### Public 変数

- FILE \* [m\\_fp](#)  
*file pointer*
- int [m\\_ncid](#)  
*netcdf file ID*
- [CDM::E\\_CDM\\_FORMAT](#) [m\\_format](#)  
*file type*
- bool [m\\_addMode](#)  
追記モードフラグ (*true*:追記モード)
- string [m\\_fname](#)  
ファイル名

### Protected メソッド

- [cdm\\_FILE](#) ()
- virtual ~[cdm\\_FILE](#) ()

#### 6.14.1 説明

[CDM](#) file pointer control class

[cdm\\_FILE.h](#) の 30 行で定義されています。

## 6.14.2 コンストラクタとデストラクタ

### 6.14.2.1 `cdm_FILE::cdm_FILE( )` `[inline]`, `[protected]`

#### コンストラクタ

`cdm_FILE.h` の 42 行で定義されています。

参照先 `CDM::E_CDM_FMT_UNKNOWN`, `m_addMode`, `m_fname`, `m_format`, `m_fp`, と `m_ncid`.

参照元 `OpenReadBinary()`, `OpenWriteAscii()`, と `OpenWriteBinary()`.

### 6.14.2.2 `virtual cdm_FILE::~~cdm_FILE( )` `[inline]`, `[protected]`, `[virtual]`

#### デストラクタ

`cdm_FILE.h` の 52 行で定義されています。

参照先 `CDM::E_CDM_FMT_NETCDF4`, `m_format`, `m_fp`, と `m_ncid`.

## 6.14.3 関数

### 6.14.3.1 `static void cdm_FILE::CloseFile( cdm_FILE * pFile )` `[inline]`, `[static]`

#### ファイルをクローズ

##### 引数

<code>in</code>	<code>fp</code>	<code>cdm_DFI</code> クラスのインスタンス
-----------------	-----------------	---------------------------------

`cdm_FILE.h` の 212 行で定義されています。

### 6.14.3.2 `static cdm_FILE* cdm_FILE::OpenReadBinary( const std::string fname, CDM::E_CDM_FORMAT format )` `[inline]`, `[static]`

#### 読み込みファイルをオープン

##### 引数

<code>in</code>	<code>fname</code>	ファイル名
<code>in</code>	<code>format</code>	ファイルタイプ

##### 戻り値

インスタンスされたクラスのポインタ

`cdm_FILE.h` の 74 行で定義されています。

参照先 `cdm_FILE()`, `CDM::E_CDM_FMT_NETCDF4`, `m_fname`, `m_format`, `m_fp`, と `m_ncid`.

### 6.14.3.3 `static cdm_FILE* cdm_FILE::OpenWriteAscii( const std::string fname, CDM::E_CDM_FORMAT format, bool addMode = false )` `[inline]`, `[static]`

#### 出力ファイルをオープン (ascii モード)

##### 引数

in	<i>fname</i>	ファイル名
in	<i>format</i>	ファイルタイプ
in	<i>addmode</i>	追記モードフラグ (true:追記)

戻り値

インスタンスされたクラスのポインタ

cdm\_FILE.h の 177 行で定義されています。

参照先 cdm\_FILE(), CDM::E\_CDM\_FMT\_NETCDF4, m\_addMode, m\_fname, m\_format, m\_fp, と m\_ncid.

**6.14.3.4 static cdm\_FILE\* cdm\_FILE::OpenWriteBinary ( const std::string *fname*, CDM::E\_CDM\_FORMAT *format*, bool *addMode* = false ) [inline],[static]**

出力ファイルをオープン (binary モード)

引数

in	<i>fname</i>	ファイル名
in	<i>format</i>	ファイルタイプ
in	<i>addmode</i>	追記モードフラグ (true:追記)

戻り値

インスタンスされたクラスのポインタ

cdm\_FILE.h の 119 行で定義されています。

参照先 cdm\_FILE(), CDM::E\_CDM\_FMT\_NETCDF4, m\_addMode, m\_fname, m\_format, m\_fp, と m\_ncid.

## 6.14.4 変数

### 6.14.4.1 bool cdm\_FILE::m\_addMode

追記モードフラグ (true:追記モード)

cdm\_FILE.h の 37 行で定義されています。

参照元 cdm\_FILE(), OpenWriteAscii(), と OpenWriteBinary().

### 6.14.4.2 string cdm\_FILE::m\_fname

ファイル名

cdm\_FILE.h の 38 行で定義されています。

参照元 cdm\_FILE(), OpenReadBinary(), OpenWriteAscii(), と OpenWriteBinary().

### 6.14.4.3 CDM::E\_CDM\_FORMAT cdm\_FILE::m\_format

file type

cdm\_FILE.h の 36 行で定義されています。

参照元 cdm\_FILE(), OpenReadBinary(), OpenWriteAscii(), OpenWriteBinary(), と ~cdm\_FILE().

#### 6.14.4.4 FILE\* cdm\_FILE::m\_fp

file pointer

cdm\_FILE.h の 34 行で定義されています。

参照元 cdm\_FILE(), OpenReadBinary(), OpenWriteAscii(), OpenWriteBinary(), と ~cdm\_FILE().

#### 6.14.4.5 int cdm\_FILE::m\_ncid

netcdf file ID

cdm\_FILE.h の 35 行で定義されています。

参照元 cdm\_FILE(), OpenReadBinary(), OpenWriteAscii(), OpenWriteBinary(), と ~cdm\_FILE().

このクラスの説明は次のファイルから生成されました:

- [cdm\\_FILE.h](#)

## 6.15 クラス cdm\_FileInfo

```
#include <cdm_FileInfo.h>
```

### Public メソッド

- [cdm\\_FileInfo \(\)](#)
- [cdm\\_FileInfo \(const CDM::E\\_CDM\\_DFITYPE \\_DFIType, const CDM::E\\_CDM\\_OUTPUT\\_FNAME - \\_FieldFilenameFormat, const std::string \\_DirectoryPath, const CDM::E\\_CDM\\_ONOFF \\_TimeSliceDirFlag, const std::string \\_Prefix, const CDM::E\\_CDM\\_FORMAT \\_FileFormat, const int \\_GuideCell, const CDM::E\\_CDM\\_DTYPE \\_DataType, const CDM::E\\_CDM\\_ENDIANTYPE \\_Endian, const CDM::E\\_CDM\\_ARRAYSHAPE \\_ArrayShape, const int \\_NumVariables, const std::string \\_RankNoPrefix=std::string\(CDM::C\\_CDM\\_RANKNOPREFIX\)\)](#)  
コンストラクタ
- [~cdm\\_FileInfo \(\)](#)
- [CDM::E\\_CDM\\_ERRORCODE Read \(cdm\\_TextParser tpCntl\)](#)  
*read FileInfo(inde.dfi)*
- [CDM::E\\_CDM\\_ERRORCODE Write \(FILE \\*fp, const unsigned tab\)](#)  
*DFI ファイル:FileInfo 要素を出力する*
- [void setVariableName \(int pvari, std::string variName\)](#)  
変数名をセットする
- [std::string getVariableName \(int pvari\)](#)  
変数名を取得する

### Public 変数

- [CDM::E\\_CDM\\_DFITYPE DFIType](#)  
*dfi 種別*
- [CDM::E\\_CDM\\_OUTPUT\\_FNAME FieldFilenameFormat](#)  
ファイル命名基準
- [std::string DirectoryPath](#)
- [CDM::E\\_CDM\\_ONOFF TimeSliceDirFlag](#)  
*TimeSlice on or off.*
- [std::string Prefix](#)

- ファイル接頭文字
- `CDM::E_CDM_FORMAT` `FileFormat`  
ファイルフォーマット `"bov", "sph",,`
- `int` `GuideCell`  
仮想セルの数
- `CDM::E_CDM_DTYPE` `DataType`  
配列のデータタイプ `"float",,`
- `CDM::E_CDM_ENDIANTYPE` `Endian`  
エンディアンタイプ `"big", "little"`
- `CDM::E_CDM_ARRAYSHAPE` `ArrayShape`  
配列形状
- `int` `NumVariables`  
変数の個数
- `vector< std::string >` `VariableName`  
変数名
- `std::string` `RankNoPrefix`  
ファイル名のランク番号前の文字列

### 6.15.1 説明

`index.dfi` ファイルの `FileInfo`

`cdm_FileInfo.h` の 20 行で定義されています。

### 6.15.2 コンストラクタとデストラクタ

#### 6.15.2.1 `cdm_FileInfo::cdm_FileInfo ( )`

コンストラクタ

6.15.2.2 `cdm_FileInfo::cdm_FileInfo ( const CDM::E_CDM_DFITYPE _DFIType, const CDM::E_CDM_OUTPUT_FNAME _FieldFilenameFormat, const std::string _DirectoryPath, const CDM::E_CDM_ONOFF _TimeSliceDirFlag, const std::string _Prefix, const CDM::E_CDM_FORMAT _FileFormat, const int _GuideCell, const CDM::E_CDM_DTYPE _DataType, const CDM::E_CDM_ENDIANTYPE _Endian, const CDM::E_CDM_ARRAYSHAPE _ArrayShape, const int _NumVariables, const std::string _RankNoPrefix = std::string(CDM::C_CDM_RANKNOPREFIX) )`

コンストラクタ

引数

<code>in</code>	<code>_DFIType</code>	dfi 種別
<code>in</code>	<code>_FieldFilenameFormat</code>	ファイル命名基準
<code>in</code>	<code>_DirectoryPath</code>	ディレクトリパス
<code>in</code>	<code>_TimeSliceDirFlag</code>	TimeSlice on or off
<code>in</code>	<code>_Prefix</code>	ファイル接頭文字
<code>in</code>	<code>_FileFormat</code>	ファイルフォーマット

in	<code>_GuideCell</code>	仮想セルの数
in	<code>_DataType</code>	配列のデータタイプ
in	<code>_Endian</code>	エンディアンタイプ
in	<code>_ArrayShape</code>	配列形状
in	<code>_NumVariables</code>	変数の個数

### 6.15.2.3 `cdm_FileInfo::~cdm_FileInfo ( )`

デストラクタ

## 6.15.3 関数

### 6.15.3.1 `std::string cdm_FileInfo::getVariableName ( int pvari )`

変数名を取得する

引数

in	<code>pvari</code>	変数位置 0:u, 1:v, 2:w
----	--------------------	--------------------

戻り値

変数名 変数名が無い場合は空白が返される

### 6.15.3.2 `CDM::E_CDM_ERRORCODE cdm_FileInfo::Read ( cdm_TextParser tpCntl )`

`read FileInfo(inde.dfi)`

引数

in	<code>tpCntl</code>	<code>cdm_TextParser</code> クラス
----	---------------------	---------------------------------

戻り値

error code

### 6.15.3.3 `void cdm_FileInfo::setVariableName ( int pvari, std::string variName )`

変数名をセットする

引数

in	<code>pvari</code>	変数位置 0:u, 1:v, 2:w
in	<code>variName</code>	変数名 "u","v","w",,,

### 6.15.3.4 `CDM::E_CDM_ERRORCODE cdm_FileInfo::Write ( FILE * fp, const unsigned tab )`

DFI ファイル:FileInfo 要素を出力する

引数

<code>in</code>	<code>fp</code>	ファイルポインタ
<code>in</code>	<code>tab</code>	インデント

戻り値

error code

#### 6.15.4 変数

##### 6.15.4.1 `CDM::E_CDM_ARRAYSHAPE` `cdm_FileInfo::ArrayShape`

配列形状

`cdm_FileInfo.h` の 37 行で定義されています。

参照元 `cdm_DFI::ReadCoordinateData()`, `cdm_DFI::ReadData()`, `cdm_DFI::WriteCoordinateData()`, `cdm_DFI::WriteData()`, `cdm_DFI::WriteFieldDataFile()`, と `cdm_DFI::WriteInit()`.

##### 6.15.4.2 `CDM::E_CDM_DTYPE` `cdm_FileInfo::DataType`

配列のデータタイプ "float", ...

`cdm_FileInfo.h` の 35 行で定義されています。

参照元 `cdm_DFI::ReadCoordinateData()`, `cdm_DFI::ReadData()`, と `cdm_DFI::WriteInit()`.

##### 6.15.4.3 `CDM::E_CDM_DFITYPE` `cdm_FileInfo::DFIType`

dfl 種別

`cdm_FileInfo.h` の 25 行で定義されています。

参照元 `cdm_DFI::WriteInit()`.

##### 6.15.4.4 `std::string` `cdm_FileInfo::DirectoryPath`

フィールドデータの存在するディレクトリパス `index.dfl` からの相対パスまたは絶対パス

`cdm_FileInfo.h` の 29 行で定義されています。

参照元 `cdm_DFI::WriteInit()`.

##### 6.15.4.5 `CDM::E_CDM_ENDIANTYPE` `cdm_FileInfo::Endian`

エンディアンタイプ "big", "little"

`cdm_FileInfo.h` の 36 行で定義されています。

参照元 `cdm_DFI::WriteInit()`.

##### 6.15.4.6 `CDM::E_CDM_OUTPUT_FNAME` `cdm_FileInfo::FieldFilenameFormat`

ファイル命名基準

`cdm_FileInfo.h` の 26 行で定義されています。

#### 6.15.4.7 CDM::E\_CDM\_FORMAT cdm\_FileInfo::FileFormat

ファイルフォーマット "bov", "sph",,,

cdm\_FileInfo.h の 33 行で定義されています。

参照元 cdm\_DFI::AddTimeSlice(), cdm\_DFI::set\_output\_fname(), cdm\_DFI::WriteData(), と cdm\_DFI::WriteInit().

#### 6.15.4.8 int cdm\_FileInfo::GuideCell

仮想セルの数

cdm\_FileInfo.h の 34 行で定義されています。

参照元 cdm\_DFI\_PLOT3D::write\_XYZ(), と cdm\_DFI::WriteInit().

#### 6.15.4.9 int cdm\_FileInfo::NumVariables

変数の個数

cdm\_FileInfo.h の 38 行で定義されています。

参照元 cdm\_DFI::AddTimeSlice(), cdm\_DFI::ReadCoordinateData(), cdm\_DFI::ReadData(), cdm\_DFI::WriteCoordinateData(), cdm\_DFI::WriteData(), cdm\_DFI::WriteFieldDataFile(), と cdm\_DFI::WriteInit().

#### 6.15.4.10 std::string cdm\_FileInfo::Prefix

ファイル接頭文字

cdm\_FileInfo.h の 32 行で定義されています。

参照元 cdm\_DFI::WriteInit().

#### 6.15.4.11 std::string cdm\_FileInfo::RankNoPrefix

ファイル名のランク番号前の文字列

cdm\_FileInfo.h の 41 行で定義されています。

#### 6.15.4.12 CDM::E\_CDM\_ONOFF cdm\_FileInfo::TimeSliceDirFlag

TimeSlice on or off.

cdm\_FileInfo.h の 31 行で定義されています。

参照元 cdm\_DFI::WriteInit().

#### 6.15.4.13 vector<std::string> cdm\_FileInfo::VariableName

変数名

cdm\_FileInfo.h の 39 行で定義されています。

参照元 cdm\_DFI::WriteData(), と cdm\_DFI::WriteFieldDataFile().

このクラスの説明は次のファイルから生成されました:

- [cdm\\_FileInfo.h](#)



## 6.16 クラス cdm\_FilePath

```
#include <cdm_FilePath.h>
```

### Public メソッド

- [cdm\\_FilePath \(\)](#)
- [cdm\\_FilePath \(const std::string \\_ProcDFIFile\)](#)  
コンストラクタ
- [~cdm\\_FilePath \(\)](#)
- [CDM::E\\_CDM\\_ERRORCODE Read \(cdm\\_TextParser tpCntl\)](#)  
*read FilePath(inde.dfi)*
- [CDM::E\\_CDM\\_ERRORCODE Write \(FILE \\*fp, const unsigned tab\)](#)  
*DFI ファイル:Process を出力する*

### Public 変数

- `std::string` [ProcDFIFile](#)  
*proc.dfi ファイル名*

#### 6.16.1 説明

index.dfi ファイルの FilePath

cdm\_FilePath.h の 19 行で定義されています。

#### 6.16.2 コンストラクタとデストラクタ

##### 6.16.2.1 cdm\_FilePath::cdm\_FilePath ( )

コンストラクタ

##### 6.16.2.2 cdm\_FilePath::cdm\_FilePath ( const std::string \_ProcDFIFile )

コンストラクタ

引数

in	<code>_ProcDFIFile</code>	proc.dfi ファイル名
----	---------------------------	----------------

##### 6.16.2.3 cdm\_FilePath::~~cdm\_FilePath ( )

デストラクタ

#### 6.16.3 関数

##### 6.16.3.1 CDM::E\_CDM\_ERRORCODE cdm\_FilePath::Read ( cdm\_TextParser tpCntl )

*read FilePath(inde.dfi)*

proc.dfi ファイル名の読み込み

引数

in	<i>tpCntl</i>	cdm_TextParser クラス
----	---------------	--------------------

戻り値

error code

#### 6.16.3.2 CDM::E\_CDM\_ERRORCODE cdm\_FilePath::Write ( FILE \* *fp*, const unsigned *tab* )

DFI ファイル:Process を出力する

proc.dfi ファイル名の出力

引数

in	<i>fp</i>	ファイルポインタ
in	<i>tab</i>	インデント

戻り値

error code

### 6.16.4 変数

#### 6.16.4.1 std::string cdm\_FilePath::ProcDFIFile

proc.dfi ファイル名

cdm\_FilePath.h の 23 行で定義されています。

参照元 cdm\_DFI::WriteInit().

このクラスの説明は次のファイルから生成されました:

- [cdm\\_FilePath.h](#)

## 6.17 クラス cdm\_MPI

```
#include <cdm_MPI.h>
```

### Public メソッド

- [cdm\\_MPI \(\)](#)
- [cdm\\_MPI \(const int \\_NumberOfRank, int \\_NumberOfGroup=0\)](#)  
コンストラクタ
- [~cdm\\_MPI \(\)](#)
- [CDM::E\\_CDM\\_ERRORCODE Read \(cdm\\_TextParser tpCntl, const cdm\\_Domain \\*domain\)](#)  
*read MPI(proc.dfi)*
- [CDM::E\\_CDM\\_ERRORCODE Write \(FILE \\*fp, const unsigned tab\)](#)  
*DFI ファイル:MPI を出力する*

## Public 変数

- int [NumberOfRank](#)  
プロセス数
- int [NumberOfGroup](#)  
グループ数

## 6.17.1 説明

proc.dfi ファイルの MPI

cdm\_MPI.h の 19 行で定義されています。

## 6.17.2 コンストラクタとデストラクタ

## 6.17.2.1 cdm\_MPI::cdm\_MPI ( )

## コンストラクタ

## 6.17.2.2 cdm\_MPI::cdm\_MPI ( const int \_NumberOfRank, int \_NumberOfGroup = 0 )

## コンストラクタ

## 引数

in	<i>_NumberOfRank</i>	プロセス数
in	<i>_NumberOfGroup</i>	グループ数

## 6.17.2.3 cdm\_MPI::~~cdm\_MPI ( )

## デストラクタ

## 6.17.3 関数

6.17.3.1 CDM::E\_CDM\_ERRORCODE cdm\_MPI::Read ( cdm\_TextParser *tpCntl*, const cdm\_Domain \* *domain* )

read MPI(proc.dfi)

## 引数

in	<i>tpCntl</i>	cdm_TextParser クラス
in	<i>domain</i>	Domain

## 戻り値

error code

6.17.3.2 CDM::E\_CDM\_ERRORCODE cdm\_MPI::Write ( FILE \* *fp*, const unsigned *tab* )

DFI ファイル:MPI を出力する

引数

in	<i>fp</i>	ファイルポインタ
in	<i>tab</i>	インデント

戻り値

error code

## 6.17.4 変数

### 6.17.4.1 int cdm\_MPI::NumberOfGroup

グループ数

cdm\_MPI.h の 24 行で定義されています。

参照元 cdm\_DFI::WriteInit().

### 6.17.4.2 int cdm\_MPI::NumberOfRank

プロセス数

cdm\_MPI.h の 23 行で定義されています。

参照元 cdm\_DFI::WriteInit().

このクラスの説明は次のファイルから生成されました:

- [cdm\\_MPI.h](#)

## 6.18 クラス テンプレート cdm\_NonUniformDomain< T >

```
#include <cdm_NonUniformDomain.h>
```

cdm\_NonUniformDomain< T > に対する継承グラフ

cdm\_NonUniformDomain< T > のコラボレーション図

### Public メソッド

- [cdm\\_NonUniformDomain \(\)](#)
- [cdm\\_NonUniformDomain \(const T \\* \\_XCoordinates, const T \\* \\_YCoordinates, const T \\* \\_ZCoordinates, const std::string \\_CoordinateFile, const CDM::E\\_CDM\\_FILE\\_TYPE \\_CoordinateFileType, const CDM::E\\_CDM\\_ENDIANTYPE \\_CoordinateFileEndian, const int \\* \\_GlobalVoxel, const int \\* \\_GlobalDivision, const int \\_gc=0\)](#)

コンストラクタ

- [cdm\\_NonUniformDomain \(const T \\* \\_GlobalOrigin, const T \\* \\_GlobalRegion, const int \\* \\_GlobalVoxel, const int \\* \\_GlobalDivision, const std::string \\_CoordinateFile, const CDM::E\\_CDM\\_FILE\\_TYPE \\_CoordinateFileType, const CDM::E\\_CDM\\_DTYPE \\_CoordinateFilePrecision, const CDM::E\\_CDM\\_ENDIANTYPE \\_CoordinateFileEndian\)](#)

コンストラクタ (*proc.dfi* 出力用)

- [~cdm\\_NonUniformDomain \(\)](#)
- double [CellX](#) (int i) const  
セル中心の X 座標を取得
- double [CellY](#) (int j) const  
セル中心の Y 座標を取得

- `double CellZ (int k) const`  
セル中心のZ 座標を取得
- `double NodeX (int i) const`  
格子点のX 座標を取得
- `double NodeY (int j) const`  
格子点のY 座標を取得
- `double NodeZ (int k) const`  
格子点のZ 座標を取得
- `std::string GetCoordinateFile () const`  
座標ファイル名を取得
- `CDM::E_CDM_FILE_TYPE GetCoordinateFileType () const`  
座標ファイルのファイルタイプを取得
- `CDM::E_CDM_DTYPE GetCoordinateFilePrecision () const`  
座標ファイルのデータ精度を取得
- `CDM::E_CDM_ENDIANTYPE GetCoordinateFileEndian () const`  
座標ファイルのエンディアンタイプを取得
- `CDM::E_CDM_ERRORCODE Read (cdm_TextParser tpCntl, std::string dirName)`  
*Domain(proc.dfi)* を読み込む
- `CDM::E_CDM_ERRORCODE Read_CoordinateFile (FILE *fp)`  
*CoordinateFile* を読み込む
- `CDM::E_CDM_ERRORCODE readCoordDataCount (FILE *fp, bool matchEndian, int globalVoxel, int *dataCount, size_t *nread)`  
*CoordinateFile* の各方向の座標データ数を読み込む
- `CDM::E_CDM_ERRORCODE readCoordData (FILE *fp, bool matchEndian, int *dataCount, double globalOrigin, double globalRegion, T *coordinates, size_t *nread)`  
*CoordinateFile* の各方向の座標データを読み込む
- `CDM::E_CDM_ERRORCODE Write (FILE *fp, const unsigned tab) const`  
*DFI* ファイル: *Domain* を出力する

## Protected メソッド

- `virtual void Clear ()`

## Private 変数

- `T * XCoordinates`  
X 座標データポインタ (*Domain* の格子点)
- `T * YCoordinates`  
Y 座標データポインタ (*Domain* の格子点)
- `T * ZCoordinates`  
Z 座標データポインタ (*Domain* の格子点)
- `std::string CoordinateFile`  
*CoordinateFile* ファイル名
- `CDM::E_CDM_FILE_TYPE CoordinateFileType`  
座標ファイルのファイルタイプ
- `CDM::E_CDM_DTYPE CoordinateFilePrecision`  
座標ファイルのデータ精度
- `CDM::E_CDM_ENDIANTYPE CoordinateFileEndian`  
座標ファイルのエンディアンタイプ
- `T pit_gcXsta`

- X方向のガイドセルの格子幅 (始点側)
- T [pit\\_gcXend](#)
  - X方向のガイドセルの格子幅 (終点側)
- T [pit\\_gcYsta](#)
  - Y方向のガイドセルの格子幅 (始点側)
- T [pit\\_gcYend](#)
  - Y方向のガイドセルの格子幅 (終点側)
- T [pit\\_gcZsta](#)
  - Z方向のガイドセルの格子幅 (始点側)
- T [pit\\_gcZend](#)
  - Z方向のガイドセルの格子幅 (終点側)

## Additional Inherited Members

### 6.18.1 説明

`template<class T>class cdm_NonUniformDomain< T >`

proc.dfi ファイルの Domain

cdm\_NonUniformDomain.h の 23 行で定義されています。

### 6.18.2 コンストラクタとデストラクタ

6.18.2.1 `template<class T> cdm_NonUniformDomain< T >::cdm_NonUniformDomain ( ) [inline]`

#### コンストラクタ

cdm\_NonUniformDomain.h の 51 行で定義されています。

参照先 `cdm_NonUniformDomain< T >::CoordinateFile`, `cdm_NonUniformDomain< T >::CoordinateFileEndian`, `cdm_NonUniformDomain< T >::CoordinateFilePrecision`, `cdm_NonUniformDomain< T >::CoordinateFileType`, `CDM::E_CDM_DTYPE_UNKNOWN`, `CDM::E_CDM_ENDIANTYPE_UNKNOWN`, と `CDM::E_CDM_FILE_TYPE_DEFAULT`.

6.18.2.2 `template<class T> cdm_NonUniformDomain< T >::cdm_NonUniformDomain ( const T * _XCoordinates, const T * _YCoordinates, const T * _ZCoordinates, const std::string _CoordinateFile, const CDM::E_CDM_FILE_TYPE _CoordinateFileType, const CDM::E_CDM_ENDIANTYPE _CoordinateFileEndian, const int * _GlobalVoxel, const int * _GlobalDivision, const int _gc = 0 ) [inline]`

#### コンストラクタ

##### 引数

in	<code>_XCoordinates</code>	X 座標データポインタ (Domain の格子点)
in	<code>_YCoordinates</code>	Y 座標データポインタ (Domain の格子点)
in	<code>_ZCoordinates</code>	Z 座標データポインタ (Domain の格子点)
in	<code>_CoordinateFile</code>	座標データ名
in	<code>_CoordinateFileType</code>	座標データのファイルタイプ

in	<code>_CoordinateFile-Endian</code>	座標データのエンディアンタイプ
in	<code>_GlobalVoxel</code>	ボクセル数
in	<code>_GlobalDivision</code>	分割数
in	<code>_gc</code>	ガイドセル数

`cdm_NonUniformDomain.h` の 71 行で定義されています。

参照先 `cdm_NonUniformDomain< T >::CoordinateFile`, `cdm_NonUniformDomain< T >::CoordinateFileEndian`, `cdm_NonUniformDomain< T >::CoordinateFilePrecision`, `cdm_NonUniformDomain< T >::CoordinateFileType`, `CDM::E_CDM_FLOAT32`, `CDM::E_CDM_FLOAT64`, `cdm_Domain::GlobalDivision`, `cdm_Domain::GlobalOrigin`, `cdm_Domain::GlobalRegion`, `cdm_Domain::GlobalVoxel`, `cdm_NonUniformDomain< T >::pit_gcXend`, `cdm_NonUniformDomain< T >::pit_gcXsta`, `cdm_NonUniformDomain< T >::pit_gcYend`, `cdm_NonUniformDomain< T >::pit_gcYsta`, `cdm_NonUniformDomain< T >::pit_gcZend`, `cdm_NonUniformDomain< T >::pit_gcZsta`, `cdm_NonUniformDomain< T >::XCoordinates`, `cdm_NonUniformDomain< T >::YCoordinates`, と `cdm_NonUniformDomain< T >::ZCoordinates`.

```
6.18.2.3 template<class T> cdm_NonUniformDomain< T >::cdm_NonUniformDomain ( const T *
    _GlobalOrigin, const T * _GlobalRegion, const int * _GlobalVoxel, const int * _GlobalDivision, const std::string
    _CoordinateFile, const CDM::E_CDM_FILE_TYPE _CoordinateFileType, const CDM::E_CDM_DTYPE
    _CoordinateFilePrecision, const CDM::E_CDM_ENDIANTYPE _CoordinateFileEndian ) [inline]
```

コンストラクタ (proc.dfi 出力用)

引数

in	<code>_GlobalOrigin</code>	起点座標
in	<code>_GlobalRegion</code>	各軸方向の長さ
in	<code>_GlobalVoxel</code>	ボクセル数
in	<code>_GlobalDivision</code>	分割数
in	<code>_CoordinateFile</code>	座標データ名
in	<code>_CoordinateFile-Type</code>	座標データのファイルタイプ
in	<code>_CoordinateFile-Precision</code>	座標データのデータ精度
in	<code>_CoordinateFile-Endian</code>	座標データのエンディアンタイプ

`cdm_NonUniformDomain.h` の 143 行で定義されています。

参照先 `cdm_NonUniformDomain< T >::CoordinateFile`, `cdm_NonUniformDomain< T >::CoordinateFileEndian`, `cdm_NonUniformDomain< T >::CoordinateFilePrecision`, `cdm_NonUniformDomain< T >::CoordinateFileType`, `cdm_Domain::GlobalDivision`, `cdm_Domain::GlobalOrigin`, `cdm_Domain::GlobalRegion`, と `cdm_Domain::GlobalVoxel`.

```
6.18.2.4 template<class T> cdm_NonUniformDomain< T >::~~cdm_NonUniformDomain ( ) [inline]
```

デストラクタ

`cdm_NonUniformDomain.h` の 176 行で定義されています。

参照先 `cdm_NonUniformDomain< T >::XCoordinates`, `cdm_NonUniformDomain< T >::YCoordinates`, と `cdm_NonUniformDomain< T >::ZCoordinates`.

## 6.18.3 関数

```
6.18.3.1 template<class T> double cdm_NonUniformDomain< T >::CellX ( int i ) const [inline],[virtual]
```

セル中心のX 座標を取得

引数

<i>in</i>	<i>i</i>	X 方向のセル番号
-----------	----------	-----------

戻り値

セル中心のX 座標

[cdm\\_Domain](#)を再定義しています。

cdm\_NonUniformDomain.h の 188 行で定義されています。

参照先 cdm\_Domain::GlobalVoxel, cdm\_NonUniformDomain< T >::pit\_gcXend, cdm\_NonUniformDomain< T >::pit\_gcXsta, と cdm\_NonUniformDomain< T >::XCoordinates.

**6.18.3.2** `template<class T> double cdm_NonUniformDomain< T >::CellY ( int j ) const` `[inline], [virtual]`

セル中心のY 座標を取得

引数

<i>in</i>	<i>j</i>	Y 方向のセル番号
-----------	----------	-----------

戻り値

セル中心のX 座標

[cdm\\_Domain](#)を再定義しています。

cdm\_NonUniformDomain.h の 203 行で定義されています。

参照先 cdm\_Domain::GlobalVoxel, cdm\_NonUniformDomain< T >::pit\_gcYend, cdm\_NonUniformDomain< T >::pit\_gcYsta, と cdm\_NonUniformDomain< T >::YCoordinates.

**6.18.3.3** `template<class T> double cdm_NonUniformDomain< T >::CellZ ( int k ) const` `[inline], [virtual]`

セル中心のZ 座標を取得

引数

<i>in</i>	<i>k</i>	Z 方向のセル番号
-----------	----------	-----------

戻り値

セル中心のX 座標

[cdm\\_Domain](#)を再定義しています。

cdm\_NonUniformDomain.h の 218 行で定義されています。

参照先 cdm\_Domain::GlobalVoxel, cdm\_NonUniformDomain< T >::pit\_gcZend, cdm\_NonUniformDomain< T >::pit\_gcZsta, と cdm\_NonUniformDomain< T >::ZCoordinates.

**6.18.3.4** `template<class T> virtual void cdm_NonUniformDomain< T >::Clear ( )` `[inline], [protected], [virtual]`

[cdm\\_Domain](#)を再定義しています。

cdm\_NonUniformDomain.h の 41 行で定義されています。

参照先 cdm\_Domain::Clear(), cdm\_NonUniformDomain< T >::XCoordinates, cdm\_NonUniformDomain< T >::YCoordinates, と cdm\_NonUniformDomain< T >::ZCoordinates.



```
6.18.3.5  template<class T> std::string cdm_NonUniformDomain< T >::GetCoordinateFile ( ) const  [inline],  
          [virtual]
```

座標ファイル名を取得

戻り値

座標ファイル名

`cdm_Domain`を再定義しています。

`cdm_NonUniformDomain.h` の 259 行で定義されています。

参照先 `cdm_NonUniformDomain< T >::CoordinateFile`.

```
6.18.3.6  template<class T> CDM::E_CDM_ENDIANTYPE cdm_NonUniformDomain< T  
          >::GetCoordinateFileEndian ( ) const  [inline],[virtual]
```

座標ファイルのエンディアンタイプを取得

戻り値

座標ファイルのエンディアンタイプ

`cdm_Domain`を再定義しています。

`cdm_NonUniformDomain.h` の 283 行で定義されています。

参照先 `cdm_NonUniformDomain< T >::CoordinateFileEndian`.

```
6.18.3.7  template<class T> CDM::E_CDM_DTYPE cdm_NonUniformDomain< T >::GetCoordinateFilePrecision ( )  
          const  [inline],[virtual]
```

座標ファイルのデータ精度を取得

戻り値

座標ファイルのデータ精度

`cdm_Domain`を再定義しています。

`cdm_NonUniformDomain.h` の 275 行で定義されています。

参照先 `cdm_NonUniformDomain< T >::CoordinateFilePrecision`.

```
6.18.3.8  template<class T> CDM::E_CDM_FILE_TYPE cdm_NonUniformDomain< T >::GetCoordinateFileType ( )  
          const  [inline],[virtual]
```

座標ファイルのファイルタイプを取得

戻り値

座標ファイルのファイルタイプ

`cdm_Domain`を再定義しています。

`cdm_NonUniformDomain.h` の 267 行で定義されています。

参照先 `cdm_NonUniformDomain< T >::CoordinateFileType`.

```
6.18.3.9  template<class T> double cdm_NonUniformDomain< T >::NodeX ( int i ) const  [inline],  
          [virtual]
```

格子点のX 座標を取得

引数

<code>in</code>	<code>i</code>	X 方向の格子番号
-----------------	----------------	-----------

戻り値

格子点のX 座標

`cdm_Domain`を再定義しています。

`cdm_NonUniformDomain.h` の 233 行で定義されています。

参照先 `cdm_NonUniformDomain< T >::XCoordinates`.

```
6.18.3.10  template<class T> double cdm_NonUniformDomain< T >::NodeY ( int j ) const  [inline],
           [virtual]
```

格子点のY 座標を取得

引数

<code>in</code>	<code>j</code>	Y 方向の格子番号
-----------------	----------------	-----------

戻り値

格子点のY 座標

`cdm_Domain`を再定義しています。

`cdm_NonUniformDomain.h` の 242 行で定義されています。

参照先 `cdm_NonUniformDomain< T >::YCoordinates`.

```
6.18.3.11  template<class T> double cdm_NonUniformDomain< T >::NodeZ ( int k ) const  [inline],
           [virtual]
```

格子点のZ 座標を取得

引数

<code>in</code>	<code>k</code>	Z 方向の格子番号
-----------------	----------------	-----------

戻り値

格子点のZ 座標

`cdm_Domain`を再定義しています。

`cdm_NonUniformDomain.h` の 251 行で定義されています。

参照先 `cdm_NonUniformDomain< T >::ZCoordinates`.

```
6.18.3.12  template<typename T> CDM::E_CDM_ERRORCODE cdm_NonUniformDomain< T >::Read (
           cdm_TextParser tpCntl, std::string dirName )  [virtual]
```

Domain(proc.dfi) を読み込む

## 引数

in	<i>tpCntl</i>	cdm_TextParser クラス
in	<i>dirName</i>	DFI のディレクトリパス

## 戻り値

error code

[cdm\\_Domain](#)を再定義しています。

cdm\_NonUniformDomain\_inline.h の 32 行で定義されています。

参照先 CDM::cdmPath\_ConnectPath(), CDM::cdmPath\_isAbsolute(), CDM::E\_CDM\_BIG, CDM::E\_CDM\_ERROR\_OPEN\_COORDINATEFILE, CDM::E\_CDM\_ERROR\_READ\_DFI\_COORDINATEFILE, CDM::E\_CDM\_ERROR\_READ\_DFI\_COORDINATEFILEENDIAN, CDM::E\_CDM\_ERROR\_READ\_DFI\_COORDINATEFILEPRECISION, CDM::E\_CDM\_ERROR\_READ\_DFI\_COORDINATEFILETYPE, CDM::E\_CDM\_ERROR\_READ\_DFI\_GLOBALDIVISION, CDM::E\_CDM\_ERROR\_READ\_DFI\_GLOBALORIGIN, CDM::E\_CDM\_ERROR\_READ\_DFI\_GLOBALREGION, CDM::E\_CDM\_ERROR\_READ\_DFI\_GLOBALVOXEL, CDM::E\_CDM\_FILE\_TYPE\_ASCII, CDM::E\_CDM\_FILE\_TYPE\_BINARY, CDM::E\_CDM\_FLOAT32, CDM::E\_CDM\_FLOAT64, CDM::E\_CDM\_LITTLE, CDM::E\_CDM\_SUCCESS, cdm\_TextParser::GetValue(), と cdm\_TextParser::GetVector().

**6.18.3.13** `template<typename T> CDM::E_CDM_ERRORCODE cdm_NonUniformDomain< T  
>::Read_CoordinateFile ( FILE * fp )`

CoordinateFile を読む

## 引数

in	<i>fp</i>	ファイルポインタ
----	-----------	----------

## 戻り値

error code

## Endian セット

cdm\_NonUniformDomain\_inline.h の 199 行で定義されています。

参照先 CDM::E\_CDM\_BIG, CDM::E\_CDM\_ENDIANTYPE\_UNKNOWN, CDM::E\_CDM\_ERROR\_READ\_COORDINATEFILE, CDM::E\_CDM\_FILE\_TYPE\_ASCII, CDM::E\_CDM\_LITTLE, と CDM::E\_CDM\_SUCCESS.

**6.18.3.14** `template<typename T> CDM_INLINE CDM::E_CDM_ERRORCODE cdm_NonUniformDomain< T  
>::readCoordData ( FILE * fp, bool matchEndian, int * dataCount, double globalOrigin, double globalRegion, T *  
coordinates, size_t * nread )`

CoordinateFile の各方向の座標データを読む

## 引数

in	<i>fp</i>	ファイルポインタ
in	<i>matchEndian</i>	true:Endian 一致
in	<i>dataCount</i>	座標データ数
in	<i>globalOrigin</i>	計算空間の起点座標
in	<i>globalRegion</i>	計算空間の各軸方向の長さ

out	<i>coordinates</i>	読み込んだ座標データのポインタ
out	<i>nread</i>	読み込みデータ数

戻り値

error code

`cdm_NonUniformDomain_inline.h` の 383 行で定義されています。

参照先 BSWAPVEC, DBSWAPVEC, `CDM::E_CDM_ERROR_READ_COORDINATEFILE`, と `CDM::E_CDM_SUCCESS`.

**6.18.3.15** `template<typename T> CDM_INLINE CDM::E_CDM_ERRORCODE cdm_NonUniformDomain< T >::readCoordDataCount ( FILE * fp, bool matchEndian, int globalVoxel, int * dataCount, size_t * nread )`

CoordinateFile の各方向の座標データ数を読み込む

引数

in	<i>fp</i>	ファイルポインタ
in	<i>matchEndian</i>	true:Endian 一致
in	<i>globalVoxel</i>	計算領域のボクセル数
out	<i>dataCount</i>	座標データ数
out	<i>nread</i>	読み込みデータ数

戻り値

error code

`cdm_NonUniformDomain_inline.h` の 358 行で定義されています。

参照先 BSWAPVEC, `CDM::E_CDM_ERROR_READ_COORDINATEFILE`, と `CDM::E_CDM_SUCCESS`.

**6.18.3.16** `template<typename T> CDM::E_CDM_ERRORCODE cdm_NonUniformDomain< T >::Write ( FILE * fp, const unsigned tab ) const [virtual]`

DFI ファイル:Domain を出力する

引数

in	<i>fp</i>	ファイルポインタ
in	<i>tab</i>	インデント

戻り値

error code

`cdm_Domain`を再定義しています。

`cdm_NonUniformDomain_inline.h` の 422 行で定義されています。

参照先 `_CDM_WRITE_TAB`, `cdm_DFI::ConvDatatypeE2S()`, `CDM::E_CDM_BIG`, `CDM::E_CDM_FILE_TYPE_ASCII`, `CDM::E_CDM_FILE_TYPE_BINARY`, `CDM::E_CDM_LITTLE`, と `CDM::E_CDM_SUCCESS`.

## 6.18.4 変数

**6.18.4.1** `template<class T> std::string cdm_NonUniformDomain< T >::CoordinateFile [private]`

CoordinateFile ファイル名

cdm\_NonUniformDomain.h の 29 行で定義されています。

参照元 cdm\_NonUniformDomain< T >::cdm\_NonUniformDomain(), と cdm\_NonUniformDomain< T >::GetCoordinateFile().

6.18.4.2 `template<class T> CDM::E_CDM_ENDIANTYPE cdm_NonUniformDomain< T >::CoordinateFileEndian [private]`

座標ファイルのエンディアンタイプ

cdm\_NonUniformDomain.h の 32 行で定義されています。

参照元 cdm\_NonUniformDomain< T >::cdm\_NonUniformDomain(), と cdm\_NonUniformDomain< T >::GetCoordinateFileEndian().

6.18.4.3 `template<class T> CDM::E_CDM_DTYPE cdm_NonUniformDomain< T >::CoordinateFilePrecision [private]`

座標ファイルのデータ精度

cdm\_NonUniformDomain.h の 31 行で定義されています。

参照元 cdm\_NonUniformDomain< T >::cdm\_NonUniformDomain(), と cdm\_NonUniformDomain< T >::GetCoordinateFilePrecision().

6.18.4.4 `template<class T> CDM::E_CDM_FILE_TYPE cdm_NonUniformDomain< T >::CoordinateFileType [private]`

座標ファイルのファイルタイプ

cdm\_NonUniformDomain.h の 30 行で定義されています。

参照元 cdm\_NonUniformDomain< T >::cdm\_NonUniformDomain(), と cdm\_NonUniformDomain< T >::GetCoordinateFileType().

6.18.4.5 `template<class T> T cdm_NonUniformDomain< T >::pit_gcXend [private]`

X 方向のガイドセルの格子幅 (終点側)

cdm\_NonUniformDomain.h の 34 行で定義されています。

参照元 cdm\_NonUniformDomain< T >::cdm\_NonUniformDomain(), と cdm\_NonUniformDomain< T >::CellX().

6.18.4.6 `template<class T> T cdm_NonUniformDomain< T >::pit_gcXsta [private]`

X 方向のガイドセルの格子幅 (始点側)

cdm\_NonUniformDomain.h の 33 行で定義されています。

参照元 cdm\_NonUniformDomain< T >::cdm\_NonUniformDomain(), と cdm\_NonUniformDomain< T >::CellX().

6.18.4.7 `template<class T> T cdm_NonUniformDomain< T >::pit_gcYend [private]`

Y 方向のガイドセルの格子幅 (終点側)

cdm\_NonUniformDomain.h の 36 行で定義されています。

参照元 cdm\_NonUniformDomain< T >::cdm\_NonUniformDomain(), と cdm\_NonUniformDomain< T >::CellY().

6.18.4.8 `template<class T> T cdm_NonUniformDomain< T >::pit_gcYsta [private]`

Y 方向のガイドセルの格子幅 (始点側)

`cdm_NonUniformDomain.h` の 35 行で定義されています。

参照元 `cdm_NonUniformDomain< T >::cdm_NonUniformDomain()`, と `cdm_NonUniformDomain< T >::CellY()`.

6.18.4.9 `template<class T> T cdm_NonUniformDomain< T >::pit_gcZend [private]`

Z 方向のガイドセルの格子幅 (終点側)

`cdm_NonUniformDomain.h` の 38 行で定義されています。

参照元 `cdm_NonUniformDomain< T >::cdm_NonUniformDomain()`, と `cdm_NonUniformDomain< T >::CellZ()`.

6.18.4.10 `template<class T> T cdm_NonUniformDomain< T >::pit_gcZsta [private]`

Z 方向のガイドセルの格子幅 (始点側)

`cdm_NonUniformDomain.h` の 37 行で定義されています。

参照元 `cdm_NonUniformDomain< T >::cdm_NonUniformDomain()`, と `cdm_NonUniformDomain< T >::CellZ()`.

6.18.4.11 `template<class T> T* cdm_NonUniformDomain< T >::XCoordinates [private]`

X 座標データポインタ (Domain の格子点)

`cdm_NonUniformDomain.h` の 26 行で定義されています。

参照元 `cdm_NonUniformDomain< T >::cdm_NonUniformDomain()`, `cdm_NonUniformDomain< T >::CellX()`, `cdm_NonUniformDomain< T >::Clear()`, `cdm_NonUniformDomain< T >::NodeX()`, と `cdm_NonUniformDomain< T >::~~cdm_NonUniformDomain()`.

6.18.4.12 `template<class T> T* cdm_NonUniformDomain< T >::YCoordinates [private]`

Y 座標データポインタ (Domain の格子点)

`cdm_NonUniformDomain.h` の 27 行で定義されています。

参照元 `cdm_NonUniformDomain< T >::cdm_NonUniformDomain()`, `cdm_NonUniformDomain< T >::CellY()`, `cdm_NonUniformDomain< T >::Clear()`, `cdm_NonUniformDomain< T >::NodeY()`, と `cdm_NonUniformDomain< T >::~~cdm_NonUniformDomain()`.

6.18.4.13 `template<class T> T* cdm_NonUniformDomain< T >::ZCoordinates [private]`

Z 座標データポインタ (Domain の格子点)

`cdm_NonUniformDomain.h` の 28 行で定義されています。

参照元 `cdm_NonUniformDomain< T >::cdm_NonUniformDomain()`, `cdm_NonUniformDomain< T >::CellZ()`, `cdm_NonUniformDomain< T >::Clear()`, `cdm_NonUniformDomain< T >::NodeZ()`, と `cdm_NonUniformDomain< T >::~~cdm_NonUniformDomain()`.

このクラスの説明は次のファイルから生成されました:

- [cdm\\_NonUniformDomain.h](#)
- [cdm\\_NonUniformDomain\\_inline.h](#)

## 6.19 クラス cdm\_Process

```
#include <cdm_Process.h>
```

### Public 型

- typedef std::map< int, int > [headT](#)

### Public メソッド

- [cdm\\_Process](#) ()
- [~cdm\\_Process](#) ()
- [CDM::E\\_CDM\\_ERRORCODE Read](#) ([cdm\\_TextParser](#) tpCntl)  
*read Rank(proc.dfi)*
- [CDM::E\\_CDM\\_ERRORCODE CheckReadRank](#) (const [cdm\\_Domain](#) \*dfi\_domain, const int head[3], const int tail[3], [CDM::E\\_CDM\\_READTYPE](#) readflag, vector< int > &readRankList)  
*読み込みランクリストの作成*
- [CDM::E\\_CDM\\_ERRORCODE CreateRankList](#) (const [cdm\\_Domain](#) \*dfi\_domain, map< int, int > &mapHeadX, map< int, int > &mapHeadY, map< int, int > &mapHeadZ)  
*DFI のProcess にHeadIndex, TailIndex 指定が無い場合*
- [CDM::E\\_CDM\\_ERRORCODE CreateRankList](#) (const int div[3], const int gvox[3], map< int, int > &mapHeadX, map< int, int > &mapHeadY, map< int, int > &mapHeadZ)  
*DFI のProcess にHeadIndex, TailIndex 指定が無い場合 渡された、subDomain をもとにCPM 同様の分割方法で RankList を生成する*
- [CDM::E\\_CDM\\_ERRORCODE CreateSubDomainInfo](#) (const [cdm\\_Domain](#) \*dfi\_domain, vector< [cdm\\_ActiveSubDomain](#) > &subDomainInfo)  
*ActiveSubDomain 情報を作成*
- int \* [CreateRankMap](#) (const int div[3], std::vector< [cdm\\_ActiveSubDomain](#) > &subDomainInfo)  
*subdomain 情報からランクマップを生成 (非活性を含む)*
- int \* [CreateRankMap](#) (const int ndiv[3], [headT](#) &mapHeadX, [headT](#) &mapHeadY, [headT](#) &mapHeadZ)  
*生成済のRankList からランクマップを生成*
- void [CreateHeadMap](#) (std::set< int > head, [headT](#) &map)  
*head map の生成*
- void [CreateHeadMap](#) (int \*head, int ndiv, [headT](#) &map)  
*head map の生成*
- [CDM::E\\_CDM\\_ERRORCODE CheckStartEnd](#) (const [cdm\\_Domain](#) \*dfi\_domain, const int head[3], const int tail[3], [CDM::E\\_CDM\\_READTYPE](#) readflag, [headT](#) mapHeadX, [headT](#) mapHeadY, [headT](#) mapHeadZ, vector< int > &readRankList)  
*読み込みランクファイルリストの作成*
- [CDM::E\\_CDM\\_ERRORCODE Write](#) (FILE \*fp, const unsigned tab)  
*DFI ファイル:Process を出力する*

### Static Public メソッド

- static int [isMatchEndianSbdmMagick](#) (int ident)  
*ActiveSubdomain ファイルのエンディアンをチェック*
- static [CDM::E\\_CDM\\_ERRORCODE ReadActiveSubdomainFile](#) (std::string subDomainFile, std::vector< [cdm\\_ActiveSubDomain](#) > &subDomainInfo, int div[3])  
*ActiveSubdomain ファイルの読み込み (static 関数)*



## Public 変数

- vector< cdm\_Rank > RankList
- int \* m\_rankMap

## 6.19.1 説明

proc.dfi ファイルのProcess

cdm\_Process.h の 61 行で定義されています。

## 6.19.2 型定義

6.19.2.1 typedef std::map<int,int> cdm\_Process::headT

cdm\_Process.h の 65 行で定義されています。

## 6.19.3 コンストラクタとデストラクタ

6.19.3.1 cdm\_Process::cdm\_Process ( )

コンストラクタ

6.19.3.2 cdm\_Process::~~cdm\_Process ( )

デストラクタ

## 6.19.4 関数

6.19.4.1 CDM::E\_CDM\_ERRORCODE cdm\_Process::CheckReadRank ( const cdm\_Domain \* dfi\_domain, const int head[3], const int tail[3], CDM::E\_CDM\_READTYPE readflag, vector< int > & readRankList )

読み込みランクリストの作成

RankList があるかないか判定しないときは新規にRankList を生成し それをもとにランクマップの生成、読み込みランクリスト readRankList を生成する

引数

in	dfi_domain	DFI の domain 情報
in	head	ソルバーのHeadIndex
in	tail	ソルバーのTailIndex
in	readflag	読み込み方法
out	readRankList	読み込みランクリスト

戻り値

error code

6.19.4.2 CDM::E\_CDM\_ERRORCODE cdm\_Process::CheckStartEnd ( const cdm\_Domain \* dfi\_domain, const int head[3], const int tail[3], CDM::E\_CDM\_READTYPE readflag, headT mapHeadX, headT mapHeadY, headT mapHeadZ, vector< int > & readRankList )

読み込みランクファイルリストの作成

## 引数

in	<i>dfi_domain</i>	DFI のDomain 情報
in	<i>head</i>	計算領域の開始位置
in	<i>tail</i>	計算領域の終了位置
in	<i>readflag</i>	粗密データ判定フラグ
in	<i>mapHeadX</i>	headX をキーにした位置情報マップ
in	<i>mapHeadY</i>	headY をキーにした位置情報マップ
in	<i>mapHeadZ</i>	headZ をキーにした位置情報マップ
out	<i>readRankList</i>	読み込みに必要なランク番号リスト

## 6.19.4.3 void cdm\_Process::CreateHeadMap ( std::set&lt; int &gt; head, headT &amp; map )

## head map の生成

## 引数

in	<i>head</i>	head インデックス
out	<i>map</i>	head map

## 6.19.4.4 void cdm\_Process::CreateHeadMap ( int \* head, int ndiv, headT &amp; map )

## head map の生成

## 引数

in	<i>head</i>	head インデックス
in	<i>ndiv</i>	分割数
out	<i>map</i>	head map

## 6.19.4.5 CDM::E\_CDM\_ERRORCODE cdm\_Process::CreateRankList ( const cdm\_Domain \* dfi\_domain, map&lt; int, int &gt; &amp; mapHeadX, map&lt; int, int &gt; &amp; mapHeadY, map&lt; int, int &gt; &amp; mapHeadZ )

## DFI のProcess にHeadIndex,TailIndex 指定が無い場合

ActiveSubDomain があれば、読み込み、なければ全て有効で subDomain を生成し、CreateRankList に渡す CPM と同じ分割で head&tail 情報を作成してRankList を作成する

## 引数

in	<i>dfi_domain</i>	DFI の domain 情報
out	<i>mapHeadX</i>	headX をキーにした位置情報マップ
out	<i>mapHeadY</i>	headX をキーにした位置情報マップ
out	<i>mapHeadZ</i>	headX をキーにした位置情報マップ

## 戻り値

error code

## 6.19.4.6 CDM::E\_CDM\_ERRORCODE cdm\_Process::CreateRankList ( const int div[3], const int gvox[3], map&lt; int, int &gt; &amp; mapHeadX, map&lt; int, int &gt; &amp; mapHeadY, map&lt; int, int &gt; &amp; mapHeadZ )

DFI のProcess にHeadIndex,TailIndex 指定が無い場合 渡された、subDomain をもとにCPM 同様の分割方法で RankList を生成する

引数

in	<i>div</i>	分割数
in	<i>gvox</i>	ボクセルサイズ
out	<i>mapHeadX</i>	headX をキーにした位置情報マップ
out	<i>mapHeadY</i>	headX をキーにした位置情報マップ
out	<i>mapHeadZ</i>	headX をキーにした位置情報マップ

戻り値

error code

6.19.4.7 int\* cdm\_Process::CreateRankMap ( const int *div*[3], std::vector< cdm\_ActiveSubDomain > & *subDomainInfo* )

subdomain 情報からランクマップを生成（非活性を含む）

引数

in	<i>div</i>	領域分割数
in	<i>subDomainInfo</i>	活性ドメイン情報

戻り値

ランクマップ  
NULL

6.19.4.8 int\* cdm\_Process::CreateRankMap ( const int *ndiv*[3], headT & *mapHeadX*, headT & *mapHeadY*, headT & *mapHeadZ* )

生成済のRankList からランクマップを生成

引数

in	<i>ndiv</i>	領域分割数
in	<i>mapHeadX</i>	headX をキーにした位置情報マップ
in	<i>mapHeadY</i>	headY をキーにした位置情報マップ
in	<i>mapHeadZ</i>	headZ をキーにした位置情報マップ

戻り値

ランクマップ  
NULL

6.19.4.9 CDM::E\_CDM\_ERRORCODE cdm\_Process::CreateSubDomainInfo ( const cdm\_Domain \* *dfi\_domain*, vector< cdm\_ActiveSubDomain > & *subDomainInfo* )

ActiveSubDomain 情報を作成

引数

in	<i>dfi_domain</i>	DFI の domain 情報
----	-------------------	-----------------

out	<i>subDomainInfo</i>	活性ドメイン情報
-----	----------------------	----------

6.19.4.10 `static int cdm_Process::isMatchEndianSbdmMagick ( int ident ) [static]`

ActiveSubdomain ファイルのエンディアンをチェック

引数

in	<i>ident</i>	ActiveSubdomain ファイルのIdentifier
----	--------------	---------------------------------

戻り値

- 1 一致
- 0 不一致
- 1 フォーマットが異なる

6.19.4.11 `CDM::E_CDM_ERRORCODE cdm_Process::Read ( cdm_TextParser tpCntl )`

read Rank(proc.dfi)

引数

in	<i>tpCntl</i>	cdm_TextParser クラス
----	---------------	--------------------

戻り値

error code

6.19.4.12 `static CDM::E_CDM_ERRORCODE cdm_Process::ReadActiveSubdomainFile ( std::string subDomainFile,  
std::vector< cdm_ActiveSubDomain > & subDomainInfo, int div[3] ) [static]`

ActiveSubdomain ファイルの読み込み (static 関数)

引数

in	<i>subDomainFile</i>	ActiveSubdomain ファイル名
out	<i>subDomainInfo</i>	活性ドメイン情報
out	<i>div</i>	ActiveSubdiomain ファイル中の領域分割数

戻り値

終了コード (CDM\_SUCCESS=正常終了)

6.19.4.13 `CDM::E_CDM_ERRORCODE cdm_Process::Write ( FILE * fp, const unsigned tab )`

DFI ファイル:Process を出力する

引数

in	<i>fp</i>	ファイルポインタ
----	-----------	----------

in	tab	インデント
----	-----	-------

戻り値

true:出力成功 false:出力失敗

### 6.19.5 変数

#### 6.19.5.1 int\* cdm\_Process::m\_rankMap

cdm\_Process.h の 69 行で定義されています。

#### 6.19.5.2 vector<cdm\_Rank> cdm\_Process::RankList

cdm\_Process.h の 67 行で定義されています。

参照元 cdm\_DFI::WriteCoordinateData(), cdm\_DFI::WriteData(), cdm\_DFI::WriteFieldDataFile(), と cdm\_DFI::WriteInit().

このクラスの説明は次のファイルから生成されました:

- [cdm\\_Process.h](#)

## 6.20 クラス cdm\_Rank

```
#include <cdm_Process.h>
```

### Public メソッド

- [cdm\\_Rank\(\)](#)
- [~cdm\\_Rank\(\)](#)
- [CDM::E\\_CDM\\_ERRORCODE Read](#) ([cdm\\_TextParser](#) tpCntl, std::string label\_leaf)  
*read Rank(proc.dfi)*
- [CDM::E\\_CDM\\_ERRORCODE Write](#) (FILE \*fp, const unsigned tab)  
*DFI ファイル:Rank 出力する*

### Public 変数

- int [RankID](#)  
ランク番号
- std::string [HostName](#)  
ホスト名
- int [VoxelSize](#) [3]  
ボクセルサイズ
- int [HeadIndex](#) [3]  
始点インデックス
- int [TailIndex](#) [3]  
終点インデックス
- int [c\\_id](#)  
*cell id*
- int [bc\\_id](#)  
境界ID

### 6.20.1 説明

proc.dfi ファイルの Rank

cdm\_Process.h の 19 行で定義されています。

### 6.20.2 コンストラクタとデストラクタ

#### 6.20.2.1 cdm\_Rank::cdm\_Rank ( )

コンストラクタ

#### 6.20.2.2 cdm\_Rank::~~cdm\_Rank ( )

デストラクタ

### 6.20.3 関数

#### 6.20.3.1 CDM::E\_CDM\_ERRORCODE cdm\_Rank::Read ( cdm\_TextParser *tpCntl*, std::string *label\_leaf* )

read Rank(proc.dfi)

引数

in	<i>tpCntl</i>	cdm_TextParser クラス
in	<i>label_leaf</i>	ベースとなる名前 ( "/Process/Rank" )

戻り値

error code

#### 6.20.3.2 CDM::E\_CDM\_ERRORCODE cdm\_Rank::Write ( FILE \* *fp*, const unsigned *tab* )

DFI ファイル:Rank 出力する

引数

in	<i>fp</i>	ファイルポインタ
in	<i>tab</i>	インデント

戻り値

true:出力成功 false:出力失敗

### 6.20.4 変数

#### 6.20.4.1 int cdm\_Rank::bc\_id

境界ID

cdm\_Process.h の 30 行で定義されています。

## 6.20.4.2 int cdm\_Rank::c\_id

cell id

cdm\_Process.h の 29 行で定義されています。

## 6.20.4.3 int cdm\_Rank::HeadIndex[3]

始点インデックス

cdm\_Process.h の 27 行で定義されています。

## 6.20.4.4 std::string cdm\_Rank::HostName

ホスト名

cdm\_Process.h の 25 行で定義されています。

## 6.20.4.5 int cdm\_Rank::RankID

ランク番号

cdm\_Process.h の 24 行で定義されています。

## 6.20.4.6 int cdm\_Rank::TailIndex[3]

終点インデックス

cdm\_Process.h の 28 行で定義されています。

## 6.20.4.7 int cdm\_Rank::VoxelSize[3]

ボクセルサイズ

cdm\_Process.h の 26 行で定義されています。

このクラスの説明は次のファイルから生成されました:

- [cdm\\_Process.h](#)

## 6.21 クラス cdm\_Slice

```
#include <cdm_TimeSlice.h>
```

### Public メソッド

- [cdm\\_Slice](#) ()
- [~cdm\\_Slice](#) ()
- [CDM::E\\_CDM\\_ERRORCODE Read](#) ([cdm\\_TextParser](#) tpCntl, std::string label\_leaf, [CDM::E\\_CDM\\_FORMAT](#) format)  
TimeSlice 要素を読み込む (*inde.dfi*)
- [CDM::E\\_CDM\\_ERRORCODE Write](#) (FILE \*fp, const unsigned tab, [CDM::E\\_CDM\\_FORMAT](#) format)  
DFI ファイル:TimeSlice 要素を出力する

## Public 変数

- int [step](#)  
ステップ番号
- double [time](#)  
時刻
- bool [avr\\_mode](#)  
*Average* 出力フラグ *true*:出力なし、*false*:出力
- int [AveragedStep](#)  
平均ステップ
- double [AveragedTime](#)  
平均タイム
- double [VectorMin](#)  
*Vector* のとき、最小値の合成値
- double [VectorMax](#)  
*Vector* のとき、最大値の合成値
- vector< double > [Min](#)  
最小値
- vector< double > [Max](#)  
最大値

### 6.21.1 説明

index.dfi ファイルの Slice

cdm\_TimeSlice.h の 19 行で定義されています。

### 6.21.2 コンストラクタとデストラクタ

#### 6.21.2.1 cdm\_Slice::cdm\_Slice ( )

コンストラクタ

#### 6.21.2.2 cdm\_Slice::~~cdm\_Slice ( )

デストラクタ

### 6.21.3 関数

#### 6.21.3.1 CDM::E\_CDM\_ERRORCODE cdm\_Slice::Read ( cdm\_TextParser *tpCntl*, std::string *label\_leaf*, CDM::E\_CDM\_FORMAT *format* )

TimeSlice 要素を読み込む (inde.dfi)

引数

in	<i>tpCntl</i>	cdm_TextParser クラス
in	<i>label_leaf</i>	ベースとなる名前 ( "/TimeSlice/Slice" )
in	<i>format</i>	ファイルフォーマット

戻り値

error code



6.21.3.2 `CDM::E_CDM_ERRORCODE cdm_Slice::Write ( FILE * fp, const unsigned tab, CDM::E_CDM_FORMAT format )`

DFI ファイル:TimeSlice 要素を出力する

引数

in	<i>fp</i>	ファイルポインタ
in	<i>tab</i>	インデント
in	<i>format</i>	ファイルフォーマット

戻り値

error code

## 6.21.4 変数

### 6.21.4.1 int cdm\_Slice::AveragedStep

平均ステップ

cdm\_TimeSlice.h の 26 行で定義されています。

### 6.21.4.2 double cdm\_Slice::AveragedTime

平均タイム

cdm\_TimeSlice.h の 27 行で定義されています。

### 6.21.4.3 bool cdm\_Slice::avr\_mode

Average 出力フラグ true:出力なし、false:出力

cdm\_TimeSlice.h の 25 行で定義されています。

### 6.21.4.4 vector<double> cdm\_Slice::Max

最大値

cdm\_TimeSlice.h の 31 行で定義されています。

### 6.21.4.5 vector<double> cdm\_Slice::Min

最小値

cdm\_TimeSlice.h の 30 行で定義されています。

### 6.21.4.6 int cdm\_Slice::step

ステップ番号

cdm\_TimeSlice.h の 23 行で定義されています。

### 6.21.4.7 double cdm\_Slice::time

時刻

cdm\_TimeSlice.h の 24 行で定義されています。

## 6.21.4.8 double cdm\_Slice::VectorMax

Vector のとき、最大値の合成値

cdm\_TimeSlice.h の 29 行で定義されています。

## 6.21.4.9 double cdm\_Slice::VectorMin

Vector のとき、最小値の合成値

cdm\_TimeSlice.h の 28 行で定義されています。

このクラスの説明は次のファイルから生成されました:

- [cdm\\_TimeSlice.h](#)

## 6.22 クラス cdm\_TextParser

```
#include <cdm_TextParser.h>
```

## Public メソッド

- [cdm\\_TextParser](#) ()
- [~cdm\\_TextParser](#) ()
- bool [GetVector](#) (const std::string label, int \*vec, const int nvec, bool checkPath=true)  
*TextParser* 入力ファイルからベクトル値を取得する ( 整数型 )
- bool [GetVector](#) (const std::string label, double \*vec, const int nvec, bool checkPath=true)  
*TextParser* 入力ファイルからベクトル値を取得する ( 実数型 )
- bool [GetVector](#) (const std::string label, std::string \*vec, const int nvec, bool checkPath=true)  
*TextParser* 入力ファイルからベクトル値を取得する ( 文字列型 )
- bool [GetValue](#) (const std::string label, int \*ct, bool checkPath=true)  
*TextParser* 入力ファイルから変数を取得する ( 整数型 )
- bool [GetValue](#) (const std::string label, double \*ct, bool checkPath=true)  
*TextParser* 入力ファイルから変数を取得する ( 実数型 )
- bool [GetValue](#) (const std::string label, std::string \*ct, bool checkPath=true)  
*TextParser* 入力ファイルから変数を取得する ( 文字列型 )
- bool [chkLabel](#) (const std::string label, bool checkPath=true)  
ラベルの有無をチェック
- bool [chkNode](#) (const std::string label)  
ノードの有無をチェック
- bool [GetNodeStr](#) (const std::string label, const int nnode, std::string \*ct)  
ノード以下の *nnode* 番目の文字列を取得する
- int [countLabels](#) (const std::string label)  
ノード以下のラベルの数を数える
- void [getTPinstance](#) ()  
*TextParserLibrary* のインスタンス生成
- int [readTPfile](#) (const std::string filename)  
*TextParser* オブジェクトに入力ファイルをセットする
- int [remove](#) ()
- TextParser \* [getTPPtr](#) ()

## Private 変数

- TextParser \* [tp](#)  
テキストパーサ

### 6.22.1 説明

cdm\_TextParser.h の 30 行で定義されています。

### 6.22.2 コンストラクタとデストラクタ

#### 6.22.2.1 cdm\_TextParser::cdm\_TextParser ( ) [inline]

##### コンストラクタ

cdm\_TextParser.h の 37 行で定義されています。

#### 6.22.2.2 cdm\_TextParser::~~cdm\_TextParser ( ) [inline]

##### デストラクタ

cdm\_TextParser.h の 40 行で定義されています。

### 6.22.3 関数

#### 6.22.3.1 bool cdm\_TextParser::chkLabel ( const std::string *label*, bool *checkPath* = true )

##### ラベルの有無をチェック

##### 引数

in	<i>label</i>	チェックするラベル ( 絶対パス )
in	<i>checkPath</i>	true: ラベル存在チェックをフルパスで行う

#### 6.22.3.2 bool cdm\_TextParser::chkNode ( const std::string *label* )

##### ノードの有無をチェック

##### 引数

in	<i>label</i>	チェックするノード ( 絶対パス )
----	--------------	--------------------

#### 6.22.3.3 int cdm\_TextParser::countLabels ( const std::string *label* )

##### ノード以下のラベルの数を数える

##### 引数

in	<i>label</i>	ラベルを数えるノードの絶対パス
----	--------------	-----------------

##### 戻り値

ラベルの数 (エラー、もしくははない場合は-1を返す)
-----------------------------

6.22.3.4 `bool cdm_TextParser::GetNodeStr ( const std::string label, const int nnode, std::string * ct )`

ノード以下の *nnode* 番目の文字列を取得する

引数

in	<i>label</i>	ノードの絶対パス
in	<i>nnode</i>	取得する文字列が現れる順番
out	<i>ct</i>	取得した文字列

6.22.3.5 `void cdm_TextParser::getTPInstance ( )`

TextParserLibrary のインスタンス生成

戻り値

エラーコード

6.22.3.6 `TextParser* cdm_TextParser::getTPPtr ( )` [inline]

`cdm_TextParser.h` の 153 行で定義されています。

6.22.3.7 `bool cdm_TextParser::GetValue ( const std::string label, int * ct, bool checkPath = true )`

TextParser 入力ファイルから変数を取得する ( 整数型 )

引数

in	<i>label</i>	取得する変数のラベル ( 絶対パス )
in	<i>checkPath</i>	true:ラベル存在チェックをフルパスで行う
out	<i>ct</i>	変数格納ポインタ

参照元 `cdm_NonUniformDomain< T >::Read()`.

6.22.3.8 `bool cdm_TextParser::GetValue ( const std::string label, double * ct, bool checkPath = true )`

TextParser 入力ファイルから変数を取得する ( 実数型 )

引数

in	<i>label</i>	取得する変数のラベル ( 絶対パス )
in	<i>checkPath</i>	true:ラベル存在チェックをフルパスで行う
out	<i>ct</i>	変数格納ポインタ

6.22.3.9 `bool cdm_TextParser::GetValue ( const std::string label, std::string * ct, bool checkPath = true )`

TextParser 入力ファイルから変数を取得する ( 文字列型 )

引数

in	<i>label</i>	取得する変数のラベル（絶対パス）
in	<i>checkPath</i>	true:ラベル存在チェックをフルパスで行う
out	<i>ct</i>	変数格納ポインタ

6.22.3.10 `bool cdm_TextParser::GetVector ( const std::string label, int * vec, const int nvec, bool checkPath = true )`

TextParser 入力ファイルからベクトル値を取得する（整数型）

引数

in	<i>label</i>	取得するベクトルのラベル（絶対パス）
out	<i>vec</i>	ベクトル格納配列ポインタ
in	<i>nvec</i>	ベクトルサイズ
in	<i>checkPath</i>	true:ラベル存在チェックをフルパスで行う

参照元 `cdm_NonUniformDomain< T >::Read()`.

6.22.3.11 `bool cdm_TextParser::GetVector ( const std::string label, double * vec, const int nvec, bool checkPath = true )`

TextParser 入力ファイルからベクトル値を取得する（実数型）

引数

in	<i>label</i>	取得するベクトルのラベル（絶対パス）
out	<i>vec</i>	ベクトル格納配列ポインタ
in	<i>nvec</i>	ベクトルサイズ
in	<i>checkPath</i>	true:ラベル存在チェックをフルパスで行う

6.22.3.12 `bool cdm_TextParser::GetVector ( const std::string label, std::string * vec, const int nvec, bool checkPath = true )`

TextParser 入力ファイルからベクトル値を取得する（文字列型）

引数

in	<i>label</i>	取得するベクトルのラベル（絶対パス）
out	<i>vec</i>	ベクトル格納配列ポインタ
in	<i>nvec</i>	ベクトルサイズ
in	<i>checkPath</i>	true:ラベル存在チェックをフルパスで行う

6.22.3.13 `int cdm_TextParser::readTPfile ( const std::string filename )`

TextParser オブジェクトに入力ファイルをセットする

引数

in	<i>filename</i>	入力ファイル名
----	-----------------	---------

戻り値

エラーコード
--------

6.22.3.14 `int cdm_TextParser::remove ( ) [inline]`

テキストパーサーの内容を破棄

cdm\_TextParser.h の 147 行で定義されています。

## 6.22.4 変数

### 6.22.4.1 TextParser\* cdm\_TextParser::tp [private]

テキストパーサ

cdm\_TextParser.h の 33 行で定義されています。

このクラスの説明は次のファイルから生成されました:

- [cdm\\_TextParser.h](#)

## 6.23 クラス cdm\_TimeSlice

```
#include <cdm_TimeSlice.h>
```

### Public メソッド

- [cdm\\_TimeSlice \(\)](#)
- [~cdm\\_TimeSlice \(\)](#)
- [CDM::E\\_CDM\\_ERRORCODE Read](#) ([cdm\\_TextParser](#) tpCntl, [CDM::E\\_CDM\\_FORMAT](#) format)  
*TimeSlice* 要素を読み込む (*index.dfi*)
- [CDM::E\\_CDM\\_ERRORCODE Write](#) (FILE \*fp, const unsigned tab, [CDM::E\\_CDM\\_FORMAT](#) format)  
*DFI* ファイル: *TimeSlice* 要素を出力する
- [CDM::E\\_CDM\\_ERRORCODE getVectorMinMax](#) (const unsigned step, double &vec\_min, double &vec\_max)  
*DFI* に出力されている *minmax* の合成値を取得
- [CDM::E\\_CDM\\_ERRORCODE getMinMax](#) (const unsigned step, const int variNo, double &min\_value, double &max\_value)
- void [AddSlice](#) (int step, double time, double \*minmax, int Nvari, [CDM::E\\_CDM\\_FORMAT](#) format, bool avr\_mode, int step\_avr, double time\_avr)  
*SliceList* への追加

### Public 変数

- vector< [cdm\\_Slice](#) > [SliceList](#)

## 6.23.1 説明

index.dfi ファイルの TimeSlice

cdm\_TimeSlice.h の 66 行で定義されています。

## 6.23.2 コンストラクタとデストラクタ

### 6.23.2.1 cdm\_TimeSlice::cdm\_TimeSlice ( )

コンストラクタ

## 6.23.2.2 cdm\_TimeSlice::~cdm\_TimeSlice ( )

デストラクタ

## 6.23.3 関数

## 6.23.3.1 void cdm\_TimeSlice::AddSlice ( int step, double time, double \* minmax, int Nvari, CDM::E\_CDM\_FORMAT format, bool avr\_mode, int step\_avr, double time\_avr )

SliceList への追加

引数

in	step	ステップ番号
in	time	時刻
in	minmax	minmax
in	Nvari	変数の個数
in	format	ファイルフォーマット
in	avr_mode	Average があるかないかのフラグ
in	step_avr	Average step
in	time_avr	Average time

参照元 cdm\_DFI::AddTimeSlice().

## 6.23.3.2 CDM::E\_CDM\_ERRORCODE cdm\_TimeSlice::getMinMax ( const unsigned step, const int variNo, double &amp; min\_value, double &amp; max\_value )

brief DFI に出力されている minmax と minmax の合成値を取得

引数

in	step	取得するステップ
in	variNo	取得する変数番号 (0 ~ n)
out	min_value	取得した min
out	max_value	取得した max

戻り値

error code 取得出来たときは E\_CDM\_SUCCESS

## 6.23.3.3 CDM::E\_CDM\_ERRORCODE cdm\_TimeSlice::getVectorMinMax ( const unsigned step, double &amp; vec\_min, double &amp; vec\_max )

DFI に出力されている minmax の合成値を取得

引数

in	step	取得するステップ
out	vec_min	取得した min の合成値
out	vec_max	取得した min の合成値

戻り値

error code 取得出来たときは E\_CDM\_SUCCESS



6.23.3.4 `CDM::E_CDM_ERRORCODE cdm_TimeSlice::Read ( cdm_TextParser tpCntl, CDM::E_CDM_FORMAT format )`

TimeSlice 要素を読み込む (inde.dfi)

引数

in	<i>tpCntl</i>	cdm_TextParser クラス
in	<i>format</i>	ファイルフォーマット

戻り値

error code

**6.23.3.5 CDM::E\_CDM\_ERRORCODE cdm\_TimeSlice::Write ( FILE \* *fp*, const unsigned *tab*, CDM::E\_CDM\_FORMAT *format* )**

DFI ファイル:TimeSlice 要素を出力する

引数

in	<i>fp</i>	ファイルポインタ
in	<i>tab</i>	インデント
in	<i>format</i>	ファイルフォーマット

戻り値

true:出力成功 false:出力失敗

## 6.23.4 変数

**6.23.4.1 vector<cdm\_Slice> cdm\_TimeSlice::SliceList**

cdm\_TimeSlice.h の 70 行で定義されています。

このクラスの説明は次のファイルから生成されました:

- [cdm\\_TimeSlice.h](#)

## 6.24 クラス テンプレート cdm\_TypeArray< T >

```
#include <cdm_TypeArray.h>
```

cdm\_TypeArray< T > に対する継承グラフ

cdm\_TypeArray< T > のコラボレーション図

### Public メソッド

- [cdm\\_TypeArray](#) (CDM::E\_CDM\_DTYPE dtype, CDM::E\_CDM\_ARRAYSHAPE shape, size\_t ix, size\_t jx, size\_t kx, size\_t gc, size\_t nvari=1)  
コンストラクタ
- [cdm\\_TypeArray](#) (T \*data, CDM::E\_CDM\_DTYPE dtype, CDM::E\_CDM\_ARRAYSHAPE shape, size\_t ix, size\_t jx, size\_t kx, size\_t gc, size\_t nvari=1)  
コンストラクタ
- virtual [~cdm\\_TypeArray](#) ()  
デストラクタ
- T \* [getData](#) (bool extract=false)  
実データのポインタを取得

- `const T & val (int i, int j, int k, int l=0) const`
- `T & val (int i, int j, int k, int l=0)`
- `const T & hval (int i, int j, int k, int l=0) const`
- `T & hval (int i, int j, int k, int l=0)`
- `const T & _val (size_t i, size_t j, size_t k, size_t l=0) const`
- `T & _val (size_t i, size_t j, size_t k, size_t l=0)`
- `virtual int copyArray (cdm_Array *dst, bool ignoreGc=false)`  
配列コピー (自信を *dst* にコピー。 *head/tail* を考慮した重複範囲をコピー)
- `virtual int copyArray (int sta[3], int end[3], cdm_Array *dst)`  
範囲指定での配列コピー (自信を *dst* にコピー。 *head/tail* を考慮した重複範囲をコピー)
- `virtual int copyArrayNvari (cdm_Array *dst, int vari, bool ignoreGc=false)`  
指定変数の配列コピー (自信を *dst* にコピー。 *head/tail* を考慮した重複範囲をコピー)
- `virtual int copyArrayNvari (int sta[3], int end[3], cdm_Array *dst, int vari)`  
指定変数の範囲指定での配列コピー (自信を *dst* にコピー。 *head/tail* を考慮した重複範囲をコピー)
- `virtual int copyArrayNvari_to_ijk (cdm_Array *dst, int vari, bool ignoreGc=false)`  
指定した変数の配列のみ取得し、*IJK* 配列で *dst* にコピー
- `virtual int copyArrayNvari_to_ijk (int sta[3], int end[3], cdm_Array *dst, int vari)`  
指定した変数の配列のみ範囲指定で取得し、*IJK* 配列で *dst* にコピー
- `virtual size_t readBinary (FILE *fp, bool bMatchEndian)`  
配列サイズ分のバイナリデータを読み込み (戻り値は読み込んだ要素数)
- `virtual size_t writeBinary (FILE *fp)`  
配列サイズ分のバイナリデータを書き出す (戻り値は読み込んだ要素数)
- `virtual size_t writeAscii (FILE *fp)`  
配列サイズ分の *ascii* データを書き出す (戻り値は読み込んだ要素数)

## Protected メソッド

- `cdm_TypeArray ()`  
デフォルトコンストラクタ

## Protected 変数

- `bool m_outptr`  
実データポインタタイプ
- `T * m_data`  
実データ配列

## Additional Inherited Members

### 6.24.1 説明

`template<class T>class cdm_TypeArray< T >`

`cdm_TypeArray.h` の 15 行で定義されています。

## 6.24.2 コンストラクタとデストラクタ

6.24.2.1 `template<class T> cdm_TypeArray< T >::cdm_TypeArray ( CDM::E_CDM_DTYPE dtype, CDM::E_CDM_ARRAYSHAPE shape, size_t ix, size_t jx, size_t kx, size_t gx, size_t nvari = 1 ) [inline]`

### コンストラクタ

cdm\_TypeArray.h の 28 行で定義されています。

参照先 cdm\_TypeArray< T >::m\_data, cdm\_Array::m\_gc, cdm\_Array::m\_nvari, cdm\_TypeArray< T >::m\_outptr, と cdm\_Array::m\_sz.

6.24.2.2 `template<class T> cdm_TypeArray< T >::cdm_TypeArray ( T * data, CDM::E_CDM_DTYPE dtype, CDM::E_CDM_ARRAYSHAPE shape, size_t ix, size_t jx, size_t kx, size_t gx, size_t nvari = 1 ) [inline]`

### コンストラクタ

cdm\_TypeArray.h の 51 行で定義されています。

参照先 cdm\_TypeArray< T >::m\_data, と cdm\_TypeArray< T >::m\_outptr.

6.24.2.3 `template<class T> virtual cdm_TypeArray< T >::~~cdm_TypeArray ( ) [inline],[virtual]`

### デストラクタ

cdm\_TypeArray.h の 68 行で定義されています。

参照先 cdm\_TypeArray< T >::m\_data, と cdm\_TypeArray< T >::m\_outptr.

6.24.2.4 `template<class T> cdm_TypeArray< T >::cdm_TypeArray ( ) [inline],[protected]`

### デフォルトコンストラクタ

cdm\_TypeArray.h の 147 行で定義されています。

参照先 cdm\_TypeArray< T >::m\_data.

## 6.24.3 関数

6.24.3.1 `template<class T> cdm_TypeArray< T >::_val ( size_t i, size_t j, size_t k, size_t l = 0 ) const`

参照 (ガイドセルを含む) ガイドセルを含む配列全体の最小インデックスを (0,0,0) とする IJKN のとき val(i,j,k,n) NIJK のとき val(n,i,j,k)

cdm\_Array\_inline.h の 365 行で定義されています。

参照元 cdm\_DFI\_PLOT3D::write\_Func().

6.24.3.2 `template<class T> cdm_TypeArray< T >::_val ( size_t i, size_t j, size_t k, size_t l = 0 )`

cdm\_Array\_inline.h の 354 行で定義されています。

6.24.3.3 `template<class T> cdm_TypeArray< T >::copyArray ( cdm_Array * dst, bool ignoreGc = false ) [virtual]`

配列コピー (自信を dst にコピー。head/tail を考慮した重複範囲をコピー)

cdm\_Arrayを実装しています。

cdm\_Array\_inline.h の 373 行で定義されています。

参照先 `cdm_Array::getGcInt()`, `cdm_Array::getHeadIndex()`, と `cdm_Array::getTailIndex()`.

6.24.3.4 `template<class T> cdm_TypeArray< T >::copyArray ( int sta[3], int end[3], cdm_Array * dst )`  
`[virtual]`

範囲指定での配列コピー (自信を `dst` にコピー。head/tail を考慮した重複範囲をコピー)

[cdm\\_Array](#)を実装しています。

`cdm_Array_inline.h` の 401 行で定義されています。

参照先 `CDM::E_CDM_IJKN`, `cdm_Array::getArrayShape()`, `cdm_Array::getDataType()`, `cdm_Array::getGcInt()`, `cdm_Array::getHeadIndex()`, `cdm_Array::getNvari()`, `cdm_Array::getTailIndex()`, と `cdm_TypeArray< T >::hval()`.

6.24.3.5 `template<class T> cdm_TypeArray< T >::copyArrayNvari ( cdm_Array * dst, int vari, bool ignoreGc = false )`  
`[virtual]`

指定変数の配列コピー (自信を `dst` にコピー。head/tail を考慮した重複範囲をコピー)

[cdm\\_Array](#)を実装しています。

`cdm_Array_inline.h` の 479 行で定義されています。

参照先 `cdm_Array::getGcInt()`, `cdm_Array::getHeadIndex()`, と `cdm_Array::getTailIndex()`.

参照元 `cdm_DFI_PLOT3D::read_Func()`.

6.24.3.6 `template<class T> cdm_TypeArray< T >::copyArrayNvari ( int sta[3], int end[3], cdm_Array * dst, int vari )`  
`[virtual]`

指定変数の範囲指定での配列コピー (自信を `dst` にコピー。head/tail を考慮した重複範囲をコピー)

[cdm\\_Array](#)を実装しています。

`cdm_Array_inline.h` の 507 行で定義されています。

参照先 `CDM::E_CDM_IJKN`, `cdm_Array::getArrayShape()`, `cdm_Array::getDataType()`, `cdm_Array::getGcInt()`, `cdm_Array::getHeadIndex()`, `cdm_Array::getNvari()`, `cdm_Array::getTailIndex()`, と `cdm_TypeArray< T >::hval()`.

6.24.3.7 `template<class T> cdm_TypeArray< T >::copyArrayNvari_to_ijk ( cdm_Array * dst, int vari, bool ignoreGc = false )`  
`[virtual]`

指定した変数の配列のみ取得し、IJK 配列で `dst` にコピー

[cdm\\_Array](#)を実装しています。

`cdm_Array_inline.h` の 582 行で定義されています。

参照先 `cdm_Array::getGcInt()`, `cdm_Array::getHeadIndex()`, と `cdm_Array::getTailIndex()`.

6.24.3.8 `template<class T> cdm_TypeArray< T >::copyArrayNvari_to_ijk ( int sta[3], int end[3], cdm_Array * dst, int vari )`  
`[virtual]`

指定した変数の配列のみ範囲指定で取得し、IJK 配列で `dst` にコピー

[cdm\\_Array](#)を実装しています。

`cdm_Array_inline.h` の 610 行で定義されています。

参照先 `CDM::E_CDM_IJKN`, `cdm_Array::getArrayShape()`, `cdm_Array::getDataType()`, `cdm_Array::getGcInt()`, `cdm_Array::getHeadIndex()`, `cdm_Array::getNvari()`, `cdm_Array::getTailIndex()`, と `cdm_TypeArray< T >::hval()`.

6.24.3.9 `template<class T> T* cdm_TypeArray<T>::getData ( bool extract = false ) [inline]`

実データのポインタを取得

cdm\_TypeArray.h の 80 行で定義されています。

参照先 cdm\_TypeArray<T>::m\_data.

参照元 cdm\_Array::getData(), と cdm\_DFI::setGridData().

6.24.3.10 `template<class T> cdm_TypeArray<T>::hval ( int i, int j, int k, int l = 0 ) const`

参照 (head インデクス考慮版) 実セルの最小インデクスを (head[0],head[1],head[2]) とする IJKN のとき val(i,j,k,n) NIJK のとき val(n,i,j,k)

cdm\_Array\_inline.h の 346 行で定義されています。

参照元 cdm\_TypeArray<T>::copyArray(), cdm\_TypeArray<T>::copyArrayNvari(), と cdm\_TypeArray<T>::copyArrayNvari\_to\_ijk().

6.24.3.11 `template<class T> cdm_TypeArray<T>::hval ( int i, int j, int k, int l = 0 )`

cdm\_Array\_inline.h の 335 行で定義されています。

6.24.3.12 `template<class T> size_t cdm_TypeArray<T>::readBinary ( FILE * fp, bool bMatchEndian ) [virtual]`

配列サイズ分のバイナリデータを読み込み (戻り値は読み込んだ要素数)

[cdm\\_Array](#)を実装しています。

cdm\_Array\_inline.h の 756 行で定義されています。

参照先 BSWAPVEC, DBSWAPVEC, と SBSWAPVEC.

参照元 cdm\_DFI\_PLOT3D::read\_Func().

6.24.3.13 `template<class T> cdm_TypeArray<T>::val ( int i, int j, int k, int l = 0 ) const`

参照 実セルの最小インデクスを (0,0,0) とする IJKN のとき val(i,j,k,n) NIJK のとき val(n,i,j,k)

cdm\_Array\_inline.h の 327 行で定義されています。

参照元 cdm\_DFI\_PLOT3D::read\_Func(), cdm\_DFI::setGridData(), と cdm\_DFI::VolumeDataDivide().

6.24.3.14 `template<class T> cdm_TypeArray<T>::val ( int i, int j, int k, int l = 0 )`

cdm\_Array\_inline.h の 316 行で定義されています。

6.24.3.15 `template<class T> size_t cdm_TypeArray<T>::writeAscii ( FILE * fp ) [virtual]`

配列サイズ分の ascii データを書き出す (戻り値は読み込んだ要素数)

[cdm\\_Array](#)を実装しています。

cdm\_Array\_inline.h の 816 行で定義されています。

6.24.3.16 `template<class T> size_t cdm_TypeArray<T>::writeBinary ( FILE * fp ) [virtual]`

配列サイズ分のバイナリデータを書き出す (戻り値は読み込んだ要素数)

[cdm\\_Array](#)を実装しています。

cdm\_Array\_inline.h の 794 行で定義されています。

#### 6.24.4 変数

6.24.4.1 `template<class T> T* cdm_TypeArray< T >::m_data` [protected]

実データ配列

cdm\_TypeArray.h の 164 行で定義されています。

参照元 `cdm_TypeArray< T >::cdm_TypeArray()`, `cdm_TypeArray< T >::getData()`, と `cdm_TypeArray< T >::~~cdm_TypeArray()`.

6.24.4.2 `template<class T> bool cdm_TypeArray< T >::m_outptr` [protected]

実データポインタタイプ

cdm\_TypeArray.h の 161 行で定義されています。

参照元 `cdm_TypeArray< T >::cdm_TypeArray()`, と `cdm_TypeArray< T >::~~cdm_TypeArray()`.

このクラスの説明は次のファイルから生成されました:

- [cdm\\_TypeArray.h](#)
- [cdm\\_Array\\_inline.h](#)

## 6.25 クラス cdm\_Unit

```
#include <cdm_Unit.h>
```

### Public メソッド

- [cdm\\_Unit](#) ()
- [~cdm\\_Unit](#) ()
- [CDM::E\\_CDM\\_ERRORCODE Read](#) ([cdm\\_TextParser](#) tpCntl)  
*read Unit(inde.dfi)*
- [CDM::E\\_CDM\\_ERRORCODE GetUnitElem](#) (const std::string Name, [cdm\\_UnitElem](#) &unit)  
*該当するUnitElem の取り出し*
- [CDM::E\\_CDM\\_ERRORCODE GetUnit](#) (const std::string Name, std::string &unit, double &ref, double &diff, bool &bSetDiff)  
*単位の取り出し ("m","cm",...)*
- [CDM::E\\_CDM\\_ERRORCODE Write](#) (FILE \*fp, const unsigned tab)  
*DFI ファイル:Unit 要素を出力する*
- void [AddUnit](#) (const std::string Name, const std::string Unit, const double reference, const double difference=0.0, const bool BsetDiff=false)  
*Unit をセットする*

### Public 変数

- map< std::string, [cdm\\_UnitElem](#) > [UnitList](#)

### 6.25.1 説明

index.dfi ファイルの Unit

cdm\_Unit.h の 68 行で定義されています。

### 6.25.2 コンストラクタとデストラクタ

#### 6.25.2.1 cdm\_Unit::cdm\_Unit ( )

コンストラクタ

#### 6.25.2.2 cdm\_Unit::~~cdm\_Unit ( )

デストラクタ

### 6.25.3 関数

#### 6.25.3.1 void cdm\_Unit::AddUnit ( const std::string *Name*, const std::string *Unit*, const double *reference*, const double *difference* = 0.0, const bool *BsetDiff* = false )

Unit をセットする

引数

in	<i>Name</i>	追加する単位系 ("Length","Velocity",,,)
in	<i>Unit</i>	単位ラベル ("M","CM","MM","M/S",,,)
in	<i>reference</i>	規格化したスケール値
in	<i>difference</i>	差の値
in	<i>BsetDiff</i>	difference の有無

#### 6.25.3.2 CDM::E\_CDM\_ERRORCODE cdm\_Unit::GetUnit ( const std::string *Name*, std::string & *unit*, double & *ref*, double & *diff*, bool & *bSetDiff* )

単位の取り出し ("m","cm",,,)

引数

in	<i>Name</i>	取り出す単位の種類
out	<i>unit</i>	単位文字列
out	<i>ref</i>	reference
out	<i>diff</i>	difference
out	<i>bSetDiff</i>	difference 有無フラグ true:あり、false:なし

戻り値

error code

#### 6.25.3.3 CDM::E\_CDM\_ERRORCODE cdm\_Unit::GetUnitElem ( const std::string *Name*, cdm\_UnitElem & *unit* )

該当するUnitElem の取り出し



引数

in	<i>Name</i>	取り出す単位の種類
out	<i>unit</i>	取得した cdm_UnitElem クラス

戻り値

error code

#### 6.25.3.4 CDM::E\_CDM\_ERRORCODE cdm\_Unit::Read ( cdm\_TextParser tpCntl )

read Unit(inde.dfi)

引数

in	<i>tpCntl</i>	cdm_TextParser クラス
----	---------------	--------------------

戻り値

error code

#### 6.25.3.5 CDM::E\_CDM\_ERRORCODE cdm\_Unit::Write ( FILE \* fp, const unsigned tab )

DFI ファイル:Unit 要素を出力する

引数

in	<i>fp</i>	ファイルポインタ
in	<i>tab</i>	インデント

戻り値

error code

### 6.25.4 変数

#### 6.25.4.1 map<std::string,cdm\_UnitElem> cdm\_Unit::UnitList

cdm\_Unit.h の 72 行で定義されています。

このクラスの説明は次のファイルから生成されました:

- [cdm\\_Unit.h](#)

## 6.26 クラス cdm\_UnitElem

```
#include <cdm_Unit.h>
```

Public メソッド

- [cdm\\_UnitElem](#) ()
- [cdm\\_UnitElem](#) (const std::string \_Name, const std::string \_Unit, const double \_reference, const double \_difference, const bool \_BsetDiff)
- [~cdm\\_UnitElem](#) ()

- [CDM::E\\_CDM\\_ERRORCODE Read](#) ([cdm\\_TextParser](#) tpCntl, const std::string label\_leaf)  
Unit 要素の読み込み
- [CDM::E\\_CDM\\_ERRORCODE Write](#) (FILE \*fp, const unsigned tab)  
DFI ファイル:Unit 要素を出力する

## Public 変数

- std::string [Name](#)  
単位の種類名 (*Length, Velocity,,*)
- std::string [Unit](#)  
単位のラベル (*m,m/s,Pa,,*)
- double [reference](#)  
規格化に用いたスケール
- double [difference](#)  
差
- bool [BsetDiff](#)  
*difference* の有無 ( *false*:なし *true*:あり )

### 6.26.1 説明

cdm\_Unit.h の 18 行で定義されています。

### 6.26.2 コンストラクタとデストラクタ

#### 6.26.2.1 cdm\_UnitElem::cdm\_UnitElem ( )

コンストラクタ

#### 6.26.2.2 cdm\_UnitElem::cdm\_UnitElem ( const std::string \_Name, const std::string \_Unit, const double \_reference, const double \_difference, const bool \_BsetDiff )

コンストラクタ

#### 6.26.2.3 cdm\_UnitElem::~~cdm\_UnitElem ( )

デストラクタ

### 6.26.3 関数

#### 6.26.3.1 CDM::E\_CDM\_ERRORCODE cdm\_UnitElem::Read ( cdm\_TextParser tpCntl, const std::string label\_leaf )

Unit 要素の読み込み

引数

in	<i>tpCntl</i>	cdm_TextParser クラス
in	<i>label_leaf</i>	

戻り値

error code

6.26.3.2 CDM::E\_CDM\_ERRORCODE cdm\_UnitElem::Write ( FILE \* *fp*, const unsigned *tab* )

DFI ファイル:Unit 要素を出力する

引数

in	<i>fp</i>	ファイルポインタ
in	<i>tab</i>	インデント

戻り値

error code

## 6.26.4 変数

### 6.26.4.1 bool cdm\_UnitElem::BsetDiff

difference の有無 ( false:なし true:あり )  
cdm\_Unit.h の 26 行で定義されています。

### 6.26.4.2 double cdm\_UnitElem::difference

差

cdm\_Unit.h の 25 行で定義されています。

### 6.26.4.3 std::string cdm\_UnitElem::Name

単位の種類名 (Length,Velovity,,)  
cdm\_Unit.h の 22 行で定義されています。

### 6.26.4.4 double cdm\_UnitElem::reference

規格化に用いたスケール

cdm\_Unit.h の 24 行で定義されています。

### 6.26.4.5 std::string cdm\_UnitElem::Unit

単位のラベル (m,m/s,Pa,,)

cdm\_Unit.h の 23 行で定義されています。

このクラスの説明は次のファイルから生成されました:

- [cdm\\_Unit.h](#)

## 6.27 クラス cdm\_VisIt

```
#include <cdm_VisIt.h>
```

Public メソッド

- [cdm\\_VisIt \(\)](#)
  - [cdm\\_VisIt \(const std::string \\_PlotGC\)](#)
- コンストラクタ

- `~cdm_VisIt()`
- `CDM::E_CDM_ERRORCODE Read (cdm_TextParser tpCntl)`  
`read VisIt` オプションの読み込み
- `CDM::E_CDM_ERRORCODE Write (FILE *fp, const unsigned tab)`  
`DFI` ファイル: `VisIt` オプションを出力する

## Public 変数

- `std::string PlotGC`  
ガイドセル描画オプション

### 6.27.1 説明

`cdm_VisIt.h` の 18 行で定義されています。

### 6.27.2 コンストラクタとデストラクタ

#### 6.27.2.1 `cdm_VisIt::cdm_VisIt ( )`

コンストラクタ

#### 6.27.2.2 `cdm_VisIt::cdm_VisIt ( const std::string _PlotGC )`

コンストラクタ

引数

<code>in</code>	<code>_PlotGC</code>	ガイドセル描画オプション
-----------------	----------------------	--------------

#### 6.27.2.3 `cdm_VisIt::~~cdm_VisIt ( )`

デストラクタ

### 6.27.3 関数

#### 6.27.3.1 `CDM::E_CDM_ERRORCODE cdm_VisIt::Read ( cdm_TextParser tpCntl )`

`read VisIt` オプションの読み込み

引数

<code>in</code>	<code>tpCntl</code>	<code>cdm_TextParser</code> クラス
-----------------	---------------------	---------------------------------

戻り値

error code

#### 6.27.3.2 `CDM::E_CDM_ERRORCODE cdm_VisIt::Write ( FILE * fp, const unsigned tab )`

`DFI` ファイル: `VisIt` オプションを出力する

引数

in	<i>fp</i>	ファイルポインタ
in	<i>tab</i>	インデント

戻り値

error code

## 6.27.4 変数

### 6.27.4.1 std::string cdm\_VisIt::PlotGC

ガイドセル描画オプション

cdm\_VisIt.h の 22 行で定義されています。

参照元 cdm\_DFI::WriteInit().

このクラスの説明は次のファイルから生成されました:

- [cdm\\_VisIt.h](#)

## 6.28 構造体 cdm\_DFI\_NETCDF::stDimInfo

```
#include <cdm_DFI_NETCDF.h>
```

### Public 変数

- int [id](#)  
*dimension ID*
- string [name](#)  
*dimension name*
- size\_t [len](#)  
*dimension*

### 6.28.1 説明

cdm\_DFI\_NETCDF.h の 25 行で定義されています。

### 6.28.2 変数

#### 6.28.2.1 int cdm\_DFI\_NETCDF::stDimInfo::id

dimension ID

cdm\_DFI\_NETCDF.h の 27 行で定義されています。

#### 6.28.2.2 size\_t cdm\_DFI\_NETCDF::stDimInfo::len

dimension

cdm\_DFI\_NETCDF.h の 29 行で定義されています。

6.28.2.3 `string cdm_DFI_NETCDF::stDimInfo::name`

dimension name

`cdm_DFI_NETCDF.h` の 28 行で定義されています。

この構造体の説明は次のファイルから生成されました:

- [cdm\\_DFI\\_NETCDF.h](#)

6.29 構造体 `cdm_DFI_NETCDF::stVarInfo`

```
#include <cdm_DFI_NETCDF.h>
```

## Public メソッド

- [stVarInfo](#) ()
- [~stVarInfo](#) ()
- void [clear](#) ()

## Public 変数

- int [id](#)  
*variable ID*
- string [name](#)  
*variable name*
- nc\_type [type](#)  
*data type*
- vector< [stDimInfo](#) > [dims](#)  
*array dimension*
- string [unit](#)  
*variable unit name*
- double \* [data](#)  
*data array for 1D array*
- int [endian](#)  
*endian type*

## 6.29.1 説明

`cdm_DFI_NETCDF.h` の 32 行で定義されています。

## 6.29.2 コンストラクタとデストラクタ

6.29.2.1 `cdm_DFI_NETCDF::stVarInfo::stVarInfo( )` [inline]

`cdm_DFI_NETCDF.h` の 42 行で定義されています。

参照先 `clear()`, と `data`.

#### 6.29.2.2 `cdm_DFI_NETCDF::stVarInfo::~stVarInfo ( ) [inline]`

`cdm_DFI_NETCDF.h` の 48 行で定義されています。

参照先 `clear()`.

### 6.29.3 関数

#### 6.29.3.1 `void cdm_DFI_NETCDF::stVarInfo::clear ( ) [inline]`

`cdm_DFI_NETCDF.h` の 53 行で定義されています。

参照先 `data`, `dims`, `endian`, `name`, `type`, と `unit`.

参照元 `stVarInfo()`, と `~stVarInfo()`.

### 6.29.4 変数

#### 6.29.4.1 `double* cdm_DFI_NETCDF::stVarInfo::data`

data array for 1D array

`cdm_DFI_NETCDF.h` の 39 行で定義されています。

参照元 `clear()`, と `stVarInfo()`.

#### 6.29.4.2 `vector<stDimInfo> cdm_DFI_NETCDF::stVarInfo::dims`

array dimension

`cdm_DFI_NETCDF.h` の 37 行で定義されています。

参照元 `clear()`.

#### 6.29.4.3 `int cdm_DFI_NETCDF::stVarInfo::endian`

endian type

`cdm_DFI_NETCDF.h` の 40 行で定義されています。

参照元 `clear()`.

#### 6.29.4.4 `int cdm_DFI_NETCDF::stVarInfo::id`

variable ID

`cdm_DFI_NETCDF.h` の 34 行で定義されています。

#### 6.29.4.5 `string cdm_DFI_NETCDF::stVarInfo::name`

variable name

`cdm_DFI_NETCDF.h` の 35 行で定義されています。

参照元 `clear()`.

#### 6.29.4.6 `nc_type cdm_DFI_NETCDF::stVarInfo::type`

data type



`cdm_DFI_NETCDF.h` の 36 行で定義されています。

参照元 `clear()`.

#### 6.29.4.7 string `cdm_DFI_NETCDF::stVarInfo::unit`

variable unit name

`cdm_DFI_NETCDF.h` の 38 行で定義されています。

参照元 `clear()`.

この構造体の説明は次のファイルから生成されました:

- [cdm\\_DFI\\_NETCDF.h](#)



## Chapter 7

# ファイル

### 7.1 cdm\_ActiveSubDomain.h

このグラフは、どのファイルから直接、間接的にインクルードされているかを示しています。

構成

- class [cdm\\_ActiveSubDomain](#)

### 7.2 cdm\_Array.h

```
#include "cdm_Define.h"
```

cdm\_Array.h のインクルード依存関係図このグラフは、どのファイルから直接、間接的にインクルードされているかを示しています。

構成

- class [cdm\\_Array](#)

関数

- void [cdm\\_interp\\_ijkn\\_r4\\_](#) (const int \*szS, const int \*gcS, const int \*szD, const int \*gcD, const int \*nvvari, float \*src, float \*dst)
- void [cdm\\_interp\\_ijkn\\_r8\\_](#) (const int \*szS, const int \*gcS, const int \*szD, const int \*gcD, const int \*nvvari, double \*src, double \*dst)
- void [cdm\\_interp\\_nijk\\_r4\\_](#) (const int \*szS, const int \*gcS, const int \*szD, const int \*gcD, const int \*nvvari, float \*src, float \*dst)
- void [cdm\\_interp\\_nijk\\_r8\\_](#) (const int \*szS, const int \*gcS, const int \*szD, const int \*gcD, const int \*nvvari, double \*src, double \*dst)

#### 7.2.1 関数

7.2.1.1 void [cdm\\_interp\\_ijkn\\_r4\\_](#) ( const int \* *szS*, const int \* *gcS*, const int \* *szD*, const int \* *gcD*, const int \* *nvvari*, float \* *src*, float \* *dst* )

参照元 [cdm\\_Array::interp\\_coarse\(\)](#).

```
7.2.1.2 void cdm_interp_ijkn_r8_ ( const int * szS, const int * gcS, const int * szD, const int * gcD, const int * nvari, double
    * src, double * dst )
```

参照元 `cdm_Array::interp_coarse()`.

```
7.2.1.3 void cdm_interp_nijk_r4_ ( const int * szS, const int * gcS, const int * szD, const int * gcD, const int * nvari, float *
    src, float * dst )
```

参照元 `cdm_Array::interp_coarse()`.

```
7.2.1.4 void cdm_interp_nijk_r8_ ( const int * szS, const int * gcS, const int * szD, const int * gcD, const int * nvari, double
    * src, double * dst )
```

参照元 `cdm_Array::interp_coarse()`.

## 7.3 cdm\_Array\_inline.h

```
#include "cdm_Array.h"
#include <typeinfo>
```

`cdm_Array_inline.h` のインクルード依存関係図このグラフは、どのファイルから直接、間接的にインクルードされているかを示しています。

### マクロ定義

- `#define CDM_INLINE inline`
- `#define CDM_MEMFUN(rettype) CDM_INLINE rettype`

#### 7.3.1 マクロ定義

```
7.3.1.1 #define CDM_INLINE inline
```

`cdm_Array_inline.h` の 20 行で定義されています。

```
7.3.1.2 #define CDM_MEMFUN( rettype ) CDM_INLINE rettype
```

`cdm_Array_inline.h` の 25 行で定義されています。

## 7.4 cdm\_Define.h

CDM の定義マクロ記述ヘッダーファイル

```
#include "mpi.h"
```

`cdm_Define.h` のインクルード依存関係図このグラフは、どのファイルから直接、間接的にインクルードされているかを示しています。

### ネームスペース

- `CDM`

## マクロ定義

- #define D\_CDM\_DFITYPE\_CARTESIAN "Cartesian"
- #define D\_CDM\_DFITYPE\_NON\_UNIFORM\_CARTESIAN "Non\_Uniform\_Cartesian"
- #define D\_CDM\_EXT\_SPH "sph"
- #define D\_CDM\_EXT\_BOV "bov"
- #define D\_CDM\_EXT\_FUNC "fun"
- #define D\_CDM\_EXT\_VTK "vtk"
- #define D\_CDM\_EXT\_NC "nc"
- #define D\_CDM\_EXT\_FUB "fub"
- #define D\_CDM\_EXT\_XYZ "xyz"
- #define D\_CDM\_EXT\_BOV\_DATAFILE "dat"
- #define D\_CDM\_ON "on"
- #define D\_CDM\_OFF "off"
- #define D\_CDM\_INT8 "Int8"
- #define D\_CDM\_INT16 "Int16"
- #define D\_CDM\_INT32 "Int32"
- #define D\_CDM\_INT64 "Int64"
- #define D\_CDM\_UINT8 "UInt8"
- #define D\_CDM\_UINT16 "UInt16"
- #define D\_CDM\_UINT32 "UInt32"
- #define D\_CDM\_UINT64 "UInt64"
- #define D\_CDM\_FLOAT32 "Float32"
- #define D\_CDM\_FLOAT64 "Float64"
- #define D\_CDM\_BYTE "BYTE"
- #define D\_CDM\_INT "INT"
- #define D\_CDM\_FLOAT "FLOAT"
- #define D\_CDM\_DOUBLE "DOUBLE"
- #define D\_CDM\_IJNK "ijkn"
- #define D\_CDM\_NIJK "nijk"
- #define D\_CDM\_LITTLE "little"
- #define D\_CDM\_BIG "big"
- #define \_CDM\_TAB\_STR " "
- #define \_CDM\_IDX\_IJK(\_I, \_J, \_K, \_NI, \_NJ, \_NK, \_VC)
- #define \_CDM\_IDX\_IJ(\_I, \_J, \_NI, \_NJ, \_VC)
- #define \_CDM\_IDX\_NIJ(\_N, \_I, \_J, \_NI, \_NJ, \_NN, \_VC)
- #define \_CDM\_IDX\_IJKN(\_I, \_J, \_K, \_N, \_NI, \_NJ, \_NK, \_VC)
- #define \_CDM\_IDX\_NIJK(\_N, \_I, \_J, \_K, \_NN, \_NI, \_NJ, \_NK, \_VC)
- #define \_CDM\_WRITE\_TAB(\_FP, \_NTAB)
- #define stmpd\_printf printf("%s (%d): ", \_\_FILE\_\_, \_\_LINE\_\_), printf

## 列挙型

- enum CDM::E\_CDM\_DFITYPE { CDM::E\_CDM\_DFITYPE\_UNKNOWN = -1, CDM::E\_CDM\_DFITYPE\_CARTESIAN, CDM::E\_CDM\_DFITYPE\_NON\_UNIFORM\_CARTESIAN }
- enum CDM::E\_CDM\_FORMAT { CDM::E\_CDM\_FMT\_UNKNOWN = -1, CDM::E\_CDM\_FMT\_SPH, CDM::E\_CDM\_FMT\_BOV, CDM::E\_CDM\_FMT\_AVS, CDM::E\_CDM\_FMT\_PLOT3D, CDM::E\_CDM\_FMT\_VTK, CDM::E\_CDM\_FMT\_NETCDF4, CDM::E\_CDM\_FMT\_FUB, CDM::E\_CDM\_FMT\_FUB\_COD }
- enum CDM::E\_CDM\_ONOFF { CDM::E\_CDM\_OFF = 0, CDM::E\_CDM\_ON }
- enum CDM::E\_CDM\_DTYPE { CDM::E\_CDM\_DTYPE\_UNKNOWN = 0, CDM::E\_CDM\_INT8, CDM::E\_CDM\_INT16, CDM::E\_CDM\_INT32, CDM::E\_CDM\_INT64, CDM::E\_CDM\_UINT8, CDM::E\_CDM\_UINT16, CDM::E\_CDM\_UINT32, CDM::E\_CDM\_UINT64, CDM::E\_CDM\_FLOAT32, CDM::E\_CDM\_FLOAT64 }

- enum CDM::E\_CDM\_ARRAYSHAPE { CDM::E\_CDM\_ARRAYSHAPE\_UNKNOWN = -1, CDM::E\_CDM\_IJKN = 0, CDM::E\_CDM\_NIJK }
- enum CDM::E\_CDM\_ENDIANTYPE { CDM::E\_CDM\_ENDIANTYPE\_UNKNOWN = -1, CDM::E\_CDM\_LITTLE = 0, CDM::E\_CDM\_BIG }
- enum CDM::E\_CDM\_READTYPE { CDM::E\_CDM\_SAMEDIV\_SAMERES = 1, CDM::E\_CDM\_SAMEDIV\_REFINEMENT, CDM::E\_CDM\_DIFFDIV\_SAMERES, CDM::E\_CDM\_DIFFDIV\_REFINEMENT, CDM::E\_CDM\_READTYPE\_UNKNOWN }
- enum CDM::E\_CDM\_FILE\_TYPE { CDM::E\_CDM\_FILE\_TYPE\_DEFAULT = -1, CDM::E\_CDM\_FILE\_TYPE\_ASCII = 0, CDM::E\_CDM\_FILE\_TYPE\_BINARY, CDM::E\_CDM\_FILE\_TYPE\_FBINARITY }
- enum CDM::E\_CDM\_OUTPUT\_FNAME { CDM::E\_CDM\_FNAME\_DEFAULT = -1, CDM::E\_CDM\_FNAME\_STEP\_RANK = 0, CDM::E\_CDM\_FNAME\_RANK\_STEP, CDM::E\_CDM\_FNAME\_RANK, CDM::E\_CDM\_FNAME\_CUSTOM }
- enum CDM::E\_CDM\_ERRORCODE { CDM::E\_CDM\_SUCCESS = 1, CDM::E\_CDM\_ERROR = -1, CDM::E\_CDM\_ERROR\_READ\_DFI\_GLOBALORIGIN = 1000, CDM::E\_CDM\_ERROR\_READ\_DFI\_GLOBALREGION = 1001, CDM::E\_CDM\_ERROR\_READ\_DFI\_GLOBALVOXEL = 1002, CDM::E\_CDM\_ERROR\_READ\_DFI\_GLOBALDIVISION = 1003, CDM::E\_CDM\_ERROR\_READ\_DFI\_DIRECTORYPATH = 1004, CDM::E\_CDM\_ERROR\_READ\_DFI\_TIMESLICEDIVISION = 1005, CDM::E\_CDM\_ERROR\_READ\_DFI\_PREFIX = 1006, CDM::E\_CDM\_ERROR\_READ\_DFI\_FILEFORMAT = 1007, CDM::E\_CDM\_ERROR\_READ\_DFI\_GUIDECCELL = 1008, CDM::E\_CDM\_ERROR\_READ\_DFI\_DATATYPE = 1009, CDM::E\_CDM\_ERROR\_READ\_DFI\_ENDIAN = 1010, CDM::E\_CDM\_ERROR\_READ\_DFI\_NUMVARIABLES = 1012, CDM::E\_CDM\_ERROR\_READ\_DFI\_FILEPATH\_PROCESS = 1013, CDM::E\_CDM\_ERROR\_READ\_DFI\_NO\_RANK = 1014, CDM::E\_CDM\_ERROR\_READ\_DFI\_ID = 1015, CDM::E\_CDM\_ERROR\_READ\_DFI\_HOSTNAME = 1016, CDM::E\_CDM\_ERROR\_READ\_DFI\_VOXELSIZE = 1017, CDM::E\_CDM\_ERROR\_READ\_DFI\_HEADINDEX = 1018, CDM::E\_CDM\_ERROR\_READ\_DFI\_TAILINDEX = 1019, CDM::E\_CDM\_ERROR\_READ\_DFI\_CELLID = 1020, CDM::E\_CDM\_ERROR\_READ\_DFI\_BCFLAGID = 1021, CDM::E\_CDM\_ERROR\_READ\_DFI\_NO\_SLICE = 1022, CDM::E\_CDM\_ERROR\_READ\_DFI\_STEP = 1023, CDM::E\_CDM\_ERROR\_READ\_DFI\_TIME = 1024, CDM::E\_CDM\_ERROR\_READ\_DFI\_NO\_MINMAX = 1025, CDM::E\_CDM\_ERROR\_READ\_DFI\_MIN = 1026, CDM::E\_CDM\_ERROR\_READ\_DFI\_MAX = 1027, CDM::E\_CDM\_ERROR\_READ\_DFI\_DFITYPE = 1028, CDM::E\_CDM\_ERROR\_READ\_DFI\_FIELDFILENAMEFORMAT = 1029, CDM::E\_CDM\_ERROR\_READ\_DFI\_COORDINATEFILE = 1030, CDM::E\_CDM\_ERROR\_READ\_DFI\_COORDINATEFILETYPE = 1031, CDM::E\_CDM\_ERROR\_READ\_DFI\_COORDINATEFILEENDIAN = 1033, CDM::E\_CDM\_ERROR\_OPEN\_COORDINATEFILE = 1034, CDM::E\_CDM\_ERROR\_READ\_COORDINATEFILE = 1035, CDM::E\_CDM\_ERROR\_READ\_INDEXFILE\_OPENERORR = 1050, CDM::E\_CDM\_ERROR\_TEXTPARSER = 1051, CDM::E\_CDM\_ERROR\_READ\_FILEINFO = 1052, CDM::E\_CDM\_ERROR\_READ\_FILEPATH = 1053, CDM::E\_CDM\_ERROR\_READ\_UNIT = 1054, CDM::E\_CDM\_ERROR\_READ\_TIMESLICE = 1055, CDM::E\_CDM\_ERROR\_READ\_PROCFILE\_OPENERORR = 1056, CDM::E\_CDM\_ERROR\_READ\_DOMAIN = 1057, CDM::E\_CDM\_ERROR\_READ\_MPI = 1058, CDM::E\_CDM\_ERROR\_READ\_NETCDF = 1059, CDM::E\_CDM\_ERROR\_READ\_DFI\_NETCDF = 1060, CDM::E\_CDM\_ERROR\_READ\_NETCDF\_MISMATCH\_TYPE = 1061, CDM::E\_CDM\_ERROR\_READ\_FILELIST\_ID = 1062, CDM::E\_CDM\_ERROR\_READ\_FILELIST\_COORDINATEFILENAME = 1063, CDM::E\_CDM\_ERROR\_READ\_FILELIST\_COORDINATEFILETYPE = 1064, CDM::E\_CDM\_ERROR\_UNDEFINED\_FIELDFILENAMEFORMAT = 1065, CDM::E\_CDM\_ERROR\_READ\_FIELDDATA\_FILE = 1900, CDM::E\_CDM\_ERROR\_READ\_SPH\_FILE = 2000, CDM::E\_CDM\_ERROR\_READ\_SPH\_REC1 = 2001, CDM::E\_CDM\_ERROR\_READ\_SPH\_REC2 = 2002, CDM::E\_CDM\_ERROR\_READ\_SPH\_REC3 = 2003, CDM::E\_CDM\_ERROR\_READ\_SPH\_REC4 = 2004, CDM::E\_CDM\_ERROR\_READ\_SPH\_REC5 = 2005, CDM::E\_CDM\_ERROR\_READ\_SPH\_REC6 = 2006, CDM::E\_CDM\_ERROR\_READ\_SPH\_REC7 = 2007, CDM::E\_CDM\_ERROR\_UNMATCH\_VOXELSIZE = 2050, CDM::E\_CDM\_ERROR\_NOMATCH\_ENDIAN =

```

2051,
CDM::E_CDM_ERROR_UNMATCH_NUM_OF_VARIABLES = 2052, CDM::E_CDM_ERROR_NUM_OF_GUIDECCELLS
= 2053, CDM::E_CDM_ERROR_READ_BOV_FILE = 2100, CDM::E_CDM_ERROR_READ_FIELD_HEADER_RECORD
= 2102,
CDM::E_CDM_ERROR_READ_FIELD_DATA_RECORD = 2103, CDM::E_CDM_ERROR_READ_FIELD_AVERAGED_RECOR
= 2104, CDM::E_CDM_ERROR_READ_DFI_FILELIST = 2105, CDM::E_CDM_ERROR_READ_NETCDF_FUNC
= 2200,
CDM::E_CDM_ERROR_READ_NETCDF_VAR_1D = 2201, CDM::E_CDM_ERROR_READ_FUB_REC1 =
2210, CDM::E_CDM_ERROR_READ_FUB_REC2 = 2211, CDM::E_CDM_ERROR_MISMATCH_NP_SUBDOMAIN
= 3003,
CDM::E_CDM_ERROR_INVALID_DIVNUM = 3011, CDM::E_CDM_ERROR_OPEN_SBDM = 3012,
CDM::E_CDM_ERROR_READ_SBDM_HEADER = 3013, CDM::E_CDM_ERROR_READ_SBDM_FORMAT
= 3014,
CDM::E_CDM_ERROR_READ_SBDM_DIV = 3015, CDM::E_CDM_ERROR_READ_SBDM_CONTENTS =
3016, CDM::E_CDM_ERROR_SBDM_NUMDOMAIN_ZERO = 3017, CDM::E_CDM_ERROR_MAKEDIRECTORY
= 3100,
CDM::E_CDM_ERROR_OPEN_FIELDDATA = 3101, CDM::E_CDM_ERROR_WRITE_FIELD_HEADER_RECORD
= 3102, CDM::E_CDM_ERROR_WRITE_FIELD_DATA_RECORD = 3103, CDM::E_CDM_ERROR_WRITE_FIELD_AVERAGED
= 3104,
CDM::E_CDM_ERROR_WRITE_SPH_REC1 = 3201, CDM::E_CDM_ERROR_WRITE_SPH_REC2 = 3202,
CDM::E_CDM_ERROR_WRITE_SPH_REC3 = 3203, CDM::E_CDM_ERROR_WRITE_SPH_REC4 = 3204,
CDM::E_CDM_ERROR_WRITE_SPH_REC5 = 3205, CDM::E_CDM_ERROR_WRITE_SPH_REC6 = 3206,
CDM::E_CDM_ERROR_WRITE_SPH_REC7 = 3207, CDM::E_CDM_ERROR_WRITE_PROCFilename_EMPTY
= 3500,
CDM::E_CDM_ERROR_WRITE_PROCFILE_OPENERERROR = 3501, CDM::E_CDM_ERROR_WRITE_DOMAIN
= 3502, CDM::E_CDM_ERROR_WRITE_MPI = 3503, CDM::E_CDM_ERROR_WRITE_PROCESS = 3504,
CDM::E_CDM_ERROR_WRITE_RANKID = 3505, CDM::E_CDM_ERROR_WRITE_INDEXFILENAME_EMPTY
= 3510, CDM::E_CDM_ERROR_WRITE_PREFIX_EMPTY = 3511, CDM::E_CDM_ERROR_WRITE_INDEXFILE_OPENERRO
= 3512,
CDM::E_CDM_ERROR_WRITE_FILEINFO = 3513, CDM::E_CDM_ERROR_WRITE_UNIT = 3514,
CDM::E_CDM_ERROR_WRITE_TIMESLICE = 3515, CDM::E_CDM_ERROR_WRITE_FILEPATH = 3516,
CDM::E_CDM_ERROR_WRITE_VISIT = 3517, CDM::E_CDM_ERROR_WRITE_GRIDFILE = 3518,
CDM::E_CDM_ERROR_WRITE_DFI_NETCDF = 3600, CDM::E_CDM_WARN_GETUNIT = 4000 }

```

## 変数

- static const char \* CDM::C\_CDM\_RANKNOPREFIX = "\_id"  
ファイル名のランク番号前の文字列

### 7.4.1 説明

CDM の定義マクロ記述ヘッダーファイル

作者

aics

cdm\_Define.h で定義されています。

### 7.4.2 マクロ定義

7.4.2.1 #define \_CDM\_IDX\_IJ( \_I, \_J, \_NI, \_NJ, \_VC )

値:

```
( (long long) ((_J)+(_VC)) * (long long) ((_NI)+2*(_VC)) \
+ (long long) ((_I)+(_VC)) \
)
```

## 2 次元 ( スカラー ) インデクス (i,j) -> 1 次元インデクス変換マクロ

引数

in	<code>_I</code>	i 方向インデクス
in	<code>_J</code>	j 方向インデクス
in	<code>_NI</code>	i 方向インデクスサイズ
in	<code>_NJ</code>	j 方向インデクスサイズ
in	<code>_VC</code>	仮想セル数

戻り値

1 次元インデクス

cdm\_Define.h の 328 行で定義されています。

**7.4.2.2** `#define _CDM_IDX_IJK( _I, _J, _K, _NI, _NJ, _NK, _VC )`

値:

```
( (long long) ((_K)+(_VC)) * (long long) ((_NI)+2*(_VC)) * (long long) ((_NJ)+2*(_VC)) \
+ (long long) ((_J)+(_VC)) * (long long) ((_NI)+2*(_VC)) \
+ (long long) ((_I)+(_VC)) \
)
```

## 3 次元 ( スカラー ) インデクス (i,j,k) -> 1 次元インデクス変換マクロ

引数

in	<code>_I</code>	i 方向インデクス
in	<code>_J</code>	j 方向インデクス
in	<code>_K</code>	k 方向インデクス
in	<code>_NI</code>	i 方向インデクスサイズ
in	<code>_NJ</code>	j 方向インデクスサイズ
in	<code>_NK</code>	k 方向インデクスサイズ
in	<code>_VC</code>	仮想セル数

戻り値

1 次元インデクス

cdm\_Define.h の 314 行で定義されています。

**7.4.2.3** `#define _CDM_IDX_IJKN( _I, _J, _K, _N, _NI, _NJ, _NK, _VC )`

値:

```
( (long long) (_N) * (long long) ((_NI)+2*(_VC)) * (long long) ((_NJ)+2*(_VC)) \
* (long long) ((_NK)+2*(_VC)) \
+ _CDM_IDX_IJK(_I, _J, _K, _NI, _NJ, _NK, _VC) \
)
```

## 3 次元 ( ベクトル ) インデクス (i,j,k,n) -> 1 次元インデクス変換マクロ



引数

in	<code>_I</code>	i 方向インデクス
in	<code>_J</code>	j 方向インデクス
in	<code>_K</code>	k 方向インデクス
in	<code>_N</code>	変数インデクス
in	<code>_NI</code>	i 方向インデクスサイズ
in	<code>_NJ</code>	j 方向インデクスサイズ
in	<code>_NK</code>	k 方向インデクスサイズ
in	<code>_VC</code>	仮想セル数

戻り値

1 次元インデクス

cdm\_Define.h の 360 行で定義されています。

7.4.2.4 `#define _CDM_IDX_IJ( _N, _I, _J, _NI, _NJ, _NN, _VC )`

値:

```
( (long long) (_NN) * _CDM_IDX_IJ(_I, _J, _NI, _NJ, _VC) \
+ (long long) (_N) \
)
```

2 次元 ( スカラー ) インデクス (n,i,j) -> 1 次元インデクス変換マクロ

引数

in	<code>_N</code>	変数インデクス
in	<code>_I</code>	i 方向インデクス
in	<code>_J</code>	j 方向インデクス
in	<code>_NI</code>	i 方向インデクスサイズ
in	<code>_NJ</code>	j 方向インデクスサイズ
in	<code>_NN</code>	変数の個数
in	<code>_VC</code>	仮想セル数

戻り値

1 次元インデクス

cdm\_Define.h の 343 行で定義されています。

7.4.2.5 `#define _CDM_IDX_IJK( _N, _I, _J, _K, _NN, _NI, _NJ, _NK, _VC )`

値:

```
( (long long) (_NN) * _CDM_IDX_IJK(_I, _J, _K, _NI, _NJ, _NK, _VC) \
+ (long long) (_N) )
```

3 次元 ( ベクトル ) インデクス (n,i,j,k) -> 1 次元インデクス変換マクロ

引数

in	<code>_N</code>	変数インデクス
in	<code>_I</code>	i 方向インデクス
in	<code>_J</code>	j 方向インデクス
in	<code>_K</code>	k 方向インデクス
in	<code>_NN</code>	変数の個数
in	<code>_NI</code>	i 方向インデクスサイズ
in	<code>_NJ</code>	j 方向インデクスサイズ
in	<code>_NK</code>	k 方向インデクスサイズ
in	<code>_VC</code>	仮想セル数

戻り値

1 次元インデクス

cdm\_Define.h の 378 行で定義されています。

7.4.2.6 `#define _CDM_TAB_STR " "`

cdm\_Define.h の 67 行で定義されています。

7.4.2.7 `#define _CDM_WRITE_TAB( _FP, _NTAB )`

値:

```
{\
  for(int _NTCNT=0; _NTCNT<_NTAB; _NTCNT++) fprintf(_FP,_CDM_TAB_STR); \
}
```

DFI ファイルのTab 出力

引数

in	<code>_FP</code>	ファイルポインタ
in	<code>_NTAB</code>	インデント数

cdm\_Define.h の 386 行で定義されています。

参照元 `cdm_NonUniformDomain< T >::Write()`.

7.4.2.8 `#define D_CDM_BIG "big"`

cdm\_Define.h の 65 行で定義されています。

7.4.2.9 `#define D_CDM_BYTE "BYTE"`

cdm\_Define.h の 56 行で定義されています。

7.4.2.10 `#define D_CDM_DFITYPE_CARTESIAN "Cartesian"`

cdm\_Define.h の 24 行で定義されています。

7.4.2.11 `#define D_CDM_DFITYPE_NON_UNIFORM_CARTESIAN "Non_Uniform_Cartesian"`

cdm\_Define.h の 25 行で定義されています。

#### 7.4.2.12 #define D\_CDM\_DOUBLE "DOUBLE"

cdm\_Define.h の 59 行で定義されています。

#### 7.4.2.13 #define D\_CDM\_EXT\_BOV "bov"

cdm\_Define.h の 28 行で定義されています。

#### 7.4.2.14 #define D\_CDM\_EXT\_BOV\_DATAFILE "dat"

cdm\_Define.h の 40 行で定義されています。

#### 7.4.2.15 #define D\_CDM\_EXT\_FUB "fub"

cdm\_Define.h の 35 行で定義されています。

#### 7.4.2.16 #define D\_CDM\_EXT\_FUNC "fun"

cdm\_Define.h の 29 行で定義されています。

#### 7.4.2.17 #define D\_CDM\_EXT\_NC "nc"

cdm\_Define.h の 32 行で定義されています。

#### 7.4.2.18 #define D\_CDM\_EXT\_SPH "sph"

cdm\_Define.h の 27 行で定義されています。

#### 7.4.2.19 #define D\_CDM\_EXT\_VTK "vtk"

cdm\_Define.h の 30 行で定義されています。

#### 7.4.2.20 #define D\_CDM\_EXT\_XYZ "xyz"

cdm\_Define.h の 36 行で定義されています。

#### 7.4.2.21 #define D\_CDM\_FLOAT "FLOAT"

cdm\_Define.h の 58 行で定義されています。

#### 7.4.2.22 #define D\_CDM\_FLOAT32 "Float32"

cdm\_Define.h の 53 行で定義されています。

#### 7.4.2.23 #define D\_CDM\_FLOAT64 "Float64"

cdm\_Define.h の 54 行で定義されています。

#### 7.4.2.24 `#define D_CDM_IJNK "ijkn"`

cdm\_Define.h の 61 行で定義されています。

#### 7.4.2.25 `#define D_CDM_INT "INT"`

cdm\_Define.h の 57 行で定義されています。

#### 7.4.2.26 `#define D_CDM_INT16 "Int16"`

cdm\_Define.h の 46 行で定義されています。

#### 7.4.2.27 `#define D_CDM_INT32 "Int32"`

cdm\_Define.h の 47 行で定義されています。

#### 7.4.2.28 `#define D_CDM_INT64 "Int64"`

cdm\_Define.h の 48 行で定義されています。

#### 7.4.2.29 `#define D_CDM_INT8 "Int8"`

cdm\_Define.h の 45 行で定義されています。

#### 7.4.2.30 `#define D_CDM_LITTLE "little"`

cdm\_Define.h の 64 行で定義されています。

#### 7.4.2.31 `#define D_CDM_NIJK "nijk"`

cdm\_Define.h の 62 行で定義されています。

#### 7.4.2.32 `#define D_CDM_OFF "off"`

cdm\_Define.h の 43 行で定義されています。

#### 7.4.2.33 `#define D_CDM_ON "on"`

cdm\_Define.h の 42 行で定義されています。

#### 7.4.2.34 `#define D_CDM_UINT16 "UInt16"`

cdm\_Define.h の 50 行で定義されています。

#### 7.4.2.35 `#define D_CDM_UINT32 "UInt32"`

cdm\_Define.h の 51 行で定義されています。

7.4.2.36 `#define D_CDM_UINT64 "UInt64"`

cdm\_Define.h の 52 行で定義されています。

7.4.2.37 `#define D_CDM_UINT8 "UInt8"`

cdm\_Define.h の 49 行で定義されています。

7.4.2.38 `#define stmpd_printf printf("%s (%d): ", __FILE__, __LINE__), printf`

cdm\_Define.h の 391 行で定義されています。

## 7.5 cdm\_DFI.h

[cdm\\_DFI](#) Class Header

```
#include "cdm_Define.h"
#include <stdlib.h>
#include <errno.h>
#include <sys/stat.h>
#include <set>
#include <map>
#include <string>
#include "cdm_Version.h"
#include "cdm_PathUtil.h"
#include "cdm_TextParser.h"
#include "cdm_ActiveSubDomain.h"
#include "cdm_endianUtil.h"
#include "cdm_TypeArray.h"
#include "cdm_FileInfo.h"
#include "cdm_FilePath.h"
#include "cdm_Visit.h"
#include "cdm_Unit.h"
#include "cdm_TimeSlice.h"
#include "cdm_Domain.h"
#include "cdm_MPI.h"
#include "cdm_Process.h"
#include "cdm_FieldFileNameFormat.h"
#include "cdm_FILE.h"
#include "inline/cdm_DFI_inline.h"
```

cdm\_DFI.h のインクルード依存関係図このグラフは、どのファイルから直接、間接的にインクルードされているかを示しています。

### 構成

- class [cdm\\_DFI](#)

#### 7.5.1 説明

[cdm\\_DFI](#) Class Header

作者

aics

[cdm\\_DFI.h](#) で定義されています。

## 7.6 cdm\_DFI\_AVS.h

[cdm\\_DFI\\_AVS](#) Class Header

```
#include "cdm_DFI.h"
```

cdm\_DFI\_AVS.h のインクルード依存関係図このグラフは、どのファイルから直接、間接的にインクルードされているかを示しています。

構成

- class [cdm\\_DFI\\_AVS](#)

### 7.6.1 説明

[cdm\\_DFI\\_AVS](#) Class Header

作者

aics

[cdm\\_DFI\\_AVS.h](#) で定義されています。

## 7.7 cdm\_DFI\_BOV.h

[cdm\\_DFI\\_BOV](#) Class Header

```
#include "cdm_DFI.h"
```

cdm\_DFI\_BOV.h のインクルード依存関係図このグラフは、どのファイルから直接、間接的にインクルードされているかを示しています。

構成

- class [cdm\\_DFI\\_BOV](#)

### 7.7.1 説明

[cdm\\_DFI\\_BOV](#) Class Header

作者

aics

[cdm\\_DFI\\_BOV.h](#) で定義されています。

## 7.8 cdm\_DFI\_FUB.h

[cdm\\_DFI\\_FUB](#) Class Header

```
#include "cdm_DFI.h"
```

cdm\_DFI\_FUB.h のインクルード依存関係図このグラフは、どのファイルから直接、間接的にインクルードされているかを示しています。

### 構成

- class [cdm\\_DFI\\_FUB](#)

### 7.8.1 説明

[cdm\\_DFI\\_FUB](#) Class Header

作者

aics

[cdm\\_DFI\\_FUB.h](#) で定義されています。

## 7.9 cdm\_DFI\_inline.h

```
#include "cdm_DFI_SPH.h"
#include "cdm_DFI_BOV.h"
#include "cdm_DFI_AVS.h"
#include "cdm_DFI_PLOT3D.h"
#include "cdm_DFI_VTK.h"
#include "cdm_DFI_NETCDF.h"
#include "cdm_DFI_FUB.h"
#include "cdm_NonUniformDomain.h"
#include <typeinfo>
```

cdm\_DFI\_inline.h のインクルード依存関係図このグラフは、どのファイルから直接、間接的にインクルードされているかを示しています。

### マクロ定義

- #define [CDM\\_INLINE](#) inline

### 7.9.1 マクロ定義

7.9.1.1 #define [CDM\\_INLINE](#) inline

cdm\_DFI\_inline.h の 37 行で定義されています。

## 7.10 cdm\_DFI\_NETCDF.h

[cdm\\_DFI\\_NETCDF](#) Class Header

```
#include "cdm_DFI.h"
#include "netcdf.h"
```

cdm\_DFI\_NETCDF.h のインクルード依存関係図このグラフは、どのファイルから直接、間接的にインクルードされているかを示しています。

## 構成

- class [cdm\\_DFI\\_NETCDF](#)
- struct [cdm\\_DFI\\_NETCDF::stDimInfo](#)
- struct [cdm\\_DFI\\_NETCDF::stVarInfo](#)

### 7.10.1 説明

[cdm\\_DFI\\_NETCDF](#) Class Header

作者

aics

[cdm\\_DFI\\_NETCDF.h](#) で定義されています。

## 7.11 cdm\_DFI\_PLOT3D.h

[cdm\\_DFI\\_PLOT3D](#) Class Header

```
#include "cdm_DFI.h"
#include "inline/cdm_Plot3d_inline.h"
```

cdm\_DFI\_PLOT3D.h のインクルード依存関係図このグラフは、どのファイルから直接、間接的にインクルードされているかを示しています。

## 構成

- class [cdm\\_DFI\\_PLOT3D](#)

### 7.11.1 説明

[cdm\\_DFI\\_PLOT3D](#) Class Header

作者

aics

[cdm\\_DFI\\_PLOT3D.h](#) で定義されています。

## 7.12 cdm\_DFI\_SPH.h

[cdm\\_DFI\\_SPH](#) Class Header

```
#include "cdm_DFI.h"
```

cdm\_DFI\_SPH.h のインクルード依存関係図このグラフは、どのファイルから直接、間接的にインクルードされているかを示しています。



## 構成

- class [cdm\\_DFI\\_SPH](#)

### 7.12.1 説明

[cdm\\_DFI\\_SPH](#) Class Header

作者

aics

[cdm\\_DFI\\_SPH.h](#) で定義されています。

## 7.13 cdm\_DFI\_VTK.h

[cdm\\_DFI\\_VTK](#) Class Header

```
#include "cdm_DFI.h"
```

cdm\_DFI\_VTK.h のインクルード依存関係図このグラフは、どのファイルから直接、間接的にインクルードされているかを示しています。

## 構成

- class [cdm\\_DFI\\_VTK](#)

### 7.13.1 説明

[cdm\\_DFI\\_VTK](#) Class Header

作者

aics

[cdm\\_DFI\\_VTK.h](#) で定義されています。

## 7.14 cdm\_Domain.h

[cdm\\_Domain](#) Class Header

このグラフは、どのファイルから直接、間接的にインクルードされているかを示しています。

## 構成

- class [cdm\\_Domain](#)

### 7.14.1 説明

[cdm\\_Domain](#) Class Header

作者

aics

[cdm\\_Domain.h](#) で定義されています。

## 7.15 cdm\_endianUtil.h

エンディアンユーティリティマクロ・関数ファイル

```
#include <stdio.h>
#include <sys/types.h>
#include <unistd.h>
#include <iostream>
#include <string>
```

cdm\_endianUtil.h のインクルード依存関係図このグラフは、どのファイルから直接、間接的にインクルードされているかを示しています。

### マクロ定義

- #define [CDM\\_INLINE](#) inline
- #define [BSWAP\\_X\\_16](#)(x)
- #define [BSWAP16](#)(x)
- #define [BSWAP\\_X\\_32](#)(x)
- #define [BSWAP32](#)(x)
- #define [BSWAP\\_X\\_64](#)(x)
- #define [BSWAP64](#)(x)
- #define [SBSWAPVEC](#)(a, n)
- #define [BSWAPVEC](#)(a, n)
- #define [DBSWAPVEC](#)(a, n)

### 7.15.1 説明

エンディアンユーティリティマクロ・関数ファイル

作者

aics

[cdm\\_endianUtil.h](#) で定義されています。

### 7.15.2 マクロ定義

#### 7.15.2.1 #define BSWAP16( x )

値:

```
{ \
    register unsigned short& _x_v = (unsigned short&)(x); \
    _x_v = BSWAP_X_16(_x_v); }
```

cdm\_endianUtil.h の 46 行で定義されています。

## 7.15.2.2 #define BSWAP32( x )

値:

```
{register unsigned int& _x_v = (unsigned int&)(x); \
  _x_v = BSWAP_X_32(_x_v);}
```

cdm\_endianUtil.h の 70 行で定義されています。

## 7.15.2.3 #define BSWAP64( x )

値:

```
{register unsigned long long& _x_v = (unsigned long long&)(x); \
  _x_v = BSWAP_X_64(_x_v);}
```

cdm\_endianUtil.h の 104 行で定義されています。

## 7.15.2.4 #define BSWAP\_X\_16( x )

値:

```
( ((x) & 0xff00) >> 8) \
| ((x) & 0x00ff) << 8)
```

cdm\_endianUtil.h の 43 行で定義されています。

## 7.15.2.5 #define BSWAP\_X\_32( x )

値:

```
( ((x) & 0xff000000) >> 24) \
| ((x) & 0x00ff0000) >> 8) \
| ((x) & 0x0000ff00) << 8) \
| ((x) & 0x000000ff) << 24)
```

cdm\_endianUtil.h の 65 行で定義されています。

## 7.15.2.6 #define BSWAP\_X\_64( x )

値:

```
( ((x) & 0xff00000000000000ull) >> 56) \
| ((x) & 0x00ff000000000000ull) >> 40) \
| ((x) & 0x0000ff0000000000ull) >> 24) \
| ((x) & 0x000000ff00000000ull) >> 8) \
| ((x) & 0x00000000ff000000ull) << 8) \
| ((x) & 0x0000000000ff0000ull) << 24) \
| ((x) & 0x000000000000ff00ull) << 40) \
| ((x) & 0x00000000000000ffull) << 56)
```

cdm\_endianUtil.h の 95 行で定義されています。

## 7.15.2.7 #define BSWAPVEC( a, n )

値:

```
do{\
    for(register unsigned int _i=0;_i<(n);_i++){BSWAP32(a[_i]);}\
}while(0)
```

cdm\_endianUtil.h の 139 行で定義されています。

参照元 cdm\_TypeArray< T >::readBinary(), cdm\_NonUniformDomain< T >::readCoordData(), と cdm\_NonUniformDomain< T >::readCoordDataCount().

#### 7.15.2.8 #define CDM\_INLINE inline

cdm\_endianUtil.h の 28 行で定義されています。

#### 7.15.2.9 #define DBSWAPVEC( a, n )

値:

```
do{\
    for(register unsigned int _i=0;_i<(n);_i++){BSWAP64(a[_i]);}\
}while(0)
```

cdm\_endianUtil.h の 156 行で定義されています。

参照元 cdm\_TypeArray< T >::readBinary(), と cdm\_NonUniformDomain< T >::readCoordData().

#### 7.15.2.10 #define SBSWAPVEC( a, n )

値:

```
do{\
    for(register unsigned int _i=0;_i<(n);_i++){BSWAP16(a[_i]);}\
}while(0)
```

cdm\_endianUtil.h の 121 行で定義されています。

参照元 cdm\_TypeArray< T >::readBinary().

## 7.16 cdm\_FieldFileNameFormat.h

```
#include <map>
```

```
#include "cdm_FieldFileNameFormatElem.h"
```

cdm\_FieldFileNameFormat.h のインクルード依存関係図このグラフは、どのファイルから直接、間接的にインクルードされているかを示しています。

### 構成

- class [cdm\\_FieldFileNameFormat](#)

### マクロ定義

- #define [\\_CDM\\_FIELDFILENAMEFORMAT\\_H\\_](#)

#### 7.16.1 マクロ定義

##### 7.16.1.1 #define \_CDM\_FIELDFILENAMEFORMAT\_H\_

cdm\_FieldFileNameFormat.h の 2 行で定義されています。

## 7.17 cdm\_FieldFileNameFormatElem.h

```
#include <stdio.h>
#include <string.h>
#include "TextParser.h"
```

cdm\_FieldFileNameFormatElem.h のインクルード依存関係図このグラフは、どのファイルから直接、間接的にインクルードされているかを示しています。

### 構成

- class `cdm_FieldFileNameFormatElem`

### マクロ定義

- #define `_FUB_TAB_STR` " "
- #define `_FUB_WRITE_TAB(_FP, _NTAB)`

### 列挙型

- enum `FUB_FNAME_TYPE` {  
`E_FUB_NONE = -1, E_FUB_STEP_RANK = 0, E_FUB_RANK_STEP = 1, E_FUB_RANK = 2,`  
`E_FUB_STEP = 3 }`

#### 7.17.1 マクロ定義

##### 7.17.1.1 #define `_FUB_TAB_STR` " "

cdm\_FieldFileNameFormatElem.h の 16 行で定義されています。

##### 7.17.1.2 #define `_FUB_WRITE_TAB( _FP, _NTAB )`

値:

```
{\
  for(int _NTCNT=0; _NTCNT<_NTAB; _NTCNT++) fprintf(_FP,_FUB_TAB_STR); \
}
```

cdm\_FieldFileNameFormatElem.h の 17 行で定義されています。

#### 7.17.2 列挙型

##### 7.17.2.1 enum `FUB_FNAME_TYPE`

列挙型の値

```
E_FUB_NONE
E_FUB_STEP_RANK
E_FUB_RANK_STEP
E_FUB_RANK
E_FUB_STEP
```

cdm\_FieldFileNameFormatElem.h の 22 行で定義されています。

## 7.18 cdm\_FILE.h

[cdm\\_FILE](#) Class Header

```
#include "cdm_Define.h"
#include <stdio.h>
#include <stdlib.h>
#include <errno.h>
#include <sys/stat.h>
#include <string>
#include "netcdf.h"
```

cdm\_FILE.h のインクルード依存関係図このグラフは、どのファイルから直接、間接的にインクルードされているかを示しています。

### 構成

- class [cdm\\_FILE](#)

### 7.18.1 説明

[cdm\\_FILE](#) Class Header

作者

aics

[cdm\\_FILE.h](#) で定義されています。

## 7.19 cdm\_FileInfo.h

[cdm\\_FileInfo](#) Class Header

このグラフは、どのファイルから直接、間接的にインクルードされているかを示しています。

### 構成

- class [cdm\\_FileInfo](#)

### 7.19.1 説明

[cdm\\_FileInfo](#) Class Header

作者

aics

[cdm\\_FileInfo.h](#) で定義されています。

## 7.20 cdm\_FilePath.h

[cdm\\_FilePath](#) Class Header

このグラフは、どのファイルから直接、間接的にインクルードされているかを示しています。

## 構成

- class [cdm\\_FilePath](#)

### 7.20.1 説明

[cdm\\_FilePath](#) Class Header

作者

aics

[cdm\\_FilePath.h](#) で定義されています。

## 7.21 cdm\_interp\_ijkn.h

### 関数

- !CDMLib Cartesian Data Management library !Copyright (c) 2013-2015 Advanced Institute for Computational Science

### 7.21.1 関数

7.21.1.1 ! CDMLib Cartesian Data Management library !Copyright ( c )

## 7.22 cdm\_interp\_nijk.h

### 関数

- !CDMLib Cartesian Data Management library !Copyright (c) 2013-2015 Advanced Institute for Computational Science

### 7.22.1 関数

7.22.1.1 ! CDMLib Cartesian Data Management library !Copyright ( c )

## 7.23 cdm\_MPI.h

[cdm\\_MPI](#) Class Header

このグラフは、どのファイルから直接、間接的にインクルードされているかを示しています。

## 構成

- class [cdm\\_MPI](#)

### 7.23.1 説明

[cdm\\_MPI](#) Class Header

作者

aics

[cdm\\_MPI.h](#) で定義されています。

## 7.24 cdm\_NonUniformDomain.h

[cdm\\_NonUniformDomain](#) Class Header

```
#include "cdm_Domain.h"
```

```
#include <typeinfo>
```

```
#include "inline/cdm_NonUniformDomain_inline.h"
```

[cdm\\_NonUniformDomain.h](#) のインクルード依存関係図このグラフは、どのファイルから直接、間接的にインクルードされているかを示しています。

構成

- class [cdm\\_NonUniformDomain](#)< T >

### 7.24.1 説明

[cdm\\_NonUniformDomain](#) Class Header

作者

aics

[cdm\\_NonUniformDomain.h](#) で定義されています。

## 7.25 cdm\_NonUniformDomain\_inline.h

[cdm\\_NonUniformDomain](#) template Header

このグラフは、どのファイルから直接、間接的にインクルードされているかを示しています。

マクロ定義

- #define [CDM\\_INLINE](#) inline

### 7.25.1 説明

[cdm\\_NonUniformDomain](#) template Header

作者

aics

[cdm\\_NonUniformDomain\\_inline.h](#) で定義されています。



## 7.25.2 マクロ定義

### 7.25.2.1 #define CDM\_INLINE inline

cdm\_NonUniformDomain\_inline.h の 23 行で定義されています。

## 7.26 cdm\_PathUtil.h

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <sstream>
```

cdm\_PathUtil.h のインクルード依存関係図このグラフは、どのファイルから直接、間接的にインクルードされているかを示しています。

### ネームスペース

- [CDM](#)

### マクロ定義

- #define [MAXPATHLEN](#) 512

### 関数

- char [CDM::cdmPath\\_getDelimChar](#) ()
- std::string [CDM::cdmPath\\_getDelimString](#) ()
- bool [CDM::cdmPath\\_hasDrive](#) (const std::string &path)
- std::string [CDM::vfvPath\\_emitDrive](#) (std::string &path)
- bool [CDM::cdmPath\\_isAbsolute](#) (const std::string &path)
- std::string [CDM::cdmPath\\_DirName](#) (const std::string &path, const char dc=cdmPath\_getDelimChar())
- std::string [CDM::cdmPath\\_FileName](#) (const std::string &path, const std::string &addext=std::string(""), const char dc=cdmPath\_getDelimChar())
- std::string [CDM::cdmPath\\_ConnectPath](#) (std::string dirName, std::string fName)
- std::string [CDM::ExtractPathWithoutExt](#) (const std::string &fn)

## 7.26.1 マクロ定義

### 7.26.1.1 #define MAXPATHLEN 512

cdm\_PathUtil.h の 17 行で定義されています。

## 7.27 cdm\_Plot3d\_inline.h

このグラフは、どのファイルから直接、間接的にインクルードされているかを示しています。

### マクロ定義

- #define [CDM\\_INLINE](#) inline

### 7.27.1 マクロ定義

#### 7.27.1.1 #define CDM\_INLINE inline

cdm\_Plot3d\_inline.h の 23 行で定義されています。

## 7.28 cdm\_Process.h

cdm\_RANK & [cdm\\_Process](#) Class Header

このグラフは、どのファイルから直接、間接的にインクルードされているかを示しています。

### 構成

- class [cdm\\_Rank](#)
- class [cdm\\_Process](#)

#### 7.28.1 説明

cdm\_RANK & [cdm\\_Process](#) Class Header

作者

aics

[cdm\\_Process.h](#) で定義されています。

## 7.29 cdm\_TextParser.h

TextParser Control class Header.

```
#include "cdm_Define.h"
#include <math.h>
#include <iostream>
#include <string>
#include <vector>
#include <map>
#include "string.h"
#include "TextParser.h"
```

cdm\_TextParser.h のインクルード依存関係図このグラフは、どのファイルから直接、間接的にインクルードされているかを示しています。

### 構成

- class [cdm\\_TextParser](#)

#### 7.29.1 説明

TextParser Control class Header.

作者

aics

[cdm\\_TextParser.h](#) で定義されています。

## 7.30 cdm\_TimeSlice.h

[cdm\\_Slice](#) & [cdm\\_TimeSliceClass](#) Header

このグラフは、どのファイルから直接、間接的にインクルードされているかを示しています。

構成

- class [cdm\\_Slice](#)
- class [cdm\\_TimeSlice](#)

### 7.30.1 説明

[cdm\\_Slice](#) & [cdm\\_TimeSliceClass](#) Header

作者

aics

[cdm\\_TimeSlice.h](#) で定義されています。

## 7.31 cdm\_TypeArray.h

```
#include "cdm_Array.h"
```

```
#include "inline/cdm_Array_inline.h"
```

[cdm\\_TypeArray.h](#) のインクルード依存関係図このグラフは、どのファイルから直接、間接的にインクルードされているかを示しています。

構成

- class [cdm\\_TypeArray< T >](#)

## 7.32 cdm\_Unit.h

[cdm\\_UnitElem](#) & [cdm\\_Unit](#) Class Header

このグラフは、どのファイルから直接、間接的にインクルードされているかを示しています。

構成

- class [cdm\\_UnitElem](#)
- class [cdm\\_Unit](#)

### 7.32.1 説明

[cdm\\_UnitElem](#) & [cdm\\_Unit](#) Class Header

作者

aics

[cdm\\_Unit.h](#) で定義されています。

## 7.33 cdm\_Version.h

このグラフは、どのファイルから直接、間接的にインクルードされているかを示しています。

### マクロ定義

- `#define CDM_VERSION_NO "0.9.4"`
- `#define CDM_REVISION "20151212_2030"`

### 7.33.1 説明

バージョン情報のヘッダーファイル

[cdm\\_Version.h](#) で定義されています。

### 7.33.2 マクロ定義

7.33.2.1 `#define CDM_REVISION "20151212_2030"`

CDM ライブラリのリビジョン

[cdm\\_Version.h](#) の 21 行で定義されています。

7.33.2.2 `#define CDM_VERSION_NO "0.9.4"`

CDM ライブラリのバージョン

[cdm\\_Version.h](#) の 18 行で定義されています。

参照元 [cdm\\_DFI::getVersionInfo\(\)](#).

## 7.34 cdm\_VisIt.h

[cdm\\_VisIt](#) Class Header

このグラフは、どのファイルから直接、間接的にインクルードされているかを示しています。

### 構成

- class [cdm\\_VisIt](#)

### 7.34.1 説明

[cdm\\_VisIt](#) Class Header

作者

aics

[cdm\\_VisIt.h](#) で定義されています。

## 7.35 mpi\_stubs.h

### マクロ定義

- `#define MPI_COMM_WORLD 0`
- `#define MPI_INT 1`
- `#define MPI_CHAR 2`
- `#define MPI_SUCCESS true`

### 型定義

- `typedef int MPI_Comm`
- `typedef int MPI_Datatype`

### 関数

- `bool MPI_Init (int *argc, char ***argv)`
- `int MPI_Comm_rank (MPI_Comm comm, int *rank)`
- `int MPI_Comm_size (MPI_Comm comm, int *size)`
- `int MPI_Allgather (void *sendbuf, int sendcount, MPI_Datatype sendtype, void *recvbuf, int recvcnt, MPI_Datatype recvtype, MPI_Comm comm)`
- `int MPI_Gather (void *sendbuf, int sendcnt, MPI_Datatype sendtype, void *recvbuf, int recvcnt, MPI_Datatype recvtype, int root, MPI_Comm comm)`

### 7.35.1 マクロ定義

#### 7.35.1.1 `#define MPI_CHAR 2`

`mpi_stubs.h` の 22 行で定義されています。

#### 7.35.1.2 `#define MPI_COMM_WORLD 0`

`mpi_stubs.h` の 20 行で定義されています。

#### 7.35.1.3 `#define MPI_INT 1`

`mpi_stubs.h` の 21 行で定義されています。

#### 7.35.1.4 `#define MPI_SUCCESS true`

`mpi_stubs.h` の 24 行で定義されています。

## 7.35.2 型定義

### 7.35.2.1 typedef int MPI\_Comm

mpi\_stubs.h の 18 行で定義されています。

### 7.35.2.2 typedef int MPI\_Datatype

mpi\_stubs.h の 19 行で定義されています。

## 7.35.3 関数

### 7.35.3.1 int MPI\_Allgather ( void \* *sendbuf*, int *sendcount*, MPI\_Datatype *sendtype*, void \* *recvbuf*, int *recvcount*, MPI\_Datatype *recvtype*, MPI\_Comm *comm* ) [inline]

mpi\_stubs.h の 40 行で定義されています。

### 7.35.3.2 int MPI\_Comm\_rank ( MPI\_Comm *comm*, int \* *rank* ) [inline]

mpi\_stubs.h の 28 行で定義されています。

参照元 cdm\_DFI::WriteInit().

### 7.35.3.3 int MPI\_Comm\_size ( MPI\_Comm *comm*, int \* *size* ) [inline]

mpi\_stubs.h の 34 行で定義されています。

参照元 cdm\_DFI::WriteInit().

### 7.35.3.4 int MPI\_Gather ( void \* *sendbuf*, int *sendcnt*, MPI\_Datatype *sendtype*, void \* *recvbuf*, int *recvcnt*, MPI\_Datatype *recvtype*, int *root*, MPI\_Comm *comm* ) [inline]

mpi\_stubs.h の 47 行で定義されています。

### 7.35.3.5 bool MPI\_Init ( int \* *argc*, char \*\*\* *argv* ) [inline]

mpi\_stubs.h の 26 行で定義されています。

# Index

- ~cdm\_ActiveSubDomain
  - cdm\_ActiveSubDomain, [20](#)
- ~cdm\_Array
  - cdm\_Array, [24](#)
- ~cdm\_DFI
  - cdm\_DFI, [39](#)
- ~cdm\_DFI\_AVS
  - cdm\_DFI\_AVS, [76](#)
- ~cdm\_DFI\_BOV
  - cdm\_DFI\_BOV, [80](#)
- ~cdm\_DFI\_FUB
  - cdm\_DFI\_FUB, [84](#)
- ~cdm\_DFI\_NETCDF
  - cdm\_DFI\_NETCDF, [91](#)
- ~cdm\_DFI\_PLOT3D
  - cdm\_DFI\_PLOT3D, [102](#)
- ~cdm\_DFI\_SPH
  - cdm\_DFI\_SPH, [109](#)
- ~cdm\_DFI\_VTK
  - cdm\_DFI\_VTK, [113](#)
- ~cdm\_Domain
  - cdm\_Domain, [118](#)
- ~cdm\_FILE
  - cdm\_FILE, [127](#)
- ~cdm\_FieldFileNameFormat
  - cdm\_FieldFileNameFormat, [122](#)
- ~cdm\_FieldFileNameFormatElem
  - cdm\_FieldFileNameFormatElem, [125](#)
- ~cdm\_FileInfo
  - cdm\_FileInfo, [133](#)
- ~cdm\_FilePath
  - cdm\_FilePath, [136](#)
- ~cdm\_MPI
  - cdm\_MPI, [138](#)
- ~cdm\_NonUniformDomain
  - cdm\_NonUniformDomain, [143](#)
- ~cdm\_Process
  - cdm\_Process, [153](#)
- ~cdm\_Rank
  - cdm\_Rank, [158](#)
- ~cdm\_Slice
  - cdm\_Slice, [160](#)
- ~cdm\_TextParser
  - cdm\_TextParser, [163](#)
- ~cdm\_TimeSlice
  - cdm\_TimeSlice, [167](#)
- ~cdm\_TypeArray
  - cdm\_TypeArray, [170](#)
- ~cdm\_Unit
  - cdm\_Unit, [174](#)
- ~cdm\_UnitElem
  - cdm\_UnitElem, [176](#)
- ~cdm\_VisIt
  - cdm\_VisIt, [179](#)
- ~stVarInfo
  - cdm\_DFI\_NETCDF::stVarInfo, [182](#)
- \_CDM\_FIELDFILENAMEFORMAT\_H
  - cdm\_FieldFileNameFormat.h, [202](#)
- \_CDM\_IDX\_IJ
  - cdm\_Define.h, [189](#)
- \_CDM\_IDX\_IJK
  - cdm\_Define.h, [190](#)
- \_CDM\_IDX\_IJKN
  - cdm\_Define.h, [190](#)
- \_CDM\_IDX\_NIJ
  - cdm\_Define.h, [191](#)
- \_CDM\_IDX\_NIJK
  - cdm\_Define.h, [191](#)
- \_CDM\_TAB\_STR
  - cdm\_Define.h, [192](#)
- \_CDM\_WRITE\_TAB
  - cdm\_Define.h, [192](#)
- \_DATA\_UNKNOWN
  - cdm\_DFI\_SPH, [108](#)
- \_DOUBLE
  - cdm\_DFI\_SPH, [108](#)
- \_FLOAT
  - cdm\_DFI\_SPH, [108](#)
- \_FUB\_TAB\_STR
  - cdm\_FieldFileNameFormatElem.h, [203](#)
- \_FUB\_WRITE\_TAB
  - cdm\_FieldFileNameFormatElem.h, [203](#)
- \_REAL\_UNKNOWN
  - cdm\_DFI\_SPH, [108](#)
- \_SCALAR
  - cdm\_DFI\_SPH, [108](#)
- \_VECTOR
  - cdm\_DFI\_SPH, [108](#)
- \_getArraySize
  - cdm\_Array, [24](#)
- \_getArraySizeInt
  - cdm\_Array, [24](#)
- \_val
  - cdm\_TypeArray, [170](#)
- ActiveSubdomainFile
  - cdm\_Domain, [121](#)
- AddFieldFileNameFormatElem
  - cdm\_FieldFileNameFormat, [123](#)

- AddNcUnit
  - cdm\_DFI\_NETCDF, 91
- AddSlice
  - cdm\_TimeSlice, 167
- AddTimeSlice
  - cdm\_DFI, 39
- AddUnit
  - cdm\_DFI, 39
  - cdm\_Unit, 174
- ArrayShape
  - cdm\_FileInfo, 134
- AveragedStep
  - cdm\_Slice, 161
- AveragedTime
  - cdm\_Slice, 161
- avr\_mode
  - cdm\_Slice, 161
- BSWAP16
  - cdm\_endianUtil.h, 200
- BSWAP32
  - cdm\_endianUtil.h, 200
- BSWAP64
  - cdm\_endianUtil.h, 201
- BSWAP\_X\_16
  - cdm\_endianUtil.h, 201
- BSWAP\_X\_32
  - cdm\_endianUtil.h, 201
- BSWAP\_X\_64
  - cdm\_endianUtil.h, 201
- BSWAPVEC
  - cdm\_endianUtil.h, 201
- bc\_id
  - cdm\_Rank, 158
- BsetDiff
  - cdm\_UnitElem, 178
- C\_CDM\_RANKNOPREFIX
  - CDM, 17
- c\_id
  - cdm\_Rank, 158
- CDM, 9
  - C\_CDM\_RANKNOPREFIX, 17
  - cdmPath\_ConnectPath, 16
  - cdmPath\_DirName, 16
  - cdmPath\_FileName, 16
  - cdmPath\_getDelimChar, 16
  - cdmPath\_getDelimString, 16
  - cdmPath\_hasDrive, 17
  - cdmPath\_isAbsolute, 17
  - E\_CDM\_ARRAYSHAPE, 11
  - E\_CDM\_ARRAYSHAPE\_UNKNOWN, 11
  - E\_CDM\_BIG, 12
  - E\_CDM\_DFITYPE, 11
  - E\_CDM\_DFITYPE\_CARTESIAN, 11
  - E\_CDM\_DFITYPE\_NON\_UNIFORM\_CARTESIAN, 11
  - E\_CDM\_DFITYPE\_UNKNOWN, 11
  - E\_CDM\_DIFFDIV\_REFINEMENT, 16
  - E\_CDM\_DIFFDIV\_SAMERES, 16
  - E\_CDM\_DTYPE, 11
  - E\_CDM\_DTYPE\_UNKNOWN, 11
  - E\_CDM\_ENDIANTYPE, 12
  - E\_CDM\_ENDIANTYPE\_UNKNOWN, 12
  - E\_CDM\_ERROR, 12
  - E\_CDM\_ERROR\_INVALID\_DIVNUM, 14
  - E\_CDM\_ERROR\_MAKEDIRECTORY, 14
  - E\_CDM\_ERROR\_MISMATCH\_NP\_SUBDOMAIN, 14
  - E\_CDM\_ERROR\_NOMATCH\_ENDIAN, 13
  - E\_CDM\_ERROR\_NUM\_OF\_GUIDECCELLS, 14
  - E\_CDM\_ERROR\_OPEN\_COORDINATEFILE, 13
  - E\_CDM\_ERROR\_OPEN\_FIELDDATA, 14
  - E\_CDM\_ERROR\_OPEN\_SBDM, 14
  - E\_CDM\_ERROR\_READ\_BOV\_FILE, 14
  - E\_CDM\_ERROR\_READ\_COORDINATEFILE, 13
  - E\_CDM\_ERROR\_READ\_DFI\_BCFLAGID, 12
  - E\_CDM\_ERROR\_READ\_DFI\_CELLID, 12
  - E\_CDM\_ERROR\_READ\_DFI\_COORDINATEFILE, 13
  - E\_CDM\_ERROR\_READ\_DFI\_COORDINATEFILE\_ENDIAN, 13
  - E\_CDM\_ERROR\_READ\_DFI\_COORDINATEFILEPRECISION, 13
  - E\_CDM\_ERROR\_READ\_DFI\_COORDINATEFILETYPE, 13
  - E\_CDM\_ERROR\_READ\_DFI\_DATATYPE, 12
  - E\_CDM\_ERROR\_READ\_DFI\_DFITYPE, 13
  - E\_CDM\_ERROR\_READ\_DFI\_DIRECTORYPATH, 12
  - E\_CDM\_ERROR\_READ\_DFI\_ENDIAN, 12
  - E\_CDM\_ERROR\_READ\_DFI\_FIELDFILENAMEFORMAT, 13
  - E\_CDM\_ERROR\_READ\_DFI\_FILEFORMAT, 12
  - E\_CDM\_ERROR\_READ\_DFI\_FILELIST, 14
  - E\_CDM\_ERROR\_READ\_DFI\_FILEPATH\_PROCESSES, 12
  - E\_CDM\_ERROR\_READ\_DFI\_GLOBALDIVISION, 12
  - E\_CDM\_ERROR\_READ\_DFI\_GLOBALORIGIN, 12
  - E\_CDM\_ERROR\_READ\_DFI\_GLOBALREGION, 12
  - E\_CDM\_ERROR\_READ\_DFI\_GLOBALVOXEL, 12
  - E\_CDM\_ERROR\_READ\_DFI\_GUIDECCELL, 12
  - E\_CDM\_ERROR\_READ\_DFI\_HEADINDEX, 12
  - E\_CDM\_ERROR\_READ\_DFI\_HOSTNAME, 12
  - E\_CDM\_ERROR\_READ\_DFI\_ID, 12
  - E\_CDM\_ERROR\_READ\_DFI\_MAX, 13
  - E\_CDM\_ERROR\_READ\_DFI\_MIN, 13
  - E\_CDM\_ERROR\_READ\_DFI\_NETCDF, 13
  - E\_CDM\_ERROR\_READ\_DFI\_NO\_MINMAX, 13
  - E\_CDM\_ERROR\_READ\_DFI\_NO\_RANK, 12
  - E\_CDM\_ERROR\_READ\_DFI\_NO\_SLICE, 13
  - E\_CDM\_ERROR\_READ\_DFI\_NUMVARIABLES, 12



- E\_CDM\_ERROR\_READ\_DFI\_PREFIX, 12
- E\_CDM\_ERROR\_READ\_DFI\_STEP, 13
- E\_CDM\_ERROR\_READ\_DFI\_TAILINDEX, 12
- E\_CDM\_ERROR\_READ\_DFI\_TIME, 13
- E\_CDM\_ERROR\_READ\_DFI\_TIMESLICEDIRECTORY, 12
- E\_CDM\_ERROR\_READ\_DFI\_VOXELSIZE, 12
- E\_CDM\_ERROR\_READ\_DOMAIN, 13
- E\_CDM\_ERROR\_READ\_FIELD\_AVERAGED\_RECORD, 14
- E\_CDM\_ERROR\_READ\_FIELD\_DATA\_RECORD, 14
- E\_CDM\_ERROR\_READ\_FIELD\_HEADER\_RECORD, 14
- E\_CDM\_ERROR\_READ\_FIELDDATA\_FILE, 13
- E\_CDM\_ERROR\_READ\_FILEINFO, 13
- E\_CDM\_ERROR\_READ\_FILELIST\_CCORDINATEFILENAME, 13
- E\_CDM\_ERROR\_READ\_FILELIST\_FIELDDATAFILENAME, 13
- E\_CDM\_ERROR\_READ\_FILELIST\_ID, 13
- E\_CDM\_ERROR\_READ\_FILEPATH, 13
- E\_CDM\_ERROR\_READ\_FUB\_REC1, 14
- E\_CDM\_ERROR\_READ\_FUB\_REC2, 14
- E\_CDM\_ERROR\_READ\_INDEXFILE\_OPENERROR, 13
- E\_CDM\_ERROR\_READ\_MPI, 13
- E\_CDM\_ERROR\_READ\_NETCDF\_FUNC, 14
- E\_CDM\_ERROR\_READ\_NETCDF\_MISMATCH\_TYPE, 13
- E\_CDM\_ERROR\_READ\_NETCDF\_VAR\_1D, 14
- E\_CDM\_ERROR\_READ\_PROCESS, 13
- E\_CDM\_ERROR\_READ\_PROCFILE\_OPENERROR, 13
- E\_CDM\_ERROR\_READ\_SBDM\_CONTENTS, 14
- E\_CDM\_ERROR\_READ\_SBDM\_DIV, 14
- E\_CDM\_ERROR\_READ\_SBDM\_FORMAT, 14
- E\_CDM\_ERROR\_READ\_SBDM\_HEADER, 14
- E\_CDM\_ERROR\_READ\_SPH\_FILE, 13
- E\_CDM\_ERROR\_READ\_SPH\_REC1, 13
- E\_CDM\_ERROR\_READ\_SPH\_REC2, 13
- E\_CDM\_ERROR\_READ\_SPH\_REC3, 13
- E\_CDM\_ERROR\_READ\_SPH\_REC4, 13
- E\_CDM\_ERROR\_READ\_SPH\_REC5, 13
- E\_CDM\_ERROR\_READ\_SPH\_REC6, 13
- E\_CDM\_ERROR\_READ\_SPH\_REC7, 13
- E\_CDM\_ERROR\_READ\_TIMESLICE, 13
- E\_CDM\_ERROR\_READ\_UNIT, 13
- E\_CDM\_ERROR\_READ\_SBDM\_NUMDOMAIN\_ZERO, 14
- E\_CDM\_ERROR\_TEXTPARSER, 13
- E\_CDM\_ERROR\_UNDEFINED\_FIELDFILENAMEFORMAT, 13
- E\_CDM\_ERROR\_UNMATCH\_NUM\_OF\_VARIABLES, 13
- E\_CDM\_ERROR\_UNMATCH\_VOXELSIZE, 13
- E\_CDM\_ERROR\_WRITE\_DFI\_NETCDF, 15
- E\_CDM\_ERROR\_WRITE\_DOMAIN, 14
- E\_CDM\_ERROR\_WRITE\_FIELD\_AVERAGED\_RECORD, 14
- E\_CDM\_ERROR\_WRITE\_FIELD\_DATA\_RECORD, 14
- E\_CDM\_ERROR\_WRITE\_FIELD\_HEADER\_RECORD, 14
- E\_CDM\_ERROR\_WRITE\_FILEINFO, 14
- E\_CDM\_ERROR\_WRITE\_FILEPATH, 14
- E\_CDM\_ERROR\_WRITE\_GRIDFILE, 15
- E\_CDM\_ERROR\_WRITE\_INDEXFILE\_OPENERROR, 14
- E\_CDM\_ERROR\_WRITE\_INDEXFILENAME\_EMPTY, 14
- E\_CDM\_ERROR\_WRITE\_MPI, 14
- E\_CDM\_ERROR\_WRITE\_PREFIX\_EMPTY, 14
- E\_CDM\_ERROR\_WRITE\_PROCESS, 14
- E\_CDM\_ERROR\_WRITE\_PROCFILE\_OPENERROR, 14
- E\_CDM\_ERROR\_WRITE\_PROCFILENAME\_EMPTY, 14
- E\_CDM\_ERROR\_WRITE\_RANKID, 14
- E\_CDM\_ERROR\_WRITE\_SPH\_REC1, 14
- E\_CDM\_ERROR\_WRITE\_SPH\_REC2, 14
- E\_CDM\_ERROR\_WRITE\_SPH\_REC3, 14
- E\_CDM\_ERROR\_WRITE\_SPH\_REC4, 14
- E\_CDM\_ERROR\_WRITE\_SPH\_REC5, 14
- E\_CDM\_ERROR\_WRITE\_SPH\_REC6, 14
- E\_CDM\_ERROR\_WRITE\_SPH\_REC7, 14
- E\_CDM\_ERROR\_WRITE\_TIMESLICE, 14
- E\_CDM\_ERROR\_WRITE\_UNIT, 14
- E\_CDM\_ERROR\_WRITE\_VISIT, 15
- E\_CDM\_ERRORCODE, 12
- E\_CDM\_FILE\_TYPE, 15
- E\_CDM\_FILE\_TYPE\_ASCII, 15
- E\_CDM\_FILE\_TYPE\_BINARY, 15
- E\_CDM\_FILE\_TYPE\_DEFAULT, 15
- E\_CDM\_FILE\_TYPE\_FBINARY, 15
- E\_CDM\_FLOAT32, 12
- E\_CDM\_FLOAT64, 12
- E\_CDM\_FMT\_AVS, 15
- E\_CDM\_FMT\_BOV, 15
- E\_CDM\_FMT\_FUB, 15
- E\_CDM\_FMT\_FUB\_COD, 15
- E\_CDM\_FMT\_NETCDF4, 15
- E\_CDM\_FMT\_PLOT3D, 15
- E\_CDM\_FMT\_SPH, 15
- E\_CDM\_FMT\_UNKNOWN, 15
- E\_CDM\_FMT\_VTK, 15
- E\_CDM\_FNAME\_CUSTOM, 16
- E\_CDM\_FNAME\_DEFAULT, 16
- E\_CDM\_FNAME\_RANK, 16
- E\_CDM\_FNAME\_RANK\_STEP, 16
- E\_CDM\_FNAME\_STEP\_RANK, 16
- E\_CDM\_FORMAT, 15
- E\_CDM\_IJKN, 11
- E\_CDM\_INT16, 11
- E\_CDM\_INT32, 11
- E\_CDM\_INT64, 11

- E\_CDM\_INT8, 11
- E\_CDM\_LITTLE, 12
- E\_CDM\_NIJK, 11
- E\_CDM\_OFF, 15
- E\_CDM\_ON, 15
- E\_CDM\_ONOFF, 15
- E\_CDM\_OUTPUT\_FNAME, 15
- E\_CDM\_READTYPE, 16
- E\_CDM\_READTYPE\_UNKNOWN, 16
- E\_CDM\_SAMEDIV\_REFINEMENT, 16
- E\_CDM\_SAMEDIV\_SAMERES, 16
- E\_CDM\_SUCCESS, 12
- E\_CDM\_UINT16, 12
- E\_CDM\_UINT32, 12
- E\_CDM\_UINT64, 12
- E\_CDM\_UINT8, 11
- E\_CDM\_WARN\_GETUNIT, 15
- ExtractPathWithoutExt, 17
- vfvPath\_emitDrive, 17
- CDM\_INLINE
  - cdm\_Array\_inline.h, 186
  - cdm\_DFI\_inline.h, 197
  - cdm\_endianUtil.h, 202
  - cdm\_NonUniformDomain\_inline.h, 207
  - cdm\_Plot3d\_inline.h, 208
- CDM\_MEMFUN
  - cdm\_Array\_inline.h, 186
- CDM\_REVISION
  - cdm\_Version.h, 210
- CDM\_VERSION\_NO
  - cdm\_Version.h, 210
- cdm\_ActiveSubDomain, 19
  - ~cdm\_ActiveSubDomain, 20
  - cdm\_ActiveSubDomain, 19
  - cdm\_ActiveSubDomain, 19
  - clear, 20
  - GetPos, 20
  - m\_pos, 21
  - operator==, 20
  - SetPos, 20
- cdm\_ActiveSubDomain.h, 185
- cdm\_Array, 21
  - ~cdm\_Array, 24
  - \_getArraySize, 24
  - \_getArraySizeInt, 24
  - cdm\_Array, 24
  - cdm\_Array, 24
  - copyArray, 24
  - copyArrayNvari, 25
  - copyArrayNvari\_to\_ijk, 25
  - getArrayLength, 25
  - getArrayShape, 25
  - getArrayShapeString, 25
  - getArraySize, 25
  - getArraySizeInt, 26
  - getData, 26
  - getDataType, 26
  - getDataTypeString, 26
- getGc, 26
- getGcInt, 26
- getHeadIndex, 27
- getNvari, 27
- getNvariInt, 27
- getTailIndex, 27
- instanceArray, 27–29
- interp\_coarse, 29
- m\_Sz, 31
- m\_SzI, 31
- m\_dtype, 29
- m\_gc, 29
- m\_gcl, 30
- m\_gcl, 30
- m\_headIndex, 30
- m\_nvari, 30
- m\_nvaril, 30
- m\_shape, 30
- m\_sz, 30
- m\_szI, 31
- m\_tailIndex, 31
- readBinary, 29
- setHeadIndex, 29
- writeAscii, 29
- writeBinary, 29
- cdm\_Array.h, 185
  - cdm\_interp\_ijkn\_r4\_, 185
  - cdm\_interp\_ijkn\_r8\_, 185
  - cdm\_interp\_nijk\_r4\_, 186
  - cdm\_interp\_nijk\_r8\_, 186
- cdm\_Array\_inline.h, 186
  - CDM\_INLINE, 186
  - CDM\_MEMFUN, 186
- cdm\_Create\_dfiProcessInfo
  - cdm\_DFI, 40
- cdm\_DFI, 31
  - ~cdm\_DFI, 39
  - AddTimeSlice, 39
  - AddUnit, 39
  - cdm\_Create\_dfiProcessInfo, 40
  - cdm\_DFI, 39
  - cdm\_DFI, 39
  - CheckAddWriteMode, 40
  - CheckReadRank, 40
  - CheckReadType, 40
  - ConvDatatypeE2S, 41
  - ConvDatatypeS2E, 41
  - CreateReadStartEnd, 41
  - DFI\_Domain, 70
  - DFI\_Finfo, 71
  - DFI\_Fpath, 71
  - DFI\_MPI, 71
  - DFI\_Process, 71
  - DFI\_TimeSlice, 71
  - DFI\_Unit, 71
  - DFI\_VisIt, 72
  - Generate\_DFI\_Name, 42
  - Generate\_Directory\_Path, 42

Generate\_FieldFileName, 42  
 Generate\_FileName, 42  
 get\_cdm\_Datasize, 43  
 get\_dfi\_fname, 43  
 GetArrayShape, 43  
 GetArrayShapeString, 43  
 getBufSize, 43  
 GetDFIGlobalDivision, 46  
 GetDFIGlobalVoxel, 46  
 GetDFIType, 46  
 GetDFITypeString, 46  
 GetDataType, 45  
 GetDataTypeString, 45  
 GetFileFormat, 46  
 GetFileFormatString, 46  
 getFileNameFromFileList, 47  
 getMinMax, 47  
 GetNumGuideCell, 47  
 GetNumVariables, 47  
 GetUnit, 47  
 GetUnitElem, 48  
 getVariableName, 48  
 getVectorMinMax, 48  
 getVersionInfo, 48  
 GetcdmDomain, 44  
 GetcdmFieldFileNameFormat, 44  
 GetcdmFileInfo, 44  
 GetcdmFilePath, 44  
 GetcdmMPI, 44  
 GetcdmProcess, 44  
 GetcdmRankNoPrefix, 45  
 GetcdmTimeSlice, 45  
 GetcdmUnit, 45  
 GetcdmVisIt, 45  
 m\_RankID, 73  
 m\_bgrid\_interp\_flag, 72  
 m\_comm, 72  
 m\_directoryPath, 72  
 m\_indexDfiName, 72  
 m\_input\_type, 72  
 m\_output\_fname, 73  
 m\_output\_type, 73  
 m\_output\_type\_coord, 73  
 m\_read\_type, 73  
 m\_readRankList, 73  
 MakeDirectory, 49  
 MakeDirectoryPath, 49  
 MakeDirectorySub, 49  
 normalizeBaseTime, 49  
 normalizeDeltaT, 49  
 normalizeIntervalTime, 50  
 normalizeLastTime, 50  
 normalizeStartTime, 50  
 normalizeTime, 50  
 read\_Datarecord, 50  
 read\_HeaderRecord, 52  
 read\_averaged, 50  
 ReadCoordinateData, 52, 53  
 ReadData, 55, 56  
 ReadFieldData, 57  
 ReadInit, 57  
 set\_RankID, 59  
 set\_input\_type, 58  
 set\_interp\_flag, 58  
 set\_output\_fname, 58  
 set\_output\_type, 58  
 set\_output\_type\_coord, 58  
 setGridData, 60  
 setIntervalStep, 61  
 setIntervalTime, 61  
 SetTimeSliceFlag, 61  
 setVariableName, 61  
 SetcdmDomain, 59  
 SetcdmFieldFileNameFormat, 59  
 SetcdmFileInfo, 59  
 SetcdmFilePath, 59  
 SetcdmMPI, 59  
 SetcdmProcess, 59  
 SetcdmRankNoPrefix, 59  
 SetcdmTimeSlice, 60  
 SetcdmUnit, 60  
 SetcdmVisIt, 60  
 VolumeDataDivide, 61  
 write\_DataRecord, 63  
 write\_GridData, 63  
 write\_HeaderRecord, 64  
 write\_ascii\_header, 63  
 write\_averaged, 63  
 WriteCoordinateData, 64  
 WriteData, 65  
 WriteFieldData, 66  
 WriteFieldDataFile, 66, 67  
 WriteGridFile, 67  
 WriteIndexDfiFile, 67  
 WriteInit, 67–70  
 WriteProcDfiFile, 70  
 cdm\_DFI.h, 195  
 cdm\_DFI\_AVS, 73  
   ~cdm\_DFI\_AVS, 76  
   cdm\_DFI\_AVS, 74  
   cdm\_DFI\_AVS, 74  
   read\_Datarecord, 76  
   read\_HeaderRecord, 77  
   read\_averaged, 76  
   write\_DataRecord, 78  
   write\_HeaderRecord, 78  
   write\_ascii\_header, 77  
   write\_averaged, 77  
   write\_avs\_cord, 78  
   write\_avs\_header, 78  
 cdm\_DFI\_AVS.h, 196  
 cdm\_DFI\_BOV, 79  
   ~cdm\_DFI\_BOV, 80  
   cdm\_DFI\_BOV, 80  
   cdm\_DFI\_BOV, 80  
   read\_Datarecord, 81

- read\_HeaderRecord, 81
- read\_averaged, 80
- write\_DataRecord, 82
- write\_HeaderRecord, 82
- write\_ascii\_header, 81
- write\_averaged, 82
- cdm\_DFI\_BOV.h, 196
- cdm\_DFI\_FUB, 83
  - ~cdm\_DFI\_FUB, 84
  - cdm\_DFI\_FUB, 84
  - cdm\_DFI\_FUB, 84
  - DFI\_FieldFileNameFormat, 88
  - getCoordinateFileName, 85
  - GetcdmFieldFileNameFormat, 85
  - read\_Datarecord, 86
  - read\_HeaderRecord, 86
  - read\_averaged, 85
  - ReadCoordinateData, 86
  - SetcdmFieldFileNameFormat, 87
  - write\_DataRecord, 87
  - write\_HeaderRecord, 88
  - write\_averaged, 87
  - WriteCoordinateData, 88
- cdm\_DFI\_FUB.h, 197
- cdm\_DFI\_NETCDF, 88
  - ~cdm\_DFI\_NETCDF, 91
  - AddNcUnit, 91
  - cdm\_DFI\_NETCDF, 91
  - cdm\_DFI\_NETCDF, 91
  - CheckAddWriteMode, 92
  - CheckArrayVarInfo, 92
  - GetCdmType, 92
  - GetDimName, 92
  - GetNcType, 93
  - GetNcUnit, 93
  - GetVarInfo, 93
  - m\_NC\_Unit, 99
  - m\_nameT, 99
  - m\_nameX, 99
  - m\_nameY, 99
  - m\_nameZ, 99
  - m\_varInfoT, 100
  - m\_varInfoX, 100
  - m\_varInfoY, 100
  - m\_varInfoZ, 100
  - m\_vecVarInfo, 100
  - m\_writeFlag, 100
  - read\_Datarecord, 94
  - read\_HeaderRecord, 94, 96
  - read\_averaged, 93
  - ReadAdditionalTP, 96
  - SetDimName, 96
  - SetWriteFlag, 97
  - write\_AttUnits, 97
  - write\_DataRecord, 97, 98
  - write\_HeaderRecord, 98
  - write\_averaged, 97
  - WriteAdditionalTP, 99
- cdm\_DFI\_NETCDF.h, 197
- cdm\_DFI\_NETCDF::stDimInfo, 180
  - id, 181
  - len, 181
  - name, 181
- cdm\_DFI\_NETCDF::stVarInfo, 181
  - ~stVarInfo, 182
  - clear, 182
  - data, 182
  - dims, 182
  - endian, 182
  - id, 182
  - name, 183
  - stVarInfo, 182
  - type, 183
  - unit, 183
- cdm\_DFI\_PLOT3D, 100
  - ~cdm\_DFI\_PLOT3D, 102
  - cdm\_DFI\_PLOT3D, 102
  - cdm\_DFI\_PLOT3D, 102
  - read\_Datarecord, 102
  - read\_Func, 104
  - read\_HeaderRecord, 104
  - read\_averaged, 102
  - write\_DataRecord, 105
  - write\_Func, 105, 106
  - write\_GridData, 106
  - write\_HeaderRecord, 106
  - write\_XYZ, 106
  - write\_averaged, 105
- cdm\_DFI\_PLOT3D.h, 198
- cdm\_DFI\_SPH, 107
  - ~cdm\_DFI\_SPH, 109
  - \_DATA\_UNKNOWN, 108
  - \_DOUBLE, 108
  - \_FLOAT, 108
  - \_REAL\_UNKNOWN, 108
  - \_SCALAR, 108
  - \_VECTOR, 108
  - cdm\_DFI\_SPH, 108
  - cdm\_DFI\_SPH, 108
  - DataDims, 108
  - read\_Datarecord, 109
  - read\_HeaderRecord, 109
  - read\_averaged, 109
  - RealType, 108
  - write\_DataRecord, 111
  - write\_HeaderRecord, 111
  - write\_averaged, 111
- cdm\_DFI\_SPH.h, 198
- cdm\_DFI\_VTK, 112
  - ~cdm\_DFI\_VTK, 113
  - cdm\_DFI\_VTK, 113
  - cdm\_DFI\_VTK, 113
  - read\_Datarecord, 114
  - read\_HeaderRecord, 114
  - read\_averaged, 113
  - write\_DataRecord, 115

- write\_HeaderRecord, 115
- write\_averaged, 115
- cdm\_DFI\_VTK.h, 199
- cdm\_DFI\_inline.h, 197
  - CDM\_INLINE, 197
- cdm\_Define.h, 186
  - \_CDM\_IDX\_IJ, 189
  - \_CDM\_IDX\_IJK, 190
  - \_CDM\_IDX\_IJKN, 190
  - \_CDM\_IDX\_NIJ, 191
  - \_CDM\_IDX\_NIJK, 191
  - \_CDM\_TAB\_STR, 192
  - \_CDM\_WRITE\_TAB, 192
  - D\_CDM\_BIG, 192
  - D\_CDM\_BYTE, 192
  - D\_CDM\_DFITYPE\_CARTESIAN, 192
  - D\_CDM\_DFITYPE\_NON\_UNIFORM\_CARTESIAN, 192
  - D\_CDM\_DOUBLE, 192
  - D\_CDM\_EXT\_BOV, 193
  - D\_CDM\_EXT\_BOV\_DATAFILE, 193
  - D\_CDM\_EXT\_FUB, 193
  - D\_CDM\_EXT\_FUNC, 193
  - D\_CDM\_EXT\_NC, 193
  - D\_CDM\_EXT\_SPH, 193
  - D\_CDM\_EXT\_VTK, 193
  - D\_CDM\_EXT\_XYZ, 193
  - D\_CDM\_FLOAT, 193
  - D\_CDM\_FLOAT32, 193
  - D\_CDM\_FLOAT64, 193
  - D\_CDM\_IJNK, 193
  - D\_CDM\_INT, 194
  - D\_CDM\_INT16, 194
  - D\_CDM\_INT32, 194
  - D\_CDM\_INT64, 194
  - D\_CDM\_INT8, 194
  - D\_CDM\_LITTLE, 194
  - D\_CDM\_NIJK, 194
  - D\_CDM\_OFF, 194
  - D\_CDM\_ON, 194
  - D\_CDM\_UINT16, 194
  - D\_CDM\_UINT32, 194
  - D\_CDM\_UINT64, 194
  - D\_CDM\_UINT8, 195
- stmpd\_printf, 195
- cdm\_Domain, 116
  - ~cdm\_Domain, 118
  - ActiveSubdomainFile, 121
  - cdm\_Domain, 117
  - cdm\_Domain, 117
  - CellX, 118
  - CellY, 118
  - CellZ, 118
  - Clear, 119
  - GetCoordinateFile, 119
  - GetCoordinateFileEndian, 119
  - GetCoordinateFilePrecision, 119
  - GetCoordinateFileType, 119
  - GlobalDivision, 121
  - GlobalOrigin, 121
  - GlobalRegion, 121
  - GlobalVoxel, 121
  - NodeX, 119
  - NodeY, 120
  - NodeZ, 120
  - Pitch, 122
  - Read, 120
  - Write, 121
- cdm\_Domain.h, 199
- cdm\_FILE, 126
  - ~cdm\_FILE, 127
  - cdm\_FILE, 127
  - cdm\_FILE, 127
  - CloseFile, 128
  - m\_addMode, 130
  - m\_fname, 130
  - m\_format, 130
  - m\_fp, 130
  - m\_ncid, 130
  - OpenReadBinary, 129
  - OpenWriteAscii, 129
  - OpenWriteBinary, 129
- cdm\_FILE.h, 204
- cdm\_FieldFileNameFormat, 122
  - ~cdm\_FieldFileNameFormat, 122
  - AddFieldFileNameFormatElem, 123
  - cdm\_FieldFileNameFormat, 122
  - cdm\_FieldFileNameFormat, 122
  - FileExist, 123
  - GenerateFileName, 123
  - GetFieldFileNameFormatElem, 123
  - GetLabelList, 123
  - LabelList, 123
  - mapElem, 123
  - Print, 123
  - Read, 123
  - Write, 123
- cdm\_FieldFileNameFormat.h, 202
  - \_CDM\_FIELDFILENAMEFORMAT\_H\_, 202
- cdm\_FieldFileNameFormatElem, 124
  - ~cdm\_FieldFileNameFormatElem, 125
  - cdm\_FieldFileNameFormatElem, 125
  - cdm\_FieldFileNameFormatElem, 125
  - FileExist, 125
  - FileName, 125
  - FileNameFormat, 126
  - FnameFormat, 126
  - FnameLabel, 126
  - GenerateFileName, 125
  - GetDisitNo, 125
  - PrintParse, 125
  - RankIdDisit, 126
  - RankIdKey, 126
  - Read, 125
  - SetDisitNoFormat, 125
  - SetFnameFormat, 125

- StepNoDisit, [126](#)
- StepNoKey, [126](#)
- Write, [125](#)
- cdm\_FieldFileNameFormatElem.h, [203](#)
  - \_FUB\_TAB\_STR, [203](#)
  - \_FUB\_WRITE\_TAB, [203](#)
  - E\_FUB\_NONE, [203](#)
  - E\_FUB\_RANK, [203](#)
  - E\_FUB\_RANK\_STEP, [203](#)
  - E\_FUB\_STEP, [203](#)
  - E\_FUB\_STEP\_RANK, [203](#)
  - FUB\_FNAME\_TYPE, [203](#)
- cdm\_FileInfo, [130](#)
  - ~cdm\_FileInfo, [133](#)
  - ArrayShape, [134](#)
  - cdm\_FileInfo, [131](#)
  - cdm\_FileInfo, [131](#)
  - DFIType, [134](#)
  - DataType, [134](#)
  - DirectoryPath, [134](#)
  - Endian, [134](#)
  - FieldFilenameFormat, [134](#)
  - FileFormat, [135](#)
  - getVariableName, [133](#)
  - GuideCell, [135](#)
  - NumVariables, [135](#)
  - Prefix, [135](#)
  - RankNoPrefix, [135](#)
  - Read, [133](#)
  - setVariableName, [133](#)
  - TimeSliceDirFlag, [135](#)
  - VariableName, [135](#)
  - Write, [134](#)
- cdm\_FileInfo.h, [204](#)
- cdm\_FilePath, [136](#)
  - ~cdm\_FilePath, [136](#)
  - cdm\_FilePath, [136](#)
  - cdm\_FilePath, [136](#)
  - ProcDFIFile, [137](#)
  - Read, [137](#)
  - Write, [137](#)
- cdm\_FilePath.h, [204](#)
- cdm\_MPI, [137](#)
  - ~cdm\_MPI, [138](#)
  - cdm\_MPI, [138](#)
  - cdm\_MPI, [138](#)
  - NumberOfGroup, [140](#)
  - NumberOfRank, [140](#)
  - Read, [138](#)
  - Write, [138](#)
- cdm\_MPI.h, [205](#)
- cdm\_NonUniformDomain
  - ~cdm\_NonUniformDomain, [143](#)
  - cdm\_NonUniformDomain, [142](#), [143](#)
  - cdm\_NonUniformDomain, [142](#), [143](#)
  - CellX, [143](#)
  - CellY, [144](#)
  - CellZ, [144](#)
- Clear, [144](#)
- CoordinateFile, [149](#)
- CoordinateFileEndian, [150](#)
- CoordinateFilePrecision, [150](#)
- CoordinateFileType, [150](#)
- GetCoordinateFile, [144](#)
- GetCoordinateFileEndian, [145](#)
- GetCoordinateFilePrecision, [145](#)
- GetCoordinateFileType, [145](#)
- NodeX, [145](#)
- NodeY, [147](#)
- NodeZ, [147](#)
- pit\_gcXend, [150](#)
- pit\_gcXsta, [150](#)
- pit\_gcYend, [150](#)
- pit\_gcYsta, [150](#)
- pit\_gcZend, [151](#)
- pit\_gcZsta, [151](#)
- Read, [147](#)
- Read\_CoordinateFile, [148](#)
- readCoordData, [148](#)
- readCoordDataCount, [149](#)
- Write, [149](#)
- XCoordinates, [151](#)
- YCoordinates, [151](#)
- ZCoordinates, [151](#)
- cdm\_NonUniformDomain< T >, [140](#)
- cdm\_NonUniformDomain.h, [206](#)
- cdm\_NonUniformDomain\_inline.h, [206](#)
  - CDM\_INLINE, [207](#)
- cdm\_PathUtil.h, [207](#)
  - MAXPATHLEN, [207](#)
- cdm\_Plot3d\_inline.h, [207](#)
  - CDM\_INLINE, [208](#)
- cdm\_Process, [152](#)
  - ~cdm\_Process, [153](#)
  - cdm\_Process, [153](#)
  - cdm\_Process, [153](#)
  - CheckReadRank, [153](#)
  - CheckStartEnd, [153](#)
  - CreateHeadMap, [154](#)
  - CreateRankList, [154](#)
  - CreateRankMap, [155](#)
  - CreateSubDomainInfo, [155](#)
  - headT, [153](#)
  - isMatchEndianSbdmMagick, [156](#)
  - m\_rankMap, [157](#)
  - RankList, [157](#)
  - Read, [156](#)
  - ReadActiveSubdomainFile, [156](#)
  - Write, [156](#)
- cdm\_Process.h, [208](#)
- cdm\_Rank, [157](#)
  - ~cdm\_Rank, [158](#)
  - bc\_id, [158](#)
  - c\_id, [158](#)
  - cdm\_Rank, [158](#)
  - cdm\_Rank, [158](#)

- HeadIndex, 159
- HostName, 159
- RankID, 159
- Read, 158
- TailIndex, 159
- VoxelSize, 159
- Write, 158
- cdm\_Slice, 159
  - ~cdm\_Slice, 160
  - AveragedStep, 161
  - AveragedTime, 161
  - avr\_mode, 161
  - cdm\_Slice, 160
  - cdm\_Slice, 160
  - Max, 161
  - Min, 161
  - Read, 160
  - step, 161
  - time, 162
  - VectorMax, 162
  - VectorMin, 162
  - Write, 161
- cdm\_TextParser, 162
  - ~cdm\_TextParser, 163
  - cdm\_TextParser, 163
  - cdm\_TextParser, 163
  - chkLabel, 163
  - chkNode, 163
  - countLabels, 164
  - GetNodeStr, 164
  - getTPPtr, 164
  - getTPinstance, 164
  - GetValue, 164, 165
  - GetVector, 165
  - readTPfile, 165
  - remove, 166
  - tp, 166
- cdm\_TextParser.h, 208
- cdm\_TimeSlice, 166
  - ~cdm\_TimeSlice, 167
  - AddSlice, 167
  - cdm\_TimeSlice, 167
  - cdm\_TimeSlice, 167
  - getMinMax, 167
  - getVectorMinMax, 167
  - Read, 168
  - SliceList, 168
  - Write, 168
- cdm\_TimeSlice.h, 209
- cdm\_TypeArray
  - ~cdm\_TypeArray, 170
  - \_val, 170
  - cdm\_TypeArray, 170
  - cdm\_TypeArray, 170
  - copyArray, 171
  - copyArrayNvari, 171
  - copyArrayNvari\_to\_ijk, 171
  - getData, 172
  - hval, 172
  - m\_data, 173
  - m\_outptr, 173
  - readBinary, 172
  - val, 172
  - writeAscii, 172
  - writeBinary, 173
- cdm\_TypeArray< T >, 168
- cdm\_TypeArray.h, 209
- cdm\_Unit, 173
  - ~cdm\_Unit, 174
  - AddUnit, 174
  - cdm\_Unit, 174
  - cdm\_Unit, 174
  - GetUnit, 174
  - GetUnitElem, 175
  - Read, 175
  - UnitList, 175
  - Write, 175
- cdm\_Unit.h, 209
- cdm\_UnitElem, 176
  - ~cdm\_UnitElem, 176
  - BsetDiff, 178
  - cdm\_UnitElem, 176
  - cdm\_UnitElem, 176
  - difference, 178
  - Name, 178
  - Read, 177
  - reference, 178
  - Unit, 178
  - Write, 178
- cdm\_Version.h, 210
  - CDM\_REVISION, 210
  - CDM\_VERSION\_NO, 210
- cdm\_VisIt, 179
  - ~cdm\_VisIt, 179
  - cdm\_VisIt, 179
  - cdm\_VisIt, 179
  - PlotGC, 180
  - Read, 179
  - Write, 180
- cdm\_VisIt.h, 210
- cdm\_endianUtil.h, 200
  - BSWAP16, 200
  - BSWAP32, 200
  - BSWAP64, 201
  - BSWAP\_X\_16, 201
  - BSWAP\_X\_32, 201
  - BSWAP\_X\_64, 201
  - BSWAPVEC, 201
  - CDM\_INLINE, 202
  - DBSWAPVEC, 202
  - SBSWAPVEC, 202
- cdm\_interp\_ijkn.h, 205
- cdm\_interp\_ijkn\_r4\_
  - cdm\_Array.h, 185
- cdm\_interp\_ijkn\_r8\_
  - cdm\_Array.h, 185

- cdm\_interp\_nijk.h, 205
- cdm\_interp\_nijk\_r4\_
  - cdm\_Array.h, 186
- cdm\_interp\_nijk\_r8\_
  - cdm\_Array.h, 186
- cdmPath\_ConnectPath
  - CDM, 16
- cdmPath\_DirName
  - CDM, 16
- cdmPath\_FileName
  - CDM, 16
- cdmPath\_getDelimChar
  - CDM, 16
- cdmPath\_getDelimString
  - CDM, 16
- cdmPath\_hasDrive
  - CDM, 17
- cdmPath\_isAbsolute
  - CDM, 17
- CellX
  - cdm\_Domain, 118
  - cdm\_NonUniformDomain, 143
- CellY
  - cdm\_Domain, 118
  - cdm\_NonUniformDomain, 144
- CellZ
  - cdm\_Domain, 118
  - cdm\_NonUniformDomain, 144
- CheckAddWriteMode
  - cdm\_DFI, 40
  - cdm\_DFI\_NETCDF, 92
- CheckArrayVarInfo
  - cdm\_DFI\_NETCDF, 92
- CheckReadRank
  - cdm\_DFI, 40
  - cdm\_Process, 153
- CheckReadType
  - cdm\_DFI, 40
- CheckStartEnd
  - cdm\_Process, 153
- chkLabel
  - cdm\_TextParser, 163
- chkNode
  - cdm\_TextParser, 163
- Clear
  - cdm\_Domain, 119
  - cdm\_NonUniformDomain, 144
- clear
  - cdm\_ActiveSubDomain, 20
  - cdm\_DFI\_NETCDF::stVarInfo, 182
- CloseFile
  - cdm\_FILE, 128
- ConvDatatypeE2S
  - cdm\_DFI, 41
- ConvDatatypeS2E
  - cdm\_DFI, 41
- CoordinateFile
  - cdm\_NonUniformDomain, 149
- CoordinateFileEndian
  - cdm\_NonUniformDomain, 150
- CoordinateFilePrecision
  - cdm\_NonUniformDomain, 150
- CoordinateFileType
  - cdm\_NonUniformDomain, 150
- copyArray
  - cdm\_Array, 24
  - cdm\_TypeArray, 171
- copyArrayNvari
  - cdm\_Array, 25
  - cdm\_TypeArray, 171
- copyArrayNvari\_to\_ijk
  - cdm\_Array, 25
  - cdm\_TypeArray, 171
- countLabels
  - cdm\_TextParser, 164
- CreateHeadMap
  - cdm\_Process, 154
- CreateRankList
  - cdm\_Process, 154
- CreateRankMap
  - cdm\_Process, 155
- CreateReadStartEnd
  - cdm\_DFI, 41
- CreateSubDomainInfo
  - cdm\_Process, 155
- D\_CDM\_BIG
  - cdm\_Define.h, 192
- D\_CDM\_BYTE
  - cdm\_Define.h, 192
- D\_CDM\_DFITYPE\_CARTESIAN
  - cdm\_Define.h, 192
- D\_CDM\_DFITYPE\_NON\_UNIFORM\_CARTESIAN
  - cdm\_Define.h, 192
- D\_CDM\_DOUBLE
  - cdm\_Define.h, 192
- D\_CDM\_EXT\_BOV
  - cdm\_Define.h, 193
- D\_CDM\_EXT\_BOV\_DATAFILE
  - cdm\_Define.h, 193
- D\_CDM\_EXT\_FUB
  - cdm\_Define.h, 193
- D\_CDM\_EXT\_FUNC
  - cdm\_Define.h, 193
- D\_CDM\_EXT\_NC
  - cdm\_Define.h, 193
- D\_CDM\_EXT\_SPH
  - cdm\_Define.h, 193
- D\_CDM\_EXT\_VTK
  - cdm\_Define.h, 193
- D\_CDM\_EXT\_XYZ
  - cdm\_Define.h, 193
- D\_CDM\_FLOAT
  - cdm\_Define.h, 193
- D\_CDM\_FLOAT32
  - cdm\_Define.h, 193
- D\_CDM\_FLOAT64



- cdm\_Define.h, [193](#)
- D\_CDM\_IJNK
  - cdm\_Define.h, [193](#)
- D\_CDM\_INT
  - cdm\_Define.h, [194](#)
- D\_CDM\_INT16
  - cdm\_Define.h, [194](#)
- D\_CDM\_INT32
  - cdm\_Define.h, [194](#)
- D\_CDM\_INT64
  - cdm\_Define.h, [194](#)
- D\_CDM\_INT8
  - cdm\_Define.h, [194](#)
- D\_CDM\_LITTLE
  - cdm\_Define.h, [194](#)
- D\_CDM\_NIJK
  - cdm\_Define.h, [194](#)
- D\_CDM\_OFF
  - cdm\_Define.h, [194](#)
- D\_CDM\_ON
  - cdm\_Define.h, [194](#)
- D\_CDM\_UINT16
  - cdm\_Define.h, [194](#)
- D\_CDM\_UINT32
  - cdm\_Define.h, [194](#)
- D\_CDM\_UINT64
  - cdm\_Define.h, [194](#)
- D\_CDM\_UINT8
  - cdm\_Define.h, [195](#)
- DBSWAPVEC
  - cdm\_endianUtil.h, [202](#)
- DFI\_Domain
  - cdm\_DFI, [70](#)
- DFI\_FieldFileNameFormat
  - cdm\_DFI\_FUB, [88](#)
- DFI\_Finfo
  - cdm\_DFI, [71](#)
- DFI\_Fpath
  - cdm\_DFI, [71](#)
- DFI\_MPI
  - cdm\_DFI, [71](#)
- DFI\_Process
  - cdm\_DFI, [71](#)
- DFI\_TimeSlice
  - cdm\_DFI, [71](#)
- DFI\_Unit
  - cdm\_DFI, [71](#)
- DFI\_VisIt
  - cdm\_DFI, [72](#)
- DFIType
  - cdm\_FileInfo, [134](#)
- data
  - cdm\_DFI\_NETCDF::stVarInfo, [182](#)
- DataDims
  - cdm\_DFI\_SPH, [108](#)
- DataType
  - cdm\_FileInfo, [134](#)
- difference
  - cdm\_UnitElem, [178](#)
- dims
  - cdm\_DFI\_NETCDF::stVarInfo, [182](#)
- DirectoryPath
  - cdm\_FileInfo, [134](#)
- E\_CDM\_ARRAYSHAPE
  - CDM, [11](#)
- E\_CDM\_ARRAYSHAPE\_UNKNOWN
  - CDM, [11](#)
- E\_CDM\_BIG
  - CDM, [12](#)
- E\_CDM\_DFITYPE
  - CDM, [11](#)
- E\_CDM\_DFITYPE\_CARTESIAN
  - CDM, [11](#)
- E\_CDM\_DFITYPE\_NON\_UNIFORM\_CARTESIAN
  - CDM, [11](#)
- E\_CDM\_DFITYPE\_UNKNOWN
  - CDM, [11](#)
- E\_CDM\_DIFFDIV\_REFINEMENT
  - CDM, [16](#)
- E\_CDM\_DIFFDIV\_SAMERES
  - CDM, [16](#)
- E\_CDM\_DTYPE
  - CDM, [11](#)
- E\_CDM\_DTYPE\_UNKNOWN
  - CDM, [11](#)
- E\_CDM\_ENDIANTYPE
  - CDM, [12](#)
- E\_CDM\_ENDIANTYPE\_UNKNOWN
  - CDM, [12](#)
- E\_CDM\_ERROR
  - CDM, [12](#)
- E\_CDM\_ERROR\_INVALID\_DIVNUM
  - CDM, [14](#)
- E\_CDM\_ERROR\_MAKEDIRECTORY
  - CDM, [14](#)
- E\_CDM\_ERROR\_MISMATCH\_NP\_SUBDOMAIN
  - CDM, [14](#)
- E\_CDM\_ERROR\_NOMATCH\_ENDIAN
  - CDM, [13](#)
- E\_CDM\_ERROR\_NUM\_OF\_GUIDECELLS
  - CDM, [14](#)
- E\_CDM\_ERROR\_OPEN\_COORDINATEFILE
  - CDM, [13](#)
- E\_CDM\_ERROR\_OPEN\_FIELDDATA
  - CDM, [14](#)
- E\_CDM\_ERROR\_OPEN\_SBDM
  - CDM, [14](#)
- E\_CDM\_ERROR\_READ\_BOV\_FILE
  - CDM, [14](#)
- E\_CDM\_ERROR\_READ\_COORDINATEFILE
  - CDM, [13](#)
- E\_CDM\_ERROR\_READ\_DFI\_BCFLAGID
  - CDM, [12](#)
- E\_CDM\_ERROR\_READ\_DFI\_CELLID
  - CDM, [12](#)
- E\_CDM\_ERROR\_READ\_DFI\_COORDINATEFILE

- CDM, [13](#)
- E\_CDM\_ERROR\_READ\_DFI\_COORDINATEFILEENDIAN
  - CDM, [13](#)
- E\_CDM\_ERROR\_READ\_DFI\_COORDINATEFILEPRECISION
  - CDM, [13](#)
- E\_CDM\_ERROR\_READ\_DFI\_COORDINATEFILETYPE
  - CDM, [13](#)
- E\_CDM\_ERROR\_READ\_DFI\_DATATYPE
  - CDM, [12](#)
- E\_CDM\_ERROR\_READ\_DFI\_DFITYPE
  - CDM, [13](#)
- E\_CDM\_ERROR\_READ\_DFI\_DIRECTORYPATH
  - CDM, [12](#)
- E\_CDM\_ERROR\_READ\_DFI\_ENDIAN
  - CDM, [12](#)
- E\_CDM\_ERROR\_READ\_DFI\_FIELDFILENAMEFORMAT
  - CDM, [13](#)
- E\_CDM\_ERROR\_READ\_DFI\_FILEFORMAT
  - CDM, [12](#)
- E\_CDM\_ERROR\_READ\_DFI\_FILELIST
  - CDM, [14](#)
- E\_CDM\_ERROR\_READ\_DFI\_FILEPATH\_PROCESS
  - CDM, [12](#)
- E\_CDM\_ERROR\_READ\_DFI\_GLOBALDIVISION
  - CDM, [12](#)
- E\_CDM\_ERROR\_READ\_DFI\_GLOBALORIGIN
  - CDM, [12](#)
- E\_CDM\_ERROR\_READ\_DFI\_GLOBALREGION
  - CDM, [12](#)
- E\_CDM\_ERROR\_READ\_DFI\_GLOBALVOXEL
  - CDM, [12](#)
- E\_CDM\_ERROR\_READ\_DFI\_GUIDECELL
  - CDM, [12](#)
- E\_CDM\_ERROR\_READ\_DFI\_HEADINDEX
  - CDM, [12](#)
- E\_CDM\_ERROR\_READ\_DFI\_HOSTNAME
  - CDM, [12](#)
- E\_CDM\_ERROR\_READ\_DFI\_ID
  - CDM, [12](#)
- E\_CDM\_ERROR\_READ\_DFI\_MAX
  - CDM, [13](#)
- E\_CDM\_ERROR\_READ\_DFI\_MIN
  - CDM, [13](#)
- E\_CDM\_ERROR\_READ\_DFI\_NETCDF
  - CDM, [13](#)
- E\_CDM\_ERROR\_READ\_DFI\_NO\_MINMAX
  - CDM, [13](#)
- E\_CDM\_ERROR\_READ\_DFI\_NO\_RANK
  - CDM, [12](#)
- E\_CDM\_ERROR\_READ\_DFI\_NO\_SLICE
  - CDM, [13](#)
- E\_CDM\_ERROR\_READ\_DFI\_NUMVARIABLES
  - CDM, [12](#)
- E\_CDM\_ERROR\_READ\_DFI\_PREFIX
  - CDM, [12](#)
- E\_CDM\_ERROR\_READ\_DFI\_STEP
  - CDM, [13](#)
- E\_CDM\_ERROR\_READ\_DFI\_TAILINDEX
  - CDM, [12](#)
- E\_CDM\_ERROR\_READ\_DFI\_TIME
  - CDM, [13](#)
- E\_CDM\_ERROR\_READ\_DFI\_TIMESLICEDIRECTORY
  - CDM, [12](#)
- E\_CDM\_ERROR\_READ\_DFI\_VOXELSIZE
  - CDM, [12](#)
- E\_CDM\_ERROR\_READ\_DOMAIN
  - CDM, [13](#)
- E\_CDM\_ERROR\_READ\_FIELD\_AVERAGED\_RECORD
  - CDM, [14](#)
- E\_CDM\_ERROR\_READ\_FIELD\_DATA\_RECORD
  - CDM, [14](#)
- E\_CDM\_ERROR\_READ\_FIELD\_HEADER\_RECORD
  - CDM, [14](#)
- E\_CDM\_ERROR\_READ\_FIELDDATA\_FILE
  - CDM, [13](#)
- E\_CDM\_ERROR\_READ\_FILEINFO
  - CDM, [13](#)
- E\_CDM\_ERROR\_READ\_FILELIST\_CCORDINATEFILENAME
  - CDM, [13](#)
- E\_CDM\_ERROR\_READ\_FILELIST\_FIELDDATAFILENAME
  - CDM, [13](#)
- E\_CDM\_ERROR\_READ\_FILELIST\_ID
  - CDM, [13](#)
- E\_CDM\_ERROR\_READ\_FILEPATH
  - CDM, [13](#)
- E\_CDM\_ERROR\_READ\_FUB\_REC1
  - CDM, [14](#)
- E\_CDM\_ERROR\_READ\_FUB\_REC2
  - CDM, [14](#)
- E\_CDM\_ERROR\_READ\_INDEXFILE\_OPENERERROR
  - CDM, [13](#)
- E\_CDM\_ERROR\_READ\_MPI
  - CDM, [13](#)
- E\_CDM\_ERROR\_READ\_NETCDF\_FUNC
  - CDM, [14](#)
- E\_CDM\_ERROR\_READ\_NETCDF\_MISMATCH\_TYPE
  - CDM, [13](#)
- E\_CDM\_ERROR\_READ\_NETCDF\_VAR\_1D
  - CDM, [14](#)
- E\_CDM\_ERROR\_READ\_PROCESS
  - CDM, [13](#)
- E\_CDM\_ERROR\_READ\_PROCFILE\_OPENERERROR
  - CDM, [13](#)
- E\_CDM\_ERROR\_READ\_SBDM\_CONTENTS
  - CDM, [14](#)
- E\_CDM\_ERROR\_READ\_SBDM\_DIV
  - CDM, [14](#)
- E\_CDM\_ERROR\_READ\_SBDM\_FORMAT
  - CDM, [14](#)

- CDM, [14](#)
- E\_CDM\_ERROR\_READ\_SBDM\_HEADER
  - CDM, [14](#)
- E\_CDM\_ERROR\_READ\_SPH\_FILE
  - CDM, [13](#)
- E\_CDM\_ERROR\_READ\_SPH\_REC1
  - CDM, [13](#)
- E\_CDM\_ERROR\_READ\_SPH\_REC2
  - CDM, [13](#)
- E\_CDM\_ERROR\_READ\_SPH\_REC3
  - CDM, [13](#)
- E\_CDM\_ERROR\_READ\_SPH\_REC4
  - CDM, [13](#)
- E\_CDM\_ERROR\_READ\_SPH\_REC5
  - CDM, [13](#)
- E\_CDM\_ERROR\_READ\_SPH\_REC6
  - CDM, [13](#)
- E\_CDM\_ERROR\_READ\_SPH\_REC7
  - CDM, [13](#)
- E\_CDM\_ERROR\_READ\_TIMESLICE
  - CDM, [13](#)
- E\_CDM\_ERROR\_READ\_UNIT
  - CDM, [13](#)
- E\_CDM\_ERROR\_SBDM\_NUMDOMAIN\_ZERO
  - CDM, [14](#)
- E\_CDM\_ERROR\_TEXTPARSER
  - CDM, [13](#)
- E\_CDM\_ERROR\_UNDEFINED\_FIELDFILENAMEFOR-  
MAT
  - CDM, [13](#)
- E\_CDM\_ERROR\_UNMATCH\_NUM\_OF\_VARIABLES
  - CDM, [13](#)
- E\_CDM\_ERROR\_UNMATCH\_VOXELSIZE
  - CDM, [13](#)
- E\_CDM\_ERROR\_WRITE\_DFI\_NETCDF
  - CDM, [15](#)
- E\_CDM\_ERROR\_WRITE\_DOMAIN
  - CDM, [14](#)
- E\_CDM\_ERROR\_WRITE\_FIELD\_AVERAGED\_RECORD
  - CDM, [14](#)
- E\_CDM\_ERROR\_WRITE\_FIELD\_DATA\_RECORD
  - CDM, [14](#)
- E\_CDM\_ERROR\_WRITE\_FIELD\_HEADER\_RECORD
  - CDM, [14](#)
- E\_CDM\_ERROR\_WRITE\_FILEINFO
  - CDM, [14](#)
- E\_CDM\_ERROR\_WRITE\_FILEPATH
  - CDM, [14](#)
- E\_CDM\_ERROR\_WRITE\_GRIDFILE
  - CDM, [15](#)
- E\_CDM\_ERROR\_WRITE\_INDEXFILE\_OPENERERROR
  - CDM, [14](#)
- E\_CDM\_ERROR\_WRITE\_INDEXFILENAME\_EMPTY
  - CDM, [14](#)
- E\_CDM\_ERROR\_WRITE\_MPI
  - CDM, [14](#)
- E\_CDM\_ERROR\_WRITE\_PREFIX\_EMPTY
  - CDM, [14](#)
- E\_CDM\_ERROR\_WRITE\_PROCESS
  - CDM, [14](#)
- E\_CDM\_ERROR\_WRITE\_PROCFILE\_OPENERERROR
  - CDM, [14](#)
- E\_CDM\_ERROR\_WRITE\_PROCFILENAME\_EMPTY
  - CDM, [14](#)
- E\_CDM\_ERROR\_WRITE\_RANKID
  - CDM, [14](#)
- E\_CDM\_ERROR\_WRITE\_SPH\_REC1
  - CDM, [14](#)
- E\_CDM\_ERROR\_WRITE\_SPH\_REC2
  - CDM, [14](#)
- E\_CDM\_ERROR\_WRITE\_SPH\_REC3
  - CDM, [14](#)
- E\_CDM\_ERROR\_WRITE\_SPH\_REC4
  - CDM, [14](#)
- E\_CDM\_ERROR\_WRITE\_SPH\_REC5
  - CDM, [14](#)
- E\_CDM\_ERROR\_WRITE\_SPH\_REC6
  - CDM, [14](#)
- E\_CDM\_ERROR\_WRITE\_SPH\_REC7
  - CDM, [14](#)
- E\_CDM\_ERROR\_WRITE\_TIMESLICE
  - CDM, [14](#)
- E\_CDM\_ERROR\_WRITE\_UNIT
  - CDM, [14](#)
- E\_CDM\_ERROR\_WRITE\_VISIT
  - CDM, [15](#)
- E\_CDM\_ERRORCODE
  - CDM, [12](#)
- E\_CDM\_FILE\_TYPE
  - CDM, [15](#)
- E\_CDM\_FILE\_TYPE\_ASCII
  - CDM, [15](#)
- E\_CDM\_FILE\_TYPE\_BINARY
  - CDM, [15](#)
- E\_CDM\_FILE\_TYPE\_DEFAULT
  - CDM, [15](#)
- E\_CDM\_FILE\_TYPE\_FBINARY
  - CDM, [15](#)
- E\_CDM\_FLOAT32
  - CDM, [12](#)
- E\_CDM\_FLOAT64
  - CDM, [12](#)
- E\_CDM\_FMT\_AVS
  - CDM, [15](#)
- E\_CDM\_FMT\_BOV
  - CDM, [15](#)
- E\_CDM\_FMT\_FUB
  - CDM, [15](#)
- E\_CDM\_FMT\_FUB\_COD
  - CDM, [15](#)
- E\_CDM\_FMT\_NETCDF4
  - CDM, [15](#)
- E\_CDM\_FMT\_PLOT3D
  - CDM, [15](#)
- E\_CDM\_FMT\_SPH

CDM, [15](#)  
 E\_CDM\_FMT\_UNKNOWN  
     CDM, [15](#)  
 E\_CDM\_FMT\_VTK  
     CDM, [15](#)  
 E\_CDM\_FNAME\_CUSTOM  
     CDM, [16](#)  
 E\_CDM\_FNAME\_DEFAULT  
     CDM, [16](#)  
 E\_CDM\_FNAME\_RANK  
     CDM, [16](#)  
 E\_CDM\_FNAME\_RANK\_STEP  
     CDM, [16](#)  
 E\_CDM\_FNAME\_STEP\_RANK  
     CDM, [16](#)  
 E\_CDM\_FORMAT  
     CDM, [15](#)  
 E\_CDM\_IJKN  
     CDM, [11](#)  
 E\_CDM\_INT16  
     CDM, [11](#)  
 E\_CDM\_INT32  
     CDM, [11](#)  
 E\_CDM\_INT64  
     CDM, [11](#)  
 E\_CDM\_INT8  
     CDM, [11](#)  
 E\_CDM\_LITTLE  
     CDM, [12](#)  
 E\_CDM\_NIJK  
     CDM, [11](#)  
 E\_CDM\_OFF  
     CDM, [15](#)  
 E\_CDM\_ON  
     CDM, [15](#)  
 E\_CDM\_ONOFF  
     CDM, [15](#)  
 E\_CDM\_OUTPUT\_FNAME  
     CDM, [15](#)  
 E\_CDM\_READTYPE  
     CDM, [16](#)  
 E\_CDM\_READTYPE\_UNKNOWN  
     CDM, [16](#)  
 E\_CDM\_SAMEDIV\_REFINEMENT  
     CDM, [16](#)  
 E\_CDM\_SAMEDIV\_SAMERES  
     CDM, [16](#)  
 E\_CDM\_SUCCESS  
     CDM, [12](#)  
 E\_CDM\_UINT16  
     CDM, [12](#)  
 E\_CDM\_UINT32  
     CDM, [12](#)  
 E\_CDM\_UINT64  
     CDM, [12](#)  
 E\_CDM\_UINT8  
     CDM, [11](#)  
 E\_CDM\_WARN\_GETUNIT

CDM, [15](#)  
 E\_FUB\_NONE  
     cdm\_FieldFileNameFormatElem.h, [203](#)  
 E\_FUB\_RANK  
     cdm\_FieldFileNameFormatElem.h, [203](#)  
 E\_FUB\_RANK\_STEP  
     cdm\_FieldFileNameFormatElem.h, [203](#)  
 E\_FUB\_STEP  
     cdm\_FieldFileNameFormatElem.h, [203](#)  
 E\_FUB\_STEP\_RANK  
     cdm\_FieldFileNameFormatElem.h, [203](#)  
 Endian  
     cdm\_FileInfo, [134](#)  
 endian  
     cdm\_DFI\_NETCDF::stVarInfo, [182](#)  
 ExtractPathWithoutExt  
     CDM, [17](#)  
  
 FUB\_FNAME\_TYPE  
     cdm\_FieldFileNameFormatElem.h, [203](#)  
 FieldFilenameFormat  
     cdm\_FileInfo, [134](#)  
 FileExist  
     cdm\_FieldFileNameFormat, [123](#)  
     cdm\_FieldFileNameFormatElem, [125](#)  
 FileFormat  
     cdm\_FileInfo, [135](#)  
 FileName  
     cdm\_FieldFileNameFormatElem, [125](#)  
 FileNameFormat  
     cdm\_FieldFileNameFormatElem, [126](#)  
 FnameFormat  
     cdm\_FieldFileNameFormatElem, [126](#)  
 FnameLabel  
     cdm\_FieldFileNameFormatElem, [126](#)  
  
 Generate\_DFI\_Name  
     cdm\_DFI, [42](#)  
 Generate\_Directory\_Path  
     cdm\_DFI, [42](#)  
 Generate\_FieldFileName  
     cdm\_DFI, [42](#)  
 Generate\_FileName  
     cdm\_DFI, [42](#)  
 GenerateFileName  
     cdm\_FieldFileNameFormat, [123](#)  
     cdm\_FieldFileNameFormatElem, [125](#)  
 get\_cdm\_Datasize  
     cdm\_DFI, [43](#)  
 get\_dfi\_fname  
     cdm\_DFI, [43](#)  
 getArrayLength  
     cdm\_Array, [25](#)  
 GetArrayShape  
     cdm\_DFI, [43](#)  
 getArrayShape  
     cdm\_Array, [25](#)  
 GetArrayShapeString  
     cdm\_DFI, [43](#)

- getArrayShapeString
  - cdm\_Array, 25
- getArraySize
  - cdm\_Array, 25
- getArraySizeInt
  - cdm\_Array, 26
- getBufSize
  - cdm\_DFI, 43
- GetCdmType
  - cdm\_DFI\_NETCDF, 92
- GetCoordinateFile
  - cdm\_Domain, 119
  - cdm\_NonUniformDomain, 144
- GetCoordinateFileEndian
  - cdm\_Domain, 119
  - cdm\_NonUniformDomain, 145
- getCoordinateFileName
  - cdm\_DFI\_FUB, 85
- GetCoordinateFilePrecision
  - cdm\_Domain, 119
  - cdm\_NonUniformDomain, 145
- GetCoordinateFileType
  - cdm\_Domain, 119
  - cdm\_NonUniformDomain, 145
- GetDFIGlobalDivision
  - cdm\_DFI, 46
- GetDFIGlobalVoxel
  - cdm\_DFI, 46
- GetDFIType
  - cdm\_DFI, 46
- GetDFITypeString
  - cdm\_DFI, 46
- getData
  - cdm\_Array, 26
  - cdm\_TypeArray, 172
- GetDataType
  - cdm\_DFI, 45
- getDataType
  - cdm\_Array, 26
- GetDataTypeString
  - cdm\_DFI, 45
- getDataTypeString
  - cdm\_Array, 26
- GetDimName
  - cdm\_DFI\_NETCDF, 92
- GetDisitNo
  - cdm\_FieldFileNameFormatElem, 125
- GetFieldFileNameFormatElem
  - cdm\_FieldFileNameFormat, 123
- GetFileFormat
  - cdm\_DFI, 46
- GetFileFormatString
  - cdm\_DFI, 46
- getFileNameFromFileList
  - cdm\_DFI, 47
- getGc
  - cdm\_Array, 26
- getGcInt
  - cdm\_Array, 26
- getHeadIndex
  - cdm\_Array, 27
- GetLabelList
  - cdm\_FieldFileNameFormat, 123
- getMinMax
  - cdm\_DFI, 47
  - cdm\_TimeSlice, 167
- GetNcType
  - cdm\_DFI\_NETCDF, 93
- GetNcUnit
  - cdm\_DFI\_NETCDF, 93
- GetNodeStr
  - cdm\_TextParser, 164
- GetNumGuideCell
  - cdm\_DFI, 47
- GetNumVariables
  - cdm\_DFI, 47
- getNvari
  - cdm\_Array, 27
- getNvariInt
  - cdm\_Array, 27
- GetPos
  - cdm\_ActiveSubDomain, 20
- getTPPtr
  - cdm\_TextParser, 164
- getTPinstance
  - cdm\_TextParser, 164
- getTailIndex
  - cdm\_Array, 27
- GetUnit
  - cdm\_DFI, 47
  - cdm\_Unit, 174
- GetUnitElem
  - cdm\_DFI, 48
  - cdm\_Unit, 175
- GetValue
  - cdm\_TextParser, 164, 165
- GetVarInfo
  - cdm\_DFI\_NETCDF, 93
- getVariableName
  - cdm\_DFI, 48
  - cdm\_FileInfo, 133
- GetVector
  - cdm\_TextParser, 165
- getVectorMinMax
  - cdm\_DFI, 48
  - cdm\_TimeSlice, 167
- getVersionInfo
  - cdm\_DFI, 48
- GetcdmDomain
  - cdm\_DFI, 44
- GetcdmFieldFileNameFormat
  - cdm\_DFI, 44
  - cdm\_DFI\_FUB, 85
- GetcdmFileInfo
  - cdm\_DFI, 44
- GetcdmFilePath

- cdm\_DFI, 44
- GetcdmMPI
  - cdm\_DFI, 44
- GetcdmProcess
  - cdm\_DFI, 44
- GetcdmRankNoPrefix
  - cdm\_DFI, 45
- GetcdmTimeSlice
  - cdm\_DFI, 45
- GetcdmUnit
  - cdm\_DFI, 45
- GetcdmVisIt
  - cdm\_DFI, 45
- GlobalDivision
  - cdm\_Domain, 121
- GlobalOrigin
  - cdm\_Domain, 121
- GlobalRegion
  - cdm\_Domain, 121
- GlobalVoxel
  - cdm\_Domain, 121
- GuideCell
  - cdm\_FileInfo, 135
- HeadIndex
  - cdm\_Rank, 159
- headT
  - cdm\_Process, 153
- HostName
  - cdm\_Rank, 159
- hval
  - cdm\_TypeArray, 172
- id
  - cdm\_DFI\_NETCDF::stDimInfo, 181
  - cdm\_DFI\_NETCDF::stVarInfo, 182
- instanceArray
  - cdm\_Array, 27–29
- interp\_coarse
  - cdm\_Array, 29
- isMatchEndianSbdmMagick
  - cdm\_Process, 156
- LabelList
  - cdm\_FieldFileNameFormat, 123
- len
  - cdm\_DFI\_NETCDF::stDimInfo, 181
- m\_NC\_Unit
  - cdm\_DFI\_NETCDF, 99
- m\_RankID
  - cdm\_DFI, 73
- m\_Sz
  - cdm\_Array, 31
- m\_SzI
  - cdm\_Array, 31
- m\_addMode
  - cdm\_FILE, 130
- m\_bgrid\_interp\_flag
  - cdm\_DFI, 72
- m\_comm
  - cdm\_DFI, 72
- m\_data
  - cdm\_TypeArray, 173
- m\_directoryPath
  - cdm\_DFI, 72
- m\_dtype
  - cdm\_Array, 29
- m\_fname
  - cdm\_FILE, 130
- m\_format
  - cdm\_FILE, 130
- m\_fp
  - cdm\_FILE, 130
- m\_gc
  - cdm\_Array, 29
- m\_gcl
  - cdm\_Array, 30
- m\_gcl
  - cdm\_Array, 30
- m\_headIndex
  - cdm\_Array, 30
- m\_indexDfiName
  - cdm\_DFI, 72
- m\_input\_type
  - cdm\_DFI, 72
- m\_nameT
  - cdm\_DFI\_NETCDF, 99
- m\_nameX
  - cdm\_DFI\_NETCDF, 99
- m\_nameY
  - cdm\_DFI\_NETCDF, 99
- m\_nameZ
  - cdm\_DFI\_NETCDF, 99
- m\_ncid
  - cdm\_FILE, 130
- m\_nvari
  - cdm\_Array, 30
- m\_nvaril
  - cdm\_Array, 30
- m\_outptr
  - cdm\_TypeArray, 173
- m\_output\_fname
  - cdm\_DFI, 73
- m\_output\_type
  - cdm\_DFI, 73
- m\_output\_type\_coord
  - cdm\_DFI, 73
- m\_pos
  - cdm\_ActiveSubDomain, 21
- m\_rankMap
  - cdm\_Process, 157
- m\_read\_type
  - cdm\_DFI, 73
- m\_readRankList
  - cdm\_DFI, 73
- m\_shape

- cdm\_Array, 30
- m\_sz
  - cdm\_Array, 30
- m\_szl
  - cdm\_Array, 31
- m\_tailIndex
  - cdm\_Array, 31
- m\_varInfoT
  - cdm\_DFI\_NETCDF, 100
- m\_varInfoX
  - cdm\_DFI\_NETCDF, 100
- m\_varInfoY
  - cdm\_DFI\_NETCDF, 100
- m\_varInfoZ
  - cdm\_DFI\_NETCDF, 100
- m\_vecVarInfo
  - cdm\_DFI\_NETCDF, 100
- m\_writeFlag
  - cdm\_DFI\_NETCDF, 100
- MAXPATHLEN
  - cdm\_PathUtil.h, 207
- MPI\_Allgather
  - mpi\_stubs.h, 212
- MPI\_CHAR
  - mpi\_stubs.h, 211
- MPI\_COMM\_WORLD
  - mpi\_stubs.h, 211
- MPI\_Comm
  - mpi\_stubs.h, 212
- MPI\_Comm\_rank
  - mpi\_stubs.h, 212
- MPI\_Comm\_size
  - mpi\_stubs.h, 212
- MPI\_Datatype
  - mpi\_stubs.h, 212
- MPI\_Gather
  - mpi\_stubs.h, 212
- MPI\_INT
  - mpi\_stubs.h, 211
- MPI\_Init
  - mpi\_stubs.h, 212
- MPI\_SUCCESS
  - mpi\_stubs.h, 211
- MakeDirectory
  - cdm\_DFI, 49
- MakeDirectoryPath
  - cdm\_DFI, 49
- MakeDirectorySub
  - cdm\_DFI, 49
- mapElem
  - cdm\_FieldFileNameFormat, 123
- Max
  - cdm\_Slice, 161
- Min
  - cdm\_Slice, 161
- mpi\_stubs.h, 211
  - MPI\_Allgather, 212
  - MPI\_CHAR, 211
  - MPI\_COMM\_WORLD, 211
  - MPI\_Comm, 212
  - MPI\_Comm\_rank, 212
  - MPI\_Comm\_size, 212
  - MPI\_Datatype, 212
  - MPI\_Gather, 212
  - MPI\_INT, 211
  - MPI\_Init, 212
  - MPI\_SUCCESS, 211
- Name
  - cdm\_UnitElem, 178
- name
  - cdm\_DFI\_NETCDF::stDimInfo, 181
  - cdm\_DFI\_NETCDF::stVarInfo, 183
- NodeX
  - cdm\_Domain, 119
  - cdm\_NonUniformDomain, 145
- NodeY
  - cdm\_Domain, 120
  - cdm\_NonUniformDomain, 147
- NodeZ
  - cdm\_Domain, 120
  - cdm\_NonUniformDomain, 147
- normalizeBaseTime
  - cdm\_DFI, 49
- normalizeDeltaT
  - cdm\_DFI, 49
- normalizeIntervalTime
  - cdm\_DFI, 50
- normalizeLastTime
  - cdm\_DFI, 50
- normalizeStartTime
  - cdm\_DFI, 50
- normalizeTime
  - cdm\_DFI, 50
- NumVariables
  - cdm\_FileInfo, 135
- NumberOfGroup
  - cdm\_MPI, 140
- NumberOfRank
  - cdm\_MPI, 140
- OpenReadBinary
  - cdm\_FILE, 129
- OpenWriteAscii
  - cdm\_FILE, 129
- OpenWriteBinary
  - cdm\_FILE, 129
- operator==
  - cdm\_ActiveSubDomain, 20
- pit\_gcXend
  - cdm\_NonUniformDomain, 150
- pit\_gcXsta
  - cdm\_NonUniformDomain, 150
- pit\_gcYend
  - cdm\_NonUniformDomain, 150
- pit\_gcYsta

- cdm\_NonUniformDomain, 150
- pit\_gcZend
  - cdm\_NonUniformDomain, 151
- pit\_gcZsta
  - cdm\_NonUniformDomain, 151
- Pitch
  - cdm\_Domain, 122
- PlotGC
  - cdm\_VisIt, 180
- Prefix
  - cdm\_FileInfo, 135
- Print
  - cdm\_FieldFileNameFormat, 123
- PrintParse
  - cdm\_FieldFileNameFormatElem, 125
- ProcDFIFile
  - cdm\_FilePath, 137
- RankID
  - cdm\_Rank, 159
- RankIdDisit
  - cdm\_FieldFileNameFormatElem, 126
- RankIdKey
  - cdm\_FieldFileNameFormatElem, 126
- RankList
  - cdm\_Process, 157
- RankNoPrefix
  - cdm\_FileInfo, 135
- Read
  - cdm\_Domain, 120
  - cdm\_FieldFileNameFormat, 123
  - cdm\_FieldFileNameFormatElem, 125
  - cdm\_FileInfo, 133
  - cdm\_FilePath, 137
  - cdm\_MPI, 138
  - cdm\_NonUniformDomain, 147
  - cdm\_Process, 156
  - cdm\_Rank, 158
  - cdm\_Slice, 160
  - cdm\_TimeSlice, 168
  - cdm\_Unit, 175
  - cdm\_UnitElem, 177
  - cdm\_VisIt, 179
- Read\_CoordinateFile
  - cdm\_NonUniformDomain, 148
- read\_Datarecord
  - cdm\_DFI, 50
  - cdm\_DFI\_AVs, 76
  - cdm\_DFI\_BOV, 81
  - cdm\_DFI\_FUB, 86
  - cdm\_DFI\_NETCDF, 94
  - cdm\_DFI\_PLOT3D, 102
  - cdm\_DFI\_SPH, 109
  - cdm\_DFI\_VTK, 114
- read\_Func
  - cdm\_DFI\_PLOT3D, 104
- read\_HeaderRecord
  - cdm\_DFI, 52
  - cdm\_DFI\_AVs, 77
  - cdm\_DFI\_BOV, 81
  - cdm\_DFI\_FUB, 86
  - cdm\_DFI\_NETCDF, 94, 96
  - cdm\_DFI\_PLOT3D, 104
  - cdm\_DFI\_SPH, 109
  - cdm\_DFI\_VTK, 114
- read\_averaged
  - cdm\_DFI, 50
  - cdm\_DFI\_AVs, 76
  - cdm\_DFI\_BOV, 80
  - cdm\_DFI\_FUB, 85
  - cdm\_DFI\_NETCDF, 93
  - cdm\_DFI\_PLOT3D, 102
  - cdm\_DFI\_SPH, 109
  - cdm\_DFI\_VTK, 113
- ReadActiveSubdomainFile
  - cdm\_Process, 156
- ReadAdditionalTP
  - cdm\_DFI\_NETCDF, 96
- readBinary
  - cdm\_Array, 29
  - cdm\_TypeArray, 172
- readCoordData
  - cdm\_NonUniformDomain, 148
- readCoordDataCount
  - cdm\_NonUniformDomain, 149
- ReadCoordinateData
  - cdm\_DFI, 52, 53
  - cdm\_DFI\_FUB, 86
- ReadData
  - cdm\_DFI, 55, 56
- ReadFieldData
  - cdm\_DFI, 57
- ReadInit
  - cdm\_DFI, 57
- readTPfile
  - cdm\_TextParser, 165
- RealType
  - cdm\_DFI\_SPH, 108
- reference
  - cdm\_UnitElem, 178
- remove
  - cdm\_TextParser, 166
- SBSWAPVEC
  - cdm\_endianUtil.h, 202
- set\_RankID
  - cdm\_DFI, 59
- set\_input\_type
  - cdm\_DFI, 58
- set\_interp\_flag
  - cdm\_DFI, 58
- set\_output\_fname
  - cdm\_DFI, 58
- set\_output\_type
  - cdm\_DFI, 58
- set\_output\_type\_coord
  - cdm\_DFI, 58
- SetDimName



- cdm\_DFI\_NETCDF, 96
- SetDisitNoFormat
  - cdm\_FieldFileNameFormatElem, 125
- SetFnameFormat
  - cdm\_FieldFileNameFormatElem, 125
- setGridData
  - cdm\_DFI, 60
- setHeadIndex
  - cdm\_Array, 29
- setIntervalStep
  - cdm\_DFI, 61
- setIntervalTime
  - cdm\_DFI, 61
- SetPos
  - cdm\_ActiveSubDomain, 20
- SetTimeSliceFlag
  - cdm\_DFI, 61
- setVariableName
  - cdm\_DFI, 61
  - cdm\_FileInfo, 133
- SetWriteFlag
  - cdm\_DFI\_NETCDF, 97
- SetcdmDomain
  - cdm\_DFI, 59
- SetcdmFieldFileNameFormat
  - cdm\_DFI, 59
  - cdm\_DFI\_FUB, 87
- SetcdmFileInfo
  - cdm\_DFI, 59
- SetcdmFilePath
  - cdm\_DFI, 59
- SetcdmMPI
  - cdm\_DFI, 59
- SetcdmProcess
  - cdm\_DFI, 59
- SetcdmRankNoPrefix
  - cdm\_DFI, 59
- SetcdmTimeSlice
  - cdm\_DFI, 60
- SetcdmUnit
  - cdm\_DFI, 60
- SetcdmVisIt
  - cdm\_DFI, 60
- SliceList
  - cdm\_TimeSlice, 168
- stVarInfo
  - cdm\_DFI\_NETCDF::stVarInfo, 182
- step
  - cdm\_Slice, 161
- StepNoDisit
  - cdm\_FieldFileNameFormatElem, 126
- StepNoKey
  - cdm\_FieldFileNameFormatElem, 126
- stmpd\_printf
  - cdm\_Define.h, 195
- TailIndex
  - cdm\_Rank, 159
- time
  - cdm\_Slice, 162
- TimeSliceDirFlag
  - cdm\_FileInfo, 135
- tp
  - cdm\_TextParser, 166
- type
  - cdm\_DFI\_NETCDF::stVarInfo, 183
- Unit
  - cdm\_UnitElem, 178
- unit
  - cdm\_DFI\_NETCDF::stVarInfo, 183
- UnitList
  - cdm\_Unit, 175
- val
  - cdm\_TypeArray, 172
- VariableName
  - cdm\_FileInfo, 135
- VectorMax
  - cdm\_Slice, 162
- VectorMin
  - cdm\_Slice, 162
- vfvPath\_emitDrive
  - CDM, 17
- VolumeDataDivide
  - cdm\_DFI, 61
- VoxelSize
  - cdm\_Rank, 159
- Write
  - cdm\_Domain, 121
  - cdm\_FieldFileNameFormat, 123
  - cdm\_FieldFileNameFormatElem, 125
  - cdm\_FileInfo, 134
  - cdm\_FilePath, 137
  - cdm\_MPI, 138
  - cdm\_NonUniformDomain, 149
  - cdm\_Process, 156
  - cdm\_Rank, 158
  - cdm\_Slice, 161
  - cdm\_TimeSlice, 168
  - cdm\_Unit, 175
  - cdm\_UnitElem, 178
  - cdm\_VisIt, 180
- write\_AttUnits
  - cdm\_DFI\_NETCDF, 97
- write\_DataRecord
  - cdm\_DFI, 63
  - cdm\_DFI\_AVS, 78
  - cdm\_DFI\_BOV, 82
  - cdm\_DFI\_FUB, 87
  - cdm\_DFI\_NETCDF, 97, 98
  - cdm\_DFI\_PLOT3D, 105
  - cdm\_DFI\_SPH, 111
  - cdm\_DFI\_VTK, 115
- write\_Func
  - cdm\_DFI\_PLOT3D, 105, 106
- write\_GridData

- cdm\_DFI, [63](#)
- cdm\_DFI\_PLOT3D, [106](#)
- write\_HeaderRecord
  - cdm\_DFI, [64](#)
  - cdm\_DFI\_AVS, [78](#)
  - cdm\_DFI\_BOV, [82](#)
  - cdm\_DFI\_FUB, [88](#)
  - cdm\_DFI\_NETCDF, [98](#)
  - cdm\_DFI\_PLOT3D, [106](#)
  - cdm\_DFI\_SPH, [111](#)
  - cdm\_DFI\_VTK, [115](#)
- write\_XYZ
  - cdm\_DFI\_PLOT3D, [106](#)
- write\_ascii\_header
  - cdm\_DFI, [63](#)
  - cdm\_DFI\_AVS, [77](#)
  - cdm\_DFI\_BOV, [81](#)
- write\_averaged
  - cdm\_DFI, [63](#)
  - cdm\_DFI\_AVS, [77](#)
  - cdm\_DFI\_BOV, [82](#)
  - cdm\_DFI\_FUB, [87](#)
  - cdm\_DFI\_NETCDF, [97](#)
  - cdm\_DFI\_PLOT3D, [105](#)
  - cdm\_DFI\_SPH, [111](#)
  - cdm\_DFI\_VTK, [115](#)
- write\_avs\_cord
  - cdm\_DFI\_AVS, [78](#)
- write\_avs\_header
  - cdm\_DFI\_AVS, [78](#)
- WriteAdditionalTP
  - cdm\_DFI\_NETCDF, [99](#)
- writeAscii
  - cdm\_Array, [29](#)
  - cdm\_TypeArray, [172](#)
- writeBinary
  - cdm\_Array, [29](#)
  - cdm\_TypeArray, [173](#)
- WriteCoordinateData
  - cdm\_DFI, [64](#)
  - cdm\_DFI\_FUB, [88](#)
- WriteData
  - cdm\_DFI, [65](#)
- WriteFieldData
  - cdm\_DFI, [66](#)
- WriteFieldDataFile
  - cdm\_DFI, [66](#), [67](#)
- WriteGridFile
  - cdm\_DFI, [67](#)
- WriteIndexDfiFile
  - cdm\_DFI, [67](#)
- WriteInit
  - cdm\_DFI, [67–70](#)
- WriteProcDfiFile
  - cdm\_DFI, [70](#)
- XCoordinates
  - cdm\_NonUniformDomain, [151](#)
- YCoordinates
  - cdm\_NonUniformDomain, [151](#)
- ZCoordinates
  - cdm\_NonUniformDomain, [151](#)