

Cartesian Data Management Library
0.9.0

作成 : Doxygen 1.8.5

Wed Nov 18 2015 13:53:39

Contents

1	ネームスペース索引	1
1.1	ネームスペース一覧	1
2	階層索引	3
2.1	クラス階層	3
3	構成索引	5
3.1	構成	5
4	ファイル索引	7
4.1	ファイル一覧	7
5	ネームスペース	9
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_ENDIANTYPE	12
5.1.2.5	E_CDM_ERRORCODE	12
5.1.2.6	E_CDM_FILE_TYPE	14
5.1.2.7	E_CDM_FORMAT	15
5.1.2.8	E_CDM_ONOFF	15
5.1.2.9	E_CDM_OUTPUT_FNAME	15
5.1.2.10	E_CDM_READTYPE	15
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	16
5.1.3.6	cdmPath_hasDrive	16

5.1.3.7	<code>cdmPath_isAbsolute</code>	16
5.1.3.8	<code>ExtractPathWithoutExt</code>	16
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
6	クラス	19
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	getArraySizeInt	26
6.2.3.14	getData	26
6.2.3.15	getDataType	26
6.2.3.16	getDataTypeString	26
6.2.3.17	getGc	26
6.2.3.18	getGcInt	26
6.2.3.19	getHeadIndex	27
6.2.3.20	getNvari	27
6.2.3.21	getNvariInt	27
6.2.3.22	getTailIndex	27
6.2.3.23	instanceArray	27
6.2.3.24	instanceArray	28
6.2.3.25	instanceArray	28
6.2.3.26	instanceArray	28
6.2.3.27	instanceArray	28
6.2.3.28	instanceArray	28
6.2.3.29	instanceArray	28
6.2.3.30	instanceArray	28
6.2.3.31	instanceArray	28
6.2.3.32	instanceArray	28
6.2.3.33	instanceArray	29
6.2.3.34	instanceArray	29
6.2.3.35	interp_coarse	29
6.2.3.36	readBinary	29
6.2.3.37	setHeadIndex	29
6.2.3.38	writeAscii	29
6.2.3.39	writeBinary	29
6.2.4	変数	29
6.2.4.1	m_dtype	29
6.2.4.2	m_gc	30
6.2.4.3	m_gcl	30
6.2.4.4	m_gcl	30
6.2.4.5	m_headIndex	30
6.2.4.6	m_nvari	30
6.2.4.7	m_nvaril	30
6.2.4.8	m_shape	30
6.2.4.9	m_sz	30
6.2.4.10	m_Sz	31
6.2.4.11	m_szl	31
6.2.4.12	m_Szl	31

6.2.4.13	m_tailIndex	31
6.3	クラス cdm_DFI	31
6.3.1	説明	38
6.3.2	コンストラクタとデストラクタ	38
6.3.2.1	cdm_DFI	38
6.3.2.2	~cdm_DFI	38
6.3.3	関数	38
6.3.3.1	AddTimeSlice	38
6.3.3.2	AddTimeSlice	38
6.3.3.3	AddUnit	39
6.3.3.4	cdm_Create_dfiProcessInfo	39
6.3.3.5	CheckAddWriteMode	39
6.3.3.6	CheckReadRank	39
6.3.3.7	CheckReadType	40
6.3.3.8	ConvDatatypeE2S	40
6.3.3.9	ConvDatatypeS2E	40
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	GetcdmFileInfo	44
6.3.3.22	GetcdmFilePath	44
6.3.3.23	GetcdmMPI	44
6.3.3.24	GetcdmProcess	44
6.3.3.25	GetcdmRankNoPrefix	45
6.3.3.26	GetcdmTimeSlice	45
6.3.3.27	GetcdmUnit	45
6.3.3.28	GetcdmVisIt	45
6.3.3.29	GetDataType	45
6.3.3.30	GetDataTypeString	45
6.3.3.31	GetDFIGlobalDivision	46
6.3.3.32	GetDFIGlobalVoxel	46
6.3.3.33	GetDFIType	46

6.3.3.34	GetDFITypeString	46
6.3.3.35	GetFileFormat	46
6.3.3.36	GetFileFormatString	46
6.3.3.37	getMinMax	46
6.3.3.38	GetNumGuideCell	47
6.3.3.39	GetNumVariables	47
6.3.3.40	GetUnit	47
6.3.3.41	GetUnitElem	47
6.3.3.42	getVariableName	47
6.3.3.43	getVectorMinMax	48
6.3.3.44	getVersionInfo	48
6.3.3.45	MakeDirectory	48
6.3.3.46	MakeDirectoryPath	48
6.3.3.47	MakeDirectorySub	48
6.3.3.48	normalizeBaseTime	49
6.3.3.49	normalizeDelteT	49
6.3.3.50	normalizeIntervalTime	49
6.3.3.51	normalizeLastTime	49
6.3.3.52	normalizeStartTime	49
6.3.3.53	normalizeTime	49
6.3.3.54	read_averaged	50
6.3.3.55	read_Datarecord	50
6.3.3.56	read_HeaderRecord	50
6.3.3.57	ReadData	51
6.3.3.58	ReadData	51
6.3.3.59	ReadData	51
6.3.3.60	ReadData	51
6.3.3.61	ReadData	53
6.3.3.62	ReadFieldData	53
6.3.3.63	ReadInit	54
6.3.3.64	set_input_type	54
6.3.3.65	set_interp_flag	54
6.3.3.66	set_output_fname	55
6.3.3.67	set_output_type	55
6.3.3.68	set_output_type_coord	55
6.3.3.69	set_RankID	55
6.3.3.70	SetcdmDomain	55
6.3.3.71	SetcdmFilePath	56
6.3.3.72	SetcdmMPI	56
6.3.3.73	SetcdmProcess	56

6.3.3.74	SetcdmRankNoPrefix	56
6.3.3.75	SetcdmTimeSlice	56
6.3.3.76	SetcdmUnit	56
6.3.3.77	SetcdmVisIt	56
6.3.3.78	setGridData	56
6.3.3.79	setGridData	57
6.3.3.80	setIntervalStep	57
6.3.3.81	setIntervalTime	57
6.3.3.82	SetTimeSliceFlag	57
6.3.3.83	setVariableName	58
6.3.3.84	VolumeDataDivide	59
6.3.3.85	VolumeDataDivide	59
6.3.3.86	write_ascii_header	59
6.3.3.87	write_averaged	59
6.3.3.88	write_DataRecord	59
6.3.3.89	write_GridData	60
6.3.3.90	write_HeaderRecord	60
6.3.3.91	WriteData	60
6.3.3.92	WriteData	61
6.3.3.93	WriteData	61
6.3.3.94	WriteFieldData	61
6.3.3.95	WriteFieldDataFile	62
6.3.3.96	WriteFieldDataFile	62
6.3.3.97	WriteFieldDataFile	62
6.3.3.98	WriteGridFile	63
6.3.3.99	WriteIndexDfiFile	63
6.3.3.100	WriteIndexDfiFile	63
6.3.3.101	WriteInit	63
6.3.3.102	WriteInit	64
6.3.3.103	WriteInit	64
6.3.3.104	WriteInit	65
6.3.3.105	WriteInit	65
6.3.3.106	WriteInit	65
6.3.3.107	WriteProcDfiFile	66
6.3.4	変数	66
6.3.4.1	DFI_Domain	66
6.3.4.2	DFI_Finfo	66
6.3.4.3	DFI_Fpath	66
6.3.4.4	DFI_MPI	67
6.3.4.5	DFI_Process	67

6.3.4.6	DFI_TimeSlice	67
6.3.4.7	DFI_Unit	67
6.3.4.8	DFI_VisIt	67
6.3.4.9	m_bgrid_interp_flag	67
6.3.4.10	m_comm	68
6.3.4.11	m_directoryPath	68
6.3.4.12	m_indexDfiName	68
6.3.4.13	m_input_type	68
6.3.4.14	m_output_fname	68
6.3.4.15	m_output_type	68
6.3.4.16	m_output_type_coord	68
6.3.4.17	m_RankID	68
6.3.4.18	m_read_type	69
6.3.4.19	m_readRankList	69
6.4	クラス cdm_DFI_AVS	69
6.4.1	説明	70
6.4.2	コンストラクタとデストラクタ	70
6.4.2.1	cdm_DFI_AVS	70
6.4.2.2	cdm_DFI_AVS	70
6.4.2.3	~cdm_DFI_AVS	70
6.4.3	関数	70
6.4.3.1	read_averaged	70
6.4.3.2	read_Datarecord	71
6.4.3.3	read_HeaderRecord	71
6.4.3.4	write_ascii_header	72
6.4.3.5	write_averaged	72
6.4.3.6	write_avs_cord	72
6.4.3.7	write_avs_header	73
6.4.3.8	write_DataRecord	73
6.4.3.9	write_HeaderRecord	73
6.5	クラス cdm_DFI_BOV	73
6.5.1	説明	74
6.5.2	コンストラクタとデストラクタ	74
6.5.2.1	cdm_DFI_BOV	74
6.5.2.2	cdm_DFI_BOV	74
6.5.2.3	~cdm_DFI_BOV	75
6.5.3	関数	75
6.5.3.1	read_averaged	75
6.5.3.2	read_Datarecord	75
6.5.3.3	read_HeaderRecord	76

6.5.3.4	write_ascii_header	76
6.5.3.5	write_averaged	76
6.5.3.6	write_DataRecord	77
6.5.3.7	write_HeaderRecord	77
6.6	クラス cdm_DFI_NETCDF	77
6.6.1	説明	80
6.6.2	コンストラクタとデストラクタ	80
6.6.2.1	cdm_DFI_NETCDF	80
6.6.2.2	cdm_DFI_NETCDF	80
6.6.2.3	~cdm_DFI_NETCDF	80
6.6.3	関数	80
6.6.3.1	AddNcUnit	80
6.6.3.2	CheckAddWriteMode	81
6.6.3.3	CheckArrayVarInfo	81
6.6.3.4	GetCdmType	81
6.6.3.5	GetDimName	81
6.6.3.6	GetNcType	82
6.6.3.7	GetNcUnit	82
6.6.3.8	GetVarInfo	82
6.6.3.9	read_averaged	82
6.6.3.10	read_Datarecord	83
6.6.3.11	read_Datarecord	83
6.6.3.12	read_HeaderRecord	84
6.6.3.13	read_HeaderRecord	85
6.6.3.14	ReadAdditionalTP	85
6.6.3.15	SetDimName	85
6.6.3.16	SetWriteFlag	86
6.6.3.17	write_AttUnits	86
6.6.3.18	write_averaged	86
6.6.3.19	write_DataRecord	86
6.6.3.20	write_DataRecord	87
6.6.3.21	write_HeaderRecord	87
6.6.3.22	write_HeaderRecord	87
6.6.3.23	WriteAdditionalTP	88
6.6.4	変数	88
6.6.4.1	m_nameT	88
6.6.4.2	m_nameX	88
6.6.4.3	m_nameY	88
6.6.4.4	m_nameZ	88
6.6.4.5	m_NC_Unit	89

6.6.4.6	m_varInfoT	89
6.6.4.7	m_varInfoX	89
6.6.4.8	m_varInfoY	89
6.6.4.9	m_varInfoZ	89
6.6.4.10	m_vecVarInfo	89
6.6.4.11	m_writeFlag	89
6.7	クラス cdm_DFI_PLOT3D	89
6.7.1	説明	90
6.7.2	コンストラクタとデストラクタ	91
6.7.2.1	cdm_DFI_PLOT3D	91
6.7.2.2	cdm_DFI_PLOT3D	91
6.7.2.3	~cdm_DFI_PLOT3D	91
6.7.3	関数	91
6.7.3.1	read_averaged	91
6.7.3.2	read_Datarecord	92
6.7.3.3	read_Func	93
6.7.3.4	read_Func	93
6.7.3.5	read_HeaderRecord	93
6.7.3.6	write_averaged	94
6.7.3.7	write_DataRecord	94
6.7.3.8	write_Func	94
6.7.3.9	write_Func	95
6.7.3.10	write_GridData	95
6.7.3.11	write_HeaderRecord	95
6.7.3.12	write_XYZ	95
6.7.3.13	write_XYZ	95
6.8	クラス cdm_DFI_SPH	96
6.8.1	説明	97
6.8.2	列挙型	97
6.8.2.1	DataDims	97
6.8.2.2	RealType	97
6.8.3	コンストラクタとデストラクタ	97
6.8.3.1	cdm_DFI_SPH	97
6.8.3.2	cdm_DFI_SPH	97
6.8.3.3	~cdm_DFI_SPH	98
6.8.4	関数	98
6.8.4.1	read_averaged	98
6.8.4.2	read_Datarecord	98
6.8.4.3	read_HeaderRecord	99
6.8.4.4	write_averaged	100

6.8.4.5	write_DataRecord	100
6.8.4.6	write_HeaderRecord	100
6.9	クラス cdm_DFI_VTK	101
6.9.1	説明	102
6.9.2	コンストラクタとデストラクタ	102
6.9.2.1	cdm_DFI_VTK	102
6.9.2.2	cdm_DFI_VTK	102
6.9.2.3	~cdm_DFI_VTK	102
6.9.3	関数	102
6.9.3.1	read_averaged	102
6.9.3.2	read_Datarecord	103
6.9.3.3	read_HeaderRecord	103
6.9.3.4	write_averaged	104
6.9.3.5	write_DataRecord	104
6.9.3.6	write_HeaderRecord	104
6.10	クラス cdm_Domain	105
6.10.1	説明	106
6.10.2	コンストラクタとデストラクタ	106
6.10.2.1	cdm_Domain	106
6.10.2.2	cdm_Domain	106
6.10.2.3	cdm_Domain	106
6.10.2.4	~cdm_Domain	106
6.10.3	関数	106
6.10.3.1	CellX	106
6.10.3.2	CellY	107
6.10.3.3	CellZ	107
6.10.3.4	Clear	107
6.10.3.5	GetCoordinateFile	107
6.10.3.6	GetCoordinateFileEndian	107
6.10.3.7	GetCoordinateFilePrecision	108
6.10.3.8	GetCoordinateFileType	108
6.10.3.9	NodeX	108
6.10.3.10	NodeY	108
6.10.3.11	NodeZ	108
6.10.3.12	Read	109
6.10.3.13	Write	109
6.10.4	変数	109
6.10.4.1	ActiveSubdomainFile	109
6.10.4.2	GlobalDivision	109
6.10.4.3	GlobalOrigin	110

6.10.4.4	GlobalRegion	110
6.10.4.5	GlobalVoxel	110
6.10.4.6	Pitch	110
6.11	クラス cdm_FILE	110
6.11.1	説明	111
6.11.2	コンストラクタとデストラクタ	111
6.11.2.1	cdm_FILE	111
6.11.2.2	~cdm_FILE	111
6.11.3	関数	111
6.11.3.1	CloseFile	111
6.11.3.2	OpenReadBinary	112
6.11.3.3	OpenWriteAscii	112
6.11.3.4	OpenWriteBinary	112
6.11.4	変数	113
6.11.4.1	m_addMode	113
6.11.4.2	m_fname	113
6.11.4.3	m_format	113
6.11.4.4	m_fp	113
6.11.4.5	m_ncid	113
6.12	クラス cdm_FileInfo	113
6.12.1	説明	114
6.12.2	コンストラクタとデストラクタ	114
6.12.2.1	cdm_FileInfo	114
6.12.2.2	cdm_FileInfo	115
6.12.2.3	~cdm_FileInfo	116
6.12.3	関数	116
6.12.3.1	getVariableName	116
6.12.3.2	Read	116
6.12.3.3	setVariableName	116
6.12.3.4	Write	117
6.12.4	変数	117
6.12.4.1	ArrayShape	117
6.12.4.2	DataType	117
6.12.4.3	DFIType	117
6.12.4.4	DirectoryPath	117
6.12.4.5	Endian	117
6.12.4.6	FieldFilenameFormat	118
6.12.4.7	FileFormat	118
6.12.4.8	GuideCell	118
6.12.4.9	NumVariables	118

6.12.4.10 Prefix	118
6.12.4.11 RankNoPrefix	118
6.12.4.12 TimeSliceDirFlag	118
6.12.4.13 VariableName	118
6.13 クラス cdm_FilePath	119
6.13.1 説明	119
6.13.2 コンストラクタとデストラクタ	119
6.13.2.1 cdm_FilePath	119
6.13.2.2 cdm_FilePath	119
6.13.2.3 ~cdm_FilePath	119
6.13.3 関数	120
6.13.3.1 Read	120
6.13.3.2 Write	120
6.13.4 変数	120
6.13.4.1 ProcDFIFile	120
6.14 クラス cdm_MPI	120
6.14.1 説明	121
6.14.2 コンストラクタとデストラクタ	121
6.14.2.1 cdm_MPI	121
6.14.2.2 cdm_MPI	121
6.14.2.3 ~cdm_MPI	121
6.14.3 関数	121
6.14.3.1 Read	121
6.14.3.2 Write	122
6.14.4 変数	123
6.14.4.1 NumberOfGroup	123
6.14.4.2 NumberOfRank	123
6.15 クラス テンプレート cdm_NonUniformDomain< T >	123
6.15.1 説明	125
6.15.2 コンストラクタとデストラクタ	125
6.15.2.1 cdm_NonUniformDomain	125
6.15.2.2 cdm_NonUniformDomain	125
6.15.2.3 cdm_NonUniformDomain	126
6.15.2.4 ~cdm_NonUniformDomain	126
6.15.3 関数	126
6.15.3.1 CellX	126
6.15.3.2 CellY	127
6.15.3.3 CellZ	127
6.15.3.4 Clear	127
6.15.3.5 GetCoordinateFile	128

6.15.3.6	GetCoordinateFileEndian	128
6.15.3.7	GetCoordinateFilePrecision	128
6.15.3.8	GetCoordinateFileType	128
6.15.3.9	NodeX	129
6.15.3.10	NodeY	130
6.15.3.11	NodeZ	130
6.15.3.12	Read	130
6.15.3.13	Read_CoordinateFile	131
6.15.3.14	readCoordData	131
6.15.3.15	readCoordDataCount	132
6.15.3.16	Write	132
6.15.4	変数	132
6.15.4.1	CoordinateFile	132
6.15.4.2	CoordinateFileEndian	133
6.15.4.3	CoordinateFilePrecision	133
6.15.4.4	CoordinateFileType	133
6.15.4.5	pit_gcXend	133
6.15.4.6	pit_gcXsta	133
6.15.4.7	pit_gcYend	133
6.15.4.8	pit_gcYsta	134
6.15.4.9	pit_gcZend	134
6.15.4.10	pit_gcZsta	134
6.15.4.11	XCoordinates	134
6.15.4.12	YCoordinates	134
6.15.4.13	ZCoordinates	134
6.16	クラス cdm_Process	135
6.16.1	説明	136
6.16.2	型定義	136
6.16.2.1	headT	136
6.16.3	コンストラクタとデストラクタ	136
6.16.3.1	cdm_Process	136
6.16.3.2	~cdm_Process	136
6.16.4	関数	136
6.16.4.1	CheckReadRank	136
6.16.4.2	CheckStartEnd	136
6.16.4.3	CreateHeadMap	137
6.16.4.4	CreateHeadMap	137
6.16.4.5	CreateRankList	137
6.16.4.6	CreateRankList	137
6.16.4.7	CreateRankMap	138

6.16.4.8	CreateRankMap	138
6.16.4.9	CreateSubDomainInfo	138
6.16.4.10	isMatchEndianSbdmMagick	139
6.16.4.11	Read	139
6.16.4.12	ReadActiveSubdomainFile	139
6.16.4.13	Write	139
6.16.5	変数	140
6.16.5.1	m_rankMap	140
6.16.5.2	RankList	140
6.17	クラス cdm_Rank	140
6.17.1	説明	141
6.17.2	コンストラクタとデストラクタ	141
6.17.2.1	cdm_Rank	141
6.17.2.2	~cdm_Rank	141
6.17.3	関数	141
6.17.3.1	Read	141
6.17.3.2	Write	141
6.17.4	変数	141
6.17.4.1	bc_id	141
6.17.4.2	c_id	142
6.17.4.3	HeadIndex	142
6.17.4.4	HostName	142
6.17.4.5	RankID	142
6.17.4.6	TailIndex	142
6.17.4.7	VoxelSize	142
6.18	クラス cdm_Slice	142
6.18.1	説明	143
6.18.2	コンストラクタとデストラクタ	143
6.18.2.1	cdm_Slice	143
6.18.2.2	~cdm_Slice	143
6.18.3	関数	143
6.18.3.1	Read	143
6.18.3.2	Write	144
6.18.4	変数	145
6.18.4.1	AveragedStep	145
6.18.4.2	AveragedTime	145
6.18.4.3	avr_mode	145
6.18.4.4	Max	145
6.18.4.5	Min	145
6.18.4.6	step	145

6.18.4.7	time	145
6.18.4.8	VectorMax	146
6.18.4.9	VectorMin	146
6.19	クラス cdm_TextParser	146
6.19.1	説明	147
6.19.2	コンストラクタとデストラクタ	147
6.19.2.1	cdm_TextParser	147
6.19.2.2	~cdm_TextParser	147
6.19.3	関数	147
6.19.3.1	chkLabel	147
6.19.3.2	chkNode	147
6.19.3.3	countLabels	147
6.19.3.4	GetNodeStr	148
6.19.3.5	getTPinstance	148
6.19.3.6	getTPPtr	148
6.19.3.7	GetValue	148
6.19.3.8	GetValue	148
6.19.3.9	GetValue	148
6.19.3.10	GetVector	149
6.19.3.11	GetVector	149
6.19.3.12	GetVector	149
6.19.3.13	readTPfile	149
6.19.3.14	remove	149
6.19.4	変数	150
6.19.4.1	tp	150
6.20	クラス cdm_TimeSlice	150
6.20.1	説明	150
6.20.2	コンストラクタとデストラクタ	150
6.20.2.1	cdm_TimeSlice	150
6.20.2.2	~cdm_TimeSlice	151
6.20.3	関数	151
6.20.3.1	AddSlice	151
6.20.3.2	getMinMax	151
6.20.3.3	getVectorMinMax	151
6.20.3.4	Read	152
6.20.3.5	Write	153
6.20.4	変数	153
6.20.4.1	SliceList	153
6.21	クラス テンプレート cdm_TypeArray< T >	153
6.21.1	説明	154

6.21.2	コンストラクタとデストラクタ	155
6.21.2.1	cdm_TypeArray	155
6.21.2.2	cdm_TypeArray	155
6.21.2.3	~cdm_TypeArray	155
6.21.2.4	cdm_TypeArray	155
6.21.3	関数	155
6.21.3.1	_val	155
6.21.3.2	_val	155
6.21.3.3	copyArray	155
6.21.3.4	copyArray	156
6.21.3.5	copyArrayNvari	156
6.21.3.6	copyArrayNvari	156
6.21.3.7	copyArrayNvari_to_ijk	156
6.21.3.8	copyArrayNvari_to_ijk	156
6.21.3.9	getData	157
6.21.3.10	hval	157
6.21.3.11	hval	157
6.21.3.12	readBinary	157
6.21.3.13	val	157
6.21.3.14	val	157
6.21.3.15	writeAscii	157
6.21.3.16	writeBinary	157
6.21.4	変数	158
6.21.4.1	m_data	158
6.21.4.2	m_outptr	158
6.22	クラス cdm_Unit	158
6.22.1	説明	159
6.22.2	コンストラクタとデストラクタ	159
6.22.2.1	cdm_Unit	159
6.22.2.2	~cdm_Unit	159
6.22.3	関数	159
6.22.3.1	AddUnit	159
6.22.3.2	GetUnit	159
6.22.3.3	GetUnitElem	159
6.22.3.4	Read	160
6.22.3.5	Write	160
6.22.4	変数	160
6.22.4.1	UnitList	160
6.23	クラス cdm_UnitElem	160
6.23.1	説明	161

6.23.2	コンストラクタとデストラクタ	161
6.23.2.1	cdm_UnitElem	161
6.23.2.2	cdm_UnitElem	161
6.23.2.3	~cdm_UnitElem	161
6.23.3	関数	161
6.23.3.1	Read	161
6.23.3.2	Write	162
6.23.4	変数	163
6.23.4.1	BsetDiff	163
6.23.4.2	difference	163
6.23.4.3	Name	163
6.23.4.4	reference	163
6.23.4.5	Unit	163
6.24	クラス cdm_VisIt	163
6.24.1	説明	164
6.24.2	コンストラクタとデストラクタ	164
6.24.2.1	cdm_VisIt	164
6.24.2.2	cdm_VisIt	164
6.24.2.3	~cdm_VisIt	164
6.24.3	関数	164
6.24.3.1	Read	164
6.24.3.2	Write	164
6.24.4	変数	165
6.24.4.1	PlotGC	165
6.25	構造体 cdm_DFI_NETCDF::stDimInfo	165
6.25.1	説明	165
6.25.2	変数	165
6.25.2.1	id	165
6.25.2.2	len	165
6.25.2.3	name	166
6.26	構造体 cdm_DFI_NETCDF::stVarInfo	166
6.26.1	説明	166
6.26.2	コンストラクタとデストラクタ	166
6.26.2.1	stVarInfo	166
6.26.2.2	~stVarInfo	167
6.26.3	関数	167
6.26.3.1	clear	167
6.26.4	変数	167
6.26.4.1	data	167
6.26.4.2	dims	167

6.26.4.3	endian	167
6.26.4.4	id	167
6.26.4.5	name	167
6.26.4.6	type	167
6.26.4.7	unit	168
7	ファイル	169
7.1	cdm_ActiveSubDomain.h	169
7.2	cdm_Array.h	169
7.2.1	関数	169
7.2.1.1	cdm_interp_ijkn_r4_	169
7.2.1.2	cdm_interp_ijkn_r8_	170
7.2.1.3	cdm_interp_nijk_r4_	170
7.2.1.4	cdm_interp_nijk_r8_	170
7.3	cdm_Array_inline.h	170
7.3.1	マクロ定義	170
7.3.1.1	CDM_INLINE	170
7.3.1.2	CDM_MEMFUN	170
7.4	cdm_Define.h	170
7.4.1	説明	173
7.4.2	マクロ定義	173
7.4.2.1	_CDM_IDX_IJ	173
7.4.2.2	_CDM_IDX_IJK	174
7.4.2.3	_CDM_IDX_IJKN	174
7.4.2.4	_CDM_IDX_NIJ	175
7.4.2.5	_CDM_IDX_NIJK	175
7.4.2.6	_CDM_TAB_STR	176
7.4.2.7	_CDM_WRITE_TAB	176
7.4.2.8	D_CDM_BIG	176
7.4.2.9	D_CDM_BYTE	176
7.4.2.10	D_CDM_DFITYPE_CARTESIAN	176
7.4.2.11	D_CDM_DFITYPE_NON_UNIFORM_CARTESIAN	176
7.4.2.12	D_CDM_DOUBLE	176
7.4.2.13	D_CDM_EXT_BOV	177
7.4.2.14	D_CDM_EXT_BOV_DATAFILE	177
7.4.2.15	D_CDM_EXT_FUNC	177
7.4.2.16	D_CDM_EXT_NC	177
7.4.2.17	D_CDM_EXT_SPH	177
7.4.2.18	D_CDM_EXT_VTK	177
7.4.2.19	D_CDM_FLOAT	177

7.4.2.20	D_CDM_FLOAT32	177
7.4.2.21	D_CDM_FLOAT64	177
7.4.2.22	D_CDM_IJNK	177
7.4.2.23	D_CDM_INT	177
7.4.2.24	D_CDM_INT16	177
7.4.2.25	D_CDM_INT32	178
7.4.2.26	D_CDM_INT64	178
7.4.2.27	D_CDM_INT8	178
7.4.2.28	D_CDM_LITTLE	178
7.4.2.29	D_CDM_NIJK	178
7.4.2.30	D_CDM_OFF	178
7.4.2.31	D_CDM_ON	178
7.4.2.32	D_CDM_UINT16	178
7.4.2.33	D_CDM_UINT32	178
7.4.2.34	D_CDM_UINT64	178
7.4.2.35	D_CDM_UINT8	178
7.4.2.36	stmpd_printf	178
7.5	cdm_DFI.h	179
7.5.1	説明	179
7.6	cdm_DFI_AVS.h	179
7.6.1	説明	180
7.7	cdm_DFI_BOV.h	180
7.7.1	説明	180
7.8	cdm_DFI_inline.h	180
7.8.1	マクロ定義	180
7.8.1.1	CDM_INLINE	180
7.9	cdm_DFI_NETCDF.h	181
7.9.1	説明	181
7.10	cdm_DFI_PLOT3D.h	181
7.10.1	説明	181
7.11	cdm_DFI_SPH.h	181
7.11.1	説明	182
7.12	cdm_DFI_VTK.h	182
7.12.1	説明	182
7.13	cdm_Domain.h	182
7.13.1	説明	182
7.14	cdm_endianUtil.h	183
7.14.1	説明	183
7.14.2	マクロ定義	183
7.14.2.1	BSWAP16	183

7.14.2.2	BSWAP32	184
7.14.2.3	BSWAP64	184
7.14.2.4	BSWAP_X_16	184
7.14.2.5	BSWAP_X_32	184
7.14.2.6	BSWAP_X_64	184
7.14.2.7	BSWAPVEC	184
7.14.2.8	CDM_INLINE	185
7.14.2.9	DBSWAPVEC	185
7.14.2.10	SBSWAPVEC	185
7.15	cdm_FILE.h	185
7.15.1	説明	186
7.16	cdm_FileInfo.h	186
7.16.1	説明	186
7.17	cdm_FilePath.h	186
7.17.1	説明	186
7.18	cdm_interp_ijkn.h	187
7.18.1	関数	187
7.18.1.1	!Copyright	187
7.19	cdm_interp_nijk.h	187
7.19.1	関数	187
7.19.1.1	!Copyright	187
7.20	cdm_MPI.h	187
7.20.1	説明	187
7.21	cdm_NonUniformDomain.h	187
7.21.1	説明	188
7.22	cdm_NonUniformDomain_inline.h	188
7.22.1	説明	188
7.22.2	マクロ定義	188
7.22.2.1	CDM_INLINE	188
7.23	cdm_PathUtil.h	188
7.23.1	マクロ定義	189
7.23.1.1	MAXPATHLEN	189
7.24	cdm_Plot3d_inline.h	189
7.24.1	マクロ定義	189
7.24.1.1	CDM_INLINE	189
7.25	cdm_Process.h	189
7.25.1	説明	190
7.26	cdm_TextParser.h	190
7.26.1	説明	190
7.27	cdm_TimeSlice.h	190

7.27.1 説明	191
7.28 cdm_TypeArray.h	191
7.29 cdm_Unit.h	191
7.29.1 説明	191
7.30 cdm_Version.h	191
7.30.1 説明	192
7.30.2 マクロ定義	192
7.30.2.1 CDM_REVISION	192
7.30.2.2 CDM_VERSION_NO	192
7.31 cdm_VisIt.h	192
7.31.1 説明	192
7.32 mpi_stubs.h	192
7.32.1 マクロ定義	193
7.32.1.1 MPI_CHAR	193
7.32.1.2 MPI_COMM_WORLD	193
7.32.1.3 MPI_INT	193
7.32.1.4 MPI_SUCCESS	193
7.32.2 型定義	193
7.32.2.1 MPI_Comm	193
7.32.2.2 MPI_Datatype	193
7.32.3 関数	193
7.32.3.1 MPI_Allgather	193
7.32.3.2 MPI_Comm_rank	194
7.32.3.3 MPI_Comm_size	194
7.32.3.4 MPI_Gather	194
7.32.3.5 MPI_Init	194

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_NETCDF	??
cdm_DFI_PLOT3D	??
cdm_DFI_SPH	??
cdm_DFI_VTK	??
cdm_Domain	??
cdm_NonUniformDomain< T >	??
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_NETCDF	??
cdm_DFI_PLOT3D	??
cdm_DFI_SPH	??
cdm_DFI_VTK	??
cdm_Domain	??
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_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_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	??
cdm_NonUniformDomain.h	
Cdm_NonUniformDomain Class Header	??
cdm_NonUniformDomain_inline.h	
Cdm_NonUniformDomain template Header	??

cdm_PathUtil.h	??
cdm_Plot3d_inline.h	??
cdm_Process.h	
Cdm_RANK & cdm_Process Class Header	??
cdm_TextParser.h	
TextParser Control class Header	??
cdm_TimeSlice.h	
Cdm_Slice & cdm_TimeSliceClass Header	??
cdm_TypeArray.h	??
cdm_Unit.h	
Cdm_UnitElem & cdm_Unit Class Header	??
cdm_Version.h	??
cdm_VisIt.h	
Cdm_VisIt Class Header	??
mpi_stubs.h	??

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 }
- 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_ENDIANATYPE { E_CDM_ENDIANATYPE_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_FBINARITY }
- 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 }
- 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_BCFID = 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_OPENERERROR
= 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_OPENERERROR = 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_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_NETCDF_FUNC = 2200, E_CDM_ERROR_READ_NETCDF_VAR_1D =
2201,
E_CDM_ERROR_MISMATCH_NP_SUBDOMAIN = 3003, E_CDM_ERROR_INVALID_DIVNUM = 3011,
E_CDM_ERROR_OPEN_SBDM = 3012, E_CDM_ERROR_READ_SBDM_HEADER = 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_FIEL
= 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_REC3 = 3203,
E_CDM_ERROR_WRITE_SPH_REC4 = 3204, E_CDM_ERROR_WRITE_SPH_REC5 = 3205, E_CDM_ERROR_WRITE_SPH
= 3206, E_CDM_ERROR_WRITE_SPH_REC7 = 3207,
E_CDM_ERROR_WRITE_PROCFILNAME_EMPTY = 3500, E_CDM_ERROR_WRITE_PROCFILE_OPENERERROR
= 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_INDEXF
= 3510, E_CDM_ERROR_WRITE_PREFIX_EMPTY = 3511,
E_CDM_ERROR_WRITE_INDEXFILE_OPENERERROR = 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_NETCDF = 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="", 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` i j k n
`E_CDM_NIJK` n i j k

cdm_Define.h の 117 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 71 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 101 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 125 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_GUIDECCELL 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 HoatName 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_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_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_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_OPENERROR 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_OPENERROR 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 161 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 143 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.

cdm_Define.h の 79 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 94 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 のみ)

cdm_Define.h の 152 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 133 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 & 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 & path, const std::string & 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 69 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()`, 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::ReadData()`, `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 682 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_sz[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_Sz[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
- 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
- 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, TimeAvrT &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 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)
- `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)
- `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)
 - 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 `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

- 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

- 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 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, 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* ヘッダーレコード出力 (*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]`)
- 読み込み判定判定
 - 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込み範囲を求める
 - `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 コミュニケーター
- `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`
読み込みタイプ
- 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`
読み込みランクリスト
- `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 47 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 241 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::FileFormat`, 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 1257 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 420 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 701 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 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

6.3.3.22 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

6.3.3.23 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

6.3.3.24 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

6.3.3.25 `std::string cdm_DFI::GetcdmRankNoPrefix ()`

cdm_FileInfo のRankNoPrefix を取得

戻り値

ランク番号前文字列

6.3.3.26 `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

6.3.3.27 `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

6.3.3.28 `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

6.3.3.29 `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)

6.3.3.30 `std::string cdm_DFI::GetDataTypeString ()`

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)

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

DFI Domain のGlobalDivision の取り出し

戻り値

GlobalDivision のポインタ

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

DFI Domain のGlobalVoxel の取り出し

戻り値

GlobalVoxel のポインタ

6.3.3.33 `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)

6.3.3.34 `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)

6.3.3.35 `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)

6.3.3.36 `std::string cdm_DFI::GetFileFormatString ()`

get FileFormat (FileFormat の取り出し関数)

戻り値

FileFormat(文字列)

6.3.3.37 `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

6.3.3.38 int cdm_DFI::GetNumGuideCell ()

6.3.3.39 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

6.3.3.40 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

6.3.3.41 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

6.3.3.42 std::string cdm_DFI::getVariableName (int *pvari*)

FileInfo の変数名を取得する

引数

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

戻り値

変数名

6.3.3.43 `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

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

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

バージョンを出力する

cdm_DFI.h の 1240 行で定義されています。

参照先 CDM_VERSION_NO.

6.3.3.45 `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

6.3.3.46 `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

6.3.3.47 `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

6.3.3.48 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
-----------	--------------	------

6.3.3.49 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
-----------	--------------	------

6.3.3.50 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
-----------	--------------	------

6.3.3.51 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
-----------	--------------	------

6.3.3.52 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
-----------	--------------	------

6.3.3.53 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

in	<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
----	--------------	-----------------------------------------------

6.3.3.54 `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

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

[cdm_DFI_NETCDF](#), [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。

6.3.3.55 `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_SPH](#), [cdm_DFI_AVS](#), [cdm_DFI_PLOT3D](#), [cdm_DFI_VTK](#), と [cdm_DFI_BOV](#) で実装されています。

6.3.3.56 `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_SPH, cdm_DFI_AVS, cdm_DFI_PLOT3D, cdm_DFI_VTK, と cdm_DFI_BOVで実装されています。

```
6.3.3.57 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 47 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().

```
6.3.3.58 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 94 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().

```
6.3.3.59 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	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 ReadData().

```
6.3.3.60 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

6.3.3.61 `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

6.3.3.62 `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

```
6.3.3.63 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

```
6.3.3.64 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 383 行で定義されています。

参照先 m_input_type.

```
6.3.3.65 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

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

cdm_DFI.h の 376 行で定義されています。

参照先 m_bgrid_interp_flag.

6.3.3.66 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

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

cdm_DFI.h の 405 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.

6.3.3.67 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

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

cdm_DFI.h の 390 行で定義されています。

参照先 m_output_type.

6.3.3.68 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

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

cdm_DFI.h の 398 行で定義されています。

参照先 m_output_type_coord.

6.3.3.69 void cdm_DFI::set_RankID (const int *rankID*) [inline]

RankID をセットする

引数

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

cdm_DFI.h の 369 行で定義されています。

参照先 m_RankID.

6.3.3.70 void cdm_DFI::SetcdmDomain (cdm_Domain * *domain*)

cdm_Domain クラスのセット

6.3.3.71 void cdm_DFI::SetcdmFilePath (cdm_FilePath *FPath*)

cdm_FilePath クラスのセット

6.3.3.72 void cdm_DFI::SetcdmMPI (cdm_MPI *mpi*)

cdm_MPI クラスセット

6.3.3.73 void cdm_DFI::SetcdmProcess (cdm_Process *Process*)

cdm_Process クラスセット

6.3.3.74 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

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

6.3.3.75 void cdm_DFI::SetcdmTimeSlice (cdm_TimeSlice *TSlice*)

[cdm_TimeSlice](#) クラスセット

6.3.3.76 void cdm_DFI::SetcdmUnit (cdm_Unit *unit*)

cdm_Unit クラスのセット

6.3.3.77 void cdm_DFI::SetcdmVisIt (cdm_VisIt *Visit*)

cdm_VisIt クラスのセット

6.3.3.78 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 278 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().

6.3.3.79 `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

6.3.3.80 `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	interval_step	インターバルステップ
in	base_step	基準となるステップ (デフォルト 0 ステップ)
in	start_step	セッション開始ステップ (デフォルト 0 ステップ)
in	last_step	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)

6.3.3.81 `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	interval_time	出力インターバルタイム
in	dt	計算の時間間隔
in	base_time	基準となるタイム (デフォルト 0.0 タイム)
in	start_time	セッション開始タイム (デフォルト 0.0 タイム)
in	last_time	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)

6.3.3.82 `void cdm_DFI::SetTimeSliceFlag (const CDM::E_CDM_ONOFF ONOFF)`

TimeSlice OnOff フラグをセットする

引数

in	ONOFF	
----	-------	--

6.3.3.83 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",,,

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

cdm_DFI_inline.h の 351 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().

6.3.3.85 `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で割る

引数

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

参照元 setGridData().

6.3.3.86 `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 1114 行で定義されています。

6.3.3.87 `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_AVS, cdm_DFI_VTK, と cdm_DFI_BOVで実装されています。

6.3.3.88 `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_SPH](#), [cdm_DFI_AVS](#), [cdm_DFI_VTK](#), と [cdm_DFI_BOV](#)で実装されています。

6.3.3.89 `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 1104 行で定義されています。

6.3.3.90 `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_SPH](#), [cdm_DFI_AVS](#), [cdm_DFI_VTK](#), と [cdm_DFI_BOV](#)で実装されています。

6.3.3.91 `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 nVar, const int gc, T * val, T * minmax, const bool avr_mode, const unsigned step_avr, TimeAvrT time_avr)`

cdm_DFI_inline.h の 140 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\(\)](#).

6.3.3.92 `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().

6.3.3.93 `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

6.3.3.94 `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

6.3.3.95 `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 200 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().

6.3.3.96 `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	<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>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 WriteFieldDataFile().

6.3.3.97 `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	<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>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

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

6.3.3.99 CDM::E_CDM_ERRORCODE cdm_DFI::WriteIndexDfiFile ()

index DFI ファイル出力 (API 関数)

6.3.3.100 CDM::E_CDM_ERRORCODE cdm_DFI::WriteIndexDfiFile (const std::string *dfi_name*) [protected]

index DFI ファイル出力

引数

in	<i>dfi_name</i>	DFI ファイル名
----	-----------------	-----------

戻り値

true:出力成功 false:出力失敗

6.3.3.101 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
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

6.3.3.102 `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

6.3.3.103 `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

6.3.3.104 `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 422 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`.

6.3.3.105 `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 479 行で定義されています。

参照先 `CDM::E_CDM_FMT_BOV`, と `CDM::E_CDM_FMT_SPH`.

6.3.3.106 `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 565 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_AVIS`, `CDM::E_CDM_FMT_BOV`, `CDM::E_CDM_FMT_N-ETCDF4`, `CDM::E_CDM_FMT_PLOT3D`, `CDM::E_CDM_FMT_SPH`, `CDM::E_CDM_FMT_VTK`, `CDM::E_CDM_IJ-KN`, `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.

6.3.3.107 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 コミュニケーター
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 コミュニケーター
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 63 行で定義されています。

参照元 cdm_DFI_AVS::cdm_DFI_AVS(), cdm_DFI_BOV::cdm_DFI_BOV(), 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 59 行で定義されています。

参照元 AddTimeSlice(), cdm_DFI_AVS::cdm_DFI_AVS(), cdm_DFI_BOV::cdm_DFI_BOV(), cdm_DFI_NETCDF::cdm_DFI_NETCDF(), cdm_DFI_PLOT3D::cdm_DFI_PLOT3D(), cdm_DFI_SPH::cdm_DFI_SPH(), cdm_DFI_VTK::cdm_DFI_VTK(), ReadData(), set_output_fname(), cdm_DFI_PLOT3D::write_XYZ(), WriteData(), と WriteFieldDataFile().

6.3.4.3 cdm_FilePath cdm_DFI::DFI_Fpath [protected]

FilePath class.

cdm_DFI.h の 60 行で定義されています。

参照元 cdm_DFI_AVS::cdm_DFI_AVS(), cdm_DFI_BOV::cdm_DFI_BOV(), 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 64 行で定義されています。

参照元 cdm_DFI_AVS::cdm_DFI_AVS(), cdm_DFI_BOV::cdm_DFI_BOV(), 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 66 行で定義されています。

参照元 cdm_DFI_AVS::cdm_DFI_AVS(), cdm_DFI_BOV::cdm_DFI_BOV(), cdm_DFI_NETCDF::cdm_DFI_NETCDF(), cdm_DFI_PLOT3D::cdm_DFI_PLOT3D(), cdm_DFI_SPH::cdm_DFI_SPH(), cdm_DFI_VTK::cdm_DFI_VTK(), WriteData(), と WriteFieldDataFile().

6.3.4.6 cdm_TimeSlice cdm_DFI::DFI_TimeSlice [protected]

TimeSlice class.

cdm_DFI.h の 65 行で定義されています。

参照元 AddTimeSlice(), cdm_DFI_AVS::cdm_DFI_AVS(), cdm_DFI_BOV::cdm_DFI_BOV(), 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 62 行で定義されています。

参照元 cdm_DFI_AVS::cdm_DFI_AVS(), cdm_DFI_BOV::cdm_DFI_BOV(), 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 61 行で定義されています。

参照元 cdm_DFI_AVS::cdm_DFI_AVS(), cdm_DFI_BOV::cdm_DFI_BOV(), 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 70 行で定義されています。

参照元 cdm_DFI_AVS::cdm_DFI_AVS(), cdm_DFI_BOV::cdm_DFI_BOV(), 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

cdm_DFI.h の 52 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 53 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 54 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,FortarnBinary)

cdm_DFI.h の 71 行で定義されています。

参照元 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 74 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,FortarnBinary)

cdm_DFI.h の 72 行で定義されています。

参照元 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 73 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 57 行で定義されています。

参照元 `set_RankID()`, `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 55 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 68 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。

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。

6.5.3.3 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_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 *dff* パラメータ読み込み
- [CDM::E_CDM_ERRORCODE WriteAdditionalTP](#) (FILE *fp, int tab)
NetCDF 特有の *dff* 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

6.6.1 説明

`cdm_DFI_NETCDF.h` の 22 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_NETCDF::cdm_DFI_NETCDF ()`

コンストラクタ

6.6.2.2 `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`.

6.6.2.3 `cdm_DFI_NETCDF::~~cdm_DFI_NETCDF ()`

デストラクタ

6.6.3 関数

6.6.3.1 `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

6.6.3.2 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。

6.6.3.3 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

6.6.3.4 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

6.6.3.5 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

6.6.3.6 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

6.6.3.7 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

6.6.3.8 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

6.6.3.9 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。

```
6.6.3.10 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

```
6.6.3.11 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。

```
6.6.3.12 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 (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

6.6.3.13 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。

6.6.3.14 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

6.6.3.15 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

6.6.3.16 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.

6.6.3.17 static void cdm_DFI_NETCDF::write_AttnUnits (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

6.6.3.18 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。

6.6.3.19 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

6.6.3.20 `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。

6.6.3.21 `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

6.6.3.22 `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。

6.6.3.23 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

6.6.4 変数

6.6.4.1 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().

6.6.4.2 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().

6.6.4.3 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().

6.6.4.4 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().

6.6.4.5 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。

6.6.4.6 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。

6.6.4.7 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。

6.6.4.8 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。

6.6.4.9 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。

6.6.4.10 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。

6.6.4.11 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](#)

6.7 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

6.7.1 説明

cdm_DFI_PLOT3D.h の 20 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_PLOT3D::cdm_DFI_PLOT3D ()

コンストラクタ

6.7.2.2 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.

6.7.2.3 cdm_DFI_PLOT3D::~cdm_DFI_PLOT3D ()

デストラクタ

6.7.3 関数

6.7.3.1 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。

6.7.3.2 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。

6.7.3.3 `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()`.

6.7.3.4 `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

6.7.3.5 `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。

6.7.3.6 `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。

6.7.3.7 `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。

6.7.3.8 `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

6.7.3.9 `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`.

6.7.3.10 `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。

6.7.3.11 `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。

6.7.3.12 `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::DFI_Finfo`, `CDM::E_CDM_FILE_TYPE_ASCII`, `CDM::E_CDM_FILE_TYPE_FBINARY`, `cdm_FileInfo::GuideCell`, と `cdm_DFI::m_output_type`.

6.7.3.13 `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](#)

6.8 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

6.8.1 説明

cdm_DFI_SPH.h の 20 行で定義されています。

6.8.2 列挙型

6.8.2.1 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。

6.8.2.2 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。

6.8.3 コンストラクタとデストラクタ

6.8.3.1 cdm_DFI_SPH::cdm_DFI_SPH ()

コンストラクタ

```
6.8.3.2 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.

6.8.3.3 cdm_DFI_SPH::~cdm_DFI_SPH ()

デストラクタ

6.8.4 関数

6.8.4.1 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。

6.8.4.2 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。

6.8.4.3 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 *voysize*[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。

6.8.4.4 `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。

6.8.4.5 `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。

6.8.4.6 `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](#)

6.9 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

6.9.1 説明

cdm_DFI_VTK.h の 20 行で定義されています。

6.9.2 コンストラクタとデストラクタ

6.9.2.1 cdm_DFI_VTK::cdm_DFI_VTK ()

コンストラクタ

6.9.2.2 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.

6.9.2.3 cdm_DFI_VTK::~cdm_DFI_VTK ()

デストラクタ

6.9.3 関数

6.9.3.1 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.

6.9.3.2 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

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.

6.9.3.3 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 *voysize*[3], double & *time*) [inline], [protected], [virtual]

vtk 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_VTK.h の 81 行で定義されています。

参照先 CDM::E_CDM_SUCCESS.

6.9.3.4 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.

6.9.3.5 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。

6.9.3.6 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](#)

6.10 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

6.10.1 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。

6.10.2 コンストラクタとデストラクタ

6.10.2.1 cdm_Domain::cdm_Domain ()

コンストラクタ

6.10.2.2 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

6.10.2.3 cdm_Domain::cdm_Domain (const float * *_GlobalOrigin*, const float * *_GlobalPitch*, const int * *_GlobalVoxel*, const int * *_GlobalDivision*)

6.10.2.4 virtual cdm_Domain::~~cdm_Domain () [virtual]

デストラクタ

6.10.3 関数

6.10.3.1 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()`.

6.10.3.2 `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()`.

6.10.3.3 `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()`.

6.10.3.4 `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()`.

6.10.3.5 `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。

6.10.3.6 `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。

6.10.3.7 `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。

6.10.3.8 `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。

6.10.3.9 `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.

6.10.3.10 `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.

6.10.3.11 `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`.

6.10.3.12 `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。

6.10.3.13 `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。

6.10.4 変数

6.10.4.1 `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。

6.10.4.2 `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()`.

6.10.4.3 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().

6.10.4.4 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().

6.10.4.5 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().

6.10.4.6 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](#)

6.11 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\(\)](#)

6.11.1 説明

[CDM](#) file pointer control class

[cdm_FILE.h](#) の 30 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_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\(\)](#).

6.11.2.2 [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](#).

6.11.3 関数

6.11.3.1 [static void cdm_FILE::CloseFile\(cdm_FILE * pFile \)](#) [inline],[static]

ファイルをクローズ

引数

in	<i>fp</i>	cdm_DFI 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。

6.11.3.2 `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

in	<i>fname</i>	ファイル名
in	<i>format</i>	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.

6.11.3.3 `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.

6.11.3.4 `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.

6.11.4 変数

6.11.4.1 `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()`.

6.11.4.2 `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()`.

6.11.4.3 `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()`.

6.11.4.4 `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()`.

6.11.4.5 `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](#)

6.12 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

6.12.1 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。

6.12.2 コンストラクタとデストラクタ

6.12.2.1 cdm_FileInfo::cdm_FileInfo ()

コンストラクタ

```
6.12.2.2 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

in	<i>_DFIType</i>	dfi 種別
in	<i>_FieldFilename- Format</i>	ファイル命名基準
in	<i>_DirectoryPath</i>	ディレクトリパス
in	<i>_TimeSliceDir- Flag</i>	TimeSlice on or off
in	<i>_Prefix</i>	ファイル接頭文字
in	<i>_FileFormat</i>	ファイルフォーマット
in	<i>_GuideCell</i>	仮想セルの数
in	<i>_DataType</i>	配列のデータタイプ
in	<i>_Endian</i>	エンディアンタイプ
in	<i>_ArrayShape</i>	配列形状
in	<i>_NumVariables</i>	変数の個数

6.12.2.3 cdm_FileInfo::~cdm_FileInfo ()

デストラクタ

6.12.3 関数

6.12.3.1 std::string cdm_FileInfo::getVariableName (int *pvari*)

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 変数名が無い場合は空白が返される

6.12.3.2 CDM::E_CDM_ERRORCODE cdm_FileInfo::Read (cdm_TextParser *tpCntl*)

read FileInfo(inde.dfi)

引数

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

戻り値

error code

6.12.3.3 void cdm_FileInfo::setVariableName (int *pvari*, std::string *variName*)

変数名をセットする

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",,,

6.12.3.4 `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

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

6.12.4 変数

6.12.4.1 `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::ReadData()`, `cdm_DFI::WriteData()`, `cdm_DFI::WriteFieldDataFile()`, と `cdm_DFI::WriteInit()`.

6.12.4.2 `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::ReadData()`, と `cdm_DFI::WriteInit()`.

6.12.4.3 `CDM::E_CDM_DFITYPE cdm_FileInfo::DFIType`

dfi 種別

`cdm_FileInfo.h` の 25 行で定義されています。

参照元 `cdm_DFI::WriteInit()`.

6.12.4.4 `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.dfi` 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()`.

6.12.4.5 `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()`.

6.12.4.6 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。

6.12.4.7 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().

6.12.4.8 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().

6.12.4.9 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::ReadData(), cdm_DFI::WriteData(), cdm_DFI::WriteFieldDataFile(), と cdm_DFI::WriteInit().

6.12.4.10 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().

6.12.4.11 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。

6.12.4.12 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().

6.12.4.13 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](#)

6.13 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(index.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

6.13.1 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。

6.13.2 コンストラクタとデストラクタ

6.13.2.1 cdm_FilePath::cdm_FilePath ()

コンストラクタ

6.13.2.2 cdm_FilePath::cdm_FilePath (const std::string _ProcDFIFile)

コンストラクタ

引数

in	_ProcDFIFile	proc.dfi ファイル名
----	------------------------------	----------------

6.13.2.3 cdm_FilePath::~cdm_FilePath ()

デストラクタ

6.13.3 関数

6.13.3.1 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	tpCntl	cdm_TextParser クラス
----	--------	--------------------

戻り値

error code

6.13.3.2 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	fp	ファイルポインタ
in	tab	インデント

戻り値

error code

6.13.4 変数

6.13.4.1 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](#)

6.14 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

6.14.1 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。

6.14.2 コンストラクタとデストラクタ

6.14.2.1 `cdm_MPI::cdm_MPI ()`

コンストラクタ

6.14.2.2 `cdm_MPI::cdm_MPI (const int _NumberOfRank, int _NumberOfGroup = 0)`

コンストラクタ

引数

in	<code>_NumberOfRank</code>	プロセス数
in	<code>_NumberOfGroup</code>	グループ数

6.14.2.3 `cdm_MPI::~~cdm_MPI ()`

デストラクタ

6.14.3 関数

6.14.3.1 `CDM::E_CDM_ERRORCODE cdm_MPI::Read (cdm_TextParser tpCntl, const cdm_Domain * domain)`

`read MPI(proc.dfi)`

引数

in	<code>tpCntl</code>	<code>cdm_TextParser</code> クラス
in	<code>domain</code>	Domain

戻り値

error code

6.14.3.2 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

<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

6.14.4 変数

6.14.4.1 `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()`.

6.14.4.2 `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](#)

6.15 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

6.15.1 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。

6.15.2 コンストラクタとデストラクタ

6.15.2.1 `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`.

6.15.2.2 `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	<i>_CoordinateFile-Endian</i>	座標データのエンディアンタイプ
in	<i>_GlobalVoxel</i>	ボクセル数
in	<i>_GlobalDivision</i>	分割数
in	<i>_gc</i>	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.

```
6.15.2.3 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	<i>_GlobalOrigin</i>	起点座標
in	<i>_GlobalRegion</i>	各軸方向の長さ
in	<i>_GlobalVoxel</i>	ボクセル数
in	<i>_GlobalDivision</i>	分割数
in	<i>_CoordinateFile</i>	座標データ名
in	<i>_CoordinateFile- Type</i>	座標データのファイルタイプ
in	<i>_CoordinateFile- Precision</i>	座標データのデータ精度
in	<i>_CoordinateFile- Endian</i>	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.

```
6.15.2.4 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.

6.15.3 関数

```
6.15.3.1 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

<code>in</code>	<code>i</code>	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`.

6.15.3.2 `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

<code>in</code>	<code>j</code>	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`.

6.15.3.3 `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

<code>in</code>	<code>k</code>	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`.

6.15.3.4 `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`.

```
6.15.3.5  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.

```
6.15.3.6  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.

```
6.15.3.7  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.

```
6.15.3.8  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.


```
6.15.3.9  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.

```
6.15.3.10 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.

```
6.15.3.11 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.

```
6.15.3.12 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().

6.15.3.13 `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.

6.15.3.14 `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.

6.15.3.15 `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.

6.15.3.16 `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.

6.15.4 変数

6.15.4.1 `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()`.

6.15.4.2 `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()`.

6.15.4.3 `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()`.

6.15.4.4 `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()`.

6.15.4.5 `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()`.

6.15.4.6 `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()`.

6.15.4.7 `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()`.

6.15.4.8 `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()`.

6.15.4.9 `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()`.

6.15.4.10 `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()`.

6.15.4.11 `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()`.

6.15.4.12 `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()`.

6.15.4.13 `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](#)

6.16 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 [isMatchEndianSbdomMagick](#) (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

6.16.1 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。

6.16.2 型定義

6.16.2.1 typedef std::map<int,int> cdm_Process::headT

cdm_Process.h の 65 行で定義されています。

6.16.3 コンストラクタとデストラクタ

6.16.3.1 cdm_Process::cdm_Process ()

コンストラクタ

6.16.3.2 cdm_Process::~~cdm_Process ()

デストラクタ

6.16.4 関数

6.16.4.1 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	<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

6.16.4.2 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

6.16.4.3 `void cdm_Process::CreateHeadMap (std::set< int > head, headT & 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

6.16.4.4 `void cdm_Process::CreateHeadMap (int * head, int ndiv, headT & 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

6.16.4.5 `CDM::E_CDM_ERRORCODE cdm_Process::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

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>	headY をキーにした位置情報マップ
out	<i>mapHeadZ</i>	headZ をキーにした位置情報マップ

戻り値

error code

6.16.4.6 `CDM::E_CDM_ERRORCODE cdm_Process::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

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

6.16.4.7 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

6.16.4.8 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

6.16.4.9 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
-----	----------------------	----------

6.16.4.10 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

6.16.4.11 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

6.16.4.12 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)

6.16.4.13 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
----	-----------	----------

<code>in</code>	<code>tab</code>	インデント
-----------------	------------------	-------

戻り値

true:出力成功 false:出力失敗

6.16.5 変数

6.16.5.1 `int* cdm_Process::m_rankMap`

`cdm_Process.h` の 69 行で定義されています。

6.16.5.2 `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::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](#)

6.17 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

6.17.1 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。

6.17.2 コンストラクタとデストラクタ

6.17.2.1 `cdm_Rank::cdm_Rank ()`

コンストラクタ

6.17.2.2 `cdm_Rank::~~cdm_Rank ()`

デストラクタ

6.17.3 関数

6.17.3.1 `CDM::E_CDM_ERRORCODE cdm_Rank::Read (cdm_TextParser tpCntl, std::string label_leaf)`

read Rank(`proc.dfi`)

引数

in	<i>tpCntl</i>	<code>cdm_TextParser</code> クラス
in	<i>label_leaf</i>	ベースとなる名前 (<code>"/Process/Rank"</code>)

戻り値

error code

6.17.3.2 `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

6.17.4 変数

6.17.4.1 `int cdm_Rank::bc_id`

境界ID

`cdm_Process.h` の 30 行で定義されています。

6.17.4.2 int cdm_Rank::c_id

cell id

cdm_Process.h の 29 行で定義されています。

6.17.4.3 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。

6.17.4.4 std::string cdm_Rank::HostName

ホスト名

cdm_Process.h の 25 行で定義されています。

6.17.4.5 int cdm_Rank::RankID

ランク番号

cdm_Process.h の 24 行で定義されています。

6.17.4.6 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。

6.17.4.7 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](#)

6.18 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

6.18.1 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。

6.18.2 コンストラクタとデストラクタ

6.18.2.1 `cdm_Slice::cdm_Slice ()`

コンストラクタ

6.18.2.2 `cdm_Slice::~~cdm_Slice ()`

デストラクタ

6.18.3 関数

6.18.3.1 `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

<code>in</code>	<code>tpCntl</code>	<code>cdm_TextParser</code> クラス
<code>in</code>	<code>label_leaf</code>	ベースとなる名前 (<code>"/TimeSlice/Slice"</code>)
<code>in</code>	<code>format</code>	ファイルフォーマット

戻り値

error code

6.18.3.2 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

<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
<code>in</code>	<code>format</code>	ファイルフォーマット

戻り値

`error code`

6.18.4 変数

6.18.4.1 `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。

6.18.4.2 `double cdm_Slice::AveragedTime`

平均タイム

`cdm_TimeSlice.h` の 27 行で定義されています。

6.18.4.3 `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。

6.18.4.4 `vector<double> cdm_Slice::Max`

最大値

`cdm_TimeSlice.h` の 31 行で定義されています。

6.18.4.5 `vector<double> cdm_Slice::Min`

最小値

`cdm_TimeSlice.h` の 30 行で定義されています。

6.18.4.6 `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。

6.18.4.7 `double cdm_Slice::time`

時刻

`cdm_TimeSlice.h` の 24 行で定義されています。

6.18.4.8 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。

6.18.4.9 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](#)

6.19 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

6.19.1 説明

`cdm_TextParser.h` の 30 行で定義されています。

6.19.2 コンストラクタとデストラクタ

6.19.2.1 `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。

6.19.2.2 `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。

6.19.3 関数

6.19.3.1 `bool cdm_TextParser::chkLabel (const std::string label, bool checkPath = true)`

ラベルの有無をチェック

引数

<code>in</code>	<code>label</code>	チェックするラベル (絶対パス)
<code>in</code>	<code>checkPath</code>	true: ラベル存在チェックをフルパスで行う

6.19.3.2 `bool cdm_TextParser::chkNode (const std::string label)`

ノードの有無をチェック

引数

<code>in</code>	<code>label</code>	チェックするノード (絶対パス)
-----------------	--------------------	--------------------

6.19.3.3 `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

<code>in</code>	<code>label</code>	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)	
-----------------------------	--

6.19.3.4 `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

6.19.3.5 `void cdm_TextParser::getTPInstance ()`

TextParserLibrary のインスタンス生成

戻り値

エラーコード

6.19.3.6 `TextParser* cdm_TextParser::getTPPtr () [inline]`

cdm_TextParser.h の 153 行で定義されています。

6.19.3.7 `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()`.

6.19.3.8 `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

6.19.3.9 `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

6.19.3.10 `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()`.

6.19.3.11 `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

6.19.3.12 `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

6.19.3.13 `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

6.19.3.14 `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。

6.19.4 変数

6.19.4.1 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](#)

6.20 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](#)

6.20.1 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。

6.20.2 コンストラクタとデストラクタ

6.20.2.1 cdm_TimeSlice::cdm_TimeSlice ()

コンストラクタ

6.20.2.2 cdm_TimeSlice::~cdm_TimeSlice ()

デストラクタ

6.20.3 関数

6.20.3.1 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().

6.20.3.2 CDM::E_CDM_ERRORCODE cdm_TimeSlice::getMinMax (const unsigned step, const int variNo, double & min_value, double & 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

6.20.3.3 CDM::E_CDM_ERRORCODE cdm_TimeSlice::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	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

6.20.3.4 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

6.20.3.5 `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

6.20.4 変数

6.20.4.1 `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](#)

6.21 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

6.21.1 説明

`template<class T>class cdm_TypeArray< T >`

`cdm_TypeArray.h` の 15 行で定義されています。

6.21.2 コンストラクタとデストラクタ

6.21.2.1 `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`.

6.21.2.2 `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`.

6.21.2.3 `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`.

6.21.2.4 `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`.

6.21.3 関数

6.21.3.1 `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()`.

6.21.3.2 `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。

6.21.3.3 `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()`.

6.21.3.4 `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()`.

6.21.3.5 `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()`.

6.21.3.6 `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()`.

6.21.3.7 `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 581 行で定義されています。

参照先 `cdm_Array::getGcInt()`, `cdm_Array::getHeadIndex()`, と `cdm_Array::getTailIndex()`.

6.21.3.8 `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 609 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()`.

6.21.3.9 `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()`.

6.21.3.10 `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()`.

6.21.3.11 `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。

6.21.3.12 `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 755 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()`.

6.21.3.13 `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()`.

6.21.3.14 `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。

6.21.3.15 `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 815 行で定義されています。

6.21.3.16 `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 793 行で定義されています。

6.21.4 変数

6.21.4.1 `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\(\)](#).

6.21.4.2 `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](#)

6.22 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](#)

6.22.1 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。

6.22.2 コンストラクタとデストラクタ

6.22.2.1 cdm_Unit::cdm_Unit ()

コンストラクタ

6.22.2.2 cdm_Unit::~~cdm_Unit ()

デストラクタ

6.22.3 関数

6.22.3.1 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

6.22.3.2 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

6.22.3.3 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

6.22.3.4 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

6.22.3.5 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

6.22.4 変数

6.22.4.1 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](#)

6.23 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)

6.23.1 説明

cdm_Unit.h の 18 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_UnitElem::cdm_UnitElem ()

コンストラクタ

6.23.2.2 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

6.23.2.3 cdm_UnitElem::~~cdm_UnitElem ()

デストラクタ

6.23.3 関数

6.23.3.1 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

6.23.3.2 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

6.23.4 変数

6.23.4.1 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。

6.23.4.2 double cdm_UnitElem::difference

差

cdm_Unit.h の 25 行で定義されています。

6.23.4.3 std::string cdm_UnitElem::Name

単位の種類名 (Length,Velocity,,)
cdm_Unit.h の 22 行で定義されています。

6.23.4.4 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。

6.23.4.5 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](#)

6.24 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

6.24.1 説明

`cdm_VisIt.h` の 18 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 `cdm_VisIt::cdm_VisIt()`

コンストラクタ

6.24.2.2 `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
-----------------	----------------------	--------------

6.24.2.3 `cdm_VisIt::~~cdm_VisIt()`

デストラクタ

6.24.3 関数

6.24.3.1 `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

6.24.3.2 `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

<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

6.24.4 変数

6.24.4.1 `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](#)

6.25 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

6.25.1 説明

`cdm_DFI_NETCDF.h` の 25 行で定義されています。

6.25.2 変数

6.25.2.1 `int cdm_DFI_NETCDF::stDimInfo::id`

dimension ID

`cdm_DFI_NETCDF.h` の 27 行で定義されています。

6.25.2.2 `size_t cdm_DFI_NETCDF::stDimInfo::len`

dimension

`cdm_DFI_NETCDF.h` の 29 行で定義されています。

6.25.2.3 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](#)

6.26 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

6.26.1 説明

cdm_DFI_NETCDF.h の 32 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_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](#).

6.26.2.2 `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()`.

6.26.3 関数

6.26.3.1 `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()`.

6.26.4 変数

6.26.4.1 `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()`.

6.26.4.2 `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()`.

6.26.4.3 `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()`.

6.26.4.4 `int cdm_DFI_NETCDF::stVarInfo::id`

variable ID

`cdm_DFI_NETCDF.h` の 34 行で定義されています。

6.26.4.5 `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()`.

6.26.4.6 `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()`.

6.26.4.7 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_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_IJKN "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_AVG, CDM::E_CDM_FMT_PLOT3D, CDM::E_CDM_FMT_VTK, CDM::E_CDM_FMT_NETCDF4 }
- 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_FBINAR } }
- 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 }
- 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_TIMESLICEDIV
 = 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_COORDINATEFI
 = 1030,
 CDM::E_CDM_ERROR_READ_DFI_COORDINATEFILETYPE = 1031, CDM::E_CDM_ERROR_READ_DFI_COORDINATEFI
 = 1032, CDM::E_CDM_ERROR_READ_DFI_COORDINATEFILEENDIAN = 1033, CDM::E_CDM_ERROR_OPEN_COORDIN
 = 1034,
 CDM::E_CDM_ERROR_READ_COORDINATEFILE = 1035, CDM::E_CDM_ERROR_READ_INDEXFILE_OPENERROR
 = 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_OPENERROR
 = 1056,
 CDM::E_CDM_ERROR_READ_DOMAIN = 1057, CDM::E_CDM_ERROR_READ_MPI = 1058, CDM::E_CDM_ERROR_READ
 = 1059, CDM::E_CDM_ERROR_READ_DFI_NETCDF = 1060,
 CDM::E_CDM_ERROR_READ_NETCDF_MISMATCH_TYPE = 1061, 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_REC
 = 2104, CDM::E_CDM_ERROR_READ_NETCDF_FUNC = 2200, CDM::E_CDM_ERROR_READ_NETCDF_VAR_1D
 = 2201,
 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_RECORD = 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_OPENERR
= 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_OPENERROR = 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 303 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 289 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 335 行で定義されています。

7.4.2.4 `#define _CDM_IDX_NIJ(_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 318 行で定義されています。

7.4.2.5 `#define _CDM_IDX_NIJK(_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 353 行で定義されています。

7.4.2.6 #define _CDM_TAB_STR " "

cdm_Define.h の 62 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 361 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 60 行で定義されています。

7.4.2.9 #define D_CDM_BYTE "BYTE"

cdm_Define.h の 51 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 54 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 35 行で定義されています。

7.4.2.15 #define D_CDM_EXT_FUNC "fun"

cdm_Define.h の 29 行で定義されています。

7.4.2.16 #define D_CDM_EXT_NC "nc"

cdm_Define.h の 32 行で定義されています。

7.4.2.17 #define D_CDM_EXT_SPH "sph"

cdm_Define.h の 27 行で定義されています。

7.4.2.18 #define D_CDM_EXT_VTK "vtk"

cdm_Define.h の 30 行で定義されています。

7.4.2.19 #define D_CDM_FLOAT "FLOAT"

cdm_Define.h の 53 行で定義されています。

7.4.2.20 #define D_CDM_FLOAT32 "Float32"

cdm_Define.h の 48 行で定義されています。

7.4.2.21 #define D_CDM_FLOAT64 "Float64"

cdm_Define.h の 49 行で定義されています。

7.4.2.22 #define D_CDM_IJNK "ijkn"

cdm_Define.h の 56 行で定義されています。

7.4.2.23 #define D_CDM_INT "INT"

cdm_Define.h の 52 行で定義されています。

7.4.2.24 #define D_CDM_INT16 "Int16"

cdm_Define.h の 41 行で定義されています。

7.4.2.25 `#define D_CDM_INT32 "Int32"`

cdm_Define.h の 42 行で定義されています。

7.4.2.26 `#define D_CDM_INT64 "Int64"`

cdm_Define.h の 43 行で定義されています。

7.4.2.27 `#define D_CDM_INT8 "Int8"`

cdm_Define.h の 40 行で定義されています。

7.4.2.28 `#define D_CDM_LITTLE "little"`

cdm_Define.h の 59 行で定義されています。

7.4.2.29 `#define D_CDM_NIJK "nijk"`

cdm_Define.h の 57 行で定義されています。

7.4.2.30 `#define D_CDM_OFF "off"`

cdm_Define.h の 38 行で定義されています。

7.4.2.31 `#define D_CDM_ON "on"`

cdm_Define.h の 37 行で定義されています。

7.4.2.32 `#define D_CDM_UINT16 "UInt16"`

cdm_Define.h の 45 行で定義されています。

7.4.2.33 `#define D_CDM_UINT32 "UInt32"`

cdm_Define.h の 46 行で定義されています。

7.4.2.34 `#define D_CDM_UINT64 "UInt64"`

cdm_Define.h の 47 行で定義されています。

7.4.2.35 `#define D_CDM_UINT8 "UInt8"`

cdm_Define.h の 44 行で定義されています。

7.4.2.36 `#define stmpd_printf printf("%s (%d): ", __FILE__, __LINE__), printf`

cdm_Define.h の 366 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_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_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_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

7.8.1 マクロ定義

7.8.1.1 #define CDM_INLINE inline

cdm_DFI_inline.h の 34 行で定義されています。

7.9 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](#)

7.9.1 説明

[cdm_DFI_NETCDF](#) Class Header

作者

aics

[cdm_DFI_NETCDF.h](#) で定義されています。

7.10 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](#)

7.10.1 説明

[cdm_DFI_PLOT3D](#) Class Header

作者

aics

[cdm_DFI_PLOT3D.h](#) で定義されています。

7.11 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](#)

7.11.1 説明

[cdm_DFI_SPH](#) Class Header

作者

aics

[cdm_DFI_SPH.h](#) で定義されています。

7.12 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](#)

7.12.1 説明

[cdm_DFI_VTK](#) Class Header

作者

aics

[cdm_DFI_VTK.h](#) で定義されています。

7.13 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](#)

7.13.1 説明

[cdm_Domain](#) Class Header

作者

aics

[cdm_Domain.h](#) で定義されています。

7.14 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)

7.14.1 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。

7.14.2 マクロ定義

7.14.2.1 #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。

7.14.2.2 #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。

7.14.2.3 #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。

7.14.2.4 #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。

7.14.2.5 #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。

7.14.2.6 #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。

7.14.2.7 #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().

7.14.2.8 #define CDM_INLINE inline

cdm_endianUtil.h の 28 行で定義されています。

7.14.2.9 #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().

7.14.2.10 #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().

7.15 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](#)

7.15.1 説明

[cdm_FILE](#) Class Header

作者

aics

[cdm_FILE.h](#) で定義されています。

7.16 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](#)

7.16.1 説明

[cdm_FileInfo](#) Class Header

作者

aics

[cdm_FileInfo.h](#) で定義されています。

7.17 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](#)

7.17.1 説明

[cdm_FilePath](#) Class Header

作者

aics

[cdm_FilePath.h](#) で定義されています。

7.18 cdm_interp_ijkn.h

関数

- !CDMLib Cartesian Data Management library !Copyright (c) 2013-2015 Advanced Institute for Computational Science

7.18.1 関数

7.18.1.1 !CDMLib Cartesian Data Management library !Copyright (c)

7.19 cdm_interp_nijk.h

関数

- !CDMLib Cartesian Data Management library !Copyright (c) 2013-2015 Advanced Institute for Computational Science

7.19.1 関数

7.19.1.1 !CDMLib Cartesian Data Management library !Copyright (c)

7.20 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](#)

7.20.1 説明

[cdm_MPI](#) Class Header

作者

aics

[cdm_MPI.h](#) で定義されています。

7.21 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 >](#)

7.21.1 説明

[cdm_NonUniformDomain](#) Class Header

作者

aics

[cdm_NonUniformDomain.h](#) で定義されています。

7.22 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

7.22.1 説明

[cdm_NonUniformDomain](#) template Header

作者

aics

[cdm_NonUniformDomain_inline.h](#) で定義されています。

7.22.2 マクロ定義

7.22.2.1 #define CDM_INLINE inline

[cdm_NonUniformDomain_inline.h](#) の 23 行で定義されています。

7.23 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)`

7.23.1 マクロ定義

7.23.1.1 `#define MAXPATHLEN 512`

cdm_PathUtil.h の 17 行で定義されています。

7.24 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`

7.24.1 マクロ定義

7.24.1.1 `#define CDM_INLINE inline`

cdm_Plot3d_inline.h の 23 行で定義されています。

7.25 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`

7.25.1 説明

cdm_RANK & cdm_Process Class Header

作者

aics

cdm_Process.h で定義されています。

7.26 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](#)

7.26.1 説明

TextParser Control class Header.

作者

aics

cdm_TextParser.h で定義されています。

7.27 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](#)

7.27.1 説明

[cdm_Slice](#) & [cdm_TimeSliceClass](#) Header

作者

aics

[cdm_TimeSlice.h](#) で定義されています。

7.28 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 >](#)

7.29 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](#)

7.29.1 説明

[cdm_UnitElem](#) & [cdm_Unit](#) Class Header

作者

aics

[cdm_Unit.h](#) で定義されています。

7.30 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](#) "0.8.1"
- #define [CDM_REVISION](#) "20150610_1300"

7.30.1 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。

7.30.2 マクロ定義

7.30.2.1 #define CDM_REVISION "20150610_1300"

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。

7.30.2.2 #define CDM_VERSION_NO "0.8.1"

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\(\)](#).

7.31 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](#)

7.31.1 説明

[cdm_VisIt](#) Class Header

作者

aics

[cdm_VisIt.h](#) で定義されています。

7.32 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 recvcount, [MPI_Datatype](#) recvttype, [MPI_Comm](#) comm)
- int [MPI_Gather](#) (void *sendbuf, int sendcnt, [MPI_Datatype](#) sendtype, void *recvbuf, int recvcnt, [MPI_Datatype](#) recvttype, int root, [MPI_Comm](#) comm)

7.32.1 マクロ定義

7.32.1.1 #define MPI_CHAR 2

mpi_stubs.h の 22 行で定義されています。

7.32.1.2 #define MPI_COMM_WORLD 0

mpi_stubs.h の 20 行で定義されています。

7.32.1.3 #define MPI_INT 1

mpi_stubs.h の 21 行で定義されています。

7.32.1.4 #define MPI_SUCCESS true

mpi_stubs.h の 24 行で定義されています。

7.32.2 型定義

7.32.2.1 typedef int MPI_Comm

mpi_stubs.h の 18 行で定義されています。

7.32.2.2 typedef int MPI_Datatype

mpi_stubs.h の 19 行で定義されています。

7.32.3 関数

7.32.3.1 int MPI_Allgather (void * sendbuf, int sendcount, MPI_Datatype sendtype, void * recvbuf, int recvcount, MPI_Datatype recvttype, MPI_Comm comm) [inline]

mpi_stubs.h の 40 行で定義されています。

7.32.3.2 `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()`.

7.32.3.3 `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()`.

7.32.3.4 `int MPI_Gather (void * sendbuf, int sendcnt, MPI_Datatype sendtype, void * recvbuf, int recvcnt, MPI_Datatype recvtpe, int root, MPI_Comm comm) [inline]`

mpi_stubs.h の 47 行で定義されています。

7.32.3.5 `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, [38](#)
- ~cdm_DFI_AVS
 - cdm_DFI_AVS, [70](#)
- ~cdm_DFI_BOV
 - cdm_DFI_BOV, [75](#)
- ~cdm_DFI_NETCDF
 - cdm_DFI_NETCDF, [80](#)
- ~cdm_DFI_PLOT3D
 - cdm_DFI_PLOT3D, [91](#)
- ~cdm_DFI_SPH
 - cdm_DFI_SPH, [98](#)
- ~cdm_DFI_VTK
 - cdm_DFI_VTK, [102](#)
- ~cdm_Domain
 - cdm_Domain, [106](#)
- ~cdm_FILE
 - cdm_FILE, [111](#)
- ~cdm_FileInfo
 - cdm_FileInfo, [116](#)
- ~cdm_FilePath
 - cdm_FilePath, [119](#)
- ~cdm_MPI
 - cdm_MPI, [121](#)
- ~cdm_NonUniformDomain
 - cdm_NonUniformDomain, [126](#)
- ~cdm_Process
 - cdm_Process, [136](#)
- ~cdm_Rank
 - cdm_Rank, [141](#)
- ~cdm_Slice
 - cdm_Slice, [143](#)
- ~cdm_TextParser
 - cdm_TextParser, [147](#)
- ~cdm_TimeSlice
 - cdm_TimeSlice, [150](#)
- ~cdm_TypeArray
 - cdm_TypeArray, [155](#)
- ~cdm_Unit
 - cdm_Unit, [159](#)
- ~cdm_UnitElem
 - cdm_UnitElem, [161](#)
- ~cdm_VisIt
 - cdm_VisIt, [164](#)
- ~stVarInfo
 - cdm_DFI_NETCDF::stVarInfo, [166](#)
- _CDM_IDX_IJ
 - cdm_Define.h, [173](#)
- _CDM_IDX_IJK
 - cdm_Define.h, [174](#)
- _CDM_IDX_IJKN
 - cdm_Define.h, [174](#)
- _CDM_IDX_NIJ
 - cdm_Define.h, [175](#)
- _CDM_IDX_NIJK
 - cdm_Define.h, [175](#)
- _CDM_TAB_STR
 - cdm_Define.h, [176](#)
- _CDM_WRITE_TAB
 - cdm_Define.h, [176](#)
- _DATA_UNKNOWN
 - cdm_DFI_SPH, [97](#)
- _DOUBLE
 - cdm_DFI_SPH, [97](#)
- _FLOAT
 - cdm_DFI_SPH, [97](#)
- _REAL_UNKNOWN
 - cdm_DFI_SPH, [97](#)
- _SCALAR
 - cdm_DFI_SPH, [97](#)
- _VECTOR
 - cdm_DFI_SPH, [97](#)
- _getArraySize
 - cdm_Array, [24](#)
- _getArraySizeInt
 - cdm_Array, [24](#)
- _val
 - cdm_TypeArray, [155](#)
- ActiveSubdomainFile
 - cdm_Domain, [109](#)
- AddNcUnit
 - cdm_DFI_NETCDF, [80](#)
- AddSlice
 - cdm_TimeSlice, [151](#)
- AddTimeSlice
 - cdm_DFI, [38](#)
- AddUnit
 - cdm_DFI, [39](#)
 - cdm_Unit, [159](#)
- ArrayShape
 - cdm_FileInfo, [117](#)
- AveragedStep
 - cdm_Slice, [145](#)
- AveragedTime

- cdm_Slice, 145
- avr_mode
 - cdm_Slice, 145
- BSWAP16
 - cdm_endianUtil.h, 183
- BSWAP32
 - cdm_endianUtil.h, 183
- BSWAP64
 - cdm_endianUtil.h, 184
- BSWAP_X_16
 - cdm_endianUtil.h, 184
- BSWAP_X_32
 - cdm_endianUtil.h, 184
- BSWAP_X_64
 - cdm_endianUtil.h, 184
- BSWAPVEC
 - cdm_endianUtil.h, 184
- bc_id
 - cdm_Rank, 141
- BsetDiff
 - cdm_UnitElem, 163
- C_CDM_RANKNOPREFIX
 - CDM, 17
- c_id
 - cdm_Rank, 141
- CDM, 9
 - C_CDM_RANKNOPREFIX, 17
 - cdmPath_ConnectPath, 16
 - cdmPath_DirName, 16
 - cdmPath_FileName, 16
 - cdmPath_getDelimChar, 16
 - cdmPath_getDelimString, 16
 - cdmPath_hasDrive, 16
 - cdmPath_isAbsolute, 16
 - 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, 15
 - E_CDM_DIFFDIV_SAMERES, 15
 - 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, 13
 - E_CDM_ERROR_NOMATCH_ENDIAN, 13
 - E_CDM_ERROR_NUM_OF_GUIDECCELLS, 13
 - E_CDM_ERROR_OPEN_COORDINATEFILE, 13
 - E_CDM_ERROR_OPEN_FIELDDATA, 14
 - E_CDM_ERROR_OPEN_SBDM, 14
 - E_CDM_ERROR_READ_BOV_FILE, 13
 - 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_FILEPATH_PROCESS, 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, 12
 - E_CDM_ERROR_READ_DFI_NO_RANK, 12
 - E_CDM_ERROR_READ_DFI_NO_SLICE, 12
 - E_CDM_ERROR_READ_DFI_NUMVARIABLES, 12
 - E_CDM_ERROR_READ_DFI_PREFIX, 12
 - E_CDM_ERROR_READ_DFI_STEP, 12
 - E_CDM_ERROR_READ_DFI_TAILINDEX, 12
 - E_CDM_ERROR_READ_DFI_TIME, 12
 - 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, 13
 - E_CDM_ERROR_READ_FIELD_DATA_RECORD, 13
 - E_CDM_ERROR_READ_FIELD_HEADER_RECORD, 13
 - E_CDM_ERROR_READ_FIELDDATA_FILE, 13

- E_CDM_ERROR_READ_FILEINFO, 13
- E_CDM_ERROR_READ_FILEPATH, 13
- E_CDM_ERROR_READ_INDEXFILE_OPENER-
ROR, 13
- E_CDM_ERROR_READ_MPI, 13
- E_CDM_ERROR_READ_NETCDF_FUNC, 13
- E_CDM_ERROR_READ_NETCDF_MISMATCH_-
TYPE, 13
- E_CDM_ERROR_READ_NETCDF_VAR_1D, 13
- E_CDM_ERROR_READ_PROCESS, 13
- E_CDM_ERROR_READ_PROCFILE_OPENER-
ROR, 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_SBDM_NUMDOMAIN_ZERO,
14
- E_CDM_ERROR_TEXTPARSER, 13
- E_CDM_ERROR_UNMATCH_NUM_OF_VARIA-
BLES, 13
- E_CDM_ERROR_UNMATCH_VOXELSIZE, 13
- E_CDM_ERROR_WRITE_DFI_NETCDF, 14
- E_CDM_ERROR_WRITE_DOMAIN, 14
- E_CDM_ERROR_WRITE_FIELD_AVERAGED_-
RECORD, 14
- E_CDM_ERROR_WRITE_FIELD_DATA_RECO-
RD, 14
- E_CDM_ERROR_WRITE_FIELD_HEADER_RE-
CORD, 14
- E_CDM_ERROR_WRITE_FILEINFO, 14
- E_CDM_ERROR_WRITE_FILEPATH, 14
- E_CDM_ERROR_WRITE_GRIDFILE, 14
- E_CDM_ERROR_WRITE_INDEXFILE_OPENER-
ROR, 14
- E_CDM_ERROR_WRITE_INDEXFILENAME_EM-
PTY, 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_OPENER-
ROR, 14
- E_CDM_ERROR_WRITE_PROCFILENAME_EM-
PTY, 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, 14
- E_CDM_ERRORCODE, 12
- E_CDM_FILE_TYPE, 14
- E_CDM_FILE_TYPE_ASCII, 14
- E_CDM_FILE_TYPE_BINARY, 14
- E_CDM_FILE_TYPE_DEFAULT, 14
- E_CDM_FILE_TYPE_FBINAR, 14
- E_CDM_FLOAT32, 11
- E_CDM_FLOAT64, 12
- E_CDM_FMT_AVS, 15
- E_CDM_FMT_BOV, 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_DEFAULT, 15
- E_CDM_FNAME_RANK, 15
- E_CDM_FNAME_RANK_STEP, 15
- E_CDM_FNAME_STEP_RANK, 15
- E_CDM_FORMAT, 14
- 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, 15
- E_CDM_READTYPE_UNKNOWN, 15
- E_CDM_SAMEDIV_REFINEMENT, 15
- E_CDM_SAMEDIV_SAMERES, 15
- E_CDM_SUCCESS, 12
- E_CDM_UINT16, 11
- E_CDM_UINT32, 11
- E_CDM_UINT64, 11
- E_CDM_UINT8, 11
- E_CDM_WARN_GETUNIT, 14
- ExtractPathWithoutExt, 16
- vvfPath_emitDrive, 16
- CDM_INLINE
 - cdm_Array_inline.h, 170
 - cdm_DFI_inline.h, 180
 - cdm_endianUtil.h, 185
 - cdm_NonUniformDomain_inline.h, 188
 - cdm_Plot3d_inline.h, 189
- CDM_MEMFUN
 - cdm_Array_inline.h, 170

CDM_REVISION
 cdm_Version.h, 192
 CDM_VERSION_NO
 cdm_Version.h, 192
 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, 169
 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, 169
 cdm_interp_ijk_r4_, 169
 cdm_interp_ijk_r8_, 169
 cdm_interp_nijk_r4_, 170
 cdm_interp_nijk_r8_, 170
 cdm_Array_inline.h, 170
 CDM_INLINE, 170
 CDM_MEMFUN, 170
 cdm_Create_dfiProcessInfo
 cdm_DFI, 39
 cdm_DFI, 31
 ~cdm_DFI, 38
 AddTimeSlice, 38
 AddUnit, 39
 cdm_Create_dfiProcessInfo, 39
 cdm_DFI, 38
 cdm_DFI, 38
 CheckAddWriteMode, 39
 CheckReadRank, 39
 CheckReadType, 40
 ConvDatatypeE2S, 40
 ConvDatatypeS2E, 40
 CreateReadStartEnd, 40
 DFI_Domain, 66
 DFI_Finfo, 66
 DFI_Fpath, 66
 DFI_MPI, 66
 DFI_Process, 67
 DFI_TimeSlice, 67
 DFI_Unit, 67
 DFI_VisIt, 67
 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, 45
 GetDFIGlobalVoxel, 46
 GetDFIType, 46
 GetDFITypeString, 46
 GetDataType, 45
 GetDataTypeString, 45
 GetFileFormat, 46
 GetFileFormatString, 46
 getMinMax, 46
 GetNumGuideCell, 47
 GetNumVariables, 47
 GetUnit, 47
 GetUnitElem, 47
 getVariableName, 47
 getVectorMinMax, 48
 getVersionInfo, 48
 GetcdmDomain, 44
 GetcdmFileInfo, 44
 GetcdmFilePath, 44
 GetcdmMPI, 44

GetcdmProcess, 44
 GetcdmRankNoPrefix, 44
 GetcdmTimeSlice, 45
 GetcdmUnit, 45
 GetcdmVisIt, 45
 m_RankID, 68
 m_bgrid_interp_flag, 67
 m_comm, 67
 m_directoryPath, 68
 m_indexDfiName, 68
 m_input_type, 68
 m_output_fname, 68
 m_output_type, 68
 m_output_type_coord, 68
 m_read_type, 69
 m_readRankList, 69
 MakeDirectory, 48
 MakeDirectoryPath, 48
 MakeDirectorySub, 48
 normalizeBaseTime, 49
 normalizeDeltaT, 49
 normalizeIntervalTime, 49
 normalizeLastTime, 49
 normalizeStartTime, 49
 normalizeTime, 49
 read_Datarecord, 50
 read_HeaderRecord, 50
 read_averaged, 50
 ReadData, 51, 53
 ReadFieldData, 53
 ReadInit, 54
 set_RankID, 55
 set_input_type, 54
 set_interp_flag, 54
 set_output_fname, 55
 set_output_type, 55
 set_output_type_coord, 55
 setGridData, 56, 57
 setIntervalStep, 57
 setIntervalTime, 57
 SetTimeSliceFlag, 57
 setVariableName, 57
 SetcdmDomain, 55
 SetcdmFilePath, 55
 SetcdmMPI, 56
 SetcdmProcess, 56
 SetcdmRankNoPrefix, 56
 SetcdmTimeSlice, 56
 SetcdmUnit, 56
 SetcdmVisIt, 56
 VolumeDataDivide, 59
 write_DataRecord, 59
 write_GridData, 60
 write_HeaderRecord, 60
 write_ascii_header, 59
 write_averaged, 59
 WriteData, 60, 61
 WriteFieldData, 61
 WriteFieldDataFile, 62
 WriteGridFile, 63
 WriteIndexDfiFile, 63
 WriteInit, 63–65
 WriteProcDfiFile, 66
 cdm_DFI.h, 179
 cdm_DFI_AVS, 69
 ~cdm_DFI_AVS, 70
 cdm_DFI_AVS, 70
 cdm_DFI_AVS, 70
 read_Datarecord, 71
 read_HeaderRecord, 71
 read_averaged, 70
 write_DataRecord, 73
 write_HeaderRecord, 73
 write_ascii_header, 72
 write_averaged, 72
 write_avs_cord, 72
 write_avs_header, 73
 cdm_DFI_AVS.h, 179
 cdm_DFI_BOV, 73
 ~cdm_DFI_BOV, 75
 cdm_DFI_BOV, 74
 cdm_DFI_BOV, 74
 read_Datarecord, 75
 read_HeaderRecord, 76
 read_averaged, 75
 write_DataRecord, 77
 write_HeaderRecord, 77
 write_ascii_header, 76
 write_averaged, 76
 cdm_DFI_BOV.h, 180
 cdm_DFI_NETCDF, 77
 ~cdm_DFI_NETCDF, 80
 AddNcUnit, 80
 cdm_DFI_NETCDF, 80
 cdm_DFI_NETCDF, 80
 CheckAddWriteMode, 81
 CheckArrayVarInfo, 81
 GetCdmType, 81
 GetDimName, 81
 GetNcType, 82
 GetNcUnit, 82
 GetVarInfo, 82
 m_NC_Unit, 88
 m_nameT, 88
 m_nameX, 88
 m_nameY, 88
 m_nameZ, 88
 m_varInfoT, 89
 m_varInfoX, 89
 m_varInfoY, 89
 m_varInfoZ, 89
 m_vecVarInfo, 89
 m_writeFlag, 89
 read_Datarecord, 83
 read_HeaderRecord, 83, 85
 read_averaged, 82

- ReadAdditionalTP, 85
- SetDimName, 85
- SetWriteFlag, 86
- write_AttUnits, 86
- write_DataRecord, 86, 87
- write_HeaderRecord, 87
- write_averaged, 86
- WriteAdditionalTP, 88
- cdm_DFI_NETCDF.h, 181
- cdm_DFI_NETCDF::stDimInfo, 165
 - id, 165
 - len, 165
 - name, 165
- cdm_DFI_NETCDF::stVarInfo, 166
 - ~stVarInfo, 166
 - clear, 167
 - data, 167
 - dims, 167
 - endian, 167
 - id, 167
 - name, 167
 - stVarInfo, 166
 - type, 167
 - unit, 168
- cdm_DFI_PLOT3D, 89
 - ~cdm_DFI_PLOT3D, 91
 - cdm_DFI_PLOT3D, 91
 - cdm_DFI_PLOT3D, 91
 - read_Datarecord, 91
 - read_Func, 93
 - read_HeaderRecord, 93
 - read_averaged, 91
 - write_DataRecord, 94
 - write_Func, 94, 95
 - write_GridData, 95
 - write_HeaderRecord, 95
 - write_XYZ, 95
 - write_averaged, 94
- cdm_DFI_PLOT3D.h, 181
- cdm_DFI_SPH, 96
 - ~cdm_DFI_SPH, 98
 - _DATA_UNKNOWN, 97
 - _DOUBLE, 97
 - _FLOAT, 97
 - _REAL_UNKNOWN, 97
 - _SCALAR, 97
 - _VECTOR, 97
 - cdm_DFI_SPH, 97
 - cdm_DFI_SPH, 97
 - DataDims, 97
 - read_Datarecord, 98
 - read_HeaderRecord, 98
 - read_averaged, 98
 - RealType, 97
 - write_DataRecord, 100
 - write_HeaderRecord, 100
 - write_averaged, 100
- cdm_DFI_SPH.h, 181
- cdm_DFI_VTK, 101
 - ~cdm_DFI_VTK, 102
 - cdm_DFI_VTK, 102
 - cdm_DFI_VTK, 102
 - read_Datarecord, 103
 - read_HeaderRecord, 103
 - read_averaged, 102
 - write_DataRecord, 104
 - write_HeaderRecord, 104
 - write_averaged, 103
- cdm_DFI_VTK.h, 182
- cdm_DFI_inline.h, 180
 - CDM_INLINE, 180
- cdm_Define.h, 170
 - _CDM_IDX_IJ, 173
 - _CDM_IDX_IJK, 174
 - _CDM_IDX_IJKN, 174
 - _CDM_IDX_NIJ, 175
 - _CDM_IDX_NIJK, 175
 - _CDM_TAB_STR, 176
 - _CDM_WRITE_TAB, 176
 - D_CDM_BIG, 176
 - D_CDM_BYTE, 176
 - D_CDM_DFITYPE_CARTESIAN, 176
 - D_CDM_DFITYPE_NON_UNIFORM_CARTESIA-
N, 176
 - D_CDM_DOUBLE, 176
 - D_CDM_EXT_BOV, 176
 - D_CDM_EXT_BOV_DATAFILE, 177
 - D_CDM_EXT_FUNC, 177
 - D_CDM_EXT_NC, 177
 - D_CDM_EXT_SPH, 177
 - D_CDM_EXT_VTK, 177
 - D_CDM_FLOAT, 177
 - D_CDM_FLOAT32, 177
 - D_CDM_FLOAT64, 177
 - D_CDM_IJNK, 177
 - D_CDM_INT, 177
 - D_CDM_INT16, 177
 - D_CDM_INT32, 177
 - D_CDM_INT64, 178
 - D_CDM_INT8, 178
 - D_CDM_LITTLE, 178
 - D_CDM_NIJK, 178
 - D_CDM_OFF, 178
 - D_CDM_ON, 178
 - D_CDM_UINT16, 178
 - D_CDM_UINT32, 178
 - D_CDM_UINT64, 178
 - D_CDM_UINT8, 178
 - stmpd_printf, 178
- cdm_Domain, 105
 - ~cdm_Domain, 106
 - ActiveSubdomainFile, 109
 - cdm_Domain, 106
 - cdm_Domain, 106
 - CellX, 106
 - CellY, 107

- CellZ, [107](#)
- Clear, [107](#)
- GetCoordinateFile, [107](#)
- GetCoordinateFileEndian, [107](#)
- GetCoordinateFilePrecision, [108](#)
- GetCoordinateFileType, [108](#)
- GlobalDivision, [109](#)
- GlobalOrigin, [109](#)
- GlobalRegion, [110](#)
- GlobalVoxel, [110](#)
- NodeX, [108](#)
- NodeY, [108](#)
- NodeZ, [108](#)
- Pitch, [110](#)
- Read, [109](#)
- Write, [109](#)
- cdm_Domain.h, [182](#)
- cdm_FILE, [110](#)
 - ~cdm_FILE, [111](#)
 - cdm_FILE, [111](#)
 - cdm_FILE, [111](#)
 - CloseFile, [111](#)
 - m_addMode, [113](#)
 - m_fname, [113](#)
 - m_format, [113](#)
 - m_fp, [113](#)
 - m_ncid, [113](#)
 - OpenReadBinary, [112](#)
 - OpenWriteAscii, [112](#)
 - OpenWriteBinary, [112](#)
- cdm_FILE.h, [185](#)
- cdm_FileInfo, [113](#)
 - ~cdm_FileInfo, [116](#)
 - ArrayShape, [117](#)
 - cdm_FileInfo, [114](#)
 - cdm_FileInfo, [114](#)
 - DFIType, [117](#)
 - DataType, [117](#)
 - DirectoryPath, [117](#)
 - Endian, [117](#)
 - FieldFilenameFormat, [117](#)
 - FileFormat, [118](#)
 - getVariableName, [116](#)
 - GuideCell, [118](#)
 - NumVariables, [118](#)
 - Prefix, [118](#)
 - RankNoPrefix, [118](#)
 - Read, [116](#)
 - setVariableName, [116](#)
 - TimeSliceDirFlag, [118](#)
 - VariableName, [118](#)
 - Write, [117](#)
- cdm_FileInfo.h, [186](#)
- cdm_FilePath, [119](#)
 - ~cdm_FilePath, [119](#)
 - cdm_FilePath, [119](#)
 - cdm_FilePath, [119](#)
 - ProcDFIFile, [120](#)
 - Read, [120](#)
 - Write, [120](#)
- cdm_FilePath.h, [186](#)
- cdm_MPI, [120](#)
 - ~cdm_MPI, [121](#)
 - cdm_MPI, [121](#)
 - cdm_MPI, [121](#)
 - NumberOfGroup, [123](#)
 - NumberOfRank, [123](#)
 - Read, [121](#)
 - Write, [121](#)
- cdm_MPI.h, [187](#)
- cdm_NonUniformDomain
 - ~cdm_NonUniformDomain, [126](#)
 - cdm_NonUniformDomain, [125](#), [126](#)
 - cdm_NonUniformDomain, [125](#), [126](#)
 - CellX, [126](#)
 - CellY, [127](#)
 - CellZ, [127](#)
 - Clear, [127](#)
 - CoordinateFile, [132](#)
 - CoordinateFileEndian, [133](#)
 - CoordinateFilePrecision, [133](#)
 - CoordinateFileType, [133](#)
 - GetCoordinateFile, [127](#)
 - GetCoordinateFileEndian, [128](#)
 - GetCoordinateFilePrecision, [128](#)
 - GetCoordinateFileType, [128](#)
 - NodeX, [128](#)
 - NodeY, [130](#)
 - NodeZ, [130](#)
 - pit_gcXend, [133](#)
 - pit_gcXsta, [133](#)
 - pit_gcYend, [133](#)
 - pit_gcYsta, [133](#)
 - pit_gcZend, [134](#)
 - pit_gcZsta, [134](#)
 - Read, [130](#)
 - Read_CoordinateFile, [131](#)
 - readCoordData, [131](#)
 - readCoordDataCount, [132](#)
 - Write, [132](#)
 - XCoordinates, [134](#)
 - YCoordinates, [134](#)
 - ZCoordinates, [134](#)
- cdm_NonUniformDomain< T >, [123](#)
- cdm_NonUniformDomain.h, [187](#)
- cdm_NonUniformDomain_inline.h, [188](#)
 - CDM_INLINE, [188](#)
- cdm_PathUtil.h, [188](#)
 - MAXPATHLEN, [189](#)
- cdm_Plot3d_inline.h, [189](#)
 - CDM_INLINE, [189](#)
- cdm_Process, [135](#)
 - ~cdm_Process, [136](#)
 - cdm_Process, [136](#)
 - cdm_Process, [136](#)
 - CheckReadRank, [136](#)

- CheckStartEnd, [136](#)
- CreateHeadMap, [137](#)
- CreateRankList, [137](#)
- CreateRankMap, [138](#)
- CreateSubDomainInfo, [138](#)
- headT, [136](#)
- isMatchEndianSbdmMagick, [139](#)
- m_rankMap, [140](#)
- RankList, [140](#)
- Read, [139](#)
- ReadActiveSubdomainFile, [139](#)
- Write, [139](#)
- cdm_Process.h, [189](#)
- cdm_Rank, [140](#)
 - ~cdm_Rank, [141](#)
 - bc_id, [141](#)
 - c_id, [141](#)
 - cdm_Rank, [141](#)
 - cdm_Rank, [141](#)
 - HeadIndex, [142](#)
 - HostName, [142](#)
 - RankID, [142](#)
 - Read, [141](#)
 - TailIndex, [142](#)
 - VoxelSize, [142](#)
 - Write, [141](#)
- cdm_Slice, [142](#)
 - ~cdm_Slice, [143](#)
 - AveragedStep, [145](#)
 - AveragedTime, [145](#)
 - avr_mode, [145](#)
 - cdm_Slice, [143](#)
 - cdm_Slice, [143](#)
 - Max, [145](#)
 - Min, [145](#)
 - Read, [143](#)
 - step, [145](#)
 - time, [145](#)
 - VectorMax, [145](#)
 - VectorMin, [146](#)
 - Write, [143](#)
- cdm_TextParser, [146](#)
 - ~cdm_TextParser, [147](#)
 - cdm_TextParser, [147](#)
 - cdm_TextParser, [147](#)
 - chkLabel, [147](#)
 - chkNode, [147](#)
 - countLabels, [147](#)
 - GetNodeStr, [148](#)
 - getTPPtr, [148](#)
 - getTPinstance, [148](#)
 - GetValue, [148](#)
 - GetVector, [149](#)
 - readTPfile, [149](#)
 - remove, [149](#)
 - tp, [150](#)
- cdm_TextParser.h, [190](#)
- cdm_TimeSlice, [150](#)
 - ~cdm_TimeSlice, [150](#)
 - AddSlice, [151](#)
 - cdm_TimeSlice, [150](#)
 - cdm_TimeSlice, [150](#)
 - getMinMax, [151](#)
 - getVectorMinMax, [151](#)
 - Read, [151](#)
 - SliceList, [153](#)
 - Write, [153](#)
- cdm_TimeSlice.h, [190](#)
- cdm_TypeArray
 - ~cdm_TypeArray, [155](#)
 - _val, [155](#)
 - cdm_TypeArray, [155](#)
 - cdm_TypeArray, [155](#)
 - copyArray, [155](#), [156](#)
 - copyArrayNvari, [156](#)
 - copyArrayNvari_to_ijk, [156](#)
 - getData, [156](#)
 - hval, [157](#)
 - m_data, [158](#)
 - m_outptr, [158](#)
 - readBinary, [157](#)
 - val, [157](#)
 - writeAscii, [157](#)
 - writeBinary, [157](#)
- cdm_TypeArray< T >, [153](#)
- cdm_TypeArray.h, [191](#)
- cdm_Unit, [158](#)
 - ~cdm_Unit, [159](#)
 - AddUnit, [159](#)
 - cdm_Unit, [159](#)
 - cdm_Unit, [159](#)
 - GetUnit, [159](#)
 - GetUnitElem, [159](#)
 - Read, [160](#)
 - UnitList, [160](#)
 - Write, [160](#)
- cdm_Unit.h, [191](#)
- cdm_UnitElem, [160](#)
 - ~cdm_UnitElem, [161](#)
 - BsetDiff, [163](#)
 - cdm_UnitElem, [161](#)
 - cdm_UnitElem, [161](#)
 - difference, [163](#)
 - Name, [163](#)
 - Read, [161](#)
 - reference, [163](#)
 - Unit, [163](#)
 - Write, [161](#)
- cdm_Version.h, [191](#)
 - CDM_REVISION, [192](#)
 - CDM_VERSION_NO, [192](#)
- cdm_VisIt, [163](#)
 - ~cdm_VisIt, [164](#)
 - cdm_VisIt, [164](#)
 - cdm_VisIt, [164](#)
 - PlotGC, [165](#)

- Read, [164](#)
- Write, [164](#)
- cdm_VisIt.h, [192](#)
- cdm_endianUtil.h, [183](#)
 - BSWAP16, [183](#)
 - BSWAP32, [183](#)
 - BSWAP64, [184](#)
 - BSWAP_X_16, [184](#)
 - BSWAP_X_32, [184](#)
 - BSWAP_X_64, [184](#)
 - BSWAPVEC, [184](#)
 - CDM_INLINE, [185](#)
 - DBSWAPVEC, [185](#)
 - SBSWAPVEC, [185](#)
- cdm_interp_ijkn.h, [187](#)
- cdm_interp_ijkn_r4_
 - cdm_Array.h, [169](#)
- cdm_interp_ijkn_r8_
 - cdm_Array.h, [169](#)
- cdm_interp_nijk.h, [187](#)
- cdm_interp_nijk_r4_
 - cdm_Array.h, [170](#)
- cdm_interp_nijk_r8_
 - cdm_Array.h, [170](#)
- cdmPath_ConnectPath
 - CDM, [16](#)
- cdmPath_DirName
 - CDM, [16](#)
- cdmPath_FileName
 - CDM, [16](#)
- cdmPath_getDelimChar
 - CDM, [16](#)
- cdmPath_getDelimString
 - CDM, [16](#)
- cdmPath_hasDrive
 - CDM, [16](#)
- cdmPath_isAbsolute
 - CDM, [16](#)
- CellX
 - cdm_Domain, [106](#)
 - cdm_NonUniformDomain, [126](#)
- CellY
 - cdm_Domain, [107](#)
 - cdm_NonUniformDomain, [127](#)
- CellZ
 - cdm_Domain, [107](#)
 - cdm_NonUniformDomain, [127](#)
- CheckAddWriteMode
 - cdm_DFI, [39](#)
 - cdm_DFI_NETCDF, [81](#)
- CheckArrayVarInfo
 - cdm_DFI_NETCDF, [81](#)
- CheckReadRank
 - cdm_DFI, [39](#)
 - cdm_Process, [136](#)
- CheckReadType
 - cdm_DFI, [40](#)
- CheckStartEnd
 - cdm_Process, [136](#)
- chkLabel
 - cdm_TextParser, [147](#)
- chkNode
 - cdm_TextParser, [147](#)
- Clear
 - cdm_Domain, [107](#)
 - cdm_NonUniformDomain, [127](#)
- clear
 - cdm_ActiveSubDomain, [20](#)
 - cdm_DFI_NETCDF::stVarInfo, [167](#)
- CloseFile
 - cdm_FILE, [111](#)
- ConvDatatypeE2S
 - cdm_DFI, [40](#)
- ConvDatatypeS2E
 - cdm_DFI, [40](#)
- CoordinateFile
 - cdm_NonUniformDomain, [132](#)
- CoordinateFileEndian
 - cdm_NonUniformDomain, [133](#)
- CoordinateFilePrecision
 - cdm_NonUniformDomain, [133](#)
- CoordinateFileType
 - cdm_NonUniformDomain, [133](#)
- copyArray
 - cdm_Array, [24](#)
 - cdm_TypeArray, [155](#), [156](#)
- copyArrayNvari
 - cdm_Array, [25](#)
 - cdm_TypeArray, [156](#)
- copyArrayNvari_to_ijk
 - cdm_Array, [25](#)
 - cdm_TypeArray, [156](#)
- countLabels
 - cdm_TextParser, [147](#)
- CreateHeadMap
 - cdm_Process, [137](#)
- CreateRankList
 - cdm_Process, [137](#)
- CreateRankMap
 - cdm_Process, [138](#)
- CreateReadStartEnd
 - cdm_DFI, [40](#)
- CreateSubDomainInfo
 - cdm_Process, [138](#)
- D_CDM_BIG
 - cdm_Define.h, [176](#)
- D_CDM_BYTE
 - cdm_Define.h, [176](#)
- D_CDM_DFITYPE_CARTESIAN
 - cdm_Define.h, [176](#)
- D_CDM_DFITYPE_NON_UNIFORM_CARTESIAN
 - cdm_Define.h, [176](#)
- D_CDM_DOUBLE
 - cdm_Define.h, [176](#)
- D_CDM_EXT_BOV
 - cdm_Define.h, [176](#)

- D_CDM_EXT_BOV_DATAFILE
 - cdm_Define.h, [177](#)
- D_CDM_EXT_FUNC
 - cdm_Define.h, [177](#)
- D_CDM_EXT_NC
 - cdm_Define.h, [177](#)
- D_CDM_EXT_SPH
 - cdm_Define.h, [177](#)
- D_CDM_EXT_VTK
 - cdm_Define.h, [177](#)
- D_CDM_FLOAT
 - cdm_Define.h, [177](#)
- D_CDM_FLOAT32
 - cdm_Define.h, [177](#)
- D_CDM_FLOAT64
 - cdm_Define.h, [177](#)
- D_CDM_IJNK
 - cdm_Define.h, [177](#)
- D_CDM_INT
 - cdm_Define.h, [177](#)
- D_CDM_INT16
 - cdm_Define.h, [177](#)
- D_CDM_INT32
 - cdm_Define.h, [177](#)
- D_CDM_INT64
 - cdm_Define.h, [178](#)
- D_CDM_INT8
 - cdm_Define.h, [178](#)
- D_CDM_LITTLE
 - cdm_Define.h, [178](#)
- D_CDM_NIJK
 - cdm_Define.h, [178](#)
- D_CDM_OFF
 - cdm_Define.h, [178](#)
- D_CDM_ON
 - cdm_Define.h, [178](#)
- D_CDM_UINT16
 - cdm_Define.h, [178](#)
- D_CDM_UINT32
 - cdm_Define.h, [178](#)
- D_CDM_UINT64
 - cdm_Define.h, [178](#)
- D_CDM_UINT8
 - cdm_Define.h, [178](#)
- DBSWAPVEC
 - cdm_endianUtil.h, [185](#)
- DFI_Domain
 - cdm_DFI, [66](#)
- DFI_Finfo
 - cdm_DFI, [66](#)
- DFI_Fpath
 - cdm_DFI, [66](#)
- DFI_MPI
 - cdm_DFI, [66](#)
- DFI_Process
 - cdm_DFI, [67](#)
- DFI_TimeSlice
 - cdm_DFI, [67](#)
- DFI_Unit
 - cdm_DFI, [67](#)
- DFI_VisIt
 - cdm_DFI, [67](#)
- DFIType
 - cdm_FileInfo, [117](#)
- data
 - cdm_DFI_NETCDF::stVarInfo, [167](#)
- DataDims
 - cdm_DFI_SPH, [97](#)
- DataType
 - cdm_FileInfo, [117](#)
- difference
 - cdm_UnitElem, [163](#)
- dims
 - cdm_DFI_NETCDF::stVarInfo, [167](#)
- DirectoryPath
 - cdm_FileInfo, [117](#)
- 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, [15](#)
- E_CDM_DIFFDIV_SAMERES
 - CDM, [15](#)
- 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, [13](#)
- E_CDM_ERROR_NOMATCH_ENDIAN
 - CDM, [13](#)
- E_CDM_ERROR_NUM_OF_GUIDECCELLS
 - CDM, [13](#)
- 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, 13
- 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_COORDINATEFILEEN-
DIAN
CDM, 13
- E_CDM_ERROR_READ_DFI_COORDINATEFILEPR-
ECISION
CDM, 13
- E_CDM_ERROR_READ_DFI_COORDINATEFILETY-
PE
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_FIELDFILENAMEFOR-
MAT
CDM, 13
- E_CDM_ERROR_READ_DFI_FILEFORMAT
CDM, 12
- 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, 12
- E_CDM_ERROR_READ_DFI_NO_RANK
CDM, 12
- E_CDM_ERROR_READ_DFI_NO_SLICE
CDM, 12
- E_CDM_ERROR_READ_DFI_NUMVARIABLES
CDM, 12
- E_CDM_ERROR_READ_DFI_PREFIX
CDM, 12
- E_CDM_ERROR_READ_DFI_STEP
CDM, 12
- E_CDM_ERROR_READ_DFI_TAILINDEX
CDM, 12
- E_CDM_ERROR_READ_DFI_TIME
CDM, 12
- E_CDM_ERROR_READ_DFI_TIMESLICEDIRECTOR-
Y
CDM, 12
- E_CDM_ERROR_READ_DFI_VOXELSIZE
CDM, 12
- E_CDM_ERROR_READ_DOMAIN
CDM, 13
- E_CDM_ERROR_READ_FIELD_AVERAGED_RECO-
RD
CDM, 13
- E_CDM_ERROR_READ_FIELD_DATA_RECORD
CDM, 13
- E_CDM_ERROR_READ_FIELD_HEADER_RECORD
CDM, 13
- E_CDM_ERROR_READ_FIELDDATA_FILE
CDM, 13
- E_CDM_ERROR_READ_FILEINFO
CDM, 13
- E_CDM_ERROR_READ_FILEPATH
CDM, 13
- E_CDM_ERROR_READ_INDEXFILE_OPENERERROR
CDM, 13
- E_CDM_ERROR_READ_MPI
CDM, 13
- E_CDM_ERROR_READ_NETCDF_FUNC
CDM, 13
- E_CDM_ERROR_READ_NETCDF_MISMATCH_TYPE
CDM, 13
- E_CDM_ERROR_READ_NETCDF_VAR_1D
CDM, 13
- 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

- 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_UNMATCH_NUM_OF_VARIABLES
CDM, 13
- E_CDM_ERROR_UNMATCH_VOXELSIZE
CDM, 13
- E_CDM_ERROR_WRITE_DFI_NETCDF
CDM, 14
- 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, 14
- 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, 14
- E_CDM_ERRORCODE
CDM, 12
- E_CDM_FILE_TYPE
CDM, 14
- E_CDM_FILE_TYPE_ASCII
CDM, 14
- E_CDM_FILE_TYPE_BINARY
CDM, 14
- E_CDM_FILE_TYPE_DEFAULT
CDM, 14
- E_CDM_FILE_TYPE_FBINARY
CDM, 14
- E_CDM_FLOAT32
CDM, 11
- E_CDM_FLOAT64
CDM, 12
- E_CDM_FMT_AVS
CDM, 15
- E_CDM_FMT_BOV
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_DEFAULT
CDM, 15
- E_CDM_FNAME_RANK
CDM, 14

- CDM, [15](#)
- E_CDM_FNAME_RANK_STEP
 - CDM, [15](#)
- E_CDM_FNAME_STEP_RANK
 - CDM, [15](#)
- E_CDM_FORMAT
 - CDM, [14](#)
- 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, [15](#)
- E_CDM_READTYPE_UNKNOWN
 - CDM, [15](#)
- E_CDM_SAMEDIV_REFINEMENT
 - CDM, [15](#)
- E_CDM_SAMEDIV_SAMERES
 - CDM, [15](#)
- E_CDM_SUCCESS
 - CDM, [12](#)
- E_CDM_UINT16
 - CDM, [11](#)
- E_CDM_UINT32
 - CDM, [11](#)
- E_CDM_UINT64
 - CDM, [11](#)
- E_CDM_UINT8
 - CDM, [11](#)
- E_CDM_WARN_GETUNIT
 - CDM, [14](#)
- Endian
 - cdm_FileInfo, [117](#)
- endian
 - cdm_DFI_NETCDF::stVarInfo, [167](#)
- ExtractPathWithoutExt
 - CDM, [16](#)
- FieldFilenameFormat
 - cdm_FileInfo, [117](#)
- FileFormat
 - cdm_FileInfo, [118](#)
- Generate_DFI_Name
 - cdm_DFI, [42](#)
- Generate_Directory_Path
 - cdm_DFI, [42](#)
- Generate_FieldFileName
 - cdm_DFI, [42](#)
- Generate_FileName
 - cdm_DFI, [42](#)
- 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, [81](#)
- GetCoordinateFile
 - cdm_Domain, [107](#)
 - cdm_NonUniformDomain, [127](#)
- GetCoordinateFileEndian
 - cdm_Domain, [107](#)
 - cdm_NonUniformDomain, [128](#)
- GetCoordinateFilePrecision
 - cdm_Domain, [108](#)
 - cdm_NonUniformDomain, [128](#)
- GetCoordinateFileType
 - cdm_Domain, [108](#)
 - cdm_NonUniformDomain, [128](#)
- GetDFIGlobalDivision
 - cdm_DFI, [45](#)
- GetDFIGlobalVoxel
 - cdm_DFI, [46](#)
- GetDFIType
 - cdm_DFI, [46](#)
- GetDFITypeString
 - cdm_DFI, [46](#)
- getData
 - cdm_Array, [26](#)
 - cdm_TypeArray, [156](#)
- GetDataType
 - cdm_DFI, [45](#)
- getDataType
 - cdm_Array, [26](#)

- GetDataTypeString
 - cdm_DFI, [45](#)
- getDataTypeString
 - cdm_Array, [26](#)
- GetDimName
 - cdm_DFI_NETCDF, [81](#)
- GetFileFormat
 - cdm_DFI, [46](#)
- GetFileFormatString
 - cdm_DFI, [46](#)
- getGc
 - cdm_Array, [26](#)
- getGcInt
 - cdm_Array, [26](#)
- getHeadIndex
 - cdm_Array, [27](#)
- getMinMax
 - cdm_DFI, [46](#)
 - cdm_TimeSlice, [151](#)
- GetNcType
 - cdm_DFI_NETCDF, [82](#)
- GetNcUnit
 - cdm_DFI_NETCDF, [82](#)
- GetNodeStr
 - cdm_TextParser, [148](#)
- GetNumGuideCell
 - cdm_DFI, [47](#)
- GetNumVariables
 - cdm_DFI, [47](#)
- getNvari
 - cdm_Array, [27](#)
- getNvariInt
 - cdm_Array, [27](#)
- GetPos
 - cdm_ActiveSubDomain, [20](#)
- getTPPtr
 - cdm_TextParser, [148](#)
- getTPInstance
 - cdm_TextParser, [148](#)
- getTailIndex
 - cdm_Array, [27](#)
- GetUnit
 - cdm_DFI, [47](#)
 - cdm_Unit, [159](#)
- GetUnitElem
 - cdm_DFI, [47](#)
 - cdm_Unit, [159](#)
- GetValue
 - cdm_TextParser, [148](#)
- GetVarInfo
 - cdm_DFI_NETCDF, [82](#)
- getVariableName
 - cdm_DFI, [47](#)
 - cdm_FileInfo, [116](#)
- GetVector
 - cdm_TextParser, [149](#)
- getVectorMinMax
 - cdm_DFI, [48](#)
- cdm_TimeSlice, [151](#)
- getVersionInfo
 - cdm_DFI, [48](#)
- GetcdmDomain
 - cdm_DFI, [44](#)
- GetcdmFileInfo
 - cdm_DFI, [44](#)
- GetcdmFilePath
 - cdm_DFI, [44](#)
- GetcdmMPI
 - cdm_DFI, [44](#)
- GetcdmProcess
 - cdm_DFI, [44](#)
- GetcdmRankNoPrefix
 - cdm_DFI, [44](#)
- GetcdmTimeSlice
 - cdm_DFI, [45](#)
- GetcdmUnit
 - cdm_DFI, [45](#)
- GetcdmVisIt
 - cdm_DFI, [45](#)
- GlobalDivision
 - cdm_Domain, [109](#)
- GlobalOrigin
 - cdm_Domain, [109](#)
- GlobalRegion
 - cdm_Domain, [110](#)
- GlobalVoxel
 - cdm_Domain, [110](#)
- GuideCell
 - cdm_FileInfo, [118](#)
- HeadIndex
 - cdm_Rank, [142](#)
- headT
 - cdm_Process, [136](#)
- HostName
 - cdm_Rank, [142](#)
- hval
 - cdm_TypeArray, [157](#)
- id
 - cdm_DFI_NETCDF::stDimInfo, [165](#)
 - cdm_DFI_NETCDF::stVarInfo, [167](#)
- instanceArray
 - cdm_Array, [27–29](#)
- interp_coarse
 - cdm_Array, [29](#)
- isMatchEndianSbdmMagick
 - cdm_Process, [139](#)
- len
 - cdm_DFI_NETCDF::stDimInfo, [165](#)
- m_NC_Unit
 - cdm_DFI_NETCDF, [88](#)
- m_RankID
 - cdm_DFI, [68](#)
- m_Sz

- cdm_Array, 31
- m_SzI
 - cdm_Array, 31
- m_addMode
 - cdm_FILE, 113
- m_bgrid_interp_flag
 - cdm_DFI, 67
- m_comm
 - cdm_DFI, 67
- m_data
 - cdm_TypeArray, 158
- m_directoryPath
 - cdm_DFI, 68
- m_dtype
 - cdm_Array, 29
- m_fname
 - cdm_FILE, 113
- m_format
 - cdm_FILE, 113
- m_fp
 - cdm_FILE, 113
- m_gc
 - cdm_Array, 29
- m_gcl
 - cdm_Array, 30
- m_gcl
 - cdm_Array, 30
- m_headIndex
 - cdm_Array, 30
- m_indexDfiName
 - cdm_DFI, 68
- m_input_type
 - cdm_DFI, 68
- m_nameT
 - cdm_DFI_NETCDF, 88
- m_nameX
 - cdm_DFI_NETCDF, 88
- m_nameY
 - cdm_DFI_NETCDF, 88
- m_nameZ
 - cdm_DFI_NETCDF, 88
- m_ncid
 - cdm_FILE, 113
- m_nvari
 - cdm_Array, 30
- m_nvaril
 - cdm_Array, 30
- m_outptr
 - cdm_TypeArray, 158
- m_output_fname
 - cdm_DFI, 68
- m_output_type
 - cdm_DFI, 68
- m_output_type_coord
 - cdm_DFI, 68
- m_pos
 - cdm_ActiveSubDomain, 21
- m_rankMap
 - cdm_Process, 140
- m_read_type
 - cdm_DFI, 69
- m_readRankList
 - cdm_DFI, 69
- m_shape
 - cdm_Array, 30
- m_sz
 - cdm_Array, 30
- m_szI
 - cdm_Array, 31
- m_tailIndex
 - cdm_Array, 31
- m_varInfoT
 - cdm_DFI_NETCDF, 89
- m_varInfoX
 - cdm_DFI_NETCDF, 89
- m_varInfoY
 - cdm_DFI_NETCDF, 89
- m_varInfoZ
 - cdm_DFI_NETCDF, 89
- m_vecVarInfo
 - cdm_DFI_NETCDF, 89
- m_writeFlag
 - cdm_DFI_NETCDF, 89
- MAXPATHLEN
 - cdm_PathUtil.h, 189
- MPI_Allgather
 - mpi_stubs.h, 193
- MPI_CHAR
 - mpi_stubs.h, 193
- MPI_COMM_WORLD
 - mpi_stubs.h, 193
- MPI_Comm
 - mpi_stubs.h, 193
- MPI_Comm_rank
 - mpi_stubs.h, 193
- MPI_Comm_size
 - mpi_stubs.h, 194
- MPI_Datatype
 - mpi_stubs.h, 193
- MPI_Gather
 - mpi_stubs.h, 194
- MPI_INT
 - mpi_stubs.h, 193
- MPI_Init
 - mpi_stubs.h, 194
- MPI_SUCCESS
 - mpi_stubs.h, 193
- MakeDirectory
 - cdm_DFI, 48
- MakeDirectoryPath
 - cdm_DFI, 48
- MakeDirectorySub
 - cdm_DFI, 48
- Max
 - cdm_Slice, 145
- Min

- cdm_Slice, 145
- mpi_stubs.h, 192
 - MPI_Allgather, 193
 - MPI_CHAR, 193
 - MPI_COMM_WORLD, 193
 - MPI_Comm, 193
 - MPI_Comm_rank, 193
 - MPI_Comm_size, 194
 - MPI_Datatype, 193
 - MPI_Gather, 194
 - MPI_INT, 193
 - MPI_Init, 194
 - MPI_SUCCESS, 193
- Name
 - cdm_UnitElem, 163
- name
 - cdm_DFI_NETCDF::stDimInfo, 165
 - cdm_DFI_NETCDF::stVarInfo, 167
- NodeX
 - cdm_Domain, 108
 - cdm_NonUniformDomain, 128
- NodeY
 - cdm_Domain, 108
 - cdm_NonUniformDomain, 130
- NodeZ
 - cdm_Domain, 108
 - cdm_NonUniformDomain, 130
- normalizeBaseTime
 - cdm_DFI, 49
- normalizeDeltaT
 - cdm_DFI, 49
- normalizeIntervalTime
 - cdm_DFI, 49
- normalizeLastTime
 - cdm_DFI, 49
- normalizeStartTime
 - cdm_DFI, 49
- normalizeTime
 - cdm_DFI, 49
- NumVariables
 - cdm_FileInfo, 118
- NumberOfGroup
 - cdm_MPI, 123
- NumberOfRank
 - cdm_MPI, 123
- OpenReadBinary
 - cdm_FILE, 112
- OpenWriteAscii
 - cdm_FILE, 112
- OpenWriteBinary
 - cdm_FILE, 112
- operator==
 - cdm_ActiveSubDomain, 20
- pit_gcXend
 - cdm_NonUniformDomain, 133
- pit_gcXsta
 - cdm_NonUniformDomain, 133
- pit_gcYend
 - cdm_NonUniformDomain, 133
- pit_gcYsta
 - cdm_NonUniformDomain, 133
- pit_gcZend
 - cdm_NonUniformDomain, 134
- pit_gcZsta
 - cdm_NonUniformDomain, 134
- Pitch
 - cdm_Domain, 110
- PlotGC
 - cdm_VisIt, 165
- Prefix
 - cdm_FileInfo, 118
- ProcDFIFile
 - cdm_FilePath, 120
- RankID
 - cdm_Rank, 142
- RankList
 - cdm_Process, 140
- RankNoPrefix
 - cdm_FileInfo, 118
- Read
 - cdm_Domain, 109
 - cdm_FileInfo, 116
 - cdm_FilePath, 120
 - cdm_MPI, 121
 - cdm_NonUniformDomain, 130
 - cdm_Process, 139
 - cdm_Rank, 141
 - cdm_Slice, 143
 - cdm_TimeSlice, 151
 - cdm_Unit, 160
 - cdm_UnitElem, 161
 - cdm_VisIt, 164
- Read_CoordinateFile
 - cdm_NonUniformDomain, 131
- read_Datarecord
 - cdm_DFI, 50
 - cdm_DFI_AVS, 71
 - cdm_DFI_BOV, 75
 - cdm_DFI_NETCDF, 83
 - cdm_DFI_PLOT3D, 91
 - cdm_DFI_SPH, 98
 - cdm_DFI_VTK, 103
- read_Func
 - cdm_DFI_PLOT3D, 93
- read_HeaderRecord
 - cdm_DFI, 50
 - cdm_DFI_AVS, 71
 - cdm_DFI_BOV, 76
 - cdm_DFI_NETCDF, 83, 85
 - cdm_DFI_PLOT3D, 93
 - cdm_DFI_SPH, 98
 - cdm_DFI_VTK, 103
- read_averaged
 - cdm_DFI, 50

- cdm_DFI_AVIS, [70](#)
- cdm_DFI_BOV, [75](#)
- cdm_DFI_NETCDF, [82](#)
- cdm_DFI_PLOT3D, [91](#)
- cdm_DFI_SPH, [98](#)
- cdm_DFI_VTK, [102](#)
- ReadActiveSubdomainFile
 - cdm_Process, [139](#)
- ReadAdditionalTP
 - cdm_DFI_NETCDF, [85](#)
- readBinary
 - cdm_Array, [29](#)
 - cdm_TypeArray, [157](#)
- readCoordData
 - cdm_NonUniformDomain, [131](#)
- readCoordDataCount
 - cdm_NonUniformDomain, [132](#)
- ReadData
 - cdm_DFI, [51](#), [53](#)
- ReadFieldData
 - cdm_DFI, [53](#)
- ReadInit
 - cdm_DFI, [54](#)
- readTPfile
 - cdm_TextParser, [149](#)
- RealType
 - cdm_DFI_SPH, [97](#)
- reference
 - cdm_UnitElem, [163](#)
- remove
 - cdm_TextParser, [149](#)
- SBSWAPVEC
 - cdm_endianUtil.h, [185](#)
- set_RankID
 - cdm_DFI, [55](#)
- set_input_type
 - cdm_DFI, [54](#)
- set_interp_flag
 - cdm_DFI, [54](#)
- set_output_fname
 - cdm_DFI, [55](#)
- set_output_type
 - cdm_DFI, [55](#)
- set_output_type_coord
 - cdm_DFI, [55](#)
- SetDimName
 - cdm_DFI_NETCDF, [85](#)
- setGridData
 - cdm_DFI, [56](#), [57](#)
- setHeadIndex
 - cdm_Array, [29](#)
- setIntervalStep
 - cdm_DFI, [57](#)
- setIntervalTime
 - cdm_DFI, [57](#)
- SetPos
 - cdm_ActiveSubDomain, [20](#)
- SetTimeSliceFlag
 - cdm_DFI, [57](#)
- setVariableName
 - cdm_DFI, [57](#)
 - cdm_FileInfo, [116](#)
- SetWriteFlag
 - cdm_DFI_NETCDF, [86](#)
- SetcdmDomain
 - cdm_DFI, [55](#)
- SetcdmFilePath
 - cdm_DFI, [55](#)
- SetcdmMPI
 - cdm_DFI, [56](#)
- SetcdmProcess
 - cdm_DFI, [56](#)
- SetcdmRankNoPrefix
 - cdm_DFI, [56](#)
- SetcdmTimeSlice
 - cdm_DFI, [56](#)
- SetcdmUnit
 - cdm_DFI, [56](#)
- SetcdmVisIt
 - cdm_DFI, [56](#)
- SliceList
 - cdm_TimeSlice, [153](#)
- stVarInfo
 - cdm_DFI_NETCDF::stVarInfo, [166](#)
- step
 - cdm_Slice, [145](#)
- stmpd_printf
 - cdm_Define.h, [178](#)
- TailIndex
 - cdm_Rank, [142](#)
- time
 - cdm_Slice, [145](#)
- TimeSliceDirFlag
 - cdm_FileInfo, [118](#)
- tp
 - cdm_TextParser, [150](#)
- type
 - cdm_DFI_NETCDF::stVarInfo, [167](#)
- Unit
 - cdm_UnitElem, [163](#)
- unit
 - cdm_DFI_NETCDF::stVarInfo, [168](#)
- UnitList
 - cdm_Unit, [160](#)
- val
 - cdm_TypeArray, [157](#)
- VariableName
 - cdm_FileInfo, [118](#)
- VectorMax
 - cdm_Slice, [145](#)
- VectorMin
 - cdm_Slice, [146](#)
- vfvPath_emitDrive
 - CDM, [16](#)

- VolumeDataDivide
 - cdm_DFI, 59
- VoxelSize
 - cdm_Rank, 142
- Write
 - cdm_Domain, 109
 - cdm_FileInfo, 117
 - cdm_FilePath, 120
 - cdm_MPI, 121
 - cdm_NonUniformDomain, 132
 - cdm_Process, 139
 - cdm_Rank, 141
 - cdm_Slice, 143
 - cdm_TimeSlice, 153
 - cdm_Unit, 160
 - cdm_UnitElem, 161
 - cdm_VisIt, 164
- write_AttUnits
 - cdm_DFI_NETCDF, 86
- write_DataRecord
 - cdm_DFI, 59
 - cdm_DFI_AVS, 73
 - cdm_DFI_BOV, 77
 - cdm_DFI_NETCDF, 86, 87
 - cdm_DFI_PLOT3D, 94
 - cdm_DFI_SPH, 100
 - cdm_DFI_VTK, 104
- write_Func
 - cdm_DFI_PLOT3D, 94, 95
- write_GridData
 - cdm_DFI, 60
 - cdm_DFI_PLOT3D, 95
- write_HeaderRecord
 - cdm_DFI, 60
 - cdm_DFI_AVS, 73
 - cdm_DFI_BOV, 77
 - cdm_DFI_NETCDF, 87
 - cdm_DFI_PLOT3D, 95
 - cdm_DFI_SPH, 100
 - cdm_DFI_VTK, 104
- write_XYZ
 - cdm_DFI_PLOT3D, 95
- write_ascii_header
 - cdm_DFI, 59
 - cdm_DFI_AVS, 72
 - cdm_DFI_BOV, 76
- write_averaged
 - cdm_DFI, 59
 - cdm_DFI_AVS, 72
 - cdm_DFI_BOV, 76
 - cdm_DFI_NETCDF, 86
 - cdm_DFI_PLOT3D, 94
 - cdm_DFI_SPH, 100
 - cdm_DFI_VTK, 103
- write_avs_cord
 - cdm_DFI_AVS, 72
- write_avs_header
 - cdm_DFI_AVS, 73
- WriteAdditionalTP
 - cdm_DFI_NETCDF, 88
- writeAscii
 - cdm_Array, 29
 - cdm_TypeArray, 157
- writeBinary
 - cdm_Array, 29
 - cdm_TypeArray, 157
- WriteData
 - cdm_DFI, 60, 61
- WriteFieldData
 - cdm_DFI, 61
- WriteFieldDataFile
 - cdm_DFI, 62
- WriteGridFile
 - cdm_DFI, 63
- WriteIndexDfiFile
 - cdm_DFI, 63
- WriteInit
 - cdm_DFI, 63–65
- WriteProcDfiFile
 - cdm_DFI, 66
- XCoordinates
 - cdm_NonUniformDomain, 134
- YCoordinates
 - cdm_NonUniformDomain, 134
- ZCoordinates
 - cdm_NonUniformDomain, 134