

Parallel File Compression Library
1.0.0

作成 : Doxygen 1.8.4

Thu Feb 13 2014 10:33:18

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 PFC	9
5.1.1	説明	10
5.1.2	列挙型	10
5.1.2.1	E_PFC_ARRAYSHAPE	10
5.1.2.2	E_PFC_COMPRESS_FORMAT	11
5.1.2.3	E_PFC_DTYPE	11
5.1.2.4	E_PFC_ENDIANTYPE	11
5.1.2.5	E_PFC_ERRORCODE	12
5.1.3	関数	14
5.1.3.1	MakeDirectory	14
5.1.3.2	PfcPath_ConnectPath	14
5.1.3.3	PfcPath_DirName	15
5.1.3.4	PfcPath_FileName	16
5.1.3.5	PfcPath_getDelimChar	16
5.1.3.6	PfcPath_getDelimString	16
5.1.3.7	PfcPath_hasDrive	17
5.1.3.8	PfcPath_isAbsolute	17
5.1.3.9	vfvPath_emitDrive	17
6	クラス	19

6.1	クラス CPfcCioDfiUtil	19
6.1.1	説明	19
6.1.2	関数	19
6.1.2.1	GetDfiDomain	19
6.1.2.2	GetDfiInfo	20
6.1.2.3	GetDfiTimeStepList	20
6.2	クラス CPfcCoeffTree	20
6.2.1	説明	21
6.2.2	コンストラクタとデストラクタ	21
6.2.2.1	CPfcCoeffTree	21
6.2.2.2	~CPfcCoeffTree	21
6.2.3	関数	21
6.2.3.1	construct_array	21
6.2.3.2	construct_child	21
6.2.3.3	construct_tree	21
6.2.3.4	deep_transerval	21
6.2.3.5	get_position	21
6.2.3.6	print_node	21
6.2.3.7	print_tree	21
6.2.3.8	read_data	21
6.2.3.9	standard_timestep_node	22
6.2.4	変数	22
6.2.4.1	data	22
6.2.4.2	data_size	22
6.2.4.3	m_numLayer	22
6.2.4.4	m_numTimeStep	22
6.2.4.5	position	22
6.2.4.6	position_interval	22
6.2.4.7	root	22
6.2.4.8	stand_space	22
6.3	クラス CPfcCoeffTreeNode	22
6.3.1	説明	23
6.3.2	コンストラクタとデストラクタ	23
6.3.2.1	CPfcCoeffTreeNode	23
6.3.2.2	CPfcCoeffTreeNode	23
6.3.2.3	~CPfcCoeffTreeNode	23
6.3.3	変数	23
6.3.3.1	is_leaf	23
6.3.3.2	left_child	23
6.3.3.3	right_child	23

6.3.3.4	value	23
6.4	クラス CPfcCompress	24
6.4.1	説明	25
6.4.2	コンストラクタとデストラクタ	25
6.4.2.1	CPfcCompress	25
6.4.2.2	~CPfcCompress	25
6.4.3	関数	26
6.4.3.1	GatherAndCreateDivisionInfo	26
6.4.3.2	Init	27
6.4.3.3	Init	27
6.4.3.4	WriteData	27
6.4.3.5	WriteIndexPfcFile	28
6.4.3.6	WriteProcFile	28
6.4.4	変数	28
6.4.4.1	m_arrayShape	28
6.4.4.2	m_comm	28
6.4.4.3	m_compressError	28
6.4.4.4	m_compressFormat	28
6.4.4.5	m_dataType	28
6.4.4.6	m_dfiFilePath	28
6.4.4.7	m_domainDivision	29
6.4.4.8	m_endianType	29
6.4.4.9	m_endStep	29
6.4.4.10	m_fileFormat	29
6.4.4.11	m_myIDinRegion	29
6.4.4.12	m_myRankID	29
6.4.4.13	m_numComponent	29
6.4.4.14	m_numParallel	29
6.4.4.15	m_numRank	29
6.4.4.16	m_numRegion	29
6.4.4.17	m_numStep	29
6.4.4.18	m_optFlags	29
6.4.4.19	m_outDirPath	30
6.4.4.20	m_pCioPod	30
6.4.4.21	m_pDfilN	30
6.4.4.22	m_prefix	30
6.4.4.23	m_regionHead	30
6.4.4.24	m_regionID	30
6.4.4.25	m_regionMasterRankID	30
6.4.4.26	m_regionSize	30

6.4.4.27	m_regionTail	30
6.4.4.28	m_startStep	30
6.4.4.29	m_stepList	30
6.5	クラス CPfcCompressCioPod	31
6.5.1	説明	32
6.5.2	コンストラクタとデストラクタ	32
6.5.2.1	CPfcCompressCioPod	32
6.5.2.2	~CPfcCompressCioPod	32
6.5.3	関数	32
6.5.3.1	Init	32
6.5.3.2	ReadCioFile	32
6.5.3.3	ReadCioFile_IJKN	33
6.5.3.4	WriteData	33
6.5.4	変数	33
6.5.4.1	m_comm	33
6.5.4.2	m_compressError	33
6.5.4.3	m_myRankID	33
6.5.4.4	m_numRank	34
6.5.4.5	m_numRegion	34
6.5.4.6	m_numStep	34
6.5.4.7	m_optFlags	34
6.5.4.8	m_outDirPath	34
6.5.4.9	m_pDfilN	34
6.5.4.10	m_pFlowData	34
6.5.4.11	m_pPod	34
6.5.4.12	m_pStepList	34
6.5.4.13	m_regionHead	34
6.5.4.14	m_regionTail	34
6.6	クラス CPfcCompressCmd	35
6.6.1	説明	35
6.6.2	コンストラクタとデストラクタ	35
6.6.2.1	CPfcCompressCmd	35
6.6.2.2	~CPfcCompressCmd	35
6.6.3	関数	35
6.6.3.1	Execute	35
6.6.3.2	ReadCompressCntl	36
6.6.4	変数	36
6.6.4.1	m_cmdList	36
6.6.4.2	m_compressCntlPath	36
6.6.4.3	m_domainDivision	36

6.7	クラス CPfcCompressCmdElm	36
6.7.1	説明	37
6.7.2	コンストラクタとデストラクタ	37
6.7.2.1	CPfcCompressCmdElm	37
6.7.2.2	~CPfcCompressCmdElm	37
6.7.3	関数	37
6.7.3.1	ReadElm	37
6.7.4	変数	37
6.7.4.1	m_compressError	37
6.7.4.2	m_compressFormat	37
6.7.4.3	m_dfiFilePath	37
6.7.4.4	m_endStep	38
6.7.4.5	m_optFlags	38
6.7.4.6	m_optSave	38
6.7.4.7	m_outDirPath	38
6.7.4.8	m_procFileSave	38
6.7.4.9	m_startStep	38
6.8	クラス CPfcCompressInfo	38
6.8.1	説明	39
6.8.2	コンストラクタとデストラクタ	39
6.8.2.1	CPfcCompressInfo	39
6.8.2.2	CPfcCompressInfo	39
6.8.2.3	~CPfcCompressInfo	39
6.8.3	関数	39
6.8.3.1	Read	39
6.8.3.2	Write	40
6.8.4	変数	41
6.8.4.1	m_calculatedLayer	41
6.8.4.2	m_compressError	41
6.8.4.3	m_compressFormat	41
6.8.4.4	m_endStep	41
6.8.4.5	m_startStep	41
6.8.4.6	m_version	41
6.9	クラス CPfcCompressPod	41
6.9.1	説明	43
6.9.2	コンストラクタとデストラクタ	43
6.9.2.1	CPfcCompressPod	43
6.9.2.2	~CPfcCompressPod	43
6.9.3	関数	43
6.9.3.1	CalcPodBase	43

6.9.3.2	CalcPodCoef	43
6.9.3.3	CheckFinish	44
6.9.3.4	GetCalculatedLayer	45
6.9.3.5	Init	45
6.9.3.6	Output	45
6.9.3.7	SwapBinaryData	46
6.9.3.8	WriteData	47
6.9.3.9	WritePodBaseFile	47
6.9.3.10	WritePodBaseFile_Debug	47
6.9.3.11	WritePodCoefFile	47
6.9.4	変数	47
6.9.4.1	m_calculatedLayer	47
6.9.4.2	m_comm	47
6.9.4.3	m_compressError	47
6.9.4.4	m_curEvaluateError	47
6.9.4.5	m_curNumSize	47
6.9.4.6	m_curPodBaseSize	47
6.9.4.7	m_layerNo	48
6.9.4.8	m_maxLayer	48
6.9.4.9	m_myIDinRegion	48
6.9.4.10	m_myNumStep	48
6.9.4.11	m_myRankID	48
6.9.4.12	m_myStartStepPos	48
6.9.4.13	m_numParallel	48
6.9.4.14	m_numRank	48
6.9.4.15	m_numRegion	48
6.9.4.16	m_numStep	48
6.9.4.17	m_optFlags	48
6.9.4.18	m_outDirPath	48
6.9.4.19	m_pCoef_a	49
6.9.4.20	m_pFlowData	49
6.9.4.21	m_pPod_base_r	49
6.9.4.22	m_prefix	49
6.9.4.23	m_regionID	49
6.9.4.24	m_regionMasterRankID	49
6.9.4.25	m_regionMaxStep	49
6.9.4.26	m_regionSize	49
6.10	クラス CPfcDivision	49
6.10.1	説明	50
6.10.2	コンストラクタとデストラクタ	50

6.10.2.1	CPfcDivision	50
6.10.2.2	~CPfcDivision	50
6.10.3	関数	50
6.10.3.1	CheckReadRegion	50
6.10.3.2	CheckReadRegion	50
6.10.3.3	CreateMinMax	51
6.10.3.4	Read	51
6.10.3.5	Write	51
6.10.4	変数	51
6.10.4.1	m_MinMax	51
6.10.4.2	m_MinMax_data	51
6.10.4.3	m_numRegion	52
6.10.4.4	m_regionList	52
6.11	クラス CPfcDomain	52
6.11.1	説明	52
6.11.2	コンストラクタとデストラクタ	52
6.11.2.1	CPfcDomain	52
6.11.2.2	CPfcDomain	53
6.11.2.3	~CPfcDomain	54
6.11.3	関数	54
6.11.3.1	Read	54
6.11.3.2	Write	54
6.11.4	変数	54
6.11.4.1	m_globalDivision	54
6.11.4.2	m_globalOrigin	54
6.11.4.3	m_globalRegion	55
6.11.4.4	m_globalVoxel	55
6.12	クラス CPfcFileInfo	55
6.12.1	説明	55
6.12.2	コンストラクタとデストラクタ	56
6.12.2.1	CPfcFileInfo	56
6.12.2.2	CPfcFileInfo	56
6.12.2.3	~CPfcFileInfo	56
6.12.3	関数	56
6.12.3.1	Read	56
6.12.3.2	Write	56
6.12.4	変数	57
6.12.4.1	m_arrayShape	57
6.12.4.2	m_dataType	57
6.12.4.3	m_dirPath	57

6.12.4.4	m_endianType	57
6.12.4.5	m_fileFormat	57
6.12.4.6	m_guideCell	57
6.12.4.7	m_numComponent	57
6.12.4.8	m_prefix	57
6.13	クラス CPfcFilePath	57
6.13.1	説明	58
6.13.2	コンストラクタとデストラクタ	58
6.13.2.1	CPfcFilePath	58
6.13.2.2	CPfcFilePath	58
6.13.2.3	~CPfcFilePath	58
6.13.3	関数	58
6.13.3.1	Read	58
6.13.3.2	Write	59
6.13.4	変数	59
6.13.4.1	m_dfiPath	59
6.13.4.2	m_procPath	59
6.14	クラス CPfcFunction	59
6.14.1	説明	60
6.14.2	関数	60
6.14.2.1	Alloc2D	60
6.14.2.2	Alloc2D	60
6.14.2.3	AndMinMax	61
6.14.2.4	CalcHeadTail	61
6.14.2.5	CalcHeadTail_block	61
6.14.2.6	CheckLapMinMax	62
6.14.2.7	CheckPntInMinMax	62
6.14.2.8	CopyData	62
6.14.2.9	GetPodBaseIndex	63
6.14.2.10	GetPodMaxLayer	63
6.14.2.11	GetPodParallel	63
6.14.2.12	GetPodRegionID	64
6.14.2.13	GetPodStepInfo	64
6.15	クラス CPfcMpiCom	64
6.15.1	説明	65
6.15.2	関数	65
6.15.2.1	GatherDataDouble	65
6.15.2.2	GatherDataInt	65
6.15.2.3	GatherV_DataDouble	66
6.15.2.4	GetMaxInt	66

6.16 クラス CPfcPodFile	66
6.16.1 説明	67
6.16.2 関数	67
6.16.2.1 CloseBaseFile	67
6.16.2.2 OpenBaseFile	68
6.16.2.3 ReadBaseFile	68
6.16.2.4 ReadBaseFile1Elm	68
6.16.2.5 ReadBaseFile_Debug	69
6.16.2.6 ReadBaseFileHeader	69
6.16.2.7 ReadCoefFile	70
6.16.2.8 WriteBaseFile	70
6.16.2.9 WriteBaseFile_Debug	71
6.16.2.10 WriteCoefFile	71
6.17 クラス CPfcRegion	72
6.17.1 説明	72
6.17.2 コンストラクタとデストラクタ	72
6.17.2.1 CPfcRegion	72
6.17.2.2 ~CPfcRegion	72
6.17.3 関数	72
6.17.3.1 Read	72
6.17.3.2 Write	73
6.17.4 変数	73
6.17.4.1 m_headIndex	73
6.17.4.2 m_regionID	73
6.17.4.3 m_tailIndex	73
6.17.4.4 m_voxelSize	73
6.18 クラス CPfcRestration	73
6.18.1 説明	75
6.18.2 コンストラクタとデストラクタ	75
6.18.2.1 CPfcRestration	75
6.18.2.2 ~CPfcRestration	75
6.18.3 関数	75
6.18.3.1 CheckCompressDataOnMem	75
6.18.3.2 DeleteCompressDataOnMem	75
6.18.3.3 GetGlobalVoxel	76
6.18.3.4 GetHeadTail	77
6.18.3.5 GetTimeStepList	77
6.18.3.6 Init	77
6.18.3.7 LoadCompressDataOnMem	77
6.18.3.8 ReadData	78

6.18.3.9	ReadData	78
6.18.4	変数	78
6.18.4.1	m_bLoadCompressData	78
6.18.4.2	m_compressInfo	79
6.18.4.3	m_division	79
6.18.4.4	m_domain	79
6.18.4.5	m_fileInfo	79
6.18.4.6	m_filePath	79
6.18.4.7	m_head	79
6.18.4.8	m_numRegion	79
6.18.4.9	m_numTimeStep	79
6.18.4.10	m_pfcFilePath	79
6.18.4.11	m_regionIdList	79
6.18.4.12	m_regionList	79
6.18.4.13	m_tail	79
6.18.4.14	m_timeSlice	80
6.18.4.15	m_timeStepList	80
6.18.4.16	m_unitList	80
6.19	クラス CPfcRestrationRegion	80
6.19.1	説明	81
6.19.2	コンストラクタとデストラクタ	81
6.19.2.1	CPfcRestrationRegion	81
6.19.2.2	~CPfcRestrationRegion	81
6.19.3	関数	81
6.19.3.1	DeleteCompressDataOnMem	81
6.19.3.2	Init	81
6.19.3.3	LoadCompressDataOnMem	81
6.19.3.4	ReadData	82
6.19.3.5	ReadDataInRange	82
6.19.4	変数	82
6.19.4.1	m_bLoadCompressData	82
6.19.4.2	m_compressForm	82
6.19.4.3	m_pCompressInfo	83
6.19.4.4	m_pDomain	83
6.19.4.5	m_pFileInfo	83
6.19.4.6	m_pRegion	83
6.19.4.7	m_pvTimeStepList	83
6.19.4.8	pPod	83
6.20	クラス CPfcRestrationRegionPod	83
6.20.1	説明	84

6.20.2	コンストラクタとデストラクタ	84
6.20.2.1	CPfcRestrictionRegionPod	84
6.20.2.2	~CPfcRestrictionRegionPod	84
6.20.3	関数	84
6.20.3.1	DeleteCompressDataOnMem	84
6.20.3.2	ExpandData	85
6.20.3.3	ExpandData	85
6.20.3.4	ExpandDataOnFile	85
6.20.3.5	GetExpandRowInfo	85
6.20.3.6	Init	86
6.20.3.7	LoadCompressDataOnMem	86
6.20.3.8	ReadData	86
6.20.3.9	ReadFieldData	87
6.20.4	変数	87
6.20.4.1	m_arrayShape	87
6.20.4.2	m_bLoadCompressData	87
6.20.4.3	m_bSingle	87
6.20.4.4	m_dirPath	87
6.20.4.5	m_numCalculatedLayer	87
6.20.4.6	m_numCoef	87
6.20.4.7	m_numComponent	88
6.20.4.8	m_numParallel	88
6.20.4.9	m_numSize	88
6.20.4.10	m_numStep	88
6.20.4.11	m_pBaseData	88
6.20.4.12	m_pBaseSizes	88
6.20.4.13	m_pCoefData	88
6.20.4.14	m_pIndexBase	88
6.20.4.15	m_pIndexCoef	88
6.20.4.16	m_prefix	88
6.20.4.17	m_regionID	88
6.21	クラス CPfcTextParser	89
6.21.1	説明	89
6.21.2	コンストラクタとデストラクタ	89
6.21.2.1	CPfcTextParser	89
6.21.2.2	~CPfcTextParser	90
6.21.3	関数	90
6.21.3.1	chkLabel	90
6.21.3.2	chkNode	90
6.21.3.3	countLabels	90

6.21.3.4	GetNodeStr	90
6.21.3.5	getTPInstance	90
6.21.3.6	GetValue	91
6.21.3.7	GetValue	91
6.21.3.8	GetValue	91
6.21.3.9	GetVector	91
6.21.3.10	GetVector	91
6.21.3.11	GetVector	91
6.21.3.12	readTPfile	92
6.21.3.13	remove	92
6.21.4	変数	92
6.21.4.1	tp	92
6.22	クラス CPfcTimeSlice	92
6.22.1	説明	93
6.22.2	コンストラクタとデストラクタ	93
6.22.2.1	CPfcTimeSlice	93
6.22.2.2	CPfcTimeSlice	93
6.22.2.3	~CPfcTimeSlice	93
6.22.3	関数	93
6.22.3.1	Select	93
6.23	クラス CPfcUnitList	94
6.23.1	説明	94
6.23.2	コンストラクタとデストラクタ	95
6.23.2.1	CPfcUnitList	95
6.23.2.2	~CPfcUnitList	95
7	ファイル	97
7.1	mpi_stubs.h	97
7.1.1	マクロ定義	97
7.1.1.1	MPI_CHAR	97
7.1.1.2	MPI_COMM_WORLD	97
7.1.1.3	MPI_INT	97
7.1.1.4	MPI_SUCCESS	98
7.1.2	型定義	98
7.1.2.1	MPI_Comm	98
7.1.2.2	MPI_Datatype	98
7.1.3	関数	98
7.1.3.1	MPI_Allgather	98
7.1.3.2	MPI_Comm_rank	98
7.1.3.3	MPI_Comm_size	98

7.1.3.4	MPI_Gather	98
7.1.3.5	MPI_Init	99
7.2	PfcCioDfiUtil.h	99
7.3	PfcCoeffTree.h	99
7.3.1	説明	100
7.4	PfcCompress.h	100
7.4.1	説明	100
7.5	PfcCompressCioPod.h	101
7.5.1	説明	101
7.6	PfcCompressCmd.h	102
7.6.1	説明	102
7.7	PfcCompressCmdElm.h	102
7.7.1	説明	103
7.7.2	マクロ定義	103
7.7.2.1	_PFC_COMPRESS_CMD_ELM_CNTL_	103
7.8	PfcCompressInfo.h	104
7.8.1	説明	104
7.9	PfcCompressPod.h	104
7.9.1	説明	105
7.10	PfcDefine.h	105
7.10.1	説明	108
7.10.2	マクロ定義	108
7.10.2.1	_PFC_IDX_IJ	108
7.10.2.2	_PFC_IDX_IJK	108
7.10.2.3	_PFC_IDX_IJKN	109
7.10.2.4	_PFC_IDX_NIJ	109
7.10.2.5	_PFC_IDX_NIJK	110
7.10.2.6	_PFC_TAB_STR	110
7.10.2.7	_PFC_WRITE_TAB	110
7.10.2.8	D_PFC_BIG	111
7.10.2.9	D_PFC_COMPRESS_ERROR_DEFAULT	111
7.10.2.10	D_PFC_EPSILON	111
7.10.2.11	D_PFC_FLOAT32	111
7.10.2.12	D_PFC_FLOAT64	111
7.10.2.13	D_PFC_IJNK	111
7.10.2.14	D_PFC_LITTLE	111
7.10.2.15	D_PFC_NIJK	111
7.10.2.16	D_PFC_OPT_FLAG_SAVE	111
7.10.2.17	DEBUG_PRINT	111
7.10.2.18	DEBUG_PRINT_AT	112

7.10.2.19 PFC_CHECK_OPT_SAVE	112
7.10.2.20 PFC_PRINT	112
7.10.2.21 PFC_PRINT_AT	112
7.11 PfcDivision.h	112
7.11.1 説明	113
7.12 PfcDomain.h	113
7.12.1 説明	114
7.13 PfcEndianChange.h	114
7.13.1 関数	115
7.13.1.1 endswap	115
7.14 PfcFileInfo.h	115
7.14.1 説明	116
7.15 PfcFilePath.h	116
7.15.1 説明	116
7.16 PfcFunction.h	117
7.16.1 関数	117
7.16.1.1 int_pow	118
7.17 PfcMatrix.h	118
7.17.1 関数	118
7.17.1.1 dgeev_	118
7.17.1.2 dgemm_	119
7.17.1.3 dgemv_	119
7.17.1.4 PFC_d_sort	119
7.17.1.5 PFC_ddot	119
7.17.1.6 PFC_dgeev	119
7.17.1.7 PFC_dgeev	119
7.17.1.8 PFC_dgemm	119
7.17.1.9 PFC_dgemv	120
7.17.1.10 PFC_matrix_Trans_d	120
7.18 PfcMpiCom.h	120
7.18.1 説明	121
7.19 PfcPathUtil.h	121
7.19.1 説明	122
7.19.2 マクロ定義	122
7.19.2.1 MAXPATHLEN	122
7.20 PfcPodFile.h	122
7.20.1 説明	123
7.21 PfcRegion.h	123
7.21.1 説明	124
7.22 PfcRestriction.h	124

7.22.1 説明	125
7.23 PfcRestrictionRegion.h	125
7.23.1 説明	126
7.24 PfcRestrictionRegionPod.h	126
7.24.1 説明	127
7.25 PfcTextParser.h	127
7.25.1 説明	128
7.26 PfcTimeSlice.h	128
7.26.1 説明	129
7.27 PfcUnitList.h	129
7.27.1 説明	130
7.28 PfcVersion.h	130
7.28.1 マクロ定義	130
7.28.1.1 PFC_REVISION	130
7.28.1.2 PFC_VERSION_NO	130
7.28.1.3 POD_COMPRESS_VERSION	130
索引	131

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。

PFC	9
-----------	---

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。

cio_TimeSlice	
CPfcTimeSlice	92
cio_Unit	
CPfcUnitList	94
CPfcCioDfiUtil	19
CPfcCoeffTree	20
CPfcCoeffTreeNode	22
CPfcCompress	24
CPfcCompressCioPod	31
CPfcCompressCmd	35
CPfcCompressCmdElm	36
CPfcCompressInfo	38
CPfcCompressPod	41
CPfcDivision	49
CPfcDomain	52
CPfcFileInfo	55
CPfcFilePath	57
CPfcFunction	59
CPfcMpiCom	64
CPfcPodFile	66
CPfcRegion	72
CPfcRestriction	73
CPfcRestrictionRegion	80
CPfcRestrictionRegionPod	83
CPfcTextParser	89

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。

CPfcCioDfiUtil	19
CPfcCoeffTree	20
CPfcCoeffTreeNode	22
CPfcCompress	24
CPfcCompressCioPod	31
CPfcCompressCmd	35
CPfcCompressCmdElm	36
CPfcCompressInfo	38
CPfcCompressPod	41
CPfcDivision	49
CPfcDomain	52
CPfcFileInfo	55
CPfcFilePath	57
CPfcFunction	59
CPfcMpiCom	64
CPfcPodFile	66
CPfcRegion	72
CPfcRestration	73
CPfcRestrationRegion	80
CPfcRestrationRegionPod	83
CPfcTextParser	89
CPfcTimeSlice	92
CPfcUnitList	94

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。

mpi_stubs.h	97
PfcCioDfiUtil.h	99
PfcCoeffTree.h	
CPfcCoeffTree Class Header	99
PfcCompress.h	
CPfcCompress Class Header	100
PfcCompressCioPod.h	
CPfcCompressCioPod Class Header	101
PfcCompressCmd.h	
CPfcCompressCmd Class Header	102
PfcCompressCmdElm.h	
CPfcCompressCmdElm Class Header	102
PfcCompressInfo.h	
CPfcCompressInfo Class Header	104
PfcCompressPod.h	
CPfcCompressPod Class Header	104
PfcDefine.h	
PFC の定義マクロ記述ヘッダーファイル	105
PfcDivision.h	
PfcDivision Class Header	112
PfcDomain.h	
PfcDomain Class Header	113
PfcEndianChange.h	114
PfcFileInfo.h	
CPfcFileInfo Class Header	115
PfcFilePath.h	
CPfcFilePath Class Header	116
PfcFunction.h	117
PfcMatrix.h	118
PfcMpiCom.h	
CPfcMpiCom MPI communication Class Header	120
PfcPathUtil.h	
Path utility 関数	121
PfcPodFile.h	
CPfcPodFile Class Header 基底ファイル、係数ファイル用	122
PfcRegion.h	
PfcRegion Class Header	123

PfcRestration.h	
CPfcRestration Class Header	124
PfcRestrationRegion.h	
CPfcRestrationRegion Class Header	125
PfcRestrationRegionPod.h	
CPfcRestrationRegionPod Class Header	126
PfcTextParser.h	
TextParser Control class Header	127
PfcTimeSlice.h	
CPfcTimeSlice Class Header	128
PfcUnitList.h	
CPfcPfcUnitList Class Header	129
PfcVersion.h	130

Chapter 5

ネームスペース

5.1 ネームスペース PFC

列挙型

- enum `E_PFC_COMPRESS_FORMAT` { `E_PFC_COMPRESS_FMT_UNKNOWN` = -1, `E_PFC_COMPRESS_FMT_POD` = 1 }
- enum `E_PFC_DTYPE` {
 `E_PFC_DTYPE_UNKNOWN` = 0, `E_PFC_INT8`, `E_PFC_INT16`, `E_PFC_INT32`,
 `E_PFC_INT64`, `E_PFC_UINT8`, `E_PFC_UINT16`, `E_PFC_UINT32`,
 `E_PFC_UINT64`, `E_PFC_FLOAT32`, `E_PFC_FLOAT64` }
- enum `E_PFC_ARRAYSHAPE` { `E_PFC_ARRAYSHAPE_UNKNOWN` = -1, `E_PFC_IJKN` = 0, `E_PFC_NIJK` }
- enum `E_PFC_ENDIANTYPE` { `E_PFC_ENDIANTYPE_UNKNOWN` = -1, `E_PFC_LITTLE` = 0, `E_PFC_BIG` }
- enum `E_PFC_ERRORCODE` {
 `E_PFC_SUCCESS` = 1, `E_PFC_ERROR` = -1, `E_PFC_ERROR_READ_CNTLFILE_OPENERERROR` = 500,
 `E_PFC_ERROR_READ_CNTL_DOMAINDIVISION` = 501,
 `E_PFC_ERROR_READ_CNTL_NO_ITEM` = 510, `E_PFC_ERROR_READ_CNTL_ITEMCNTL` = 511,
 `E_PFC_ERROR_READ_CNTL_DFI_PATH` = 512, `E_PFC_ERROR_READ_CNTL_OUTDIR_PATH` = 513,
 `E_PFC_ERROR_READ_CNTL_COMPRESS_FMT` = 514, `E_PFC_ERROR_READ_CNTL_PROCFILE_SAVE`
 = 515, `E_PFC_ERROR_READ_INDEXFILE_OPENERERROR` = 1000, `E_PFC_ERROR_READ_FILEINFO` =
 1010,
 `E_PFC_ERROR_READ_PFC_DIRECTORYPATH` = 1011, `E_PFC_ERROR_READ_PFC_PREFIX` = 1012,
 `E_PFC_ERROR_READ_PFC_FILEFORMAT` = 1013, `E_PFC_ERROR_READ_PFC_GUIDECCELL` = 1014,
 `E_PFC_ERROR_READ_PFC_DATATYPE` = 1015, `E_PFC_ERROR_READ_PFC_ENDIAN` = 1016,
 `E_PFC_ERROR_READ_PFC_ARRAYSHAPE` = 1017, `E_PFC_ERROR_READ_PFC_COMPONENT` =
 1018,
 `E_PFC_ERROR_READ_COMPRESSINFO` = 1030, `E_PFC_ERROR_READ_PFC_COMPRESSFORMAT` =
 1031, `E_PFC_ERROR_READ_PFC_COMPRESSERROR` = 1032, `E_PFC_ERROR_READ_PFC_CALCULATEDLAYER`
 = 1033,
 `E_PFC_ERROR_READ_PFC_VERSION` = 1034, `E_PFC_ERROR_READ_PFC_STARTSTEP` = 1035,
 `E_PFC_ERROR_READ_PFC_ENDSTEP` = 1036, `E_PFC_ERROR_READ_FILEPATH` = 1040,
 `E_PFC_ERROR_READ_PFC_DFIPATH` = 1041, `E_PFC_ERROR_READ_PFC_PFCPROCESS` = 1042,
 `E_PFC_ERROR_READ_UNITLIST` = 1050, `E_PFC_ERROR_READ_TIMESLICE` = 1060,
 `E_PFC_ERROR_WRITE_INDEXFILENAME_EMPTY` = 1100, `E_PFC_ERROR_WRITE_INDEXFILE_OPENERERROR`
 = 1101, `E_PFC_ERROR_WRITE_FILEINFO` = 1110, `E_PFC_ERROR_WRITE_PFC_DATATYPE` = 1111,
 `E_PFC_ERROR_WRITE_COMPRESSINFO` = 1130, `E_PFC_ERROR_WRITE_FILEPATH` = 1140,
 `E_PFC_ERROR_WRITE_UNITLIST` = 1150, `E_PFC_ERROR_WRITE_TIMESLICE` = 1160,
 `E_PFC_ERROR_READ_PROCFILE_OPENERERROR` = 1200, `E_PFC_ERROR_READ_DOMAIN` = 1210,
 `E_PFC_ERROR_READ_PFC_GLOBALORIGIN` = 1211, `E_PFC_ERROR_READ_PFC_GLOBALREGION`
 = 1212,
 `E_PFC_ERROR_READ_PFC_GLOBALVOXEL` = 1213, `E_PFC_ERROR_READ_PFC_GLOBALDIVISION`

```
= 1214, E_PFC_ERROR_READ_DIVISION = 1220, E_PFC_ERROR_READ_PFC_NO_REGION = 1221,
E_PFC_ERROR_READ_PFC_REGION_ID = 1222, E_PFC_ERROR_READ_PFC_REGION_VOXELSIZE =
1223, E_PFC_ERROR_READ_PFC_REGION_HEADINDEX = 1224, E_PFC_ERROR_READ_PFC_REGION_TAILINDEX
= 1225,
E_PFC_ERROR_WRITE_PROCFILE_OPENERERROR = 1300, E_PFC_ERROR_WRITE_DOMAIN = 1310,
E_PFC_ERROR_WRITE_DIVISION = 1320, E_PFC_ERROR_READ_PODBASE_OPENERERROR = 2000,
E_PFC_ERROR_READ_PODBASE_HEADER = 2005, E_PFC_ERROR_READ_PODBASE_DATA = 2010,
E_PFC_ERROR_WRITE_PODBASE_OPENERERROR = 2100, E_PFC_ERROR_WRITE_PODBASE_DATA
= 2110,
E_PFC_ERROR_READ_PODCOEF_OPENERERROR = 2200, E_PFC_ERROR_READ_PODCOEF_DATA =
2210, E_PFC_ERROR_WRITE_PODCOEF_OPENERERROR = 2300, E_PFC_ERROR_WRITE_PODCOEF_DATA
= 2310,
E_PFC_ERROR_PFC_COMPRESSFORMAT = 3000, E_PFC_ERROR_OUT_OF_RANGE = 3100 }
```

関数

- char [PfcPath_getDelimChar](#) ()
- std::string [PfcPath_getDelimString](#) ()
- bool [PfcPath_hasDrive](#) (const std::string &path)
- std::string [PfcPath_emitDrive](#) (std::string &path)
- bool [PfcPath_isAbsolute](#) (const std::string &path)
- std::string [PfcPath_DirName](#) (const std::string &path, const char dc=[PfcPath_getDelimChar](#)())
- std::string [PfcPath_FileName](#) (const std::string &path, const std::string &addext=std::string(""), const char dc=[PfcPath_getDelimChar](#)())
- std::string [PfcPath_ConnectPath](#) (std::string dirName, std::string fName)
- [PFC::E_PFC_ERRORCODE MakeDirectory](#) (const std::string &path)

5.1.1 説明

namespace の設定

5.1.2 列挙型

5.1.2.1 enum [PFC::E_PFC_ARRAYSHAPE](#)

配列形式

列挙型の値

[E_PFC_ARRAYSHAPE_UNKNOWN](#) 未定

[E_PFC_IJKN](#) ijkn

[E_PFC_NIJK](#) nijk

[PfcDefine.h](#) の 119 行で定義されています。

```
120 {
121     E\_PFC\_ARRAYSHAPE\_UNKNOWN=-1,
122     E\_PFC\_IJKN=0,
123     E\_PFC\_NIJK
124 };
```

5.1.2.2 enum PFC::E_PFC_COMPRESS_FORMAT

圧縮形式

列挙型の値

E_PFC_COMPRESS_FMT_UNKNOWN 未定
E_PFC_COMPRESS_FMT_POD POD format.

PfcDefine.h の 95 行で定義されています。

```
96 {
97     E_PFC_COMPRESS_FMT_UNKNOWN = -1,
98     E_PFC_COMPRESS_FMT_POD = 1
99 };
```

5.1.2.3 enum PFC::E_PFC_DTYPE

データ形式

列挙型の値

E_PFC_DTYPE_UNKNOWN 未定
E_PFC_INT8 char
E_PFC_INT16 short
E_PFC_INT32 int
E_PFC_INT64 long long
E_PFC_UINT8 unsigned char
E_PFC_UINT16 unsigned short
E_PFC_UINT32 unsigned int
E_PFC_UINT64 unsigned long long
E_PFC_FLOAT32 float
E_PFC_FLOAT64 double

PfcDefine.h の 102 行で定義されています。

```
103 {
104     // 実際使われるのは E_PFC_FLOAT64 のみ
105     E_PFC_DTYPE_UNKNOWN = 0,
106     E_PFC_INT8,
107     E_PFC_INT16,
108     E_PFC_INT32,
109     E_PFC_INT64,
110     E_PFC_UINT8,
111     E_PFC_UINT16,
112     E_PFC_UINT32,
113     E_PFC_UINT64,
114     E_PFC_FLOAT32,
115     E_PFC_FLOAT64
116 };
```

5.1.2.4 enum PFC::E_PFC_ENDIAN_TYPE

Endian 形式

列挙型の値

E_PFC_ENDIAN_TYPE_UNKNOWN

E_PFC_LITTLE

E_PFC_BIG

PfcDefine.h の 127 行で定義されています。

```
128 {
129     E_PFC_ENDIANATYPE_UNKNOWN=-1,
130     E_PFC_LITTLE=0,
131     E_PFC_BIG
132 };
```

5.1.2.5 enum PFC::E_PFC_ERRORCODE

PFC のエラーコード

列挙型の値

E_PFC_SUCCESS 正常終了
E_PFC_ERROR エラー終了
E_PFC_ERROR_READ_CNTLFILE_OPENERERROR
E_PFC_ERROR_READ_CNTL_DOMAINDIVISION
E_PFC_ERROR_READ_CNTL_NO_ITEM
E_PFC_ERROR_READ_CNTL_ITEMCNTL
E_PFC_ERROR_READ_CNTL_DFI_PATH
E_PFC_ERROR_READ_CNTL_OUTDIR_PATH
E_PFC_ERROR_READ_CNTL_COMPRESS_FMT
E_PFC_ERROR_READ_CNTL_PROCFILE_SAVE
E_PFC_ERROR_READ_INDEXFILE_OPENERERROR
E_PFC_ERROR_READ_FILEINFO
E_PFC_ERROR_READ_PFC_DIRECTORYPATH
E_PFC_ERROR_READ_PFC_PREFIX
E_PFC_ERROR_READ_PFC_FILEFORMAT
E_PFC_ERROR_READ_PFC_GUIDECELL
E_PFC_ERROR_READ_PFC_DATATYPE
E_PFC_ERROR_READ_PFC_ENDIAN
E_PFC_ERROR_READ_PFC_ARRAYSHAPE
E_PFC_ERROR_READ_PFC_COMPONENT
E_PFC_ERROR_READ_COMPRESSINFO
E_PFC_ERROR_READ_PFC_COMPRESSFORMAT
E_PFC_ERROR_READ_PFC_COMPRESSERROR
E_PFC_ERROR_READ_PFC_CALCULATEDLAYER
E_PFC_ERROR_READ_PFC_VERSION
E_PFC_ERROR_READ_PFC_STARTSTEP
E_PFC_ERROR_READ_PFC_ENDSTEP
E_PFC_ERROR_READ_FILEPATH
E_PFC_ERROR_READ_PFC_DFIPATH
E_PFC_ERROR_READ_PFC_PFCPROCESS
E_PFC_ERROR_READ_UNITLIST
E_PFC_ERROR_READ_TIMESLICE

```

E_PFC_ERROR_WRITE_INDEXFILENAME_EMPTY
E_PFC_ERROR_WRITE_INDEXFILE_OPENERERROR
E_PFC_ERROR_WRITE_FILEINFO
E_PFC_ERROR_WRITE_PFC_DATATYPE
E_PFC_ERROR_WRITE_COMPRESSINFO
E_PFC_ERROR_WRITE_FILEPATH
E_PFC_ERROR_WRITE_UNITLIST
E_PFC_ERROR_WRITE_TIMESLICE
E_PFC_ERROR_READ_PROCFILE_OPENERERROR
E_PFC_ERROR_READ_DOMAIN
E_PFC_ERROR_READ_PFC_GLOBALORIGIN
E_PFC_ERROR_READ_PFC_GLOBALREGION
E_PFC_ERROR_READ_PFC_GLOBALVOXEL
E_PFC_ERROR_READ_PFC_GLOBALDIVISION
E_PFC_ERROR_READ_DIVISION
E_PFC_ERROR_READ_PFC_NO_REGION
E_PFC_ERROR_READ_PFC_REGION_ID
E_PFC_ERROR_READ_PFC_REGION_VOXELSIZE
E_PFC_ERROR_READ_PFC_REGION_HEADINDEX
E_PFC_ERROR_READ_PFC_REGION_TAILINDEX
E_PFC_ERROR_WRITE_PROCFILE_OPENERERROR
E_PFC_ERROR_WRITE_DOMAIN
E_PFC_ERROR_WRITE_DIVISION
E_PFC_ERROR_READ_PODBASE_OPENERERROR
E_PFC_ERROR_READ_PODBASE_HEADER
E_PFC_ERROR_READ_PODBASE_DATA
E_PFC_ERROR_WRITE_PODBASE_OPENERERROR
E_PFC_ERROR_WRITE_PODBASE_DATA
E_PFC_ERROR_READ_PODCOEF_OPENERERROR
E_PFC_ERROR_READ_PODCOEF_DATA
E_PFC_ERROR_WRITE_PODCOEF_OPENERERROR
E_PFC_ERROR_WRITE_PODCOEF_DATA
E_PFC_ERROR_PFC_COMPRESSFORMAT
E_PFC_ERROR_OUT_OF_RANGE

```

PfcDefine.h の 147 行で定義されています。

```

148 {
149     E_PFC_SUCCESS = 1,
150     E_PFC_ERROR = -1,
151
152     // READ 系 (pfc_cntl)
153     E_PFC_ERROR_READ_CNTLFILE_OPENERERROR = 500,
154     E_PFC_ERROR_READ_CNTL_DOMAINDIVISION = 501, // PfcCompressCntl/DomainDivision
155     E_PFC_ERROR_READ_CNTL_NO_ITEM = 510,
156     E_PFC_ERROR_READ_CNTL_ITEMCNTL = 511, // PfcCompressCntl/ItemCntl
157     E_PFC_ERROR_READ_CNTL_DFI_PATH = 512, // PfcCompressCntl/ItemCntl/InputDfiPath
158     E_PFC_ERROR_READ_CNTL_OUTDIR_PATH = 513, // PfcCompressCntl/ItemCntl/OutputDirectoryPath
159     E_PFC_ERROR_READ_CNTL_COMPRESS_FMT = 514, // PfcCompressCntl/ItemCntl/CompressFormat
160     E_PFC_ERROR_READ_CNTL_PROCFILE_SAVE = 515, // PfcCompressCntl/ItemCntl/ProcFileSave
161
162     // READ 系 (index.pfc)
163     E_PFC_ERROR_READ_INDEXFILE_OPENERERROR = 1000,
164     E_PFC_ERROR_READ_FILEINFO = 1010, // FileInfo

```

```

165 E_PFC_ERROR_READ_PFC_DIRECTORYPATH = 1011, // FileInfo/DirectoryPath 読みエラー
166 E_PFC_ERROR_READ_PFC_PREFIX        = 1012, // FileInfo/Prefix
167 E_PFC_ERROR_READ_PFC_FILEFORMAT    = 1013, // FileInfo/FileFormat
168 E_PFC_ERROR_READ_PFC_GUIDECCELL    = 1014, // FileInfo/GuideCell
169 E_PFC_ERROR_READ_PFC_DATATYPE      = 1015, // FileInfo/DataType
170 E_PFC_ERROR_READ_PFC_ENDIAN        = 1016, // FileInfo/Endian
171 E_PFC_ERROR_READ_PFC_ARRAYSHAPE     = 1017, // FileInfo/ArrayShape
172 E_PFC_ERROR_READ_PFC_COMPONENT     = 1018, // FileInfo/Component
173 E_PFC_ERROR_READ_PFC_COMPRESSINFO  = 1030, // CompressInfo
174 E_PFC_ERROR_READ_PFC_COMPRESSFORMAT = 1031, // CompressInfo/CompressFormat
175 E_PFC_ERROR_READ_PFC_COMPRESSERROR  = 1032, // CompressInfo/CompressError
176 E_PFC_ERROR_READ_PFC_CALCULATEDLAYER = 1033, // CompressInfo/CalculatedLayer
177 E_PFC_ERROR_READ_PFC_VERSION       = 1034, // CompressInfo/Version
178 E_PFC_ERROR_READ_PFC_STARTSTEP     = 1035, // CompressInfo/StartStep
179 E_PFC_ERROR_READ_PFC_ENDSTEP       = 1036, // CompressInfo/EndStep
180 E_PFC_ERROR_READ_PFC_FILEPATH      = 1040, // FilePath
181 E_PFC_ERROR_READ_PFC_DFIPATH       = 1041, // FilePath/DfiPath
182 E_PFC_ERROR_READ_PFC_PFCPROCESS    = 1042, // FilePath/PfcProcess
183 E_PFC_ERROR_READ_PFC_UNITLIST      = 1050, // UnitList
184 E_PFC_ERROR_READ_PFC_TIMESLICE     = 1060, // TimeSlice
185
186 // WRITE 系 (index.pfc)
187 E_PFC_ERROR_WRITE_INDEXFILENAME_EMPTY = 1100,
188 E_PFC_ERROR_WRITE_INDEXFILE_OPENERROR = 1101,
189 E_PFC_ERROR_WRITE_FILEINFO            = 1110, // FileInfo
190 E_PFC_ERROR_WRITE_PFC_DATATYPE        = 1111, // FileInfo/DataType
191 E_PFC_ERROR_WRITE_PFC_COMPRESSINFO    = 1130, // CompressInfo
192 E_PFC_ERROR_WRITE_PFC_FILEPATH        = 1140, // FilePath
193 E_PFC_ERROR_WRITE_PFC_UNITLIST        = 1150, // UnitList
194 E_PFC_ERROR_WRITE_PFC_TIMESLICE       = 1160, // TimeSlice
195
196 // READ 系 (proc.pfc)
197 E_PFC_ERROR_READ_PFC_PROCFILE_OPENERROR = 1200,
198 E_PFC_ERROR_READ_PFC_DOMAIN              = 1210, // Domain
199 E_PFC_ERROR_READ_PFC_GLOBALORIGIN        = 1211, // Domain/GlobalOrigin
200 E_PFC_ERROR_READ_PFC_GLOBALREGION        = 1212, // Domain/GlobalRegion
201 E_PFC_ERROR_READ_PFC_GLOBALVOXEL        = 1213, // Domain/GlobalVoxel
202 E_PFC_ERROR_READ_PFC_GLOBALDIVISION      = 1214, // Domain/GlobalDivision
203 E_PFC_ERROR_READ_PFC_DIVISION            = 1220, // Division
204 E_PFC_ERROR_READ_PFC_NO_REGION           = 1221, // Division/Region 読みエラー
205 E_PFC_ERROR_READ_PFC_REGION_ID           = 1222, // Division/Region/ID
206 E_PFC_ERROR_READ_PFC_REGION_VOXELSIZE   = 1223, // Division/Region/VoxelSize
207 E_PFC_ERROR_READ_PFC_REGION_HEADINDEX    = 1224, // Division/Region/HeadIndex
208 E_PFC_ERROR_READ_PFC_REGION_TAILINDEX    = 1225, // Division/Region/TailIndex
209
210 // WRITE 系 (proc.pfc)
211 E_PFC_ERROR_WRITE_PFC_PROCFILE_OPENERROR = 1300,
212 E_PFC_ERROR_WRITE_PFC_DOMAIN              = 1310,
213 E_PFC_ERROR_WRITE_PFC_DIVISION            = 1320,
214
215 // READ 系 (基底ファイル)
216 E_PFC_ERROR_READ_PFC_PODBASE_OPENERROR   = 2000,
217 E_PFC_ERROR_READ_PFC_PODBASE_HEADER      = 2005,
218 E_PFC_ERROR_READ_PFC_PODBASE_DATA        = 2010,
219
220 // WRITE 系 (基底ファイル)
221 E_PFC_ERROR_WRITE_PFC_PODBASE_OPENERROR  = 2100,
222 E_PFC_ERROR_WRITE_PFC_PODBASE_DATA       = 2110,
223
224 // READ 系 (係数ファイル)
225 E_PFC_ERROR_READ_PFC_PODCOEF_OPENERROR   = 2200,
226 E_PFC_ERROR_READ_PFC_PODCOEF_DATA        = 2210,
227
228 // WRITE 系 (係数ファイル)
229 E_PFC_ERROR_WRITE_PFC_PODCOEF_OPENERROR  = 2300,
230 E_PFC_ERROR_WRITE_PFC_PODCOEF_DATA       = 2310,
231
232 // その他のエラー
233 E_PFC_ERROR_PFC_COMPRESSFORMAT           = 3000,
234 E_PFC_ERROR_OUT_OF_RANGE                 = 3100
235
236 };

```

5.1.3 関数

5.1.3.1 PFC::E_PFC_ERRORCODE PFC::MakeDirectory (const std::string & path)

5.1.3.2 std::string PFC::PfcPath_ConnectPath (std::string dirName, std::string fname) [inline]

PfcPathUtil.h の 191 行で定義されています。

参照先 PfcPath_getDelimChar(), と PfcPath_getDelimString().

```

192 {
193     std::string path = dirName;
194
195     const char *p = dirName.c_str();
196     if( p[strlen(p)-1] != PFC::PfcPath_getDelimChar() )
197     {
198         path += PFC::PfcPath_getDelimString();
199     }
200
201     path += fname;
202
203     return path;
204 }
```

5.1.3.3 `std::string PFC::PfcPath_DirName(const std::string & path, const char dc = PfcPath_getDelimChar())`
[inline]

PfcPathUtil.h の 87 行で定義されています。

参照先 PfcPath_isAbsolute().

```

88                                     {
89     char* name = strdup( path.c_str() );
90     char* p = name;
91
92     for ( ; ; ++p ) {
93         if ( ! *p ) {
94             if ( p > name ) {
95                 char rs[2] = {dc, '\0'};
96                 return rs;
97             } else {
98                 char rs[3] = {'.', dc, '\0'};
99                 return rs;
100             }
101         }
102         if ( *p != dc ) break;
103     }
104
105     for ( ; *p; ++p );
106     while ( *--p == dc ) continue;
107     *++p = '\0';
108
109     while ( --p >= name )
110         if ( *p == dc ) break;
111     ++p;
112     if ( p == name )
113     {
114         char rs[3] = {'.', dc, '\0'};
115         return rs;
116     }
117
118     while ( --p >= name )
119         if ( *p != dc ) break;
120     ++p;
121
122     *p = '\0';
123     if( p == name ) {
124         char rs[2] = {dc, '\0'};
125         return rs;
126     } else {
127         std::string s( name );
128         free( name );
129         if( !PFC::PfcPath_isAbsolute(s) )
130         {
131             const char *q = s.c_str();
132             if( q[0] != '.' && q[1] != '/' )
133             {
134                 char rs[3] = {'.', dc, '\0'};
135                 s = std::string(rs) + s;
136             }
137         }
138         return s;
139     }
140 }
```

5.1.3.4 `std::string PFC::PfcPath_FileName (const std::string & path, const std::string & addext = std::string(""), const char dc = PfcPath_getDelimChar()) [inline]`

PfcPathUtil.h の 143 行で定義されています。

```

145                                     {
146     char* name = strdup( path.c_str() );
147     char* p = name;
148
149     for ( ; ; ++p ) {
150         if ( ! *p ) {
151             if ( p > name ) {
152                 char rs[2] = {dc, '\0'};
153                 return rs;
154             } else
155                 return "";
156         }
157         if ( *p != dc ) break;
158     }
159
160     for ( ; *p; ++p ) continue;
161     while ( *--p == dc ) continue;
162     ++p = '\0';
163
164     while ( --p >= name )
165         if ( *p == dc ) break;
166     ++p;
167
168     bool add = false;
169     if ( addext.length() > 0 ) {
170         const int suffixlen = addext.length();
171         const int stringlen = strlen( p );
172         if ( suffixlen < stringlen ) {
173             const int off = stringlen - suffixlen;
174             if ( strcasecmp( p + off, addext.c_str() ) != 0 )
175                 add = true;
176         }
177         else
178         {
179             add = true;
180         }
181     }
182
183     std::string s( p );
184     if( add ) s += addext;
185
186     free( name );
187     return s;
188 }

```

5.1.3.5 `char PFC::PfcPath_getDelimChar () [inline]`

PfcPathUtil.h の 37 行で定義されています。

参照元 PfcPath_ConnectPath(), PfcPath_getDelimString(), と PfcPath_isAbsolute().

```

38 {
39 #ifdef WIN32
40     return '\\';
41 #else
42     return '/';
43 #endif
44 }

```

5.1.3.6 `std::string PFC::PfcPath_getDelimString () [inline]`

PfcPathUtil.h の 47 行で定義されています。

参照先 PfcPath_getDelimChar().

参照元 PfcPath_ConnectPath().

```

48 {
49     const char dc = PFC::PfcPath_getDelimChar();

```

```

50     char rs[2] = {dc, '\\0'};
51     return rs;
52 }

```

5.1.3.7 bool PFC::PfcPath_hasDrive (const std::string & path) [inline]

PfcPathUtil.h の 55 行で定義されています。

参照元 vfvPath_emitDrive().

```

55                                     {
56     if ( path.size() < 2 ) return false;
57     char x = path[0];
58     if ( ((x >= 'A' && x <= 'Z') || (x >= 'a' && x <= 'z')) &&
59         path[1] == ':' )
60         return true;
61     return false;
62 }

```

5.1.3.8 bool PFC::PfcPath_isAbsolute (const std::string & path) [inline]

PfcPathUtil.h の 76 行で定義されています。

参照先 PfcPath_getDelimChar(), と vfvPath_emitDrive().

参照元 PfcPath_DirName().

```

77 {
78     std::string xpath(path);
79     vfvPath_emitDrive(xpath);
80     char c1, c2;
81     c1 = xpath[0];
82     c2 = PfcPath_getDelimChar();
83     return (c1 == c2);
84 }

```

5.1.3.9 std::string PFC::vfvPath_emitDrive (std::string & path) [inline]

PfcPathUtil.h の 65 行で定義されています。

参照先 PfcPath_hasDrive().

参照元 PfcPath_isAbsolute().

```

66 {
67     // returns drive (ex. 'C:')
68     if ( ! PfcPath_hasDrive(path) ) return std::string();
69     std::string driveStr = path.substr(0, 2);
70     path = path.substr(2);
71     return driveStr;
72 }

```


Chapter 6

クラス

6.1 クラス CPfcCioDfiUtil

```
#include <PfcCioDfiUtil.h>
```

Static Public メソッド

- static [PFC::E_PFC_ERRORCODE GetDfiDomain](#) (const std::string &dfiFilePath, cio_Domain &domain)
DFI Domain 情報取得
- static [PFC::E_PFC_ERRORCODE GetDfiTimeStepList](#) (const std::string &dfiFilePath, vector< int > &time-StepList)
DFI タイムステップ リスト 情報取得

Static Protected メソッド

- static [PFC::E_PFC_ERRORCODE GetDfiInfo](#) (const std::string &dfiFilePath, cio_Domain &domain, cio_TimeSlice &timeSlice)
内部ルーチン: DFI Dfi 情報取得

6.1.1 説明

[PFC](#) CIO Dfi utility class

PfcCioDfiUtil.h の 34 行で定義されています。

6.1.2 関数

6.1.2.1 static [PFC::E_PFC_ERRORCODE CPfcCioDfiUtil::GetDfiDomain](#) (const std::string & *dfiFilePath*, cio_Domain & *domain*) [static]

DFI Domain 情報取得

引数

in	<i>dfiFilePath</i>	DFI ファイルパス
----	--------------------	------------

out	<i>domain</i>	ドメイン情報
-----	---------------	--------

戻り値

終了コード

```
6.1.2.2 static PFC::E_PFC_ERRORCODE CPfcCioDfiUtil::GetDfiInfo ( const std::string & dfiFilePath, cio_Domain &
domain, cio_TimeSlice & timeSlice ) [static],[protected]
```

内部ルーチン: DFI Dfi 情報取得

引数

in	<i>dfiFilePath</i>	DFI ファイルパス
out	<i>domain</i>	ドメイン情報
out	<i>timeSlice</i>	時系列データ

戻り値

終了コード

```
6.1.2.3 static PFC::E_PFC_ERRORCODE CPfcCioDfiUtil::GetDfiTimeStepList ( const std::string & dfiFilePath, vector<
int > & timeStepList ) [static]
```

DFI タイムステップ リスト 情報取得

引数

in	<i>dfiFilePath</i>	DFI ファイルパス
out	<i>timeStepList</i>	タイムステップ リスト

戻り値

終了コード

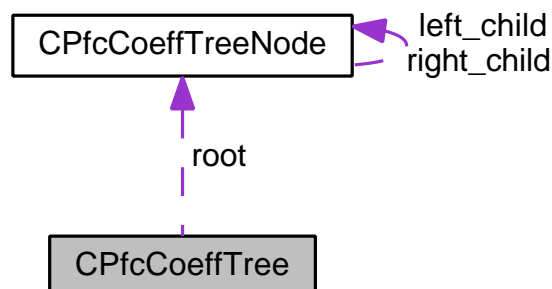
このクラスの説明は次のファイルから生成されました:

- [PfcCioDfiUtil.h](#)

6.2 クラス CPfcCoeffTree

```
#include <PfcCoeffTree.h>
```

CPfcCoeffTree のコラボレーション図



Public メソッド

- `CPfcCoeffTree` (int numTimeStep, int numLayer)
- `~CPfcCoeffTree` ()
- void `construct_tree` (int timestep)
- void `construct_child` (CPfcCoeffTreeNode *p, int curr_layer, int timestep_node)
- void `print_tree` ()
- void `print_node` (int c_layer)
- void `read_data` (CPfcCoeffTreeNode *p, int pos)
- int `get_position` (CPfcCoeffTreeNode *p, int pos)
- void `construct_array` ()
- void `deep_transerval` (int *result, int index)
- int `standard_timestep_node` (int timestep)

Public 変数

- `CPfcCoeffTreeNode * root`
- int `m_numTimeStep`
- int `m_numLayer`
- int `stand_space`
- int ** `data`
- int `data_size`
- int * `position`
- int `position_interval`

6.2.1 説明

PfcCoeffTree.h の 39 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 `CPfcCoeffTree::CPfcCoeffTree (int numTimeStep, int numLayer)`

6.2.2.2 `CPfcCoeffTree::~~CPfcCoeffTree ()`

6.2.3 関数

6.2.3.1 void `CPfcCoeffTree::construct_array ()`

6.2.3.2 void `CPfcCoeffTree::construct_child (CPfcCoeffTreeNode * p, int curr_layer, int timestep_node)`

6.2.3.3 void `CPfcCoeffTree::construct_tree (int timestep)`

6.2.3.4 void `CPfcCoeffTree::deep_transerval (int * result, int index)`

6.2.3.5 int `CPfcCoeffTree::get_position (CPfcCoeffTreeNode * p, int pos)`

6.2.3.6 void `CPfcCoeffTree::print_node (int c_layer)`

6.2.3.7 void `CPfcCoeffTree::print_tree ()`

6.2.3.8 void `CPfcCoeffTree::read_data (CPfcCoeffTreeNode * p, int pos)`

6.2.3.9 `int CPfcCoeffTree::standard_timestep_node (int timestep)`

6.2.4 変数

6.2.4.1 `int** CPfcCoeffTree::data`

PfcCoeffTree.h の 49 行で定義されています。

6.2.4.2 `int CPfcCoeffTree::data_size`

PfcCoeffTree.h の 50 行で定義されています。

6.2.4.3 `int CPfcCoeffTree::m_numLayer`

PfcCoeffTree.h の 46 行で定義されています。

6.2.4.4 `int CPfcCoeffTree::m_numTimeStep`

PfcCoeffTree.h の 44 行で定義されています。

6.2.4.5 `int* CPfcCoeffTree::position`

PfcCoeffTree.h の 51 行で定義されています。

6.2.4.6 `int CPfcCoeffTree::position_interval`

PfcCoeffTree.h の 52 行で定義されています。

6.2.4.7 `CPfcCoeffTreeNode* CPfcCoeffTree::root`

PfcCoeffTree.h の 42 行で定義されています。

6.2.4.8 `int CPfcCoeffTree::stand_space`

PfcCoeffTree.h の 48 行で定義されています。

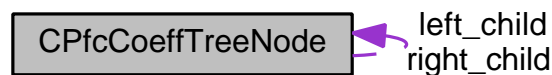
このクラスの説明は次のファイルから生成されました:

- [PfcCoeffTree.h](#)

6.3 クラス CPfcCoeffTreeNode

```
#include <PfcCoeffTree.h>
```

CPfcCoeffTreeNode のコラボレーション図



Public メソッド

- [CPfcCoeffTreeNode \(\)](#)
- [CPfcCoeffTreeNode \(int v0, int v1\)](#)
- [~CPfcCoeffTreeNode \(\)](#)

Public 変数

- bool [is_leaf](#)
- int * [value](#)
- [CPfcCoeffTreeNode *](#) [left_child](#)
- [CPfcCoeffTreeNode *](#) [right_child](#)

6.3.1 説明

PfcCoeffTree.h の 24 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 CPfcCoeffTreeNode::CPfcCoeffTreeNode () [inline]

PfcCoeffTree.h の 34 行で定義されています。

```
34 {};
```

6.3.2.2 CPfcCoeffTreeNode::CPfcCoeffTreeNode (int v0, int v1)

6.3.2.3 CPfcCoeffTreeNode::~~CPfcCoeffTreeNode ()

6.3.3 変数

6.3.3.1 bool CPfcCoeffTreeNode::is_leaf

PfcCoeffTree.h の 28 行で定義されています。

6.3.3.2 CPfcCoeffTreeNode* CPfcCoeffTreeNode::left_child

PfcCoeffTree.h の 30 行で定義されています。

6.3.3.3 CPfcCoeffTreeNode* CPfcCoeffTreeNode::right_child

PfcCoeffTree.h の 31 行で定義されています。

6.3.3.4 int* CPfcCoeffTreeNode::value

PfcCoeffTree.h の 29 行で定義されています。

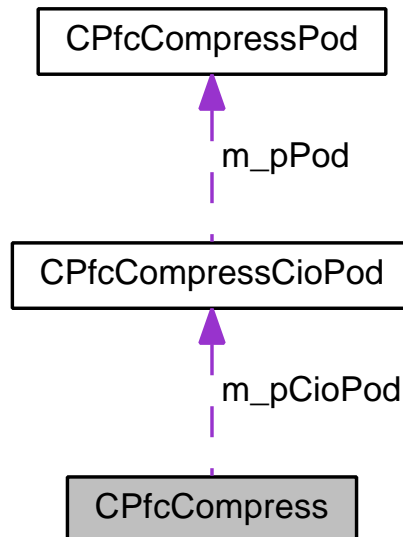
このクラスの説明は次のファイルから生成されました:

- [PfcCoeffTree.h](#)

6.4 クラス CPfcCompress

```
#include <PfcCompress.h>
```

CPfcCompress のコラボレーション図



Public メソッド

- [CPfcCompress \(\)](#)
- [~CPfcCompress \(\)](#)
- [PFC::E_PFC_ERRORCODE Init](#) (const [MPI_Comm](#) comm, const std::string &dfiFilePath, const std::string &outDirPath, const std::string &compressFormat, const double compressError, const int domainDivision[3])
圧縮クラス 初期化
- [PFC::E_PFC_ERRORCODE Init](#) (const [MPI_Comm](#) comm, const std::string &dfiFilePath, const std::string &outDirPath, const std::string &compressFormat, const double compressError, const int domainDivision[3], const int startStep, const int endStep, const int optFlags)
圧縮クラス 初期化
- [PFC::E_PFC_ERRORCODE WriteData](#) (void)
圧縮 & 圧縮ファイル出力
- [PFC::E_PFC_ERRORCODE WriteProcFile](#) (void)
proc.pfc ファイル出力
- [PFC::E_PFC_ERRORCODE WriteIndexPfcFile](#) (void)
index.pfc ファイル出力

Protected メソッド

- [PFC::E_PFC_ERRORCODE GatherAndCreateDivisionInfo](#) ([CPfcDivision](#) &division)
proc.pfc Division&Region 情報の収集 & 作成

Protected 変数

- [CPfcCompressCioPod * m_pCioPod](#)
- [MPI_Comm m_comm](#)
MPI コミュニケーター

- std::string `m_dfiFilePath`
CIO の DFI ファイル
- std::string `m_outDirPath`
フィールドデータの出力ディレクトリパス
- std::string `m_compressFormat`
- double `m_compressError`
- int `m_domainDivision` [3]
計算領域の分割数
- int `m_startStep`
開始タイムステップ
- int `m_endStep`
終了タイムステップ
- int `m_optFlags`
- cio_DFI * `m_pDfiIn`
- std::string `m_prefix`
- std::string `m_fileFormat`
- PFC::E_PFC_DTYPE `m_dataType`
- PFC::E_PFC_ENDIANTYPE `m_endianType`
- PFC::E_PFC_ARRAYSHAPE `m_arrayShape`
- int `m_numComponent`
- int `m_numRank`
- int `m_numRegion`
- int `m_numParallel`
- int `m_numStep`
- vector< int > `m_stepList`
- int `m_regionID`
- int `m_regionHead` [3]
- int `m_regionTail` [3]
- int `m_regionSize`
- int `m_regionMasterRankID`
- int `m_myRankID`
- int `m_myIDinRegion`

6.4.1 説明

PFC Compress main class

PfcCompress.h の 32 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 CPfcCompress::CPfcCompress ()

コンストラクタ

6.4.2.2 CPfcCompress::~CPfcCompress ()

デストラクタ

6.4.3 関数

6.4.3.1 **PFC::E_PFC_ERRORCODE** CPfcCompress::GatherAndCreateDivisionInfo (CPfcDivision & *division*)
[protected]

proc.pfc Division&Region 情報の収集 & 作成

引数

out	division	proc.pfc Division&Region クラス
-----	----------	------------------------------

戻り値

終了コード

6.4.3.2 PFC::E_PFC_ERRORCODE CPfcCompress::Init (const MPI_Comm comm, const std::string & dfiFilePath, const std::string & outDirPath, const std::string & compressFormat, const double compressError, const int domainDivision[3])

圧縮クラス 初期化

引数

in	comm	MPI コミュニケータ
in	dfiFilePath	DFI ファイルパス
in	outDirPath	フィールドデータ出力ディレクトリパス
in	compressFormat	圧縮フォーマット "pod" POD 圧縮
in	compressError	圧縮率
in	domainDivision	領域の分割数

戻り値

終了コード

6.4.3.3 PFC::E_PFC_ERRORCODE CPfcCompress::Init (const MPI_Comm comm, const std::string & dfiFilePath, const std::string & outDirPath, const std::string & compressFormat, const double compressError, const int domainDivision[3], const int startStep, const int endStep, const int optFlags)

圧縮クラス 初期化

引数

in	comm	MPI コミュニケータ
in	dfiFilePath	DFI ファイルパス
in	outDirPath	フィールドデータ出力ディレクトリパス
in	compressFormat	圧縮フォーマット "pod" POD 圧縮
in	compressError	誤差率 (%)
in	domainDivision	領域の分割数
in	startStep	開始ステップ -1 全タイムステップ
in	endStep	終了ステップ -1 全タイムステップ
in	optFlags	オプション 0 オプションなし

戻り値

終了コード

6.4.3.4 PFC::E_PFC_ERRORCODE CPfcCompress::WriteData (void)

圧縮 & 圧縮ファイル出力

引数

out	ret	終了コード
-----	-----	-------

戻り値

終了コード

6.4.3.5 PFC::E_PFC_ERRORCODE CPfcCompress::WriteIndexPfcFile (void)

index.pfc ファイル出力

戻り値

終了コード

6.4.3.6 PFC::E_PFC_ERRORCODE CPfcCompress::WriteProcFile (void)

proc.pfc ファイル出力

戻り値

終了コード

6.4.4 変数

6.4.4.1 PFC::E_PFC_ARRAYSHAPE CPfcCompress::m_arrayShape [protected]

PfcCompress.h の 66 行で定義されています。

6.4.4.2 MPI_Comm CPfcCompress::m_comm [protected]

MPI コミュニケータ

PfcCompress.h の 42 行で定義されています。

6.4.4.3 double CPfcCompress::m_compressError [protected]

PfcCompress.h の 46 行で定義されています。

6.4.4.4 std::string CPfcCompress::m_compressFormat [protected]

PfcCompress.h の 45 行で定義されています。

6.4.4.5 PFC::E_PFC_DTYPE CPfcCompress::m_dataType [protected]

PfcCompress.h の 62 行で定義されています。

6.4.4.6 std::string CPfcCompress::m_dfiFilePath [protected]

CIO のDFI ファイル

PfcCompress.h の 43 行で定義されています。

6.4.4.7 `int CPfcCompress::m_domainDivision[3] [protected]`

計算領域の分割数

PfcCompress.h の 47 行で定義されています。

6.4.4.8 `PFC::E_PFC_ENDIANTYPE CPfcCompress::m_endianType [protected]`

PfcCompress.h の 65 行で定義されています。

6.4.4.9 `int CPfcCompress::m_endStep [protected]`

終了タイムステップ

PfcCompress.h の 50 行で定義されています。

6.4.4.10 `std::string CPfcCompress::m_fileFormat [protected]`

PfcCompress.h の 61 行で定義されています。

6.4.4.11 `int CPfcCompress::m_myIDinRegion [protected]`

PfcCompress.h の 99 行で定義されています。

6.4.4.12 `int CPfcCompress::m_myRankID [protected]`

PfcCompress.h の 98 行で定義されています。

6.4.4.13 `int CPfcCompress::m_numComponent [protected]`

PfcCompress.h の 67 行で定義されています。

6.4.4.14 `int CPfcCompress::m_numParallel [protected]`

PfcCompress.h の 76 行で定義されています。

6.4.4.15 `int CPfcCompress::m_numRank [protected]`

PfcCompress.h の 72 行で定義されています。

6.4.4.16 `int CPfcCompress::m_numRegion [protected]`

PfcCompress.h の 74 行で定義されています。

6.4.4.17 `int CPfcCompress::m_numStep [protected]`

PfcCompress.h の 79 行で定義されています。

6.4.4.18 `int CPfcCompress::m_optFlags [protected]`

PfcCompress.h の 52 行で定義されています。

6.4.4.19 `std::string CPfcCompress::m_outDirPath` [protected]

フィールドデータの出力ディレクトリパス
PfcCompress.h の 44 行で定義されています。

6.4.4.20 `CPfcCompressCioPod* CPfcCompress::m_pCioPod` [protected]

PfcCompress.h の 38 行で定義されています。

6.4.4.21 `cio_DFI* CPfcCompress::m_pDfiIN` [protected]

PfcCompress.h の 56 行で定義されています。

6.4.4.22 `std::string CPfcCompress::m_prefix` [protected]

PfcCompress.h の 60 行で定義されています。

6.4.4.23 `int CPfcCompress::m_regionHead[3]` [protected]

PfcCompress.h の 86 行で定義されています。

6.4.4.24 `int CPfcCompress::m_regionID` [protected]

PfcCompress.h の 85 行で定義されています。

6.4.4.25 `int CPfcCompress::m_regionMasterRankID` [protected]

PfcCompress.h の 92 行で定義されています。

6.4.4.26 `int CPfcCompress::m_regionSize` [protected]

PfcCompress.h の 88 行で定義されています。

6.4.4.27 `int CPfcCompress::m_regionTail[3]` [protected]

PfcCompress.h の 87 行で定義されています。

6.4.4.28 `int CPfcCompress::m_startStep` [protected]

開始タイムステップ
PfcCompress.h の 48 行で定義されています。

6.4.4.29 `vector<int> CPfcCompress::m_stepList` [protected]

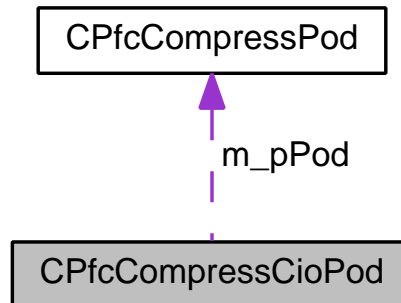
PfcCompress.h の 80 行で定義されています。
このクラスの説明は次のファイルから生成されました:

- [PfcCompress.h](#)

6.5 クラス CPfcCompressCioPod

```
#include <PfcCompressCioPod.h>
```

CPfcCompressCioPod のコラボレーション図



Public メソッド

- [CPfcCompressCioPod \(\)](#)
- [~CPfcCompressCioPod \(\)](#)
- [PFC::E_PFC_ERRORCODE Init](#) (const [MPI_Comm](#) comm, cio_DFI *pDfilN, const std::string &outDirPath, const double compressError, const int numRegion, const int head[3], const int tail[3], const vector< int > *pStepList, const int optFlags)
圧縮クラス 初期化
- [PFC::E_PFC_ERRORCODE WriteData](#) (void)
圧縮 & 圧縮ファイル出力

Public 変数

- [CPfcCompressPod * m_pPod](#)

Protected メソッド

- [PFC::E_PFC_ERRORCODE ReadCioFile](#) (double *pFlowData, const [PFC::E_PFC_DTYPE](#) dataType, const int startStep, const int numStep, const int unit)
CIO ファイル Read original 配列形状 (NIJK/IJKN) 順
- [PFC::E_PFC_ERRORCODE ReadCioFile_IJKN](#) (double *pFlowData, const [PFC::E_PFC_DTYPE](#) dataType, const int startStep, const int numStep, const int unit)
CIO ファイル Read 配列形状 (IJKN) 順 POD 圧縮内部データをIJKN 順とするため

Protected 変数

- double * [m_pFlowData](#)
- [MPI_Comm](#) [m_comm](#)
MPI コミュニケータ
- cio_DFI * [m_pDfilN](#)
- std::string [m_outDirPath](#)
フィールドデータの出力ディレクトリパス
- double [m_compressError](#)
- int [m_numRegion](#)
- int [m_regionHead](#) [3]

- int `m_regionTail` [3]
- int `m_numStep`
- const vector< int > * `m_pStepList`
- int `m_optFlags`
- int `m_numRank`
- int `m_myRankID`

6.5.1 説明

PFC CompressCioPod main class

PfcCompressCioPod.h の 31 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 CPfcCompressCioPod::CPfcCompressCioPod ()

コンストラクタ

6.5.2.2 CPfcCompressCioPod::~CPfcCompressCioPod ()

デストラクタ

6.5.3 関数

6.5.3.1 PFC::E_PFC_ERRORCODE CPfcCompressCioPod::Init (const MPI_Comm *comm*, cio_DFI * *pDfilN*, const std::string & *outDirPath*, const double *compressError*, const int *numRegion*, const int *head*[3], const int *tail*[3], const vector< int > * *pStepList*, const int *optFlags*)

圧縮クラス 初期化

引数

in	<i>comm</i>	MPI コミュニケータ
in	<i>pDfilN</i>	CIO DFI クラス
in	<i>outDirPath</i>	フィールドデータ出力ディレクトリパス
in	<i>compressError</i>	誤差率 (%)
in	<i>numRegion</i>	分割領域数
in	<i>head</i>	分割計算領域の開始位置
in	<i>tail</i>	分割計算領域の終了位置
in	<i>pStepList</i>	タイムステップリスト
in	<i>optFlags</i>	オプション 0 オプションなし

戻り値

終了コード

6.5.3.2 PFC::E_PFC_ERRORCODE CPfcCompressCioPod::ReadCioFile (double * *pFlowData*, const PFC::E_PFC_DTYPE *dataType*, const int *startStep*, const int *numStep*, const int *unit*) [protected]

CIO ファイル Read original 配列形状 (NIJK/IJKN) 順

引数

in	<i>pFlowData</i>	読み込みデータ
in	<i>dataType</i>	配列のデータタイプ
in	<i>startStep</i>	読み込み開始ステップ (0 からの連番)
in	<i>numStep</i>	時間軸方向のサイズ
in	<i>unit</i>	領域方向のサイズ

戻り値

終了コード

6.5.3.3 PFC::E_PFC_ERRORCODE CPfcCompressCioPod::ReadCioFile_IJKN (double * *pFlowData*, const PFC::E_PFC_DTYPE *dataType*, const int *startStep*, const int *numStep*, const int *unit*) [protected]

CIO ファイル Read 配列形状 (IJKN) 順 POD 圧縮内部データをIJKN 順とするため

引数

in	<i>pFlowData</i>	読み込みデータ 配列形状 (IJKN) 順
in	<i>dataType</i>	配列のデータタイプ
in	<i>startStep</i>	読み込み開始ステップ (0 からの連番)
in	<i>numStep</i>	時間軸方向のサイズ
in	<i>unit</i>	領域方向のサイズ

戻り値

終了コード

6.5.3.4 PFC::E_PFC_ERRORCODE CPfcCompressCioPod::WriteData (void)

圧縮 & 圧縮ファイル出力

引数

out	<i>ret</i>	終了コード
-----	------------	-------

戻り値

終了コード

6.5.4 変数

6.5.4.1 MPI_Comm CPfcCompressCioPod::m_comm [protected]

MPI コミュニケータ

PfcCompressCioPod.h の 41 行で定義されています。

6.5.4.2 double CPfcCompressCioPod::m_compressError [protected]

PfcCompressCioPod.h の 44 行で定義されています。

6.5.4.3 int CPfcCompressCioPod::m_myRankID [protected]

PfcCompressCioPod.h の 55 行で定義されています。

6.5.4.4 `int CPfcCompressCioPod::m_numRank` `[protected]`

PfcCompressCioPod.h の 54 行で定義されています。

6.5.4.5 `int CPfcCompressCioPod::m_numRegion` `[protected]`

PfcCompressCioPod.h の 45 行で定義されています。

6.5.4.6 `int CPfcCompressCioPod::m_numStep` `[protected]`

PfcCompressCioPod.h の 48 行で定義されています。

6.5.4.7 `int CPfcCompressCioPod::m_optFlags` `[protected]`

PfcCompressCioPod.h の 51 行で定義されています。

6.5.4.8 `std::string CPfcCompressCioPod::m_outDirPath` `[protected]`

フィールドデータの出力ディレクトリパス

PfcCompressCioPod.h の 43 行で定義されています。

6.5.4.9 `cio_DFI* CPfcCompressCioPod::m_pDfiIN` `[protected]`

PfcCompressCioPod.h の 42 行で定義されています。

6.5.4.10 `double* CPfcCompressCioPod::m_pFlowData` `[protected]`

PfcCompressCioPod.h の 39 行で定義されています。

6.5.4.11 `CPfcCompressPod* CPfcCompressCioPod::m_pPod`

PfcCompressCioPod.h の 35 行で定義されています。

6.5.4.12 `const vector<int>* CPfcCompressCioPod::m_pStepList` `[protected]`

PfcCompressCioPod.h の 49 行で定義されています。

6.5.4.13 `int CPfcCompressCioPod::m_regionHead[3]` `[protected]`

PfcCompressCioPod.h の 46 行で定義されています。

6.5.4.14 `int CPfcCompressCioPod::m_regionTail[3]` `[protected]`

PfcCompressCioPod.h の 47 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:

- [PfcCompressCioPod.h](#)

6.6 クラス CPfcCompressCmd

```
#include <PfcCompressCmd.h>
```

Public メソッド

- [CPfcCompressCmd](#) (const std::string &compressCntlPath)
- [~CPfcCompressCmd](#) ()
- [PFC::E_PFC_ERRORCODE Execute](#) (void)
圧縮実行

Protected メソッド

- [PFC::E_PFC_ERRORCODE ReadCompressCntl](#) (void)
圧縮制御情報ファイル読み込み

Protected 変数

- std::string [m_compressCntlPath](#)
圧縮制御情報ファイルのディレクトリパス
- int [m_domainDivision](#) [3]
計算領域の分割数
- vector< [CPfcCompressCmdElm](#) > [m_cmdList](#)
圧縮制御情報リスト

6.6.1 説明

[PFC](#) Compress command main class

PfcCompressCmd.h の 25 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 [CPfcCompressCmd::CPfcCompressCmd](#) (const std::string & *compressCntlPath*)

コンストラクタ

6.6.2.2 [CPfcCompressCmd::~~CPfcCompressCmd](#) ()

デストラクタ

6.6.3 関数

6.6.3.1 [PFC::E_PFC_ERRORCODE CPfcCompressCmd::Execute](#) (void)

圧縮実行

戻り値

エラーコード

6.6.3.2 `PFC::E_PFC_ERRORCODE CPfcCompressCmd::ReadCompressCntl (void)` `[protected]`

圧縮制御情報ファイル読み込み

戻り値

エラーコード

6.6.4 変数

6.6.4.1 `vector<CPfcCompressCmdElm> CPfcCompressCmd::m_cmdList` `[protected]`

圧縮制御情報リスト

PfcCompressCmd.h の 37 行で定義されています。

6.6.4.2 `std::string CPfcCompressCmd::m_compressCntlPath` `[protected]`

圧縮制御情報ファイルのディレクトリパス

PfcCompressCmd.h の 31 行で定義されています。

6.6.4.3 `int CPfcCompressCmd::m_domainDivision[3]` `[protected]`

計算領域の分割数

PfcCompressCmd.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:

- [PfcCompressCmd.h](#)

6.7 クラス CPfcCompressCmdElm

```
#include <PfcCompressCmdElm.h>
```

Public メソッド

- [CPfcCompressCmdElm \(\)](#)
- [~CPfcCompressCmdElm \(\)](#)
- [PFC::E_PFC_ERRORCODE ReadElm \(CPfcTextParser tpCntl, const std::string &label_leaf\)](#)
ItemCntl 要素を読み込む (pfc_cntl)

Public 変数

- [std::string m_dfiFilePath](#)
CIO の DFI ファイル
- [std::string m_outDirPath](#)
フィールドデータの出力ディレクトリパス
- [std::string m_compressFormat](#)
- [double m_compressError](#)
- [int m_startStep](#)
開始タイムステップ

- int `m_endStep`
終了タイムステップ
- std::string `m_procFileSave`
- std::string `m_optSave`
- int `m_optFlags`

6.7.1 説明

pfc_cntl ファイルの圧縮制御データ要素

PfcCompressCmdElm.h の 23 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 CPfcCompressCmdElm::CPfcCompressCmdElm ()

コンストラクタ

6.7.2.2 CPfcCompressCmdElm::~~CPfcCompressCmdElm ()

デストラクタ

6.7.3 関数

6.7.3.1 PFC::E_PFC_ERRORCODE CPfcCompressCmdElm::ReadElm (CPfcTextParser *tpCntl*, const std::string & *label_leaf*)

ItemCntl 要素を読み込む (pfc_cntl)

引数

in	<i>tpCntl</i>	CPfcTextParser クラス
in	<i>label_leaf</i>	ベースとなる名前 ("/PfcCompressCntl/ItemCntl")

戻り値

error code

6.7.4 変数

6.7.4.1 double CPfcCompressCmdElm::m_compressError

PfcCompressCmdElm.h の 31 行で定義されています。

6.7.4.2 std::string CPfcCompressCmdElm::m_compressFormat

PfcCompressCmdElm.h の 30 行で定義されています。

6.7.4.3 std::string CPfcCompressCmdElm::m_dfiFilePath

CIO のDFI ファイル

PfcCompressCmdElm.h の 27 行で定義されています。

6.7.4.4 int CPfcCompressCmdElm::m_endStep

終了タイムステップ

PfcCompressCmdElm.h の 37 行で定義されています。

6.7.4.5 int CPfcCompressCmdElm::m_optFlags

PfcCompressCmdElm.h の 46 行で定義されています。

6.7.4.6 std::string CPfcCompressCmdElm::m_optSave

PfcCompressCmdElm.h の 42 行で定義されています。

6.7.4.7 std::string CPfcCompressCmdElm::m_outDirPath

フィールドデータの出力ディレクトリパス

PfcCompressCmdElm.h の 28 行で定義されています。

6.7.4.8 std::string CPfcCompressCmdElm::m_procFileSave

PfcCompressCmdElm.h の 40 行で定義されています。

6.7.4.9 int CPfcCompressCmdElm::m_startStep

開始タイムステップ

PfcCompressCmdElm.h の 34 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:

- [PfcCompressCmdElm.h](#)

6.8 クラス CPfcCompressInfo

```
#include <PfcCompressInfo.h>
```

Public メソッド

- [CPfcCompressInfo \(\)](#)
- [CPfcCompressInfo \(const std::string &compressFormat, const double compressError, const int calculatedLayer, const std::string &version, const int startStep, const int endStep\)](#)
コンストラクタ
- [~CPfcCompressInfo \(\)](#)
- [PFC::E_PFC_ERRORCODE Read \(CPfcTextParser tpCntl\)](#)
read CompressInfo(index.pfc)
- [PFC::E_PFC_ERRORCODE Write \(FILE *fp, const unsigned tab\)](#)
write CompressInfo(index.pfc)

Public 変数

- std::string [m_compressFormat](#)
- double [m_compressError](#)
- int [m_calculatedLayer](#)
- std::string [m_version](#)
- int [m_startStep](#)
- int [m_endStep](#)

6.8.1 説明

index.pfc ファイルの CompressInfo

PfcCompressInfo.h の 21 行で定義されています。

6.8.2 コンストラクタとデストラクタ

6.8.2.1 CPfcCompressInfo::CPfcCompressInfo ()

コンストラクタ

6.8.2.2 CPfcCompressInfo::CPfcCompressInfo (const std::string & *compressFormat*, const double *compressError*, const int *calculatedLayer*, const std::string & *version*, const int *startStep*, const int *endStep*)

コンストラクタ

引数

in	<i>compressFormat</i>	圧縮形式
in	<i>compressError</i>	誤差率 (%)
in	<i>calculatedLayer</i>	計算レイヤー数
in	<i>version</i>	圧縮バージョン
in	<i>startStep</i>	開始 step 数
in	<i>endStep</i>	終了 step 数

6.8.2.3 CPfcCompressInfo::~CPfcCompressInfo ()

デストラクタ

6.8.3 関数

6.8.3.1 PFC::E_PFC_ERRORCODE CPfcCompressInfo::Read (CPfcTextParser *tpCntl*)

read CompressInfo(index.pfc)

引数

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

戻り値

error code

6.8.3.2 PFC::E_PFC_ERRORCODE CPfcCompressInfo::Write (FILE * *fp*, const unsigned *tab*)

write CompressInfo(index.pfc)

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

6.8.4 変数

6.8.4.1 int CPfcCompressInfo::m_calculatedLayer

PfcCompressInfo.h の 27 行で定義されています。

6.8.4.2 double CPfcCompressInfo::m_compressError

PfcCompressInfo.h の 26 行で定義されています。

6.8.4.3 std::string CPfcCompressInfo::m_compressFormat

PfcCompressInfo.h の 25 行で定義されています。

6.8.4.4 int CPfcCompressInfo::m_endStep

PfcCompressInfo.h の 30 行で定義されています。

6.8.4.5 int CPfcCompressInfo::m_startStep

PfcCompressInfo.h の 29 行で定義されています。

6.8.4.6 std::string CPfcCompressInfo::m_version

PfcCompressInfo.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:

- [PfcCompressInfo.h](#)

6.9 クラス CPfcCompressPod

```
#include <PfcCompressPod.h>
```

Public メソッド

- [CPfcCompressPod \(\)](#)
- [~CPfcCompressPod \(\)](#)
- [PFC::E_PFC_ERRORCODE Init](#) (const [MPI_Comm](#) comm, const std::string &outDirPath, const std::string &prefix, const double compressError, const int numRegion, const int regionSize, const int numStep, const int optFlags, double *pFlowData, const int myNumStep)
圧縮クラス 初期化

- [PFC::E_PFC_ERRORCODE WriteData](#) (void)
圧縮 & 圧縮ファイル出力
- int [GetCalculatedLayer](#) (void)
POD 計算レイヤー数取得

Protected メソッド

- [PFC::E_PFC_ERRORCODE Output](#) (void)
POD ファイル出力 (基底ファイル, 係数ファイル)
- [PFC::E_PFC_ERRORCODE CalcPodBase](#) (double *pFlowData, const int unit, const int size, double *pPod_base_r)
基底計算
- void [CalcPodCoef](#) (double *pPod_base_r, double *pFlowData, const int unit, const int size, double *pCoef_a)
係数計算
- [PFC::E_PFC_ERRORCODE CheckFinish](#) (const int layerNo, const int base_size, const int num_size, const int nalloc_size, const double userInputError, double &evaluationError, const double *pFlowData, const double *pPod_base_r, const double *pCoef_a, int &finish_flg)
圧縮終了判定
- [PFC::E_PFC_ERRORCODE SwapBinaryData](#) (double *pPod_base_r, int &base_size, double *&pFlowData)
バイナリスワップ
- [PFC::E_PFC_ERRORCODE WritePodBaseFile](#) (double *pPod_base_r)
- [PFC::E_PFC_ERRORCODE WritePodBaseFile_Debug](#) (double *pPod_base_r)
- [PFC::E_PFC_ERRORCODE WritePodCoefFile](#) (double *pCoef_a)

Protected 変数

- [MPI_Comm m_comm](#)
- [std::string m_outDirPath](#)
- [std::string m_prefix](#)
- [double m_compressError](#)
- [int m_numRegion](#)
- [int m_regionSize](#)
- [int m_numStep](#)
- [int m_optFlags](#)
- [double * m_pFlowData](#)
- [int m_myNumStep](#)
- [int m_numRank](#)
- [int m_numParallel](#)
- [int m_maxLayer](#)
- [int m_calculatedLayer](#)
- [int m_regionID](#)
- [int m_regionMasterRankID](#)
- [int m_regionMaxStep](#)
- [int m_myRankID](#)
- [int m_myIDinRegion](#)
- [int m_myStartStepPos](#)
- [int m_layerNo](#)
- [int m_curPodBaseSize](#)
- [int m_curNumSize](#)
- [double m_curEvaluateError](#)
- [double * m_pPod_base_r](#)
- [double * m_pCoef_a](#)

6.9.1 説明

PFC Compress main class

PfcCompressPod.h の 28 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 CPfcCompressPod::CPfcCompressPod ()

コンストラクタ

6.9.2.2 CPfcCompressPod::~CPfcCompressPod ()

デストラクタ

6.9.3 関数

6.9.3.1 PFC::E_PFC_ERRORCODE CPfcCompressPod::CalcPodBase (double * *pFlowData*, const int *unit*, const int *size*, double * *pPod_base_r*) [protected]

基底計算

引数

in	<i>pFlowData</i>	[size][unit]
in	<i>unit</i>	領域内のサイズ
in	<i>size</i>	タイムステップ方向のサイズ
out	<i>pPod_base_r</i>	[unit]

戻り値

終了コード

6.9.3.2 void CPfcCompressPod::CalcPodCoef (double * *pPod_base_r*, double * *pFlowData*, const int *unit*, const int *size*, double * *pCoef_a*) [protected]

係数計算

引数

in	<i>pPod_base_r</i>	[unit]
in	<i>pFlowData</i>	[size][unit]
in	<i>unit</i>	領域内のサイズ
in	<i>size</i>	タイムステップ方向のサイズ
out	<i>pCoef_a</i>	[size]

戻り値

終了コード

6.9.3.3 PFC::E_PFC_ERRORCODE CPfcCompressPod::CheckFinish (const int *layerNo*, const int *base_size*, const int *num_size*, const int *nalloc_size*, const double *userInputError*, double & *evaluationError*, const double * *pFlowData*, const double * *pPod_base_r*, const double * *pCoef_a*, int & *finish_flg*) [protected]

圧縮終了判定

引数

in	<i>layerNo</i>	レイヤー No
in	<i>base_size</i>	基底データサイズ
in	<i>num_size</i>	タイムステップ方向のサイズ
in	<i>nalloc_size</i>	タイムステップ方向のアロケーションサイズ
in	<i>userInputError</i>	ユーザ入力誤差率
in, out	<i>evaluationError</i>	積算誤差率
in	<i>pFlowData</i>	基底データ (バイナリスワップ後)
in	<i>pPod_base_r</i>	基底データ
in	<i>pCoef_a</i>	係数データ
out	<i>finish_flg</i>	終了フラグ =0 not finished =1 finish

戻り値

終了コード

6.9.3.4 int CPfcCompressPod::GetCalculatedLayer (void)

POD 計算レイヤー数取得

戻り値

計算レイヤー数

6.9.3.5 PFC::E_PFC_ERRORCODE CPfcCompressPod::Init (const MPI_Comm comm, const std::string & outDirPath, const std::string & prefix, const double compressError, const int numRegion, const int regionSize, const int numStep, const int optFlags, double * pFlowData, const int myNumStep)

圧縮クラス 初期化

引数

in	<i>comm</i>	MPI コミュニケータ
in	<i>outDirPath</i>	フィールドデータ出力ディレクトリパス
in	<i>compressError</i>	誤差率 (%) ユーザ指定誤差
in	<i>numRegion</i>	領域の分割数
in	<i>regionSize</i>	分割領域サイズ (成分数含む)
in	<i>numStep</i>	タイムステップ数
in	<i>optFlags</i>	オプション
in	<i>pFlowData</i>	圧縮するデータ領域 (サイズ: regionSize*myNumStep)
in	<i>myNumStep</i>	自身が担当するステップ数

戻り値

終了コード

6.9.3.6 PFC::E_PFC_ERRORCODE CPfcCompressPod::Output (void) [protected]

POD ファイル出力 (基底ファイル, 係数ファイル)

戻り値

終了コード

6.9.3.7 **PFC::E_PFC_ERRORCODE** CPfcCompressPod::SwapBinaryData (double * *pPod_base_r*, int & *base_size*,
double *& *pFlowData*) [protected]

バイナリスワップ

引数

in	<i>pPod_base_r</i>	基底データ
in, out	<i>base_size</i>	基底データサイズ
in, out	<i>pFlowData</i>	基底データ（バイナリスワップ後）ポインタのアドレスも更新される

戻り値

終了コード

6.9.3.8 PFC::E_PFC_ERRORCODE CPfcCompressPod::WriteData (void)

圧縮 & 圧縮ファイル出力

戻り値

終了コード

6.9.3.9 PFC::E_PFC_ERRORCODE CPfcCompressPod::WritePodBaseFile (double * *pPod_base_r*) [protected]

6.9.3.10 PFC::E_PFC_ERRORCODE CPfcCompressPod::WritePodBaseFile_Debug (double * *pPod_base_r*) [protected]

6.9.3.11 PFC::E_PFC_ERRORCODE CPfcCompressPod::WritePodCoefFile (double * *pCoef_a*) [protected]

6.9.4 変数

6.9.4.1 int CPfcCompressPod::m_calculatedLayer [protected]

PfcCompressPod.h の 60 行で定義されています。

6.9.4.2 MPI_Comm CPfcCompressPod::m_comm [protected]

PfcCompressPod.h の 34 行で定義されています。

6.9.4.3 double CPfcCompressPod::m_compressError [protected]

PfcCompressPod.h の 37 行で定義されています。

6.9.4.4 double CPfcCompressPod::m_curEvaluateError [protected]

PfcCompressPod.h の 80 行で定義されています。

6.9.4.5 int CPfcCompressPod::m_curNumSize [protected]

PfcCompressPod.h の 79 行で定義されています。

6.9.4.6 int CPfcCompressPod::m_curPodBaseSize [protected]

PfcCompressPod.h の 78 行で定義されています。

6.9.4.7 `int CPfcCompressPod::m_layerNo` [protected]

PfcCompressPod.h の 77 行で定義されています。

6.9.4.8 `int CPfcCompressPod::m_maxLayer` [protected]

PfcCompressPod.h の 59 行で定義されています。

6.9.4.9 `int CPfcCompressPod::m_myIDinRegion` [protected]

PfcCompressPod.h の 73 行で定義されています。

6.9.4.10 `int CPfcCompressPod::m_myNumStep` [protected]

PfcCompressPod.h の 46 行で定義されています。

6.9.4.11 `int CPfcCompressPod::m_myRankID` [protected]

PfcCompressPod.h の 72 行で定義されています。

6.9.4.12 `int CPfcCompressPod::m_myStartStepPos` [protected]

PfcCompressPod.h の 74 行で定義されています。

6.9.4.13 `int CPfcCompressPod::m_numParallel` [protected]

PfcCompressPod.h の 56 行で定義されています。

6.9.4.14 `int CPfcCompressPod::m_numRank` [protected]

PfcCompressPod.h の 53 行で定義されています。

6.9.4.15 `int CPfcCompressPod::m_numRegion` [protected]

PfcCompressPod.h の 38 行で定義されています。

6.9.4.16 `int CPfcCompressPod::m_numStep` [protected]

PfcCompressPod.h の 40 行で定義されています。

6.9.4.17 `int CPfcCompressPod::m_optFlags` [protected]

PfcCompressPod.h の 41 行で定義されています。

6.9.4.18 `std::string CPfcCompressPod::m_outDirPath` [protected]

PfcCompressPod.h の 35 行で定義されています。

6.9.4.19 `double* CPfcCompressPod::m_pCoef_a` [protected]

PfcCompressPod.h の 84 行で定義されています。

6.9.4.20 `double* CPfcCompressPod::m_pFlowData` [protected]

PfcCompressPod.h の 43 行で定義されています。

6.9.4.21 `double* CPfcCompressPod::m_pPod_base_r` [protected]

PfcCompressPod.h の 83 行で定義されています。

6.9.4.22 `std::string CPfcCompressPod::m_prefix` [protected]

PfcCompressPod.h の 36 行で定義されています。

6.9.4.23 `int CPfcCompressPod::m_regionID` [protected]

PfcCompressPod.h の 64 行で定義されています。

6.9.4.24 `int CPfcCompressPod::m_regionMasterRankID` [protected]

PfcCompressPod.h の 65 行で定義されています。

6.9.4.25 `int CPfcCompressPod::m_regionMaxStep` [protected]

PfcCompressPod.h の 68 行で定義されています。

6.9.4.26 `int CPfcCompressPod::m_regionSize` [protected]

PfcCompressPod.h の 39 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:

- [PfcCompressPod.h](#)

6.10 クラス CPfcDivision

```
#include <PfcDivision.h>
```

Public メソッド

- [CPfcDivision \(\)](#)
- [~CPfcDivision \(\)](#)
- [PFC::E_PFC_ERRORCODE Read \(CPfcTextParser tpCntl\)](#)
read Rank(proc.dfi)
- [PFC::E_PFC_ERRORCODE CheckReadRegion](#) (const int head[3], const int tail[3], vector< int > &read-RegionIdList)
読み込み領域リストの取得
- [PFC::E_PFC_ERRORCODE CheckReadRegion](#) (const int ix, const int iy, const int iz, int ®ionID)

読み込み領域IDの取得

- `PFC::E_PFC_ERRORCODE Write` (FILE *fp, const unsigned tab)
PFC ファイル:Division を出力する

Public 変数

- int `m_numRegion`
- vector< `CPfcRegion` > `m_regionList`
- int ** `m_MinMax`
- int * `m_MinMax_data`

Protected メソッド

- void `CreateMinMax` (void)
MinMax 領域設定

6.10.1 説明

proc.pfc ファイルのDivision

PfcDivision.h の 23 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 CPfcDivision::CPfcDivision ()

コンストラクタ

6.10.2.2 CPfcDivision::~~CPfcDivision ()

デストラクタ

6.10.3 関数

6.10.3.1 PFC::E_PFC_ERRORCODE CPfcDivision::CheckReadRegion (const int *head*[3], const int *tail*[3], vector< int > & *readRegionIdList*)

読み込み領域リストの取得

引数

in	<i>head</i>	HeadIndex
in	<i>tail</i>	TailIndex
out	<i>regionIdList</i>	読み込み領域ID リスト

戻り値

error code

6.10.3.2 PFC::E_PFC_ERRORCODE CPfcDivision::CheckReadRegion (const int *ix*, const int *iy*, const int *iz*, int & *regionID*)

読み込み領域IDの取得

引数

in	<i>ix</i>	X インデックス
in	<i>iy</i>	Y インデックス
in	<i>iz</i>	Z インデックス
out	<i>regionID</i>	読み込み領域ID

戻り値

error code

6.10.3.3 void CPfcDivision::CreateMinMax (void) [protected]

MinMax 領域設定

戻り値

error code

6.10.3.4 PFC::E_PFC_ERRORCODE CPfcDivision::Read (CPfcTextParser tpCntl)

read Rank(proc.dfi)

引数

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

戻り値

error code

6.10.3.5 PFC::E_PFC_ERRORCODE CPfcDivision::Write (FILE * fp, const unsigned tab)

PFC ファイル:Division 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

true:出力成功 false:出力失敗

6.10.4 変数

6.10.4.1 int** CPfcDivision::m_MinMax

PfcDivision.h の 30 行で定義されています。

6.10.4.2 int* CPfcDivision::m_MinMax_data

PfcDivision.h の 34 行で定義されています。

6.10.4.3 int CPfcDivision::m_numRegion

PfcDivision.h の 27 行で定義されています。

6.10.4.4 vector<CPfcRegion> CPfcDivision::m_regionList

PfcDivision.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:

- [PfcDivision.h](#)

6.11 クラス CPfcDomain

```
#include <PfcDomain.h>
```

Public メソッド

- [CPfcDomain](#) ()
- [CPfcDomain](#) (const double *globalOrigin, const double *globalRegion, const int *globalVoxel, const int *globalDivision)
コンストラクタ
- [~CPfcDomain](#) ()
- [PFC::E_PFC_ERRORCODE Read](#) (CPfcTextParser tpCntl)
read Domain(proc.dfi)
- [PFC::E_PFC_ERRORCODE Write](#) (FILE *fp, const unsigned tab)
DFI ファイル:Domain を出力する

Public 変数

- double [m_globalOrigin](#) [3]
計算空間の起点座標
- double [m_globalRegion](#) [3]
計算空間の各軸方向の長さ
- int [m_globalVoxel](#) [3]
計算領域全体のボクセル数
- int [m_globalDivision](#) [3]
計算領域の分割数

6.11.1 説明

proc.pfc ファイルの Domain

PfcDomain.h の 21 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 CPfcDomain::CPfcDomain ()

コンストラクタ

6.11.2.2 CPfcDomain::CPfcDomain (const double * *globalOrigin*, const double * *globalRegion*, const int * *globalVoxel*, const int * *globalDivision*)

コンストラクタ

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

6.11.2.3 CPfcDomain::~CPfcDomain ()

デストラクタ

6.11.3 関数

6.11.3.1 PFC::E_PFC_ERRORCODE CPfcDomain::Read (CPfcTextParser *tpCntl*)

read Domain(proc.dfi)

引数

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

戻り値

error code

6.11.3.2 PFC::E_PFC_ERRORCODE CPfcDomain::Write (FILE * *fp*, const unsigned *tab*)

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

true:出力成功 false:出力失敗

6.11.4 変数

6.11.4.1 int CPfcDomain::m_globalDivision[3]

計算領域の分割数

PfcDomain.h の 28 行で定義されています。

6.11.4.2 double CPfcDomain::m_globalOrigin[3]

計算空間の起点座標

PfcDomain.h の 25 行で定義されています。

6.11.4.3 double CPfcDomain::m_globalRegion[3]

計算空間の各軸方向の長さ

PfcDomain.h の 26 行で定義されています。

6.11.4.4 int CPfcDomain::m_globalVoxel[3]

計算領域全体のボクセル数

PfcDomain.h の 27 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:

- [PfcDomain.h](#)

6.12 クラス CPfcFileInfo

```
#include <PfcFileInfo.h>
```

Public メソッド

- [CPfcFileInfo \(\)](#)
- [CPfcFileInfo \(const std::string &dirPath, const std::string &prefix, const std::string &fileFormat, const int guideCell, const PFC::E_PFC_DTYPE dataType, const PFC::E_PFC_ENDIANTYPE endianType, const PFC::E_PFC_ARRAYSHAPE arrayShape, const int numComponent\)](#)
コンストラクタ
- [~CPfcFileInfo \(\)](#)
- [PFC::E_PFC_ERRORCODE Read \(CPfcTextParser tpCntl\)](#)
read FileInfo(index.pfc)
- [PFC::E_PFC_ERRORCODE Write \(FILE *fp, const unsigned tab\)](#)
write FileInfo(index.pfc)

Public 変数

- std::string [m_dirPath](#)
- std::string [m_prefix](#)
- std::string [m_fileFormat](#)
ファイルフォーマット "pod"
- int [m_guideCell](#)
仮想セルの数 (0 固定)
- [PFC::E_PFC_DTYPE m_dataType](#)
配列のデータタイプ (PFC::E_PFC_FLOAT64)
- [PFC::E_PFC_ENDIANTYPE m_endianType](#)
エンディアンタイプ E_PFC_LITTLE / E_PFC_BIG
- [PFC::E_PFC_ARRAYSHAPE m_arrayShape](#)
- int [m_numComponent](#)

6.12.1 説明

index.pfc ファイルの FileInfo

PfcFileInfo.h の 22 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 CPfcFileInfo::CPfcFileInfo ()

コンストラクタ

6.12.2.2 CPfcFileInfo::CPfcFileInfo (const std::string & *dirPath*, const std::string & *prefix*, const std::string & *fileFormat*, const int *guideCell*, const PFC::E_PFC_DTYPE *dataType*, const PFC::E_PFC_ENDIANTYPE *endianType*, const PFC::E_PFC_ARRAYSHAPE *arrayShape*, const int *numComponent*)

コンストラクタ

引数

in	<i>dirPath</i>	フィールドデータの存在するディレクトリパス
in	<i>prefix</i>	ベースファイル名（属性名）
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>endianType</i>	エンディアンタイプ
in	<i>arrayShape</i>	配列形状
in	<i>numComponent</i>	成分数

6.12.2.3 CPfcFileInfo::~~CPfcFileInfo ()

デストラクタ

6.12.3 関数

6.12.3.1 PFC::E_PFC_ERRORCODE CPfcFileInfo::Read (CPfcTextParser *tpCntl*)

read FileInfo(index.pfc)

引数

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

戻り値

error code

6.12.3.2 PFC::E_PFC_ERRORCODE CPfcFileInfo::Write (FILE * *fp*, const unsigned *tab*)

write FileInfo(index.pfc)

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 PFC::E_PFC_ARRAYSHAPE CPfcFileInfo::m_arrayShape

PfcFileInfo.h の 33 行で定義されています。

6.12.4.2 PFC::E_PFC_DTYPE CPfcFileInfo::m_dataType

配列のデータタイプ (PFC::E_PFC_FLOAT64)

PfcFileInfo.h の 31 行で定義されています。

6.12.4.3 std::string CPfcFileInfo::m_dirPath

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

PfcFileInfo.h の 26 行で定義されています。

6.12.4.4 PFC::E_PFC_ENDIANTYPE CPfcFileInfo::m_endianType

エンディアンタイプ E_PFC_LITTLE / E_PFC_BIG

PfcFileInfo.h の 32 行で定義されています。

6.12.4.5 std::string CPfcFileInfo::m_fileFormat

ファイルフォーマット "pod"

PfcFileInfo.h の 29 行で定義されています。

6.12.4.6 int CPfcFileInfo::m_guideCell

仮想セルの数 (0 固定)

PfcFileInfo.h の 30 行で定義されています。

6.12.4.7 int CPfcFileInfo::m_numComponent

PfcFileInfo.h の 34 行で定義されています。

6.12.4.8 std::string CPfcFileInfo::m_prefix

PfcFileInfo.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:

- [PfcFileInfo.h](#)

6.13 クラス CPfcFilePath

```
#include <PfcFilePath.h>
```

Public メソッド

- [CPfcFilePath](#) ()
- [CPfcFilePath](#) (const std::string &dfiPath, const std::string &procPath)
コンストラクタ
- [~CPfcFilePath](#) ()
- [PFC::E_PFC_ERRORCODE Read](#) ([CPfcTextParser](#) tpCntl)
read FilePath(index.pfc)
- [PFC::E_PFC_ERRORCODE Write](#) (FILE *fp, const unsigned tab)
write FilePath(index.pfc)

Public 変数

- std::string [m_dfiPath](#)
- std::string [m_procPath](#)

6.13.1 説明

index.pfc ファイルの FilePath

PfcFilePath.h の 23 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 CPfcFilePath::CPfcFilePath ()

コンストラクタ

6.13.2.2 CPfcFilePath::CPfcFilePath (const std::string & dfiPath, const std::string & procPath)

コンストラクタ

引数

in	<i>dfiPath</i>	CIO DFI ファイルパス
in	<i>procPath</i>	proc.pfc ファイルパス

6.13.2.3 CPfcFilePath::~~CPfcFilePath ()

デストラクタ

6.13.3 関数

6.13.3.1 PFC::E_PFC_ERRORCODE CPfcFilePath::Read (CPfcTextParser tpCntl)

read FilePath(index.pfc)

引数

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

戻り値

error code

6.13.3.2 PFC::E_PFC_ERRORCODE CPfcFilePath::Write (FILE * *fp*, const unsigned *tab*)

write FilePath(index.pfc)

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

6.13.4 変数

6.13.4.1 std::string CPfcFilePath::m_dfiPath

PfcFilePath.h の 27 行で定義されています。

6.13.4.2 std::string CPfcFilePath::m_procPath

PfcFilePath.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:

- [PfcFilePath.h](#)

6.14 クラス CPfcFunction

```
#include <PfcFunction.h>
```

Static Public メソッド

- static [PFC::E_PFC_ERRORCODE CalcHeadTail](#) (const int voxel[3], const int div[3], const int regionID, int head[3], int tail[3])
領域範囲 (Head/Tail) 計算 ex) 95 を 10 で分割する場合 -> 10 10 10 10 10 9 9 9 9 9
- static [PFC::E_PFC_ERRORCODE CalcHeadTail_block](#) (const int voxel[3], const int div[3], const int regionID, int head[3], int tail[3])
領域範囲 (Head/Tail) 計算 (単純ブロック分割) ex) 95 を 10 で分割する場合 -> 10 10 10 10 10 10 10 10 10 5
- static void [CopyData](#) (double *to, const int nsize_to[3], const int offset_to[3], const double *from, const int nsize_from[3], const int offset_from[3], const int nsize_copy[3], const int num_component, [PFC::E_PFC_ARRAYSHAPE](#) array_shape)
データコピー (範囲指定)
- static [PFC::E_PFC_ERRORCODE AndMinMax](#) (const int head1[3], const int tail1[3], const int head2[3], const int tail2[3], int lap_offset1[3], int lap_offset2[3], int lap_size[3])

- Min/Max(Head/Tail) 領域のAND (重なり) 取得*
 - static [PFC::E_PFC_ERRORCODE CheckLapMinMax](#) (const int head1[3], const int tail1[3], const int head2[3], const int tail2[3])
- Min/Max(Head/Tail) 領域の重なりチェック*
 - static [PFC::E_PFC_ERRORCODE CheckPntInMinMax](#) (const int head[3], const int tail[3], const int ix, const int iy, const int iz)
- 点のMin/Max(Head/Tail) 領域内チェック*
 - static void [Alloc2D](#) (const int size1, const int size2, int *&data, int **&index)
- 2次元固定長のアロケーション (整数用)*
 - static void [Alloc2D](#) (const int size1, const int size2, double *&data, double **&index)
- 2次元固定長のアロケーション (倍精度用)*
 - static int [GetPodMaxLayer](#) (const int numStep)
- POD Max Layer 数取得*
 - static int [GetPodParallel](#) (const int numStep)
- POD 並列数取得*
 - static [PFC::E_PFC_ERRORCODE GetPodStepInfo](#) (const int numStep, const int id, int &numStepInId, int &startStepPos, int &maxStep)
- POD 担当するステップ数 & ステップ開始位置取得*
 - static [PFC::E_PFC_ERRORCODE GetPodRegionID](#) (const int rankID, const int numStep, int ®ionID, int &idInRegion, int &masterRankID)
- POD の regionID (領域ID) 取得*
 - static int [GetPodBaseIndex](#) (const int numStep, const int numParallel, const int numCalculatedLayer, const int stepID)
- POD の base index 取得 base index: pod base ファイルを出力したランク番号*

6.14.1 説明

[PFC](#) POD function class

PfcFunction.h の 33 行で定義されています。

6.14.2 関数

6.14.2.1 static void CPfcFunction::Alloc2D (const int *size1*, const int *size2*, int *& *data*, int **& *index*) [static]

2次元固定長のアロケーション (整数用)

引数

in	<i>size1</i>	1次元目サイズ
in	<i>size2</i>	2次元目サイズ
out	<i>data</i>	アロケーションされた領域 (1次元連続領域) サイズ: <i>size2</i> * <i>size1</i> ポインタの参照型であることに注意 使用後, delete [] <i>data</i>
out	<i>index</i>	<i>data</i> へのアクセス用インデックス [<i>size1</i>][<i>size2</i>] ダブルポインタの参照型であることに注意 使用後, delete [] <i>index</i>

戻り値

なし

6.14.2.2 static void CPfcFunction::Alloc2D (const int *size1*, const int *size2*, double *& *data*, double **& *index*) [static]

2次元固定長のアロケーション (倍精度用)

引数

in	<i>size1</i>	1次元目サイズ
in	<i>size2</i>	2次元目サイズ
out	<i>data</i>	アロケーションされた領域 (1次元連続領域) サイズ: <i>size2</i> * <i>size1</i> ポインタの参照型であることに注意 使用後, delete [] <i>data</i>
out	<i>index</i>	<i>data</i> へのアクセス用インデックス [<i>size1</i>][<i>size2</i>] ダブルポインタの参照型であることに注意 使用後, delete [] <i>index</i>

戻り値

なし

6.14.2.3 static PFC::E_PFC_ERRORCODE CPfcFunction::AndMinMax (const int *head1*[3], const int *tail1*[3], const int *head2*[3], const int *tail2*[3], int *lap_offset1*[3], int *lap_offset2*[3], int *lap_size*[3]) [static]

Min/Max(Head/Tail) 領域 の A N D (重なり) 取得

引数

in	<i>head1</i>	計算領域 1 の開始位置
in	<i>tail1</i>	計算領域 1 の終了位置
in	<i>head2</i>	計算領域 2 の開始位置
in	<i>tail2</i>	計算領域 2 の終了位置
out	<i>lap_offset1</i>	計算領域 1 の重なり区間までのオフセット
out	<i>lap_offset2</i>	計算領域 2 の重なり区間までのオフセット
out	<i>lap_size</i>	A N D (重なり) 領域のサイズ

戻り値

終了コード E_PFC_SUCCESS 重なりあり E_PFC_ERROR 重なりなし

6.14.2.4 static PFC::E_PFC_ERRORCODE CPfcFunction::CalcHeadTail (const int *voxel*[3], const int *div*[3], const int *regionID*, int *head*[3], int *tail*[3]) [static]

領域範囲 (Head/Tail) 計算 ex) 95 を 10 で分割する場合 -> 10 10 10 10 10 9 9 9 9 9

引数

in	<i>voxel</i>	ボクセル数
in	<i>div</i>	計算空間の領域分割数
in	<i>regionID</i>	領域ID (1次元) : 0 ~ (<i>div</i> [0]* <i>div</i> [1]* <i>div</i> [2]-1)
out	<i>head</i>	計算領域の開始位置 (1 起点)
out	<i>tail</i>	計算領域の終了位置 (1 起点)

戻り値

終了コード

6.14.2.5 static PFC::E_PFC_ERRORCODE CPfcFunction::CalcHeadTail_block (const int *voxel*[3], const int *div*[3], const int *regionID*, int *head*[3], int *tail*[3]) [static]

領域範囲 (Head/Tail) 計算 (単純ブロック分割) ex) 95 を 10 で分割する場合 -> 10 10 10 10 10 10 10 10 10 5

引数

in	<i>voxel</i>	ボクセル数
in	<i>div</i>	計算空間の領域分割数
in	<i>regionID</i>	領域ID (1 次元) : 0 ~ (div[0]*div[1]*div[2]-1)
out	<i>head</i>	計算領域の開始位置 (1 起点)
out	<i>tail</i>	計算領域の終了位置 (1 起点)

戻り値

終了コード

6.14.2.6 `static PFC::E_PFC_ERRORCODE CPfcFunction::CheckLapMinMax (const int head1[3], const int tail1[3], const int head2[3], const int tail2[3]) [static]`

Min/Max(Head/Tail) 領域 の重なりチェック

引数

in	<i>head1</i>	計算領域 1 の開始位置
in	<i>tail1</i>	計算領域 1 の終了位置
in	<i>head2</i>	計算領域 2 の開始位置
in	<i>tail2</i>	計算領域 2 の終了位置

戻り値

終了コード E_PFC_SUCCESS 重なりあり E_PFC_ERROR 重なりなし

6.14.2.7 `static PFC::E_PFC_ERRORCODE CPfcFunction::CheckPntInMinMax (const int head[3], const int tail[3], const int ix, const int iy, const int iz) [static]`

点のMin/Max(Head/Tail) 領域内チェック

引数

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>ix</i>	点のX 座標
in	<i>iy</i>	点のY 座標
in	<i>iz</i>	点のZ 座標

戻り値

終了コード E_PFC_SUCCESS 領域内 E_PFC_ERROR 領域外

6.14.2.8 `static void CPfcFunction::CopyData (double * to, const int nsize_to[3], const int offset_to[3], const double * from, const int nsize_from[3], const int offset_from[3], const int nsize_copy[3], const int num_component, PFC::E_PFC_ARRAYSHAPE array_shape) [static]`

データコピー (範囲指定)

引数

out	<i>to</i>	コピー先配列のアドレス
in	<i>nsizeto</i>	コピー先配列のサイズ
in	<i>offset_to</i>	コピー先の配列先頭からのオフセット (0 ~)
in	<i>from</i>	コピー元配列のアドレス
in	<i>nsizefrom</i>	コピー元配列のサイズ
in	<i>offset_from</i>	コピー元の配列先頭からのオフセット (0 ~)
in	<i>nsizecopy</i>	コピーサイズ
in	<i>num_component</i>	成分数
in	<i>array_shape</i>	配列形状 E_PFC_IJKN / E_PFC_NIJK

戻り値

終了コード

6.14.2.9 static int CPfcFunction::GetPodBaseIndex (const int *numStep*, const int *numParallel*, const int *numCalculatedLayer*, const int *stepID*) [static]

POD の base index 取得 base index: pod base ファイルを出力したランク番号

引数

in	<i>numStep</i>	ステップ数
in	<i>numParallel</i>	並列数
in	<i>numCalculated-Layer</i>	計算レイヤー数
in	<i>stepID</i>	ステップID (0 起点)

戻り値

base index

6.14.2.10 static int CPfcFunction::GetPodMaxLayer (const int *numStep*) [static]

POD Max Layer 数取得

引数

in	<i>numStep</i>	ステップ数
----	----------------	-------

戻り値

Max Layer 数

6.14.2.11 static int CPfcFunction::GetPodParallel (const int *numStep*) [static]

POD 並列数取得

引数

in	<i>numStep</i>	ステップ数
----	----------------	-------

戻り値

並列数

6.14.2.12 **static PFC::E_PFC_ERRORCODE CPfcFunction::GetPodRegionID (const int *rankID*, const int *numStep*, int & *regionID*, int & *idInRegion*, int & *masterRankID*)** [static]

POD の regionID (領域ID) 取得

引数

in	<i>rankID</i>	ランクID
in	<i>numStep</i>	ステップ数
out	<i>regionID</i>	領域ID (0 起点)
out	<i>idInRegion</i>	領域内のID (0 起点)
out	<i>masterRankID</i>	分割領域内のマスターランクID (0 起点)

戻り値

終了コード

6.14.2.13 **static PFC::E_PFC_ERRORCODE CPfcFunction::GetPodStepInfo (const int *numStep*, const int *id*, int & *numStepInId*, int & *startStepPos*, int & *maxStep*)** [static]

POD 担当するステップ数 & ステップ開始位置取得

引数

in	<i>numStep</i>	ステップ数
in	<i>id</i>	担当ID (並列内のID)
out	<i>numStepInId</i>	担当するステップ数
out	<i>startStepPos</i>	担当する開始ステップ位置 (0 ~)
out	<i>maxStep</i>	領域内の各ランクが担当する最大ステップ数

戻り値

終了コード

このクラスの説明は次のファイルから生成されました:

- [PfcFunction.h](#)

6.15 クラス CPfcMpiCom

```
#include <PfcMpiCom.h>
```

Static Public メソッド

- **static PFC::E_PFC_ERRORCODE GatherDataDouble (MPI_Comm comm, const int masterRankID, const int numRank, const int myRankID, const int nSize, double *localData, double *gatherData)**

データギャザー *Double*

- static `PFC::E_PFC_ERRORCODE GatherDataInt (MPI_Comm comm, const int masterRankID, const int numRank, const int myRankID, const int nSize, int *localData, int *gatherData)`

データギャザー *Int*

- static `PFC::E_PFC_ERRORCODE GatherV_DataDouble (MPI_Comm comm, const int masterRankID, const int numRank, const int myRankID, const int nSize, double *localData, double *gatherData, int *nSizeRanks)`

データギャザー *Double* 各ランク違うサイズ

- static `PFC::E_PFC_ERRORCODE GetMaxInt (MPI_Comm comm, const int masterRankID, const int numRank, const int myRankID, int val, int &maxVal)`

MAX 値取得 *Int*.

6.15.1 説明

PFC MPI communication class

PfcMpiCom.h の 24 行で定義されています。

6.15.2 関数

- 6.15.2.1 static `PFC::E_PFC_ERRORCODE CPfcMpiCom::GatherDataDouble (MPI_Comm comm, const int masterRankID, const int numRank, const int myRankID, const int nSize, double * localData, double * gatherData)` [static]

データギャザー *Double*

引数

in	<i>comm</i>	MPI コミュニケータ
in	<i>masterRankID</i>	マスターランクID (データ収集ランク) 収集対象ランクの最小ランクNo
in	<i>numRank</i>	収集対象ランク数 (masterRankID) ~ (masterRankID+numRank-1)
in	<i>myRankID</i>	自身のランクID
in	<i>nSize</i>	データサイズ (全ランク同一サイズ)
in	<i>localData</i>	ローカルデータ
out	<i>gatherData</i>	収集データ (マスターランクのみ設定)

戻り値

終了コード

- 6.15.2.2 static `PFC::E_PFC_ERRORCODE CPfcMpiCom::GatherDataInt (MPI_Comm comm, const int masterRankID, const int numRank, const int myRankID, const int nSize, int * localData, int * gatherData)` [static]

データギャザー *Int*

引数

in	<i>comm</i>	MPI コミュニケータ
in	<i>masterRankID</i>	マスターランクID (データ収集ランク) 収集対象ランクの最小ランクNo
in	<i>numRank</i>	収集対象ランク数 (masterRankID) ~ (masterRankID+numRank-1)
in	<i>myRankID</i>	自身のランクID
in	<i>nSize</i>	データサイズ (全ランク同一サイズ)

in	<i>localData</i>	ローカルデータ
out	<i>gatherData</i>	収集データ (マスターランクのみ設定)

戻り値

終了コード

6.15.2.3 `static PFC::E_PFC_ERRORCODE CPfcMpiCom::GatherV_DataDouble (MPI_Comm comm, const int masterRankID, const int numRank, const int myRankID, const int nSize, double * localData, double * gatherData, int * nSizeRanks) [static]`

データギャザー Double 各ランク違うサイズ

引数

in	<i>comm</i>	MPI コミュニケーター
in	<i>masterRankID</i>	マスターランクID (データ収集ランク) 収集対象ランクの最小ランクNo
in	<i>numRank</i>	収集対象ランク数 (masterRankID) ~ (masterRankID+numRank-1)
in	<i>myRankID</i>	自身のランクID
in	<i>nSize</i>	データサイズ (全ランク同一サイズ)
in	<i>localData</i>	ローカルデータ
out	<i>gatherData</i>	収集データ (マスターランクのみ設定)
out	<i>nSizeRanks</i>	各ランクのサイズ (マスターランクのみ設定)

戻り値

終了コード

6.15.2.4 `static PFC::E_PFC_ERRORCODE CPfcMpiCom::GetMaxInt (MPI_Comm comm, const int masterRankID, const int numRank, const int myRankID, int val, int & maxVal) [static]`

MAX 値取得 Int.

引数

in	<i>comm</i>	MPI コミュニケーター
in	<i>masterRankID</i>	マスターランクID (データ収集ランク) 収集対象ランクの最小ランクNo
in	<i>numRank</i>	収集対象ランク数 (masterRankID) ~ (masterRankID+numRank-1)
in	<i>myRankID</i>	自身のランクID
in	<i>val</i>	自ランクの値
out	<i>maxVal</i>	max 値 (マスターランクのみ設定)

戻り値

終了コード

このクラスの説明は次のファイルから生成されました:

- [PfcMpiCom.h](#)

6.16 クラス CPfcPodFile

```
#include <PfcPodFile.h>
```

Static Public メソッド

- static [PFC::E_PFC_ERRORCODE WriteBaseFile](#) (const std::string &outDirPath, const std::string &prefix, const int regionID, const bool bSingle, const int numStep, const int numParallel, const int numCalculatedLayer, const int *nSizes, double *pPod_base)
基底ファイル出力処理
- static [PFC::E_PFC_ERRORCODE WriteBaseFile_Debug](#) (const std::string &outDirPath, const std::string &prefix, const int regionID, const int idInRegion, const int layerNo, const int numStep, const int numParallel, const int numCalculatedLayer, const int nSize, double *pPod_base)
基底ファイル出力処理 (デバッグ用) 途中のレイヤーの基底ファイルを出力する
- static [PFC::E_PFC_ERRORCODE ReadBaseFile](#) (const std::string &dirPath, const std::string &prefix, const int regionID, const bool bSingle, int &numStep, int &numParallel, int &numCalculatedLayer, int *&pSizes, double *&pPod_base, double **&pIndex)
基底ファイル入力 (読み込み) 処理 *open/read/close*
- static [PFC::E_PFC_ERRORCODE OpenBaseFile](#) (const std::string &dirPath, const std::string &prefix, const int regionID, const bool bSingle, ifstream &if_)
基底ファイルOpen 処理 (*read mode*)
- static void [CloseBaseFile](#) (ifstream &if_)
基底ファイルClose 処理
- static [PFC::E_PFC_ERRORCODE ReadBaseFileHeader](#) (ifstream &if_, int &numStep, int &numParallel, int &numCalculatedLayer, int *&pSizes, int &total_header_bsize, int &total_base_size, bool &endian_chg)
基底ファイルヘッダ部読み込み処理
- static [PFC::E_PFC_ERRORCODE ReadBaseFile1Elm](#) (ifstream &if_, bool endian_chg, int total_header_bsize, int index, double &v)
基底データ (1 成分) 読み込み処理
- static [PFC::E_PFC_ERRORCODE ReadBaseFile_Debug](#) (const std::string &dirPath, const std::string &prefix, const int regionID, const int idInRegion, const int layerNo, int &numStep, int &numParallel, int &numCalculatedLayer, int &nSize, double *pPod_base)
基底ファイル入力 (読み込み) 処理 (デバッグ用) デバッグ用の基底ファイルを読み込む
- static [PFC::E_PFC_ERRORCODE WriteCoefFile](#) (const std::string &outDirPath, const std::string &prefix, const int regionID, const bool bSingle, const int numStep, const int numCoef, const int numCalculatedLayer, double *pCoef_a_layer)
係数ファイル出力処理
- static [PFC::E_PFC_ERRORCODE ReadCoefFile](#) (const std::string &dirPath, const std::string &prefix, const int regionID, const bool bSingle, int &numStep, int &numCoef, int &numCalculatedLayer, double *&pCoef_a_layer, double **&pIndex)
係数ファイル入力 (読み込み) 処理

6.16.1 説明

[PFC](#) POD File class

PfcPodFile.h の 33 行で定義されています。

6.16.2 関数

6.16.2.1 static void CPfcPodFile::CloseBaseFile (ifstream & if_) [static]

基底ファイルClose 処理

引数

in	<i>ifs</i>	入力ファイルストリーム
----	------------	-------------

戻り値

なし

6.16.2.2 static PFC::E_PFC_ERRORCODE CPfcPodFile::OpenBaseFile (const std::string & *dirPath*, const std::string & *prefix*, const int *regionID*, const bool *bSingle*, ifstream & *ifs*) [static]

基底ファイルOpen 処理 (read mode)

引数

in	<i>dirPath</i>	入力ディレクトリ
in	<i>prefix</i>	属性名
in	<i>regionID</i>	領域ID
in	<i>bSingle</i>	単一領域フラグ
out	<i>ifs</i>	入力ファイルストリーム

戻り値

終了コード

6.16.2.3 static PFC::E_PFC_ERRORCODE CPfcPodFile::ReadBaseFile (const std::string & *dirPath*, const std::string & *prefix*, const int *regionID*, const bool *bSingle*, int & *numStep*, int & *numParallel*, int & *numCalculatedLayer*, int *& *pSizes*, double *& *pPod_base*, double **& *pIndex*) [static]

基底ファイル入力 (読み込み) 処理 open/read/close

引数

in	<i>dirPath</i>	入力ディレクトリ
in	<i>prefix</i>	属性名
in	<i>regionID</i>	領域ID
in	<i>bSingle</i>	単一領域フラグ
	[out]	numStep タイムステップ数
out	<i>numParallel</i>	圧縮時並列数
out	<i>numCalculatedLayer</i>	圧縮時計算レイヤー数
out	<i>pSizes</i>	各ランク内の要素数 使用後, delete [] pSizes 要
out	<i>pPod_base</i>	基底データ 使用後, delete [] pPod_base 要
out	<i>pIndex</i>	各ランク内の基底データへのポインタ 使用後, delete [] pIndex 要

戻り値

終了コード

6.16.2.4 static PFC::E_PFC_ERRORCODE CPfcPodFile::ReadBaseFile1Elm (ifstream & *ifs*, bool *endian_chg*, int *total_header_bsize*, int *index*, double & *v*) [static]

基底データ (1 成分) 読み込み処理

引数

in	<i>ifs</i>	入力ファイルストリーム
in	<i>endian_chg</i>	エンディアン変換フラグ
in	<i>total_header_ - bsize</i>	ヘッダ部のトータルサイズ (byte)
in	<i>index</i> , 基底データ部の読み込みインデックス	
out	<i>v</i>	1 成分データ

戻り値

終了コード

6.16.2.5 static PFC::E_PFC_ERRORCODE CPfcPodFile::ReadBaseFile_Debug (const std::string & *dirPath*, const std::string & *prefix*, const int *regionID*, const int *idInRegion*, const int *layerNo*, int & *numStep*, int & *numParallel*, int & *numCalculatedLayer*, int & *nSize*, double *& *pPod_base*) [static]

基底ファイル入力（読み込み）処理（デバッグ用）デバッグ用の基底ファイルを読み込む

引数

in	<i>dirPath</i>	入力ディレクトリ
in	<i>prefix</i>	属性名
in	<i>regionID</i>	領域ID
in	<i>layerNo</i>	レイヤーNo（0 起点）
	[out]	numStep タイムステップ数
out	<i>numParallel</i>	圧縮時並列数
out	<i>numCalculated- Layer</i>	圧縮時計算レイヤー数
out	<i>nSize</i>	各ランクの要素数
out	<i>pPod_base</i>	基底データ 使用後, delete [] pPod_base 要

戻り値

終了コード

6.16.2.6 static PFC::E_PFC_ERRORCODE CPfcPodFile::ReadBaseFileHeader (ifstream & *ifs*, int & *numStep*, int & *numParallel*, int & *numCalculatedLayer*, int *& *pSizes*, int & *total_header_bsize*, int & *total_base_size*, bool & *endian_chg*) [static]

基底ファイルヘッダ部読み込み処理

引数

in	<i>ifs</i>	入力ファイルストリーム
out	<i>numStep</i> , タイムステップ数	
out	<i>numParallel</i> , 圧縮時並列数	

out	<i>numCalculated-Layer</i>	圧縮時計算レイヤー数
out	<i>pSizes</i>	各ランク内の要素数 ポインタの参照型であることに注意 使用後, delete [] pSizes 必要
out	<i>total_header_bsize</i>	ヘッダ部のトータルサイズ (byte)
out	<i>total_base_size</i>	基底データのトータルサイズ
out	<i>endian_chg</i>	エンディアン変換フラグ

戻り値

終了コード

6.16.2.7 static PFC::E_PFC_ERRORCODE CPfcPodFile::ReadCoefFile (const std::string & *dirPath*, const std::string & *prefix*, const int *regionID*, const bool *bSingle*, int & *numStep*, int & *numCoef*, int & *numCalculatedLayer*, double *& *pCoef_a_layer*, double **& *pIndex*) [static]

係数ファイル入力 (読み込み) 処理

引数

in	<i>dirPath</i>	入力ディレクトリ
in	<i>prefix</i>	属性名
in	<i>regionID</i>	領域ID
out	<i>numStep</i>	タイムステップ数 (Layer0 係数出力数)
out	<i>numCoef</i>	係数出力数 (Layer1 以降)
out	<i>numCalculated-Layer</i>	圧縮時計算レイヤー数
out	<i>pCoef_a_layer</i>	係数データ 使用後, delete [] pCoef_a_layer 要
out	<i>pIndex</i>	各ランク内の係数データへのポインタ 使用後, delete [] pIndex 要

戻り値

終了コード

6.16.2.8 static PFC::E_PFC_ERRORCODE CPfcPodFile::WriteBaseFile (const std::string & *outDirPath*, const std::string & *prefix*, const int *regionID*, const bool *bSingle*, const int *numStep*, const int *numParallel*, const int *numCalculatedLayer*, const int * *nSizes*, double * *pPod_base*) [static]

基底ファイル出力処理

引数

in	<i>outDirPath</i>	出力ディレクトリ
in	<i>prefix</i>	属性名
in	<i>regionID</i>	領域ID
in	<i>bSingle</i>	単一領域フラグ
in	<i>numStep</i>	タイムステップ数
in	<i>numParallel</i>	圧縮時並列数
in	<i>numCalculated-Layer</i>	圧縮時計算レイヤー数

in	<i>nSizes</i>	各ランク内の要素数
in	<i>pPod_base</i>	基底データ

戻り値

終了コード

6.16.2.9 static PFC::E_PFC_ERRORCODE CPfcPodFile::WriteBaseFile_Debug (const std::string & *outDirPath*, const std::string & *prefix*, const int *regionID*, const int *idInRregion*, const int *layerNo*, const int *numStep*, const int *numParallel*, const int *numCalculatedLayer*, const int *nSize*, double * *pPod_base*) [static]

基底ファイル出力処理（デバッグ用）途中のレイヤーの基底ファイルを出力する

引数

in	<i>outDirPath</i>	出力ディレクトリ
in	<i>prefix</i>	属性名
in	<i>regionID</i>	領域ID
in	<i>layerNo</i>	レイヤーNo（0 起点）
in	<i>numStep</i>	タイムステップ数
in	<i>numParallel</i>	圧縮時並列数
in	<i>numCalculated-Layer</i>	圧縮時計算レイヤー数
in	<i>nSize</i>	各ランクの要素数
in	<i>pPod_base</i>	各ランクの基底データ

戻り値

終了コード

6.16.2.10 static PFC::E_PFC_ERRORCODE CPfcPodFile::WriteCoefFile (const std::string & *outDirPath*, const std::string & *prefix*, const int *regionID*, const bool *bSingle*, const int *numStep*, const int *numCoef*, const int *numCalculatedLayer*, double * *pCoef_a_layer*) [static]

係数ファイル出力処理

引数

in	<i>outDirPath</i>	出力ディレクトリ
in	<i>prefix</i>	属性名
in	<i>regionID</i>	領域ID
in	<i>bSingle</i>	単一領域フラグ
in	<i>numStep</i>	タイムステップ数
in	<i>numCoef</i>	係数出力数（Layer1 以降）
in	<i>numCalculated-Layer</i>	圧縮時計算レイヤー数
in	<i>pCoef_a_layer</i>	係数データ

戻り値

終了コード

このクラスの説明は次のファイルから生成されました:

- [PfcPodFile.h](#)

6.17 クラス CPfcRegion

```
#include <PfcRegion.h>
```

Public メソッド

- [CPfcRegion \(\)](#)
- [~CPfcRegion \(\)](#)
- [PFC::E_PFC_ERRORCODE Read \(CPfcTextParser tpCntl, std::string label_leaf\)](#)
read Region(proc.pfc)
- [PFC::E_PFC_ERRORCODE Write \(FILE *fp, const unsigned tab\)](#)
proc.pfc ファイル: Region 出力する

Public 変数

- [int m_regionID](#)
分割した領域のID
- [int m_voxelSize \[3\]](#)
ボクセルサイズ
- [int m_headIndex \[3\]](#)
始点インデックス
- [int m_tailIndex \[3\]](#)
終点インデックス

6.17.1 説明

proc.pfc ファイルの Region

PfcRegion.h の 22 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 CPfcRegion::CPfcRegion ()

コンストラクタ

6.17.2.2 CPfcRegion::~~CPfcRegion ()

デストラクタ

6.17.3 関数

6.17.3.1 PFC::E_PFC_ERRORCODE CPfcRegion::Read (CPfcTextParser tpCntl, std::string label_leaf)

read Region(proc.pfc)

引数

in	<i>tpCntl</i>	PfcTextParser クラス
in	<i>label_leaf</i>	ベースとなる名前 ("/Division/Region")

戻り値

error code

6.17.3.2 PFC::E_PFC_ERRORCODE CPfcRegion::Write (FILE * *fp*, const unsigned *tab*)

proc.pfc ファイル: Region 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 CPfcRegion::m_headIndex[3]

始点インデックス

PfcRegion.h の 29 行で定義されています。

6.17.4.2 int CPfcRegion::m_regionID

分割した領域のID

PfcRegion.h の 27 行で定義されています。

6.17.4.3 int CPfcRegion::m_tailIndex[3]

終点インデックス

PfcRegion.h の 30 行で定義されています。

6.17.4.4 int CPfcRegion::m_voxelSize[3]

ボクセルサイズ

PfcRegion.h の 28 行で定義されています。

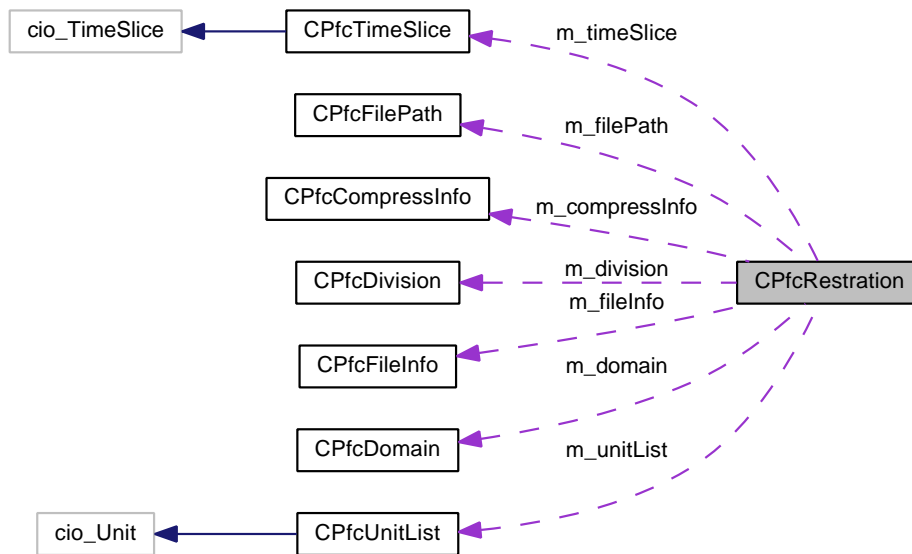
このクラスの説明は次のファイルから生成されました:

- [PfcRegion.h](#)

6.18 クラス CPfcRestration

```
#include <PfcRestration.h>
```

CPfcRestration のコラボレーション図



Public メソッド

- [CPfcRestration \(\)](#)
- [~CPfcRestration \(\)](#)
- [PFC::E_PFC_ERRORCODE Init](#) (const std::string &pfcFilePath)
展開クラス 初期化
- void [GetGlobalVoxel](#) (int voxel[3])
計算領域サイズ取得
- [PFC::E_PFC_ERRORCODE GetHeadTail](#) (const int gDiv[3], const int regionID, int head[3], int tail[3])
領域範囲 (Head/Tail) 計算
- [PFC::E_PFC_ERRORCODE CheckCompressDataOnMem](#) (const int memUseMax, const int head[3], const int tail[3], double &loadRatio)
圧縮データのメモリロード チェック タイムステップは全展開される前提
- [PFC::E_PFC_ERRORCODE LoadCompressDataOnMem](#) (const int head[3], const int tail[3])
圧縮データ ロード
- void [DeleteCompressDataOnMem](#) (void)
ロードした圧縮データ削除
- [PFC::E_PFC_ERRORCODE GetTimeStepList](#) (vector< int > &timeStepList)
タイムステップリスト取得
- [PFC::E_PFC_ERRORCODE ReadData](#) (double *v, const int step, const int head[3], const int tail[3])
データ読み込み (範囲指定)
- [PFC::E_PFC_ERRORCODE ReadData](#) (double *v, const int step, const int ix, const int iy, const int iz)
データ読み込み (位置指定)

Protected 変数

- std::string [m_pfcFilePath](#)
- [CPfcFileInfo m_fileInfo](#)
- [CPfcCompressInfo m_compressInfo](#)
- [CPfcFilePath m_filePath](#)
- [CPfcUnitList m_unitList](#)
- [CPfcTimeSlice m_timeSlice](#)

- int `m_numTimeStep`
- vector< int > `m_timeStepList`
- CPfcDomain `m_domain`
- CPfcDivision `m_division`
- bool `m_bLoadCompressData`
- int `m_head` [3]
- int `m_tail` [3]
- int `m_numRegion`
- vector< int > `m_regionIdList`
- vector< CPfcRestrationRegion * > `m_regionList`

6.18.1 説明

PFC restration class

CPfcRestration.h の 40 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 CPfcRestration::CPfcRestration ()

コンストラクタ

6.18.2.2 CPfcRestration::~~CPfcRestration ()

デストラクタ

6.18.3 関数

6.18.3.1 PFC::E_PFC_ERRORCODE CPfcRestration::CheckCompressDataOnMem (const int *memUseMax*, const int *head*[3], const int *tail*[3], double & *loadRatio*)

圧縮データのメモリロードチェック タイムステップは全展開される前提

引数

in	<i>memUseMax</i>	使用メモリMAX 単位 (Mib)
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>loadRatio</i>	ロード可能な割合 (0.0 - 1.0)

戻り値

終了コード PFC::E_PFC_SUCCESS ロード可 PFC::E_PFC_ERROR ロード不可

6.18.3.2 void CPfcRestration::DeleteCompressDataOnMem (void)

ロードした圧縮データ削除

戻り値

なし

6.18.3.3 void CPfcRestration::GetGlobalVoxel (int voxel[3])

計算領域サイズ取得

引数

out	voxel	計算領域サイズ
-----	-------	---------

戻り値

終了コード

6.18.3.4 PFC::E_PFC_ERRORCODE CPfcRestration::GetHeadTail (const int *gDiv*[3], const int *regionID*, int *head*[3], int *tail*[3])

領域範囲 (Head/Tail) 計算

引数

in	<i>gDiv</i>	計算空間の領域分割数
in	<i>regionID</i>	領域ID: 0 ~ (gDiv[0]*gDiv[1]*gDiv[2]-1)
out	<i>head</i>	計算領域の開始位置
out	<i>tail</i>	計算領域の終了位置

戻り値

終了コード

6.18.3.5 PFC::E_PFC_ERRORCODE CPfcRestration::GetTimeStepList (vector< int > & *timeStepList*)

タイムステップリスト取得

引数

out	<i>timeStepList</i>	タイムステップリスト index.pfc に記載されているStep の値 (複数) を取得
-----	---------------------	---

戻り値

終了コード

6.18.3.6 PFC::E_PFC_ERRORCODE CPfcRestration::Init (const std::string & *pfcFilePath*)

展開クラス 初期化

引数

in	<i>pfcFilePath</i>	index.pfc のファイルパス
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

6.18.3.7 PFC::E_PFC_ERRORCODE CPfcRestration::LoadCompressDataOnMem (const int *head*[3], const int *tail*[3])

圧縮データ ロード

引数

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

戻り値

終了コード

6.18.3.8 PFC::E_PFC_ERRORCODE CPfcRestration::ReadData (double * *v*, const int *step*, const int *head*[3], const int *tail*[3])

データ読み込み (範囲指定)

[LoadCompressDataOnMem\(\)](#) が実行されている場合、メモリから展開する

引数

out	<i>v</i>	展開後の出力領域 メモリは確保済みであること head-tail 間のボクセルサイズ * 成分数
in	<i>step</i>	ステップ番号
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

戻り値

終了コード E_PFC_SUCCESS : OK E_PFC_OUT_OF_RANGE : [LoadCompressDataOnMem\(\)](#) で設定された領域範囲外 E_PFC_ERROR : NG

6.18.3.9 PFC::E_PFC_ERRORCODE CPfcRestration::ReadData (double * *v*, const int *step*, const int *ix*, const int *iy*, const int *iz*)

データ読み込み (位置指定)

[LoadCompressDataOnMem\(\)](#) が実行されている場合、メモリから展開する

引数

out	<i>v</i>	展開後の出力領域 (メモリは確保済みであること)
in	<i>step</i>	ステップ番号
in	<i>ix</i>	x position (>= 1) head の開始は 1 からのため、それに合わせる
in	<i>iy</i>	y position (>= 1) head の開始は 1 からのため、それに合わせる
in	<i>iz</i>	z position (>= 1) head の開始は 1 からのため、それに合わせる

戻り値

終了コード E_PFC_SUCCESS : OK (全タイムステップ可能) E_PFC_OUT_OF_RANGE : [LoadCompressDataOnMem\(\)](#) で設定された領域範囲外 E_PFC_ERROR : NG

6.18.4 変数

6.18.4.1 bool CPfcRestration::m_bLoadCompressData [protected]

PfcRestration.h の 61 行で定義されています。

6.18.4.2 CPfcCompressInfo CPfcRestration::m_compressInfo [protected]

PfcRestration.h の 49 行で定義されています。

6.18.4.3 CPfcDivision CPfcRestration::m_division [protected]

PfcRestration.h の 58 行で定義されています。

6.18.4.4 CPfcDomain CPfcRestration::m_domain [protected]

PfcRestration.h の 57 行で定義されています。

6.18.4.5 CPfcFileInfo CPfcRestration::m_fileInfo [protected]

PfcRestration.h の 48 行で定義されています。

6.18.4.6 CPfcFilePath CPfcRestration::m_filePath [protected]

PfcRestration.h の 50 行で定義されています。

6.18.4.7 int CPfcRestration::m_head[3] [protected]

PfcRestration.h の 64 行で定義されています。

6.18.4.8 int CPfcRestration::m_numRegion [protected]

PfcRestration.h の 68 行で定義されています。

6.18.4.9 int CPfcRestration::m_numTimeStep [protected]

PfcRestration.h の 53 行で定義されています。

6.18.4.10 std::string CPfcRestration::m_pfcFilePath [protected]

PfcRestration.h の 45 行で定義されています。

6.18.4.11 vector<int> CPfcRestration::m_regionIdList [protected]

PfcRestration.h の 69 行で定義されています。

6.18.4.12 vector<CPfcRestrationRegion*> CPfcRestration::m_regionList [protected]

PfcRestration.h の 71 行で定義されています。

6.18.4.13 int CPfcRestration::m_tail[3] [protected]

PfcRestration.h の 65 行で定義されています。

6.18.4.14 CPfcTimeSlice CPfcRestration::m_timeSlice [protected]

PfcRestration.h の 52 行で定義されています。

6.18.4.15 vector<int> CPfcRestration::m_timeStepList [protected]

PfcRestration.h の 54 行で定義されています。

6.18.4.16 CPfcUnitList CPfcRestration::m_unitList [protected]

PfcRestration.h の 51 行で定義されています。

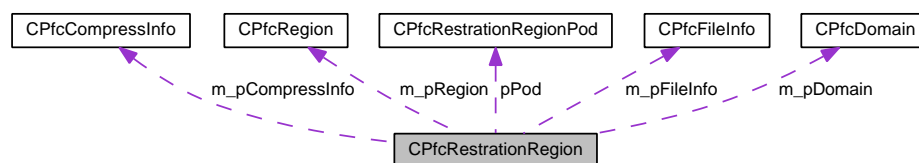
このクラスの説明は次のファイルから生成されました:

- [PfcRestration.h](#)

6.19 クラス CPfcRestrationRegion

```
#include <PfcRestrationRegion.h>
```

CPfcRestrationRegion のコラボレーション図



Public メソッド

- [CPfcRestrationRegion \(\)](#)
- [~CPfcRestrationRegion \(\)](#)
- [PFC::E_PFC_ERRORCODE Init \(CPfcFileInfo *pFileInfo, CPfcCompressInfo *pCompressInfo, vector< int > *pvTimeStepList, CPfcDomain *pDomain, CPfcRegion *pRegion\)](#)
展開クラス 初期化
- [PFC::E_PFC_ERRORCODE LoadCompressDataOnMem \(void\)](#)
圧縮データのメモリロード (分割領域の全データロード)
- [void DeleteCompressDataOnMem \(void\)](#)
ロードした圧縮データ削除
- [PFC::E_PFC_ERRORCODE ReadDataInRange \(double *v, const int step, const int head\[3\], const int tail\[3\], int out_offset_st\[3\], int out_size\[3\]\)](#)
データ読み込み (範囲指定)
- [PFC::E_PFC_ERRORCODE ReadData \(double *v, const int step, const int ix, const int iy, const int iz\)](#)
データ読み込み (位置指定)

Protected 変数

- [CPfcFileInfo * m_pFileInfo](#)
- [CPfcCompressInfo * m_pCompressInfo](#)
- [vector< int > * m_pvTimeStepList](#)
- [CPfcDomain * m_pDomain](#)

- CPfcRegion * m_pRegion
- PFC::E_PFC_COMPRESS_FORMAT m_compressForm
- CPfcRestrationRegionPod * pPod
- bool m_bLoadCompressData

6.19.1 説明

PFC restraion 領域 class

PfcRestrationRegion.h の 37 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 CPfcRestrationRegion::CPfcRestrationRegion ()

コンストラクタ

6.19.2.2 CPfcRestrationRegion::~CPfcRestrationRegion ()

デストラクタ

6.19.3 関数

6.19.3.1 void CPfcRestrationRegion::DeleteCompressDataOnMem (void)

ロードした圧縮データ削除

戻り値

なし

6.19.3.2 PFC::E_PFC_ERRORCODE CPfcRestrationRegion::Init (CPfcFileInfo * pFileInfo, CPfcCompressInfo * pCompressInfo, vector< int > * pvTimeStepList, CPfcDomain * pDomain, CPfcRegion * pRegion)

展開クラス 初期化

引数

in	<i>pFileInfo</i>	File 情報 (属性 etc.)
in	<i>pCompressInfo</i>	圧縮情報
in	<i>pvTimeStepList</i>	タイムステップリスト
in	<i>pDomain</i>	ドメイン情報
in	<i>pRegion</i>	分割領域情報 (1 領域)

戻り値

終了コード

6.19.3.3 PFC::E_PFC_ERRORCODE CPfcRestrationRegion::LoadCompressDataOnMem (void)

圧縮データのメモリロード (分割領域の全データロード)

戻り値

終了コード

6.19.3.4 PFC::E_PFC_ERRORCODE CPfcRestrationRegion::ReadData (double * v, const int step, const int ix, const int iy, const int iz)

データ読み込み (位置指定)

[LoadCompressDataOnMem\(\)](#) が実行されている場合、メモリから展開する

引数

out	v	展開後の出力領域 (1 要素)
in	step	ステップ番号
in	ix	x position (≥ 1) head の開始は 1 からのため、それに合わせる
in	iy	y position (≥ 1) head の開始は 1 からのため、それに合わせる
in	iz	z position (≥ 1) head の開始は 1 からのため、それに合わせる

戻り値

終了コード

6.19.3.5 PFC::E_PFC_ERRORCODE CPfcRestrationRegion::ReadDataInRange (double * v, const int step, const int head[3], const int tail[3], int out_offset_st[3], int out_size[3])

データ読み込み (範囲指定)

[LoadCompressDataOnMem\(\)](#) が実行されている場合、メモリから展開する 対象計算領域と自身が管理している領域が一致していない場合、AND 領域のみが設定される

引数

out	v	展開後の出力領域 領域サイズ : head-tail 間のボクセルサイズ * 成分数 設定領域 : out_offset_st から out_size 分
in	step	ステップ番号
in	head	計算領域の開始位置 (自身の領域のサイズではない)
in	tail	計算領域の終了位置 (自身の領域のサイズではない)
out	out_offset_st	計算領域の終了位置
out	out_size	計算領域の終了位置

戻り値

終了コード

6.19.4 変数

6.19.4.1 bool CPfcRestrationRegion::m_bLoadCompressData [protected]

PfcRestrationRegion.h の 57 行で定義されています。

6.19.4.2 PFC::E_PFC_COMPRESS_FORMAT CPfcRestrationRegion::m_compressForm [protected]

PfcRestrationRegion.h の 53 行で定義されています。

6.19.4.3 CPfcCompressInfo* CPfcRestrationRegion::m_pCompressInfo [protected]

PfcRestrationRegion.h の 45 行で定義されています。

6.19.4.4 CPfcDomain* CPfcRestrationRegion::m_pDomain [protected]

PfcRestrationRegion.h の 49 行で定義されています。

6.19.4.5 CPfcFileInfo* CPfcRestrationRegion::m_pFileInfo [protected]

PfcRestrationRegion.h の 44 行で定義されています。

6.19.4.6 CPfcRegion* CPfcRestrationRegion::m_pRegion [protected]

PfcRestrationRegion.h の 50 行で定義されています。

6.19.4.7 vector<int>* CPfcRestrationRegion::m_pvTimeStepList [protected]

PfcRestrationRegion.h の 46 行で定義されています。

6.19.4.8 CPfcRestrationRegionPod* CPfcRestrationRegion::pPod [protected]

PfcRestrationRegion.h の 55 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:

- [PfcRestrationRegion.h](#)

6.20 クラス CPfcRestrationRegionPod

```
#include <PfcRestrationRegionPod.h>
```

Public メソッド

- [CPfcRestrationRegionPod \(\)](#)
- [~CPfcRestrationRegionPod \(\)](#)
- [PFC::E_PFC_ERRORCODE Init](#) (const std::string &dirPath, const std::string &prefix, const int regionID, const int numSize, [PFC::E_PFC_ARRAYSHAPE](#) arrayShape, int numComponent, int numStep, bool bSingle)
展開クラス 初期化
- [PFC::E_PFC_ERRORCODE LoadCompressDataOnMem](#) (void)
圧縮データのメモリロード (分割領域の全データロード)
- void [DeleteCompressDataOnMem](#) (void)
ロードした圧縮データ削除
- [PFC::E_PFC_ERRORCODE ReadData](#) ([PFC::E_PFC_ARRAYSHAPE](#) arrayShape, double *v, const int stepID)
フィールドデータ読み込み (領域内全情報) 配列形状指定
- [PFC::E_PFC_ERRORCODE ExpandData](#) (double *v, const int stepID)
圧縮データ展開 (領域内全情報, Load 済みのデータから展開)
- [PFC::E_PFC_ERRORCODE ExpandData](#) (double *v, const int stepID, const int index)
圧縮データ展開 (Load 済みのデータから展開)
- [PFC::E_PFC_ERRORCODE ExpandDataOnFile](#) (double *v, const int stepID, const int index)
圧縮データ展開 (ファイルから展開)

Protected メソッド

- [PFC::E_PFC_ERRORCODE ReadFieldData](#) (double *v, const int stepID)
フィールドデータ読み込み (領域内全情報) *POD* 圧縮時の配列形状で返す (*IJKN* 順)
- [PFC::E_PFC_ERRORCODE GetExpandRowInfo](#) (const int stepID, const int row_num, double *coef, int *base_index, int *row_st, int *row_size)
圧縮データ展開経路情報取得 係数ファイルは読み込み済みであること 基底ファイルのヘッダは読み込み済みであること

Protected 変数

- bool [m_bLoadCompressData](#)
- std::string [m_dirPath](#)
- std::string [m_prefix](#)
- int [m_regionID](#)
- int [m_numSize](#)
- [PFC::E_PFC_ARRAYSHAPE](#) [m_arrayShape](#)
- int [m_numComponent](#)
- int [m_numStep](#)
- bool [m_bSingle](#)
- int [m_numCalculatedLayer](#)
- int [m_numParallel](#)
- double ** [m_pIndexBase](#)
- double * [m_pBaseData](#)
- int * [m_pBaseSizes](#)
- int [m_numCoef](#)
- double ** [m_pIndexCoef](#)
- double * [m_pCoefData](#)

6.20.1 説明

[PFC](#) restration class

PfcRestrationRegionPod.h の 34 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 CPfcRestrationRegionPod::CPfcRestrationRegionPod ()

コンストラクタ

6.20.2.2 CPfcRestrationRegionPod::~CPfcRestrationRegionPod ()

デストラクタ

6.20.3 関数

6.20.3.1 void CPfcRestrationRegionPod::DeleteCompressDataOnMem (void)

ロードした圧縮データ削除

戻り値

なし

6.20.3.2 PFC::E_PFC_ERRORCODE CPfcRestrationRegionPod::ExpandData (double * v, const int stepID)

圧縮データ展開 (領域内全情報, Load 済みのデータから展開)

引数

out	v	展開後の出力領域 メモリは確保済みであること ボクセルサイズ * 成分数
in	stepID	ステップID (0 起点)

戻り値

終了コード

6.20.3.3 PFC::E_PFC_ERRORCODE CPfcRestrationRegionPod::ExpandData (double * v, const int stepID, const int index)

圧縮データ展開 (Load 済みのデータから展開)

引数

out	v	展開後の出力領域 (1 要素)
in	stepID	ステップID (0 起点)
in	index	領域先頭からの要素の 1 次元Index

戻り値

終了コード

6.20.3.4 PFC::E_PFC_ERRORCODE CPfcRestrationRegionPod::ExpandDataOnFile (double * v, const int stepID, const int index)

圧縮データ展開 (ファイルから展開)

引数

out	v	展開後の出力領域 (1 要素)
in	stepID	ステップID (0 起点)
in	index	領域先頭からの要素の 1 次元Index

戻り値

終了コード

6.20.3.5 PFC::E_PFC_ERRORCODE CPfcRestrationRegionPod::GetExpandRowInfo (const int stepID, const int row_num, double * coef, int * base_index, int * row_st, int * row_size) [protected]

圧縮データ展開用経路情報取得 係数ファイルは読み込み済みであること 基底ファイルのヘッダは読み込み済みであること

引数

in	stepID	ステップID (0 起点)
----	--------	---------------

in	<i>row_num</i>	経路の数
out	<i>coef</i>	経路毎の係数 size:[row_num]
out	<i>base_index</i>	経路毎の基底のインデックス size:[row_num] pod base ファイルを出力したランク番号
out	<i>row_st</i>	経路毎の開始位置 (0 起点) size:[row_num]
out	<i>row_size</i>	経路毎のサイズ size:[row_num] == base_index 毎のサイズ

戻り値

終了コード

6.20.3.6 PFC::E_PFC_ERRORCODE CPfcRestrationRegionPod::Init (const std::string & *dirPath*, const std::string & *prefix*, const int *regionID*, const int *numSize*, PFC::E_PFC_ARRAYSHAPE *arrayShape*, int *numComponent*, int *numStep*, bool *bSingle*)

展開クラス 初期化

引数

in	<i>dirPath</i>	フィールドデータ出力ディレクトリパス
in	<i>prefix</i>	属性名
in	<i>regionID</i>	領域ID
in	<i>numSize</i>	要素数 (voxel サイズ)
in	<i>arrayShape</i>	配列形状 E_PFC_IJKN / E_PFC_NIJK
in	<i>numComponent</i>	成分数
in	<i>numStep</i>	タイムステップ数
in	<i>arrayShape</i>	配列形状 E_PFC_IJKN / E_PFC_NIJK
in	<i>bSingle</i>	単一領域フラグ true / false

戻り値

終了コード

6.20.3.7 PFC::E_PFC_ERRORCODE CPfcRestrationRegionPod::LoadCompressDataOnMem (void)

圧縮データのメモリロード (分割領域の全データロード)

戻り値

終了コード

6.20.3.8 PFC::E_PFC_ERRORCODE CPfcRestrationRegionPod::ReadData (PFC::E_PFC_ARRAYSHAPE *arrayShape*, double * *v*, const int *stepID*)

フィールドデータ読み込み (領域内全情報) 配列形状指定

指定された配列形状に従ってデータを返す

引数

in	<i>arrayShape</i>	配列形状 (E_PFC_IJKN/E_PFC_NIJK)
----	-------------------	------------------------------

out	v	展開後の出力領域 メモリは確保済みであること region 内のボクセルサイズ * 成分数
in	stepID	ステップID (0 起点)

戻り値

終了コード

6.20.3.9 PFC::E_PFC_ERRORCODE CPfcRestrationRegionPod::ReadFieldData (double * v, const int stepID)
[protected]

フィールドデータ読み込み (領域内全情報) POD 圧縮時の配列形状で返す (IJKN 順)

[LoadCompressDataOnMem\(\)](#) が実行されている場合、メモリから展開する

引数

out	v	展開後の出力領域 メモリは確保済みであること region 内のボクセルサイズ * 成分数
in	stepID	ステップID (0 起点)

戻り値

終了コード

6.20.4 変数

6.20.4.1 PFC::E_PFC_ARRAYSHAPE CPfcRestrationRegionPod::m_arrayShape [protected]

PfcRestrationRegionPod.h の 47 行で定義されています。

6.20.4.2 bool CPfcRestrationRegionPod::m_bLoadCompressData [protected]

PfcRestrationRegionPod.h の 40 行で定義されています。

6.20.4.3 bool CPfcRestrationRegionPod::m_bSingle [protected]

PfcRestrationRegionPod.h の 50 行で定義されています。

6.20.4.4 std::string CPfcRestrationRegionPod::m_dirPath [protected]

PfcRestrationRegionPod.h の 42 行で定義されています。

6.20.4.5 int CPfcRestrationRegionPod::m_numCalculatedLayer [protected]

PfcRestrationRegionPod.h の 55 行で定義されています。

6.20.4.6 int CPfcRestrationRegionPod::m_numCoef [protected]

PfcRestrationRegionPod.h の 66 行で定義されています。

6.20.4.7 `int CPfcRestrationRegionPod::m_numComponent` [protected]

PfcRestrationRegionPod.h の 48 行で定義されています。

6.20.4.8 `int CPfcRestrationRegionPod::m_numParallel` [protected]

PfcRestrationRegionPod.h の 56 行で定義されています。

6.20.4.9 `int CPfcRestrationRegionPod::m_numSize` [protected]

PfcRestrationRegionPod.h の 45 行で定義されています。

6.20.4.10 `int CPfcRestrationRegionPod::m_numStep` [protected]

PfcRestrationRegionPod.h の 49 行で定義されています。

6.20.4.11 `double* CPfcRestrationRegionPod::m_pBaseData` [protected]

PfcRestrationRegionPod.h の 61 行で定義されています。

6.20.4.12 `int* CPfcRestrationRegionPod::m_pBaseSizes` [protected]

PfcRestrationRegionPod.h の 62 行で定義されています。

6.20.4.13 `double* CPfcRestrationRegionPod::m_pCoefData` [protected]

PfcRestrationRegionPod.h の 69 行で定義されています。

6.20.4.14 `double** CPfcRestrationRegionPod::m_pIndexBase` [protected]

PfcRestrationRegionPod.h の 59 行で定義されています。

6.20.4.15 `double** CPfcRestrationRegionPod::m_pIndexCoef` [protected]

PfcRestrationRegionPod.h の 67 行で定義されています。

6.20.4.16 `std::string CPfcRestrationRegionPod::m_prefix` [protected]

PfcRestrationRegionPod.h の 43 行で定義されています。

6.20.4.17 `int CPfcRestrationRegionPod::m_regionID` [protected]

PfcRestrationRegionPod.h の 44 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:

- [PfcRestrationRegionPod.h](#)

6.21 クラス CPfcTextParser

```
#include <PfcTextParser.h>
```

Public メソッド

- [CPfcTextParser](#) ()
- [~CPfcTextParser](#) ()
- bool [GetVector](#) (const std::string &label, int *vec, const int nvec)
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)
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)
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)
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)
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)
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)
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](#) ()

Private 変数

- [TextParser](#) * [tp](#)
テキストパーサ

6.21.1 説明

PfcTextParser.h の 30 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 CPfcTextParser::CPfcTextParser () [inline]

コンストラクタ

PfcTextParser.h の 37 行で定義されています。

```
37 {};
```

6.21.2.2 CPfcTextParser::~CPfcTextParser () [inline]

デストラクタ

PfcTextParser.h の 40 行で定義されています。

```
40 {};
```

6.21.3 関数

6.21.3.1 bool CPfcTextParser::chkLabel (const std::string & label)

ラベルの有無をチェック

引数

in	label	チェックするラベル (絶対パス)
----	-------	--------------------

6.21.3.2 bool CPfcTextParser::chkNode (const std::string & label)

ノードの有無をチェック

引数

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

6.21.3.3 int CPfcTextParser::countLabels (const std::string & label)

ノード以下のラベルの数を数える

引数

in	label	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、もしくはない場合は -1 を返す)

6.21.3.4 bool CPfcTextParser::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	label	ノードの絶対パス
in	nnode	取得する文字列が現れる順番
out	ct	取得した文字列

6.21.3.5 void CPfcTextParser::getTPInstance ()

TextParserLibrary のインスタンス生成

戻り値

エラーコード

6.21.3.6 bool CPfcTextParser::GetValue (const std::string & *label*, int * *ct*)

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）
out	<i>ct</i>	変数格納ポインタ

6.21.3.7 bool CPfcTextParser::GetValue (const std::string & *label*, double * *ct*)

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）
out	<i>ct</i>	変数格納ポインタ

6.21.3.8 bool CPfcTextParser::GetValue (const std::string & *label*, std::string * *ct*)

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）
out	<i>ct</i>	変数格納ポインタ

6.21.3.9 bool CPfcTextParser::GetVector (const std::string & *label*, int * *vec*, const int *nvec*)

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

6.21.3.10 bool CPfcTextParser::GetVector (const std::string & *label*, double * *vec*, const int *nvec*)

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

6.21.3.11 bool CPfcTextParser::GetVector (const std::string & *label*, std::string * *vec*, const int *nvec*)

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

6.21.3.12 `int CPfcTextParser::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

6.21.3.13 `int CPfcTextParser::remove () [inline]`

テキストパーサーの内容を破棄

PfcTextParser.h の 151 行で定義されています。

```

152 {
153     return tp->remove();
154 }
```

6.21.4 変数

6.21.4.1 `TextParser* CPfcTextParser::tp [private]`

テキストパーサ

PfcTextParser.h の 33 行で定義されています。

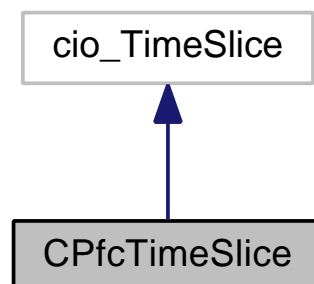
このクラスの説明は次のファイルから生成されました:

- [PfcTextParser.h](#)

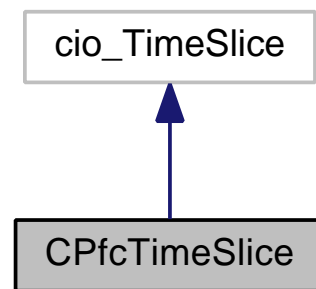
6.22 クラス CPfcTimeSlice

```
#include <PfcTimeSlice.h>
```

CPfcTimeSlice に対する継承グラフ



CPfcTimeSlice のコラボレーション図



Public メソッド

- [CPfcTimeSlice \(\)](#)
- [CPfcTimeSlice \(const cio_TimeSlice *cioTimeSlice\)](#)
- [~CPfcTimeSlice \(\)](#)
- [PFC::E_PFC_ERRORCODE Select \(int startStep, int endStep\)](#)
タイムスライス情報を指定された開始ステップ、終了ステップに絞る

6.22.1 説明

index.pfc ファイルの TimeSlice

PfcTimeSlice.h の 25 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 CPfcTimeSlice::CPfcTimeSlice ()

コンストラクタ

6.22.2.2 CPfcTimeSlice::CPfcTimeSlice (const cio_TimeSlice * cioTimeSlice)

コンストラクタ

6.22.2.3 CPfcTimeSlice::~~CPfcTimeSlice ()

デストラクタ

6.22.3 関数

6.22.3.1 PFC::E_PFC_ERRORCODE CPfcTimeSlice::Select (int startStep, int endStep)

タイムスライス情報を指定された開始ステップ、終了ステップに絞る

引数

in	<i>startStep</i>	開始ステップ
in	<i>endStep</i>	終了ステップ

戻り値

error code

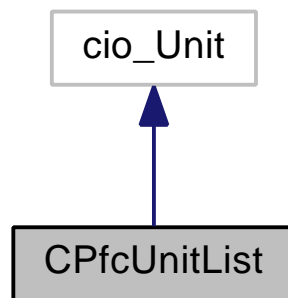
このクラスの説明は次のファイルから生成されました:

- [PfcTimeSlice.h](#)

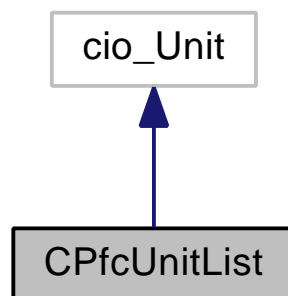
6.23 クラス CPfcUnitList

```
#include <PfcUnitList.h>
```

CPfcUnitList に対する継承グラフ



CPfcUnitList のコラボレーション図



Public メソッド

- [CPfcUnitList \(\)](#)
- [~CPfcUnitList \(\)](#)

6.23.1 説明

index.pfc ファイルの TimeSlice

PfcUnitList.h の 27 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 CPfcUnitList::CPfcUnitList ()

コンストラクタ

6.23.2.2 CPfcUnitList::~~CPfcUnitList ()

デストラクタ

このクラスの説明は次のファイルから生成されました:

- [PfcUnitList.h](#)

Chapter 7

ファイル

7.1 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 recvtype, MPI_Comm comm)`
- `int MPI_Gather (void *sendbuf, int sendcnt, MPI_Datatype sendtype, void *recvbuf, int recvcnt, MPI_Datatype recvtype, int root, MPI_Comm comm)`

7.1.1 マクロ定義

7.1.1.1 `#define MPI_CHAR 2`

mpi_stubs.h の 21 行で定義されています。

7.1.1.2 `#define MPI_COMM_WORLD 0`

mpi_stubs.h の 19 行で定義されています。

7.1.1.3 `#define MPI_INT 1`

mpi_stubs.h の 20 行で定義されています。

7.1.1.4 #define MPI_SUCCESS true

mpi_stubs.h の 23 行で定義されています。

7.1.2 型定義

7.1.2.1 typedef int MPI_Comm

mpi_stubs.h の 17 行で定義されています。

7.1.2.2 typedef int MPI_Datatype

mpi_stubs.h の 18 行で定義されています。

7.1.3 関数

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

mpi_stubs.h の 39 行で定義されています。

```
42 {  
43     return 0;  
44 }
```

7.1.3.2 int MPI_Comm_rank (MPI_Comm *comm*, int * *rank*) [inline]

mpi_stubs.h の 27 行で定義されています。

```
28 {  
29     *rank = 0;  
30     return 0;  
31 }
```

7.1.3.3 int MPI_Comm_size (MPI_Comm *comm*, int * *size*) [inline]

mpi_stubs.h の 33 行で定義されています。

```
34 {  
35     *size = 1;  
36     return 0;  
37 }
```

7.1.3.4 int MPI_Gather (void * *sendbuf*, int *sendcnt*, MPI_Datatype *sendtype*, void * *recvbuf*, int *recvcnt*, MPI_Datatype *recvtype*, int *root*, MPI_Comm *comm*) [inline]

mpi_stubs.h の 46 行で定義されています。

```
49 {  
50     return 0;  
51 }
```

7.1.3.5 `bool MPI_Init (int * argc, char *** argv) [inline]`

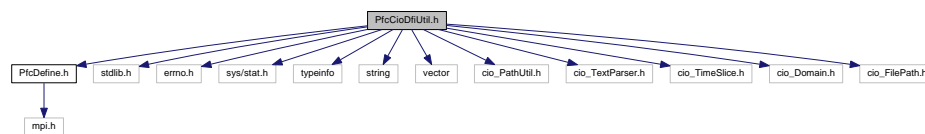
mpi_stubs.h の 25 行で定義されています。

```
25 { return true; }
```

7.2 PfcCioDfiUtil.h

```
#include "PfcDefine.h"
#include <stdlib.h>
#include <errno.h>
#include <sys/stat.h>
#include <typeinfo>
#include <string>
#include <vector>
#include "cio_PathUtil.h"
#include "cio_TextParser.h"
#include "cio_TimeSlice.h"
#include "cio_Domain.h"
#include "cio_FilePath.h"
```

PfcCioDfiUtil.h のインクルード依存関係図



構成

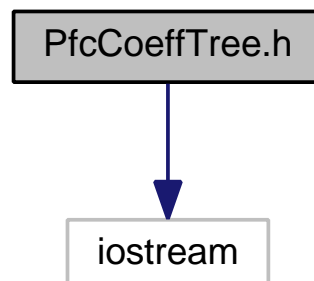
- class [CPfcCioDfiUtil](#)

7.3 PfcCoeffTree.h

[CPfcCoeffTree](#) Class Header.

```
#include <iostream>
```

PfcCoeffTree.h のインクルード依存関係図



構成

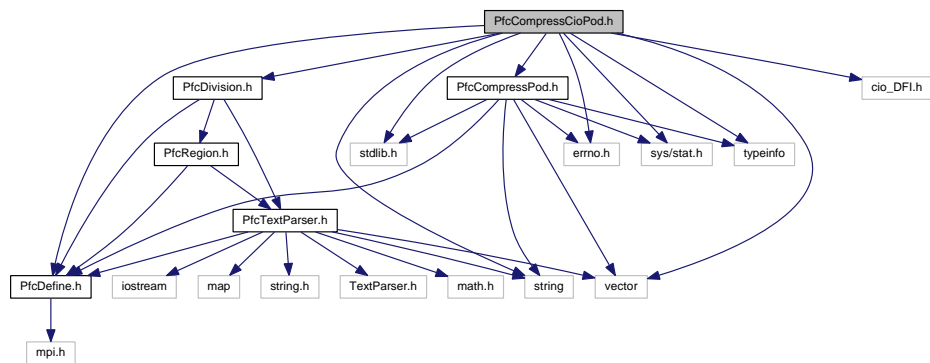
- class [CPfcCoeffTreeNode](#)
- class [CPfcCoeffTree](#)

7.5 PfcCompressCioPod.h

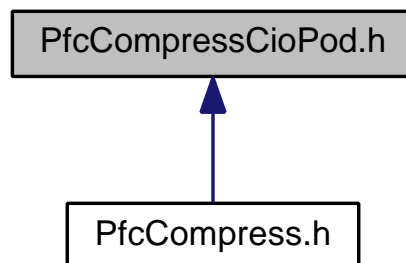
[CPfcCompressCioPod](#) Class Header.

```
#include "PfcDefine.h"
#include "cio_DFI.h"
#include "PfcDivision.h"
#include "PfcCompressPod.h"
#include <stdlib.h>
#include <errno.h>
#include <sys/stat.h>
#include <typeinfo>
#include <string>
#include <vector>
```

PfcCompressCioPod.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 [CPfcCompressCioPod](#)

7.5.1 説明

[CPfcCompressCioPod](#) Class Header.

作者

aics

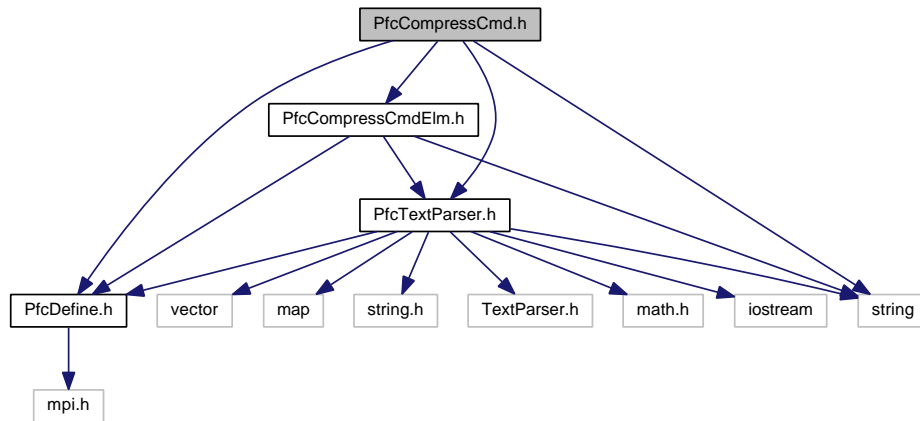
[PfcCompressCioPod.h](#) で定義されています。

7.6 PfcCompressCmd.h

[CPfcCompressCmd](#) Class Header.

```
#include "PfcDefine.h"
#include <string>
#include "PfcCompressCmdElm.h"
#include "PfcTextParser.h"
```

PfcCompressCmd.h のインクルード依存関係図



構成

- class [CPfcCompressCmd](#)

7.6.1 説明

[CPfcCompressCmd](#) Class Header.

作者

aics

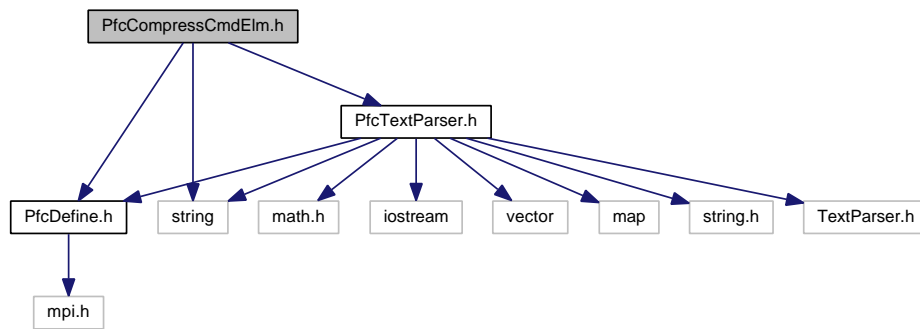
[PfcCompressCmd.h](#) で定義されています。

7.7 PfcCompressCmdElm.h

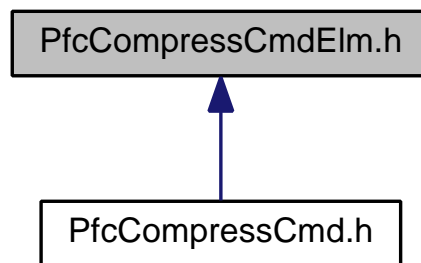
[CPfcCompressCmdElm](#) Class Header.

```
#include "PfcDefine.h"
#include <string>
#include "PfcTextParser.h"
```


PfcCompressCmdElm.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 [CPfcCompressCmdElm](#)

マクロ定義

- `#define _PFC_COMPRESS_CMD_ELM_CNTL_`

7.7.1 説明

[CPfcCompressCmdElm](#) Class Header.

作者

aics

[PfcCompressCmdElm.h](#) で定義されています。

7.7.2 マクロ定義

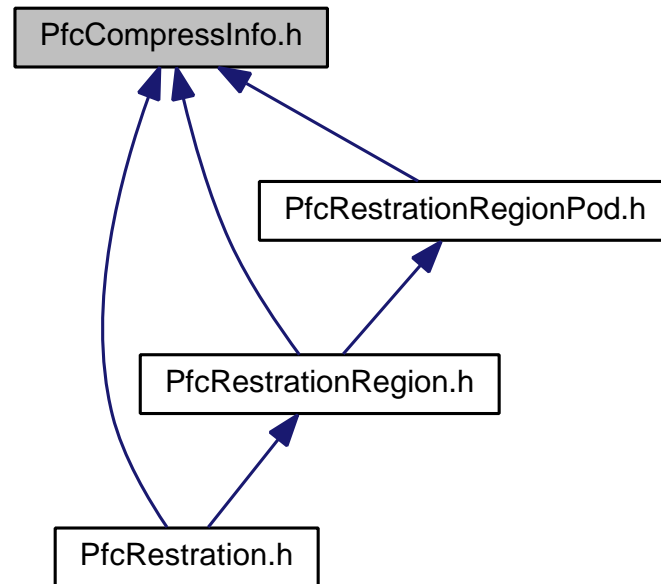
7.7.2.1 `#define _PFC_COMPRESS_CMD_ELM_CNTL_`

[PfcCompressCmdElm.h](#) の 2 行で定義されています。

7.8 PfcCompressInfo.h

[CPfcCompressInfo](#) 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 [CPfcCompressInfo](#)

7.8.1 説明

[CPfcCompressInfo](#) Class Header.

作者

aics

[PfcCompressInfo.h](#) で定義されています。

7.9 PfcCompressPod.h

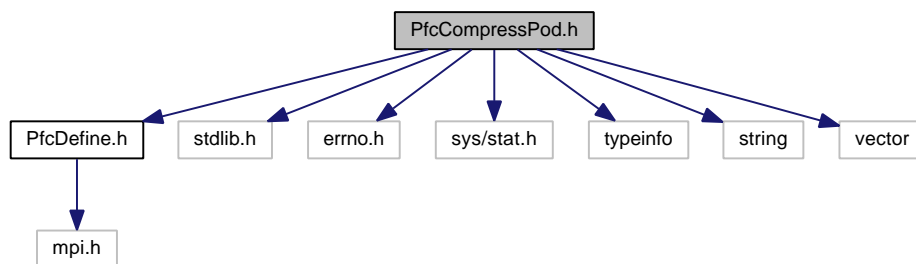
[CPfcCompressPod](#) Class Header.

```

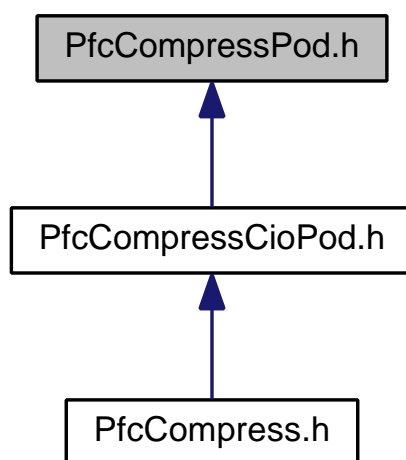
#include "PfcDefine.h"
#include <stdlib.h>
#include <errno.h>
#include <sys/stat.h>
#include <typeinfo>
#include <string>
#include <vector>

```

PfcCompressPod.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 [CPfcCompressPod](#)

7.9.1 説明

[CPfcCompressPod](#) Class Header.

作者

aics

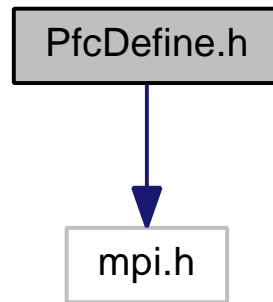
[PfcCompressPod.h](#) で定義されています。

7.10 PfcDefine.h

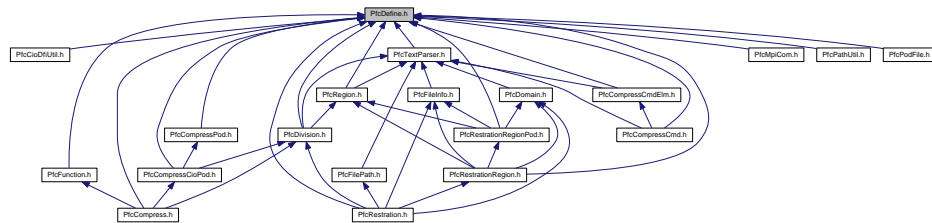
PFC の定義マクロ記述ヘッダーファイル

```
#include "mpi.h"
```

PfcDefine.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

- [PFC](#)

Constant Groups

- [PFC](#)

マクロ定義

- `#define PFC_PRINT PFC_PRINT_AT(__FILE__, __LINE__);printf`
- `#define PFC_PRINT_AT(file, lineno)`
- `#define DEBUG_PRINT if(0) printf`
- `#define DEBUG_PRINT_AT(file, lineno)`
- `#define D_PFC_EPSILON (1.0e-9)`
- `#define D_PFC_COMPRESS_ERROR_DEFAULT (0.01)`
- `#define D_PFC_FLOAT32 "Float32"`
- `#define D_PFC_FLOAT64 "Float64"`
- `#define D_PFC_IJNK "ijkn"`
- `#define D_PFC_NIJK "nijk"`
- `#define D_PFC_LITTLE "little"`
- `#define D_PFC_BIG "big"`
- `#define _PFC_TAB_STR " "`
- `#define D_PFC_OPT_FLAG_SAVE (0x00000001)`
- `#define PFC_CHECK_OPT_SAVE(opt_flags) (opt_flags & D_PFC_OPT_FLAG_SAVE)`
- `#define _PFC_IDX_IJK(_I, _J, _K, _NI, _NJ, _NK, _VC)`
- `#define _PFC_IDX_IJ(_I, _J, _NI, _NJ, _VC)`
- `#define _PFC_IDX_NIJ(_N, _I, _J, _NI, _NJ, _NN, _VC)`

- #define `_PFC_IDX_IJKN`(_I, _J, _K, _N, _NI, _NJ, _NK, _VC)
- #define `_PFC_IDX_NIJK`(_N, _I, _J, _K, _NN, _NI, _NJ, _NK, _VC)
- #define `_PFC_WRITE_TAB`(_FP, _NTAB)

列挙型

- enum `PFC::E_PFC_COMPRESS_FORMAT` { `PFC::E_PFC_COMPRESS_FMT_UNKNOWN` = -1, `PFC::E_PFC_COMPRESS_FMT_POD` = 1 }
- enum `PFC::E_PFC_DTYPE` { `PFC::E_PFC_DTYPE_UNKNOWN` = 0, `PFC::E_PFC_INT8`, `PFC::E_PFC_INT16`, `PFC::E_PFC_INT32`, `PFC::E_PFC_INT64`, `PFC::E_PFC_UINT8`, `PFC::E_PFC_UINT16`, `PFC::E_PFC_UINT32`, `PFC::E_PFC_UINT64`, `PFC::E_PFC_FLOAT32`, `PFC::E_PFC_FLOAT64` }
- enum `PFC::E_PFC_ARRAYSHAPE` { `PFC::E_PFC_ARRAYSHAPE_UNKNOWN` = -1, `PFC::E_PFC_IJKN` = 0, `PFC::E_PFC_NIJK` }
- enum `PFC::E_PFC_ENDIANTYPE` { `PFC::E_PFC_ENDIANTYPE_UNKNOWN` = -1, `PFC::E_PFC_LITTLE` = 0, `PFC::E_PFC_BIG` }
- enum `PFC::E_PFC_ERRORCODE` { `PFC::E_PFC_SUCCESS` = 1, `PFC::E_PFC_ERROR` = -1, `PFC::E_PFC_ERROR_READ_CNTLFILE_OPENERERROR` = 500, `PFC::E_PFC_ERROR_READ_CNTL_DOMAINDIVISION` = 501, `PFC::E_PFC_ERROR_READ_CNTL_NO_ITEM` = 510, `PFC::E_PFC_ERROR_READ_CNTL_ITEMCNTL` = 511, `PFC::E_PFC_ERROR_READ_CNTL_DFI_PATH` = 512, `PFC::E_PFC_ERROR_READ_CNTL_OUTDIR_PATH` = 513, `PFC::E_PFC_ERROR_READ_CNTL_COMPRESS_FMT` = 514, `PFC::E_PFC_ERROR_READ_CNTL_PROCFILE_SAVE` = 515, `PFC::E_PFC_ERROR_READ_INDEXFILE_OPENERERROR` = 1000, `PFC::E_PFC_ERROR_READ_FILEINFO` = 1010, `PFC::E_PFC_ERROR_READ_PFC_DIRECTORYPATH` = 1011, `PFC::E_PFC_ERROR_READ_PFC_PREFIX` = 1012, `PFC::E_PFC_ERROR_READ_PFC_FILEFORMAT` = 1013, `PFC::E_PFC_ERROR_READ_PFC_GUIDECCELL` = 1014, `PFC::E_PFC_ERROR_READ_PFC_DATATYPE` = 1015, `PFC::E_PFC_ERROR_READ_PFC_ENDIAN` = 1016, `PFC::E_PFC_ERROR_READ_PFC_ARRAYSHAPE` = 1017, `PFC::E_PFC_ERROR_READ_PFC_COMPONENT` = 1018, `PFC::E_PFC_ERROR_READ_COMPRESSINFO` = 1030, `PFC::E_PFC_ERROR_READ_PFC_COMPRESSFORMAT` = 1031, `PFC::E_PFC_ERROR_READ_PFC_COMPRESSERROR` = 1032, `PFC::E_PFC_ERROR_READ_PFC_CALCULATED` = 1033, `PFC::E_PFC_ERROR_READ_PFC_VERSION` = 1034, `PFC::E_PFC_ERROR_READ_PFC_STARTSTEP` = 1035, `PFC::E_PFC_ERROR_READ_PFC_ENDSTEP` = 1036, `PFC::E_PFC_ERROR_READ_FILEPATH` = 1040, `PFC::E_PFC_ERROR_READ_PFC_DFIPATH` = 1041, `PFC::E_PFC_ERROR_READ_PFC_PFCPROCESS` = 1042, `PFC::E_PFC_ERROR_READ_UNITLIST` = 1050, `PFC::E_PFC_ERROR_READ_TIMESLICE` = 1060, `PFC::E_PFC_ERROR_WRITE_INDEXFILENAME_EMPTY` = 1100, `PFC::E_PFC_ERROR_WRITE_INDEXFILE_OPENERERROR` = 1101, `PFC::E_PFC_ERROR_WRITE_FILEINFO` = 1110, `PFC::E_PFC_ERROR_WRITE_PFC_DATATYPE` = 1111, `PFC::E_PFC_ERROR_WRITE_COMPRESSINFO` = 1130, `PFC::E_PFC_ERROR_WRITE_FILEPATH` = 1140, `PFC::E_PFC_ERROR_WRITE_UNITLIST` = 1150, `PFC::E_PFC_ERROR_WRITE_TIMESLICE` = 1160, `PFC::E_PFC_ERROR_READ_PROCFILE_OPENERERROR` = 1200, `PFC::E_PFC_ERROR_READ_DOMAIN` = 1210, `PFC::E_PFC_ERROR_READ_PFC_GLOBALORIGIN` = 1211, `PFC::E_PFC_ERROR_READ_PFC_GLOBALREGION` = 1212, `PFC::E_PFC_ERROR_READ_PFC_GLOBALVOXEL` = 1213, `PFC::E_PFC_ERROR_READ_PFC_GLOBALDIVISION` = 1214, `PFC::E_PFC_ERROR_READ_DIVISION` = 1220, `PFC::E_PFC_ERROR_READ_PFC_NO_REGION` = 1221, `PFC::E_PFC_ERROR_READ_PFC_REGION_ID` = 1222, `PFC::E_PFC_ERROR_READ_PFC_REGION_VOXELSIZE` = 1223, `PFC::E_PFC_ERROR_READ_PFC_REGION_HEADINDEX` = 1224, `PFC::E_PFC_ERROR_READ_PFC_REGION_T` = 1225, `PFC::E_PFC_ERROR_WRITE_PROCFILE_OPENERERROR` = 1300, `PFC::E_PFC_ERROR_WRITE_DOMAIN` = 1310, `PFC::E_PFC_ERROR_WRITE_DIVISION` = 1320, `PFC::E_PFC_ERROR_READ_PODBASE_OPENERERROR`

```

= 2000,
PFC::E_PFC_ERROR_READ_PODBASE_HEADER = 2005, PFC::E_PFC_ERROR_READ_PODBASE_DATA
= 2010, PFC::E_PFC_ERROR_WRITE_PODBASE_OPENERROR = 2100, PFC::E_PFC_ERROR_WRITE_PODBASE_DATA
= 2110,
PFC::E_PFC_ERROR_READ_PODCOEF_OPENERROR = 2200, PFC::E_PFC_ERROR_READ_PODCOEF_DATA
= 2210, PFC::E_PFC_ERROR_WRITE_PODCOEF_OPENERROR = 2300, PFC::E_PFC_ERROR_WRITE_PODCOEF_DATA
= 2310,
PFC::E_PFC_ERROR_PFC_COMPRESSFORMAT = 3000, PFC::E_PFC_ERROR_OUT_OF_RANGE =
3100 }

```

7.10.1 説明

PFC の定義マクロ記述ヘッダーファイル

作者

aics

[PfcDefine.h](#) で定義されています。

7.10.2 マクロ定義

7.10.2.1 #define _PFC_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

[PfcDefine.h](#) の 264 行で定義されています。

7.10.2.2 #define _PFC_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

PfcDefine.h の 250 行で定義されています。

7.10.2.3 `#define _PFC_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)) \
+ _PFC_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>	成分インデクス
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

PfcDefine.h の 296 行で定義されています。

7.10.2.4 `#define _PFC_IDX_NIJ(_N, _I, _J, _NI, _NJ, _NN, _VC)`

値:

```
( (long long) (_NN) * _PFC_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>	成分インデクス
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>	成分数
in	<code>_VC</code>	仮想セル数

戻り値

1 次元インデクス

PfcDefine.h の 279 行で定義されています。

7.10.2.5 `#define _PFC_IDX_NIJK(_N, _I, _J, _K, _NN, _NI, _NJ, _NK, _VC)`

値:

```
( (long long) (_NN) * _PFC_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>	成分インデクス
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>	成分数
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

PfcDefine.h の 314 行で定義されています。

7.10.2.6 `#define _PFC_TAB_STR " "`

PfcDefine.h の 73 行で定義されています。

7.10.2.7 `#define _PFC_WRITE_TAB(_FP, _NTAB)`

値:

```
{\
  for(int _NTCNT=0; _NTCNT<_NTAB; _NTCNT++) fprintf(_FP, _PFC_TAB_STR); \
}
```

PFC 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

PfcDefine.h の 323 行で定義されています。

7.10.2.8 `#define D_PFC_BIG "big"`

PfcDefine.h の 71 行で定義されています。

7.10.2.9 `#define D_PFC_COMPRESS_ERROR_DEFAULT (0.01)`

PfcDefine.h の 54 行で定義されています。

7.10.2.10 `#define D_PFC_EPSILON (1.0e-9)`

PfcDefine.h の 51 行で定義されています。

7.10.2.11 `#define D_PFC_FLOAT32 "Float32"`

PfcDefine.h の 64 行で定義されています。

7.10.2.12 `#define D_PFC_FLOAT64 "Float64"`

PfcDefine.h の 65 行で定義されています。

7.10.2.13 `#define D_PFC_IJNK "ijkn"`

PfcDefine.h の 67 行で定義されています。

7.10.2.14 `#define D_PFC_LITTLE "little"`

PfcDefine.h の 70 行で定義されています。

7.10.2.15 `#define D_PFC_NIJK "nijk"`

PfcDefine.h の 68 行で定義されています。

7.10.2.16 `#define D_PFC_OPT_FLAG_SAVE (0x00000001)`

PfcDefine.h の 77 行で定義されています。

7.10.2.17 `#define DEBUG_PRINT if(0) printf`

PfcDefine.h の 39 行で定義されています。

7.10.2.18 #define DEBUG_PRINT_AT(file, lineno)

値:

```
{ \
    printf ( "DBG [%s:%4d] ",file,lineno ); \
}
```

PfcDefine.h の 42 行で定義されています。

7.10.2.19 #define PFC_CHECK_OPT_SAVE(opt_flags) (opt_flags&D_PFC_OPT_FLAG_SAVE)

PfcDefine.h の 79 行で定義されています。

7.10.2.20 #define PFC_PRINT PFC_PRINT_AT(__FILE__,__LINE__);printf

PfcDefine.h の 28 行で定義されています。

7.10.2.21 #define PFC_PRINT_AT(file, lineno)

値:

```
{ \
    printf ( "[%s:%4d] ",file,lineno ); \
}
```

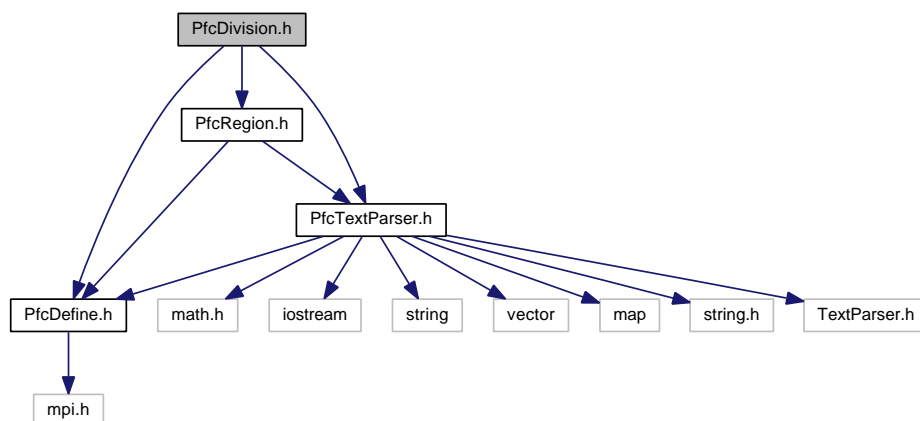
PfcDefine.h の 31 行で定義されています。

7.11 PfcDivision.h

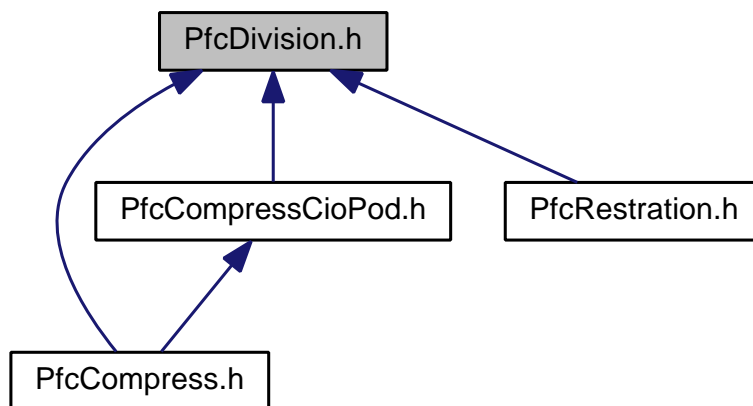
PfcDivision Class Header.

```
#include "PfcDefine.h"
#include "PfcTextParser.h"
#include "PfcRegion.h"
```

PfcDivision.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 [CPfcDivision](#)

7.11.1 説明

PfcDivision Class Header.

作者

aics

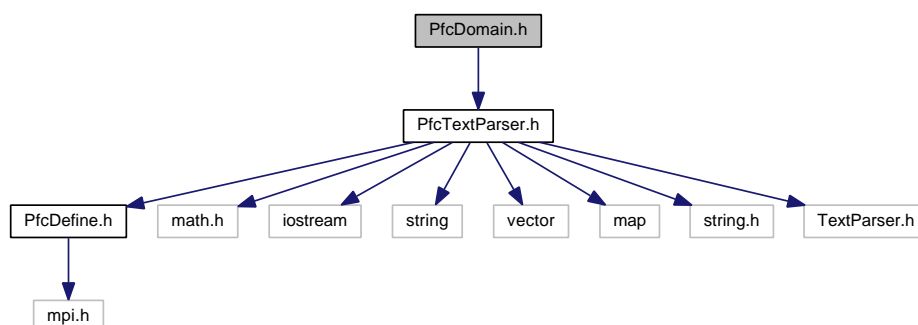
[PfcDivision.h](#) で定義されています。

7.12 PfcDomain.h

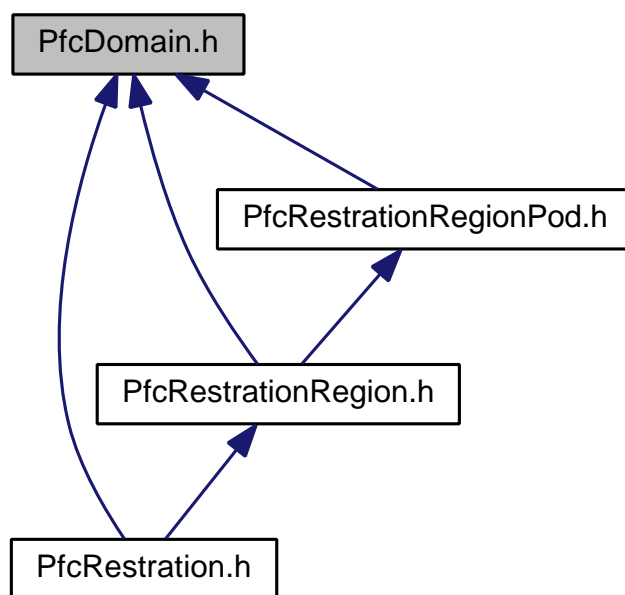
PfcDomain Class Header.

```
#include "PfcTextParser.h"
```

PfcDomain.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 [CPfcDomain](#)

7.12.1 説明

PfcDomain Class Header.

作者

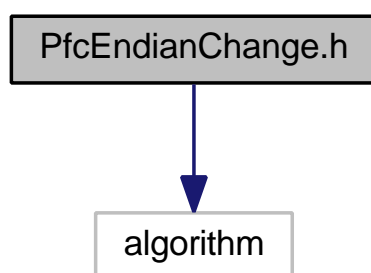
aics

[PfcDomain.h](#) で定義されています。

7.13 PfcEndianChange.h

```
#include <algorithm>
```

PfcEndianChange.h のインクルード依存関係図



関数

- `template<class T >`
`void endswap (T *objp)`

7.13.1 関数

7.13.1.1 `template<class T > void endswap (T * objp)`

PfcEndianChange.h の 9 行で定義されています。

```

9      {
10     unsigned char *memp = reinterpret_cast<unsigned char*>(objp);
11     reverse(memp, memp + sizeof(T));
12 }

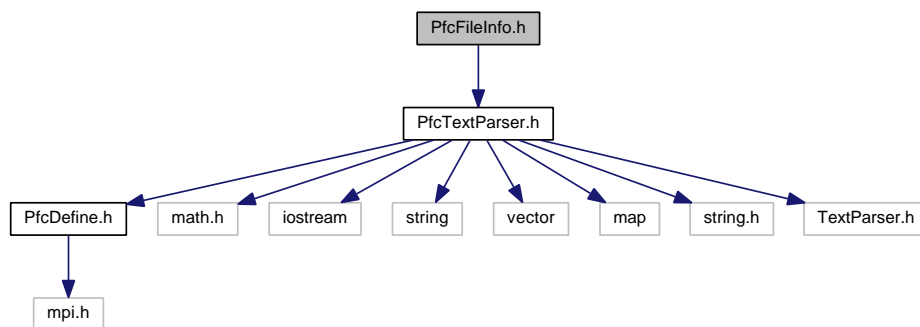
```

7.14 PfcFileInfo.h

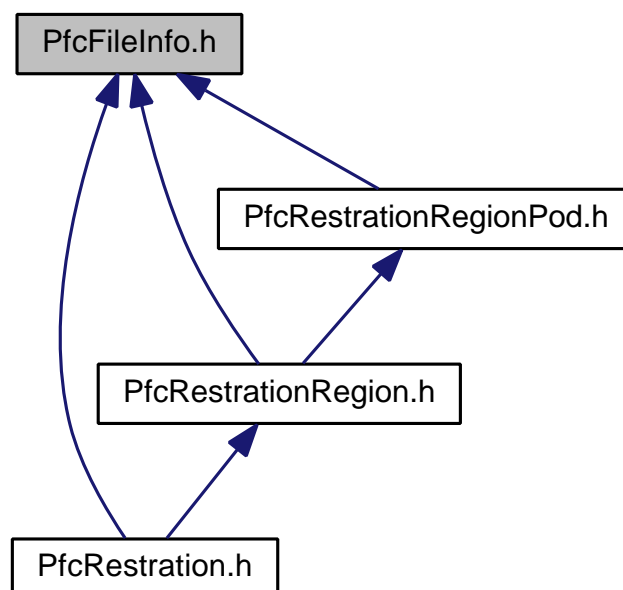
CPfcFileInfo Class Header.

```
#include "PfcTextParser.h"
```

PfcFileInfo.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 [CPfcFileInfo](#)

7.14.1 説明

[PfcFileInfo](#) Class Header.

作者

aics

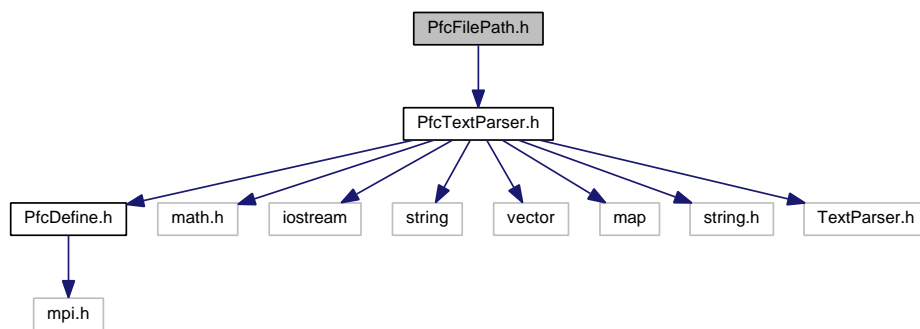
[PfcFileInfo.h](#) で定義されています。

7.15 PfcFilePath.h

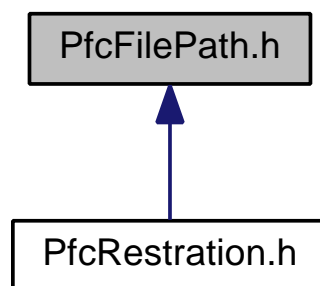
[CPfcFilePath](#) Class Header.

```
#include "PfcTextParser.h"
```

PfcFilePath.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 [CPfcFilePath](#)

7.15.1 説明

[CPfcFilePath](#) Class Header.

作者

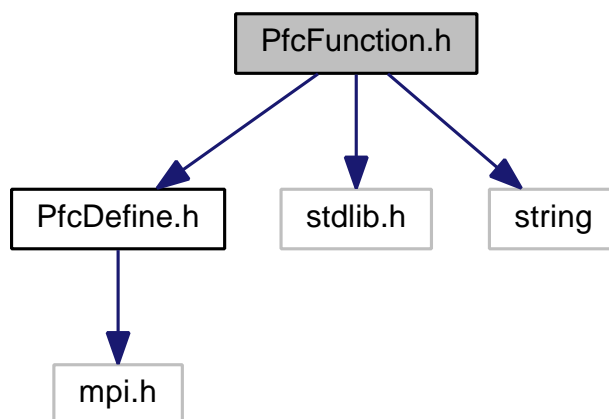
aics

[PfcFilePath.h](#) で定義されています。

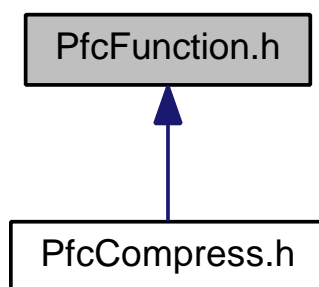
7.16 PfcFunction.h

```
#include "PfcDefine.h"  
#include <stdlib.h>  
#include <string>
```

PfcFunction.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 [CPfcFunction](#)

関数

- int [int_pow](#) (int x, int index)

7.16.1 関数

7.16.1.1 int int_pow (int x, int index) [inline]

PfcFunction.h の 23 行で定義されています。

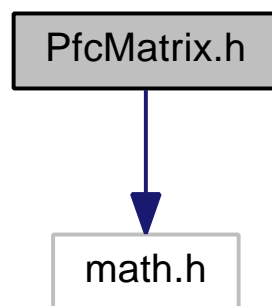
```

24 {
25     int result = 1;
26     for(int i = 0; i < index; i++)
27         result = result * x;
28     return result;
29 }
```

7.17 PfcMatrix.h

```
#include <math.h>
```

PfcMatrix.h のインクルード依存関係図



関数

- void **dgemm_** (char *transa, char *transb, int *m, int *n, int *k, double *alpha, double *A, int *lda, double *B, int *ldb, double *beta, double *C, int *ldc)
- void **dgemv_** (char *trans, int *m, int *n, double *alpha, double *A, int *lda, double *x, int *incx, double *beta, double *y, int *incy)
- int **dgeev_** (char *jobvl, char *jobvr, int *n, double *a, int *lda, double *wr, double *wi, double *vl, int *ldvl, double *vr, int *ldvr, double *work, int *lwork, int *info)
- void **PFC_dgemm** (int M, int K, double alpha, double *A, double *B, double beta, double *C)
dgemm (行列と行列の乗算) のラッパー関数
- void **PFC_dgemv** (int M, int N, double alpha, double *A, double *X, double beta, double *Y)
dgemv (行列-ベクトル積) のラッパー関数
- void **PFC_matrix_Trans_d** (double *dest, double *src, const int dest_row_size, const int dest_col_size)
次元入れ替え (一次元の密データ)
- double **PFC_ddot** (int N, double *X, double *Y)
内積
- void **PFC_dgeev** (double *H, int n, double *Er, double *Evecs)
dgeev (一般行列 (n 行 n 列の非対称行列) の固有値と左右の固有ベクトル) のラッパー関数
- void **PFC_dgeev** (double *H, int n, double *Er, double *Ei, double *Evecs)
- void **PFC_d_sort** (double *Er, double *Ei, double *Evecs, int N)
ソート (Er*Er+Ei*Ei) の絶対値の小さい順 (昇順) に並べる

7.17.1 関数

7.17.1.1 int dgeev_ (char * jobvl, char * jobvr, int * n, double * a, int * lda, double * wr, double * wi, double * vl, int * ldvl, double * vr, int * ldvr, double * work, int * lwork, int * info)

7.17.1.2 void dgemm_ (char * *transa*, char * *transb*, int * *m*, int * *n*, int * *k*, double * *alpha*, double * *A*, int * *ldA*, double * *B*, int * *ldB*, double * *beta*, double * *C*, int * *ldC*)

7.17.1.3 void dgemv_ (char * *trans*, int * *m*, int * *n*, double * *alpha*, double * *A*, int * *ldA*, double * *x*, int * *incx*, double * *beta*, double * *y*, int * *incy*)

7.17.1.4 void PFC_d_sort (double * *Er*, double * *Ei*, double * *Evecs*, int *N*)

ソート ($Er \cdot Er + Ei \cdot Ei$) の絶対値の小さい順 (昇順) に並べる

引数

in, out	$Er[N]$	
in, out	$Ei[N]$	
in, out	$Evecs[N][N]$	
in	N	

戻り値

なし

7.17.1.5 double PFC_ddot (int *N*, double * *X*, double * *Y*)

内積

引数

in	N	サイズ
in	X	$X[N]$
in	Y	$Y[N]$

戻り値

内積値

7.17.1.6 void PFC_dgeev (double * *H*, int *n*, double * *Er*, double * *Evecs*)

dgeev (一般行列 (n 行 n 列の非対称行列) の固有値と左右の固有ベクトル) のラッパー関数

引数

in	$H[n][n]$	
in	n	
out	$Er[n]$	
out	$Evecs[n][n]$	

戻り値

なし

7.17.1.7 void PFC_dgeev (double * *H*, int *n*, double * *Er*, double * *Ei*, double * *Evecs*)

7.17.1.8 void PFC_dgemm (int *M*, int *K*, double *alpha*, double * *A*, double * *B*, double *beta*, double * *C*)

dgemm (行列と行列の乗算) のラッパー関数

引数

in	M	
in	K	
in	α	
in	$A(*,M)$	
in	$B(*,K)$	
in	β	
out	$C(*,M)$	

戻り値

なし

7.17.1.9 void PFC_dgemv (int M , int N , double α , double * A , double * X , double β , double * Y)

dgemv (行列-ベクトル積) のラッパー関数

引数

in	M	
in	N	
in	α	
in	$A(M,N)$	
in	$X(N)$	
in	β	
out	$Y(N)$	

戻り値

なし

7.17.1.10 void PFC_matrix_Trans_d (double * $dest$, double * src , const int $dest_row_size$, const int $dest_col_size$)

次元入れ替え (一次元の密データ)

引数

out	$dest$	転送先
in	src	転送元 (ソース)
in	$dest_row$	転送先行数
in	$dest_col$	転送先列数

戻り値

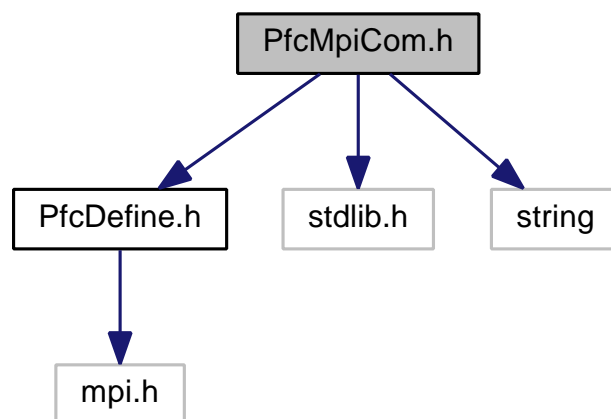
なし

7.18 PfcMpiCom.h

[CPfcMpiCom](#) MPI communication Class Header.

```
#include "PfcDefine.h"
#include <stdlib.h>
#include <string>
```

PfcMpiCom.h のインクルード依存関係図



構成

- class [CPfcMpiCom](#)

7.18.1 説明

[CPfcMpiCom](#) MPI communication Class Header.

作者

aics

[PfcMpiCom.h](#) で定義されています。

7.19 PfcPathUtil.h

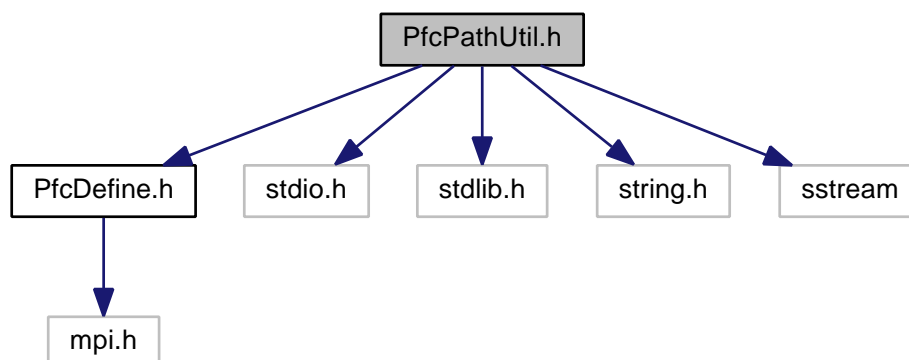
Path utility 関数

```

#include "PfcDefine.h"
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <sstream>

```

PfcPathUtil.h のインクルード依存関係図



ネームスペース

- [PFC](#)

Constant Groups

- [PFC](#)

マクロ定義

- `#define` [MAXPATHLEN](#) 512

関数

- `char` [PFC::PfcPath_getDelimChar](#) ()
- `std::string` [PFC::PfcPath_getDelimString](#) ()
- `bool` [PFC::PfcPath_hasDrive](#) (const `std::string` &path)
- `std::string` [PFC::vfvPath_emitDrive](#) (std::string &path)
- `bool` [PFC::PfcPath_isAbsolute](#) (const `std::string` &path)
- `std::string` [PFC::PfcPath_DirName](#) (const `std::string` &path, const `char` dc=[PfcPath_getDelimChar](#)())
- `std::string` [PFC::PfcPath_FileName](#) (const `std::string` &path, const `std::string` &addext=`std::string`(""), const `char` dc=[PfcPath_getDelimChar](#)())
- `std::string` [PFC::PfcPath_ConnectPath](#) (std::string dirName, std::string fname)
- [PFC::E_PFC_ERRORCODE](#) [PFC::MakeDirectory](#) (const `std::string` &path)

7.19.1 説明

Path utility 関数

作者

aics

[PfcPathUtil.h](#) で定義されています。

7.19.2 マクロ定義

7.19.2.1 `#define` [MAXPATHLEN](#) 512

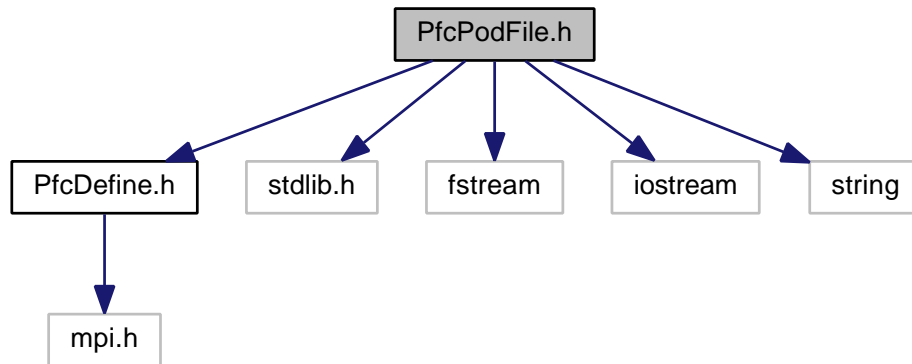
[PfcPathUtil.h](#) の 32 行で定義されています。

7.20 [PfcPodFile.h](#)

[CPfcPodFile](#) Class Header 基底ファイル、係数ファイル用

```
#include "PfcDefine.h"
#include <stdlib.h>
#include <fstream>
#include <iostream>
#include <string>
```

PfcPodFile.h のインクルード依存関係図



構成

- class [CPfcPodFile](#)

7.20.1 説明

[CPfcPodFile](#) Class Header 基底ファイル、係数ファイル用

作者

aics

[PfcPodFile.h](#) で定義されています。

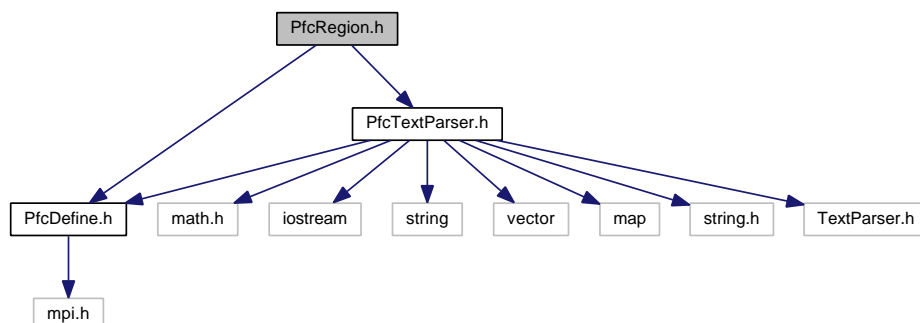
7.21 PfcRegion.h

PfcRegion Class Header.

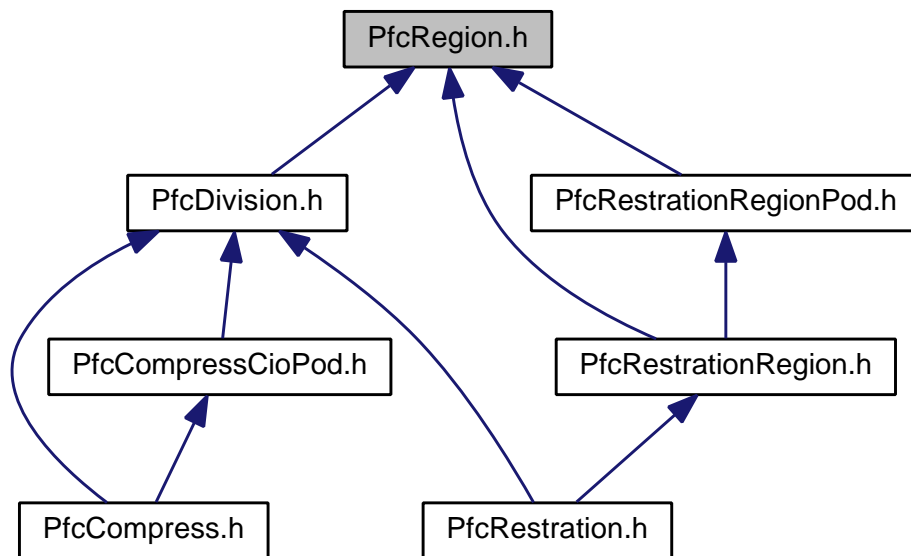
```
#include "PfcDefine.h"
```

```
#include "PfcTextParser.h"
```

PfcRegion.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 [CPfcRegion](#)

7.21.1 説明

PfcRegion Class Header.

作者

aics

[PfcRegion.h](#) で定義されています。

7.22 PfcRestration.h

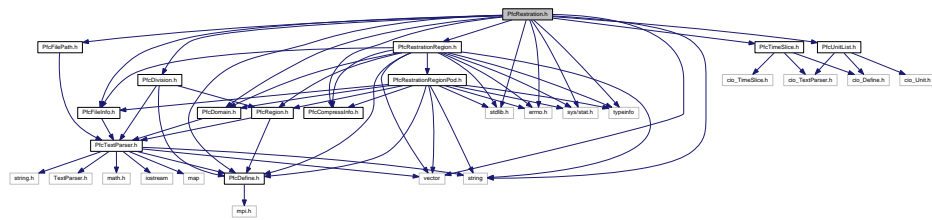
[CPfcRestration](#) Class Header.

```

#include "PfcDefine.h"
#include <stdlib.h>
#include <errno.h>
#include <sys/stat.h>
#include <typeinfo>
#include <vector>
#include <string>
#include "PfcFileInfo.h"
#include "PfcCompressInfo.h"
#include "PfcFilePath.h"
#include "PfcUnitList.h"
#include "PfcTimeSlice.h"
#include "PfcDomain.h"
#include "PfcDivision.h"
#include "PfcRestrationRegion.h"

```

PfcRestration.h のインクルード依存関係図



構成

- class CPfcRestriction

7.22.1 説明

CPfcRestration Class Header.

作者

aics

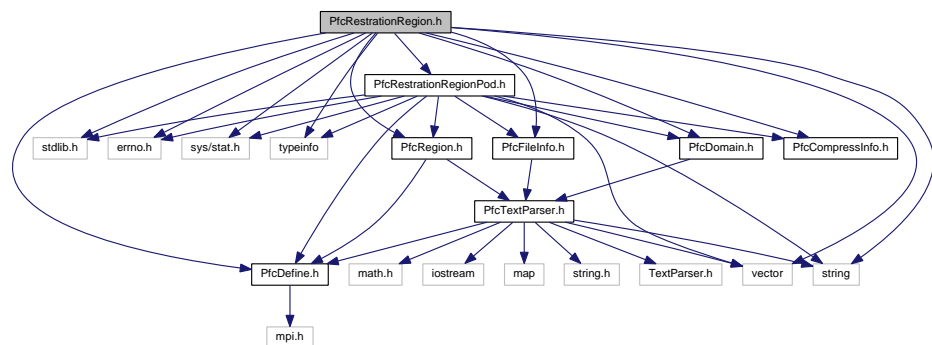
PfcRestrstration.h で定義されています。

7.23 PfcRestrictionRegion.h

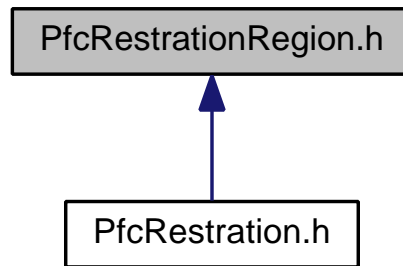
CPfcRestrationRegion Class Header.

```
#include "PfcDefine.h"
#include <stdlib.h>
#include <errno.h>
#include <sys/stat.h>
#include <typeinfo>
#include <vector>
#include <string>
#include "PfcFileInfo.h"
#include "PfcCompressInfo.h"
#include "PfcDomain.h"
#include "PfcRegion.h"
#include "PfcRestrictionRegionPod.h"
```

PfcRestrationRegion.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 [CPfcRestrationRegion](#)

7.23.1 説明

[CPfcRestrationRegion](#) Class Header. 分割領域クラス

作者

aics

[PfcRestrationRegion.h](#) で定義されています。

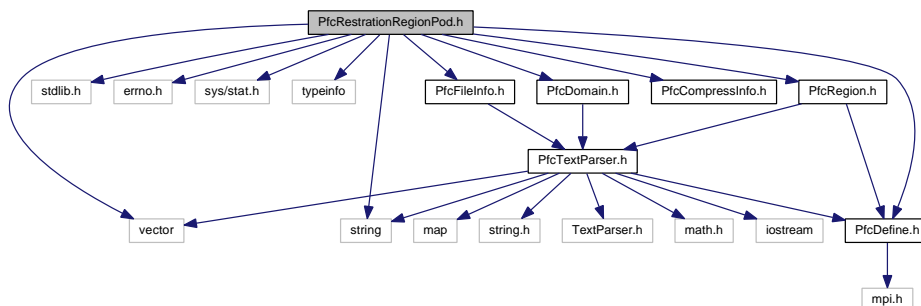
7.24 PfcRestrationRegionPod.h

[CPfcRestrationRegionPod](#) Class Header.

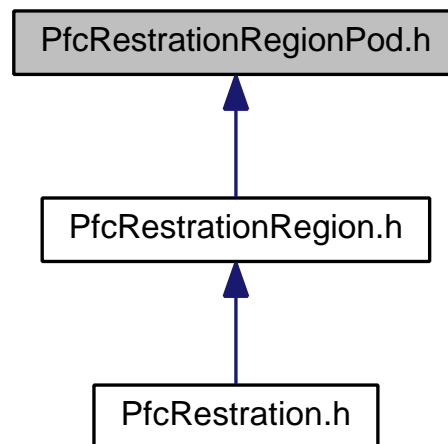
```

#include "PfcDefine.h"
#include <stdlib.h>
#include <errno.h>
#include <sys/stat.h>
#include <typeinfo>
#include <vector>
#include <string>
#include "PfcFileInfo.h"
#include "PfcCompressInfo.h"
#include "PfcDomain.h"
#include "PfcRegion.h"
  
```

PfcRestrationRegionPod.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 [CPfcRestrationRegionPod](#)

7.24.1 説明

[CPfcRestrationRegionPod](#) Class Header. 分割領域クラス（１次元で操作する）

作者

aics

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

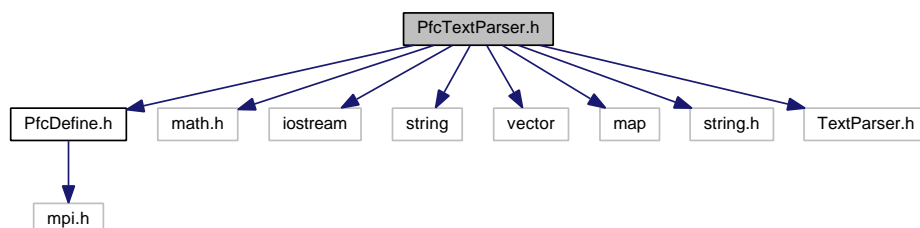
7.25 PfcTextParser.h

TextParser Control class Header.

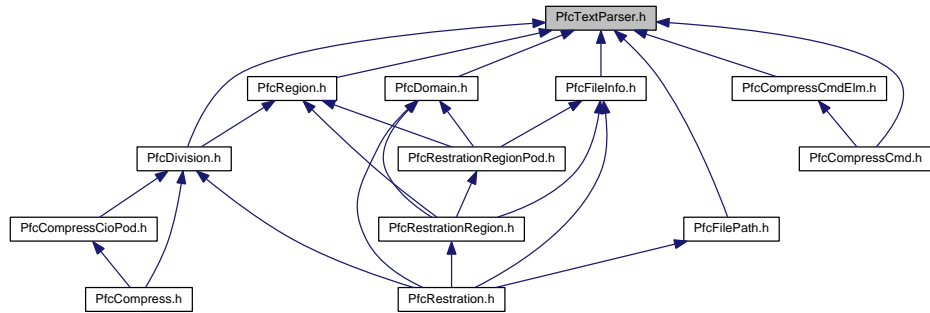
```

#include "PfcDefine.h"
#include <math.h>
#include <iostream>
#include <string>
#include <vector>
#include <map>
#include "string.h"
#include "TextParser.h"
  
```

PfcTextParser.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 [CPfcTextParser](#)

7.25.1 説明

TextParser Control class Header.

作者

aics

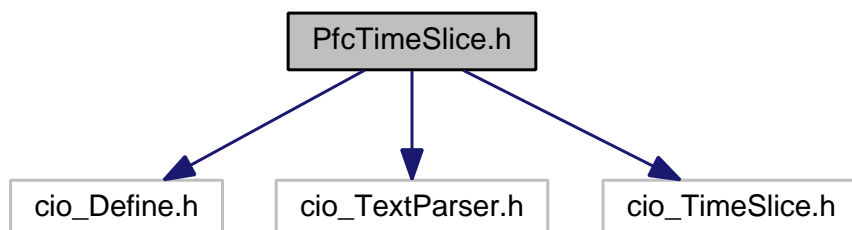
[PfcTextParser.h](#) で定義されています。

7.26 PfcTimeSlice.h

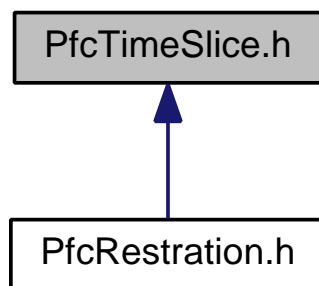
[CPfcTimeSlice](#) Class Header.

```
#include "cio_Define.h"
#include "cio_TextParser.h"
#include "cio_TimeSlice.h"
```

PfcTimeSlice.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 [CPfcTimeSlice](#)

7.26.1 説明

[CPfcTimeSlice](#) Class Header.

作者

aics

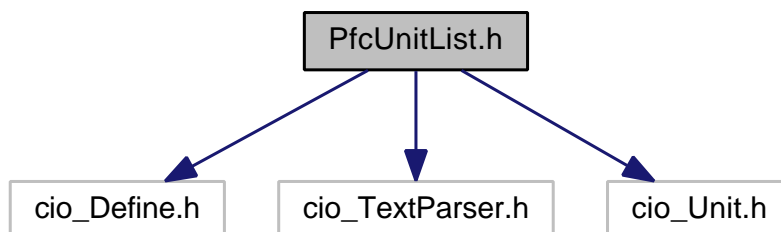
[PfcTimeSlice.h](#) で定義されています。

7.27 PfcUnitList.h

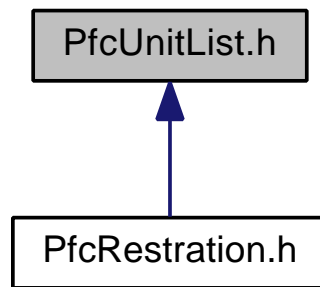
CPfcPfcUnitList Class Header.

```
#include "cio_Define.h"
#include "cio_TextParser.h"
#include "cio_Unit.h"
```

PfcUnitList.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 [CPfcUnitList](#)

7.27.1 説明

CPfcPfcUnitList Class Header.

作者

aics

[PfcUnitList.h](#) で定義されています。

7.28 PfcVersion.h

マクロ定義

- #define [PFC_VERSION_NO](#) "0.7.1"
- #define [PFC_REVISION](#) "20140209_1700"
- #define [POD_COMPRESS_VERSION](#) "1.0.0"

7.28.1 マクロ定義

7.28.1.1 #define [PFC_REVISION](#) "20140209_1700"

PFC ライブラリのリビジョン

PfcVersion.h の 21 行で定義されています。

7.28.1.2 #define [PFC_VERSION_NO](#) "0.7.1"

PFC ライブラリのバージョン

PfcVersion.h の 18 行で定義されています。

7.28.1.3 #define [POD_COMPRESS_VERSION](#) "1.0.0"

[PFC](#) POD 圧縮形式バージョン

PfcVersion.h の 24 行で定義されています。

Index

- ~CPfcCoeffTree
 - CPfcCoeffTree, [21](#)
- ~CPfcCoeffTreeNode
 - CPfcCoeffTreeNode, [23](#)
- ~CPfcCompress
 - CPfcCompress, [25](#)
- ~CPfcCompressCioPod
 - CPfcCompressCioPod, [32](#)
- ~CPfcCompressCmd
 - CPfcCompressCmd, [35](#)
- ~CPfcCompressCmdElm
 - CPfcCompressCmdElm, [37](#)
- ~CPfcCompressInfo
 - CPfcCompressInfo, [39](#)
- ~CPfcCompressPod
 - CPfcCompressPod, [43](#)
- ~CPfcDivision
 - CPfcDivision, [50](#)
- ~CPfcDomain
 - CPfcDomain, [54](#)
- ~CPfcFileInfo
 - CPfcFileInfo, [56](#)
- ~CPfcFilePath
 - CPfcFilePath, [58](#)
- ~CPfcRegion
 - CPfcRegion, [72](#)
- ~CPfcRestration
 - CPfcRestration, [75](#)
- ~CPfcRestrationRegion
 - CPfcRestrationRegion, [81](#)
- ~CPfcRestrationRegionPod
 - CPfcRestrationRegionPod, [84](#)
- ~CPfcTextParser
 - CPfcTextParser, [89](#)
- ~CPfcTimeSlice
 - CPfcTimeSlice, [93](#)
- ~CPfcUnitList
 - CPfcUnitList, [95](#)
- _PFC_COMPRESS_CMD_ELM_CNTL_
 - PfcCompressCmdElm.h, [103](#)
- _PFC_IDX_IJ
 - PfcDefine.h, [108](#)
- _PFC_IDX_IJK
 - PfcDefine.h, [108](#)
- _PFC_IDX_IJKN
 - PfcDefine.h, [109](#)
- _PFC_IDX_NIJ
 - PfcDefine.h, [109](#)
- _PFC_IDX_NIJK
 - PfcDefine.h, [110](#)
- _PFC_TAB_STR
 - PfcDefine.h, [110](#)
- _PFC_WRITE_TAB
 - PfcDefine.h, [110](#)
- Alloc2D
 - CPfcFunction, [60](#)
- AndMinMax
 - CPfcFunction, [61](#)
- CPfcCioDfiUtil, [19](#)
 - GetDfiDomain, [19](#)
 - GetDfiInfo, [20](#)
 - GetDfiTimeStepList, [20](#)
- CPfcCoeffTree, [20](#)
 - ~CPfcCoeffTree, [21](#)
 - CPfcCoeffTree, [21](#)
 - construct_array, [21](#)
 - construct_child, [21](#)
 - construct_tree, [21](#)
 - CPfcCoeffTree, [21](#)
 - data, [22](#)
 - data_size, [22](#)
 - deep_transerval, [21](#)
 - get_position, [21](#)
 - m_numLayer, [22](#)
 - m_numTimeStep, [22](#)
 - position, [22](#)
 - position_interval, [22](#)
 - print_node, [21](#)
 - print_tree, [21](#)
 - read_data, [21](#)
 - root, [22](#)
 - stand_space, [22](#)
 - standard_timestep_node, [21](#)
- CPfcCoeffTreeNode, [22](#)
 - ~CPfcCoeffTreeNode, [23](#)
 - CPfcCoeffTreeNode, [23](#)
 - CPfcCoeffTreeNode, [23](#)
 - is_leaf, [23](#)
 - left_child, [23](#)
 - right_child, [23](#)
 - value, [23](#)
- CPfcCompress, [24](#)
 - ~CPfcCompress, [25](#)
 - CPfcCompress, [25](#)
 - CPfcCompress, [25](#)
 - GatherAndCreateDivisionInfo, [26](#)
 - Init, [27](#)

- m_arrayShape, 28
- m_comm, 28
- m_compressError, 28
- m_compressFormat, 28
- m_dataType, 28
- m_dfiFilePath, 28
- m_domainDivision, 28
- m_endStep, 29
- m_endianType, 29
- m_fileFormat, 29
- m_myIDinRegion, 29
- m_myRankID, 29
- m_numComponent, 29
- m_numParallel, 29
- m_numRank, 29
- m_numRegion, 29
- m_numStep, 29
- m_optFlags, 29
- m_outDirPath, 29
- m_pCioPod, 30
- m_pDfiIN, 30
- m_prefix, 30
- m_regionHead, 30
- m_regionID, 30
- m_regionMasterRankID, 30
- m_regionSize, 30
- m_regionTail, 30
- m_startStep, 30
- m_stepList, 30
- WriteData, 27
- WriteIndexPfcFile, 28
- WriteProcFile, 28
- CPfcCompressCioPod, 31
 - ~CPfcCompressCioPod, 32
 - CPfcCompressCioPod, 32
 - CPfcCompressCioPod, 32
 - Init, 32
 - m_comm, 33
 - m_compressError, 33
 - m_myRankID, 33
 - m_numRank, 33
 - m_numRegion, 34
 - m_numStep, 34
 - m_optFlags, 34
 - m_outDirPath, 34
 - m_pDfiIN, 34
 - m_pFlowData, 34
 - m_pPod, 34
 - m_pStepList, 34
 - m_regionHead, 34
 - m_regionTail, 34
 - ReadCioFile, 32
 - ReadCioFile_IJKN, 33
 - WriteData, 33
- CPfcCompressCmd, 35
 - ~CPfcCompressCmd, 35
 - CPfcCompressCmd, 35
 - CPfcCompressCmd, 35
- Execute, 35
- m_cmdList, 36
- m_compressCntlPath, 36
- m_domainDivision, 36
- ReadCompressCntl, 35
- CPfcCompressCmdElm, 36
 - ~CPfcCompressCmdElm, 37
 - CPfcCompressCmdElm, 37
 - CPfcCompressCmdElm, 37
 - m_compressError, 37
 - m_compressFormat, 37
 - m_dfiFilePath, 37
 - m_endStep, 37
 - m_optFlags, 38
 - m_optSave, 38
 - m_outDirPath, 38
 - m_procFileSave, 38
 - m_startStep, 38
 - ReadElm, 37
- CPfcCompressInfo, 38
 - ~CPfcCompressInfo, 39
 - CPfcCompressInfo, 39
 - CPfcCompressInfo, 39
 - m_calculatedLayer, 41
 - m_compressError, 41
 - m_compressFormat, 41
 - m_endStep, 41
 - m_startStep, 41
 - m_version, 41
 - Read, 39
 - Write, 39
- CPfcCompressPod, 41
 - ~CPfcCompressPod, 43
 - CPfcCompressPod, 43
 - CalcPodBase, 43
 - CalcPodCoef, 43
 - CheckFinish, 43
 - CPfcCompressPod, 43
 - GetCalculatedLayer, 45
 - Init, 45
 - m_calculatedLayer, 47
 - m_comm, 47
 - m_compressError, 47
 - m_curEvaluateError, 47
 - m_curNumSize, 47
 - m_curPodBaseSize, 47
 - m_layerNo, 47
 - m_maxLayer, 48
 - m_myIDinRegion, 48
 - m_myNumStep, 48
 - m_myRankID, 48
 - m_myStartStepPos, 48
 - m_numParallel, 48
 - m_numRank, 48
 - m_numRegion, 48
 - m_numStep, 48
 - m_optFlags, 48
 - m_outDirPath, 48

- m_pCoef_a, [48](#)
- m_pFlowData, [49](#)
- m_pPod_base_r, [49](#)
- m_prefix, [49](#)
- m_regionID, [49](#)
- m_regionMasterRankID, [49](#)
- m_regionMaxStep, [49](#)
- m_regionSize, [49](#)
- Output, [45](#)
- SwapBinaryData, [45](#)
- WriteData, [47](#)
- WritePodBaseFile, [47](#)
- WritePodBaseFile_Debug, [47](#)
- WritePodCoefFile, [47](#)
- CPfcDivision, [49](#)
 - ~CPfcDivision, [50](#)
 - CPfcDivision, [50](#)
 - CheckReadRegion, [50](#)
 - CPfcDivision, [50](#)
 - CreateMinMax, [51](#)
 - m_MinMax, [51](#)
 - m_MinMax_data, [51](#)
 - m_numRegion, [51](#)
 - m_regionList, [52](#)
 - Read, [51](#)
 - Write, [51](#)
- CPfcDomain, [52](#)
 - ~CPfcDomain, [54](#)
 - CPfcDomain, [52](#)
 - CPfcDomain, [52](#)
 - m_globalDivision, [54](#)
 - m_globalOrigin, [54](#)
 - m_globalRegion, [54](#)
 - m_globalVoxel, [55](#)
 - Read, [54](#)
 - Write, [54](#)
- CPfcFileInfo, [55](#)
 - ~CPfcFileInfo, [56](#)
 - CPfcFileInfo, [56](#)
 - CPfcFileInfo, [56](#)
 - m_arrayShape, [57](#)
 - m_dataType, [57](#)
 - m_dirPath, [57](#)
 - m_endianType, [57](#)
 - m_fileFormat, [57](#)
 - m_guideCell, [57](#)
 - m_numComponent, [57](#)
 - m_prefix, [57](#)
 - Read, [56](#)
 - Write, [56](#)
- CPfcFilePath, [57](#)
 - ~CPfcFilePath, [58](#)
 - CPfcFilePath, [58](#)
 - CPfcFilePath, [58](#)
 - m_dfiPath, [59](#)
 - m_procPath, [59](#)
 - Read, [58](#)
 - Write, [59](#)
- CPfcFunction, [59](#)
 - Alloc2D, [60](#)
 - AndMinMax, [61](#)
 - CalcHeadTail, [61](#)
 - CalcHeadTail_block, [61](#)
 - CheckLapMinMax, [62](#)
 - CheckPntInMinMax, [62](#)
 - CopyData, [62](#)
 - GetPodBaseIndex, [63](#)
 - GetPodMaxLayer, [63](#)
 - GetPodParallel, [63](#)
 - GetPodRegionID, [64](#)
 - GetPodStepInfo, [64](#)
- CPfcMpiCom, [64](#)
 - GatherDataDouble, [65](#)
 - GatherDataInt, [65](#)
 - GatherV_DataDouble, [66](#)
 - GetMaxInt, [66](#)
- CPfcPodFile, [66](#)
 - CloseBaseFile, [67](#)
 - OpenBaseFile, [68](#)
 - ReadBaseFile, [68](#)
 - ReadBaseFile1Elm, [68](#)
 - ReadBaseFile_Debug, [69](#)
 - ReadBaseFileHeader, [69](#)
 - ReadCoefFile, [70](#)
 - WriteBaseFile, [70](#)
 - WriteBaseFile_Debug, [71](#)
 - WriteCoefFile, [71](#)
- CPfcRegion, [72](#)
 - ~CPfcRegion, [72](#)
 - CPfcRegion, [72](#)
 - CPfcRegion, [72](#)
 - m_headIndex, [73](#)
 - m_regionID, [73](#)
 - m_tailIndex, [73](#)
 - m_voxelSize, [73](#)
 - Read, [72](#)
 - Write, [73](#)
- CPfcRestration, [73](#)
 - ~CPfcRestration, [75](#)
 - CPfcRestration, [75](#)
 - CheckCompressDataOnMem, [75](#)
 - CPfcRestration, [75](#)
 - DeleteCompressDataOnMem, [75](#)
 - GetGlobalVoxel, [75](#)
 - GetHeadTail, [77](#)
 - GetTimeStepList, [77](#)
 - Init, [77](#)
 - LoadCompressDataOnMem, [77](#)
 - m_bLoadCompressData, [78](#)
 - m_compressInfo, [78](#)
 - m_division, [79](#)
 - m_domain, [79](#)
 - m_fileInfo, [79](#)
 - m_filePath, [79](#)
 - m_head, [79](#)
 - m_numRegion, [79](#)

- m_numTimeStep, 79
- m_pfcFilePath, 79
- m_regionIdList, 79
- m_regionList, 79
- m_tail, 79
- m_timeSlice, 79
- m_timeStepList, 80
- m_unitList, 80
- ReadData, 78
- CPfcRestrationRegion, 80
 - ~CPfcRestrationRegion, 81
 - CPfcRestrationRegion, 81
 - CPfcRestrationRegion, 81
 - DeleteCompressDataOnMem, 81
 - Init, 81
 - LoadCompressDataOnMem, 81
 - m_bLoadCompressData, 82
 - m_compressForm, 82
 - m_pCompressInfo, 82
 - m_pDomain, 83
 - m_pFileInfo, 83
 - m_pRegion, 83
 - m_pvTimeStepList, 83
 - pPod, 83
 - ReadData, 82
 - ReadDataInRange, 82
- CPfcRestrationRegionPod, 83
 - ~CPfcRestrationRegionPod, 84
 - CPfcRestrationRegionPod, 84
 - CPfcRestrationRegionPod, 84
 - DeleteCompressDataOnMem, 84
 - ExpandData, 84, 85
 - ExpandDataOnFile, 85
 - GetExpandRowInfo, 85
 - Init, 86
 - LoadCompressDataOnMem, 86
 - m_arrayShape, 87
 - m_bLoadCompressData, 87
 - m_bSingle, 87
 - m_dirPath, 87
 - m_numCalculatedLayer, 87
 - m_numCoef, 87
 - m_numComponent, 87
 - m_numParallel, 88
 - m_numSize, 88
 - m_numStep, 88
 - m_pBaseData, 88
 - m_pBaseSizes, 88
 - m_pCoefData, 88
 - m_pIndexBase, 88
 - m_pIndexCoef, 88
 - m_prefix, 88
 - m_regionID, 88
 - ReadData, 86
 - ReadFieldData, 87
- CPfcTextParser, 89
 - ~CPfcTextParser, 89
 - CPfcTextParser, 89
- chkLabel, 90
- chkNode, 90
- countLabels, 90
- CPfcTextParser, 89
- GetNodeStr, 90
- getTPInstance, 90
- GetValue, 90, 91
- GetVector, 91
- readTPfile, 92
- remove, 92
- tp, 92
- CPfcTimeSlice, 92
 - ~CPfcTimeSlice, 93
 - CPfcTimeSlice, 93
 - CPfcTimeSlice, 93
 - Select, 93
- CPfcUnitList, 94
 - ~CPfcUnitList, 95
 - CPfcUnitList, 95
 - CPfcUnitList, 95
- CalcHeadTail
 - CPfcFunction, 61
- CalcHeadTail_block
 - CPfcFunction, 61
- CalcPodBase
 - CPfcCompressPod, 43
- CalcPodCoef
 - CPfcCompressPod, 43
- CheckCompressDataOnMem
 - CPfcRestration, 75
- CheckFinish
 - CPfcCompressPod, 43
- CheckLapMinMax
 - CPfcFunction, 62
- CheckPntInMinMax
 - CPfcFunction, 62
- CheckReadRegion
 - CPfcDivision, 50
- chkLabel
 - CPfcTextParser, 90
- chkNode
 - CPfcTextParser, 90
- CloseBaseFile
 - CPfcPodFile, 67
- construct_array
 - CPfcCoeffTree, 21
- construct_child
 - CPfcCoeffTree, 21
- construct_tree
 - CPfcCoeffTree, 21
- CopyData
 - CPfcFunction, 62
- countLabels
 - CPfcTextParser, 90
- CreateMinMax
 - CPfcDivision, 51
- D_PFC_BIG
 - PfcDefine.h, 111

- D_PFC_COMPRESS_ERROR_DEFAULT
 - PfcDefine.h, [111](#)
- D_PFC_EPSILON
 - PfcDefine.h, [111](#)
- D_PFC_FLOAT32
 - PfcDefine.h, [111](#)
- D_PFC_FLOAT64
 - PfcDefine.h, [111](#)
- D_PFC_IJNK
 - PfcDefine.h, [111](#)
- D_PFC_LITTLE
 - PfcDefine.h, [111](#)
- D_PFC_NIJK
 - PfcDefine.h, [111](#)
- D_PFC_OPT_FLAG_SAVE
 - PfcDefine.h, [111](#)
- DEBUG_PRINT
 - PfcDefine.h, [111](#)
- DEBUG_PRINT_AT
 - PfcDefine.h, [111](#)
- data
 - CPfcCoeffTree, [22](#)
- data_size
 - CPfcCoeffTree, [22](#)
- deep_transerval
 - CPfcCoeffTree, [21](#)
- DeleteCompressDataOnMem
 - CPfcRestriction, [75](#)
 - CPfcRestrictionRegion, [81](#)
 - CPfcRestrictionRegionPod, [84](#)
- dgeev_
 - PfcMatrix.h, [118](#)
- dgemm_
 - PfcMatrix.h, [118](#)
- dgemv_
 - PfcMatrix.h, [119](#)
- E_PFC_ARRAYSHAPE
 - PFC, [10](#)
- E_PFC_ARRAYSHAPE_UNKNOWN
 - PFC, [10](#)
- E_PFC_BIG
 - PFC, [12](#)
- E_PFC_COMPRESS_FMT_POD
 - PFC, [11](#)
- E_PFC_COMPRESS_FMT_UNKNOWN
 - PFC, [11](#)
- E_PFC_COMPRESS_FORMAT
 - PFC, [10](#)
- E_PFC_DTYPE
 - PFC, [11](#)
- E_PFC_DTYPE_UNKNOWN
 - PFC, [11](#)
- E_PFC_ENDIANTYPE
 - PFC, [11](#)
- E_PFC_ENDIANTYPE_UNKNOWN
 - PFC, [11](#)
- E_PFC_ERROR
 - PFC, [12](#)
- E_PFC_ERROR_OUT_OF_RANGE
 - PFC, [13](#)
- E_PFC_ERROR_PFC_COMPRESSFORMAT
 - PFC, [13](#)
- E_PFC_ERROR_READ_CNTL_COMPRESS_FMT
 - PFC, [12](#)
- E_PFC_ERROR_READ_CNTL_DFI_PATH
 - PFC, [12](#)
- E_PFC_ERROR_READ_CNTL_DOMAINDIVISION
 - PFC, [12](#)
- E_PFC_ERROR_READ_CNTL_ITEMCNTL
 - PFC, [12](#)
- E_PFC_ERROR_READ_CNTL_NO_ITEM
 - PFC, [12](#)
- E_PFC_ERROR_READ_CNTL_OUTDIR_PATH
 - PFC, [12](#)
- E_PFC_ERROR_READ_CNTL_PROCFILE_SAVE
 - PFC, [12](#)
- E_PFC_ERROR_READ_CNTLFILE_OPENERERROR
 - PFC, [12](#)
- E_PFC_ERROR_READ_COMPRESSINFO
 - PFC, [12](#)
- E_PFC_ERROR_READ_DIVISION
 - PFC, [13](#)
- E_PFC_ERROR_READ_DOMAIN
 - PFC, [13](#)
- E_PFC_ERROR_READ_FILEINFO
 - PFC, [12](#)
- E_PFC_ERROR_READ_FILEPATH
 - PFC, [12](#)
- E_PFC_ERROR_READ_INDEXFILE_OPENERERROR
 - PFC, [12](#)
- E_PFC_ERROR_READ_PFC_ARRAYSHAPE
 - PFC, [12](#)
- E_PFC_ERROR_READ_PFC_CALCULATEDLAYER
 - PFC, [12](#)
- E_PFC_ERROR_READ_PFC_COMPONENT
 - PFC, [12](#)
- E_PFC_ERROR_READ_PFC_COMPRESSERROR
 - PFC, [12](#)
- E_PFC_ERROR_READ_PFC_COMPRESSFORMAT
 - PFC, [12](#)
- E_PFC_ERROR_READ_PFC_DATATYPE
 - PFC, [12](#)
- E_PFC_ERROR_READ_PFC_DFI_PATH
 - PFC, [12](#)
- E_PFC_ERROR_READ_PFC_DIRECTORYPATH
 - PFC, [12](#)
- E_PFC_ERROR_READ_PFC_ENDIAN
 - PFC, [12](#)
- E_PFC_ERROR_READ_PFC_ENDSTEP
 - PFC, [12](#)
- E_PFC_ERROR_READ_PFC_FILEFORMAT
 - PFC, [12](#)
- E_PFC_ERROR_READ_PFC_GLOBALDIVISION
 - PFC, [13](#)
- E_PFC_ERROR_READ_PFC_GLOBALORIGIN
 - PFC, [13](#)

- E_PFC_ERROR_READ_PFC_GLOBALREGION
PFC, [13](#)
- E_PFC_ERROR_READ_PFC_GLOBALVOXEL
PFC, [13](#)
- E_PFC_ERROR_READ_PFC_GUIDECELL
PFC, [12](#)
- E_PFC_ERROR_READ_PFC_NO_REGION
PFC, [13](#)
- E_PFC_ERROR_READ_PFC_PFCPROCESS
PFC, [12](#)
- E_PFC_ERROR_READ_PFC_PREFIX
PFC, [12](#)
- E_PFC_ERROR_READ_PFC_REGION_HEADINDEX
PFC, [13](#)
- E_PFC_ERROR_READ_PFC_REGION_ID
PFC, [13](#)
- E_PFC_ERROR_READ_PFC_REGION_TAILINDEX
PFC, [13](#)
- E_PFC_ERROR_READ_PFC_REGION_VOXELSIZE
PFC, [13](#)
- E_PFC_ERROR_READ_PFC_STARTSTEP
PFC, [12](#)
- E_PFC_ERROR_READ_PFC_VERSION
PFC, [12](#)
- E_PFC_ERROR_READ_PODBASE_DATA
PFC, [13](#)
- E_PFC_ERROR_READ_PODBASE_HEADER
PFC, [13](#)
- E_PFC_ERROR_READ_PODBASE_OPENERERROR
PFC, [13](#)
- E_PFC_ERROR_READ_PODCOEF_DATA
PFC, [13](#)
- E_PFC_ERROR_READ_PODCOEF_OPENERERROR
PFC, [13](#)
- E_PFC_ERROR_READ_PROCFILE_OPENERERROR
PFC, [13](#)
- E_PFC_ERROR_READ_TIMESLICE
PFC, [12](#)
- E_PFC_ERROR_READ_UNITLIST
PFC, [12](#)
- E_PFC_ERROR_WRITE_COMPRESSINFO
PFC, [13](#)
- E_PFC_ERROR_WRITE_DIVISION
PFC, [13](#)
- E_PFC_ERROR_WRITE_DOMAIN
PFC, [13](#)
- E_PFC_ERROR_WRITE_FILEINFO
PFC, [13](#)
- E_PFC_ERROR_WRITE_FILEPATH
PFC, [13](#)
- E_PFC_ERROR_WRITE_INDEXFILE_OPENERERROR
PFC, [13](#)
- E_PFC_ERROR_WRITE_INDEXFILENAME_EMPTY
PFC, [12](#)
- E_PFC_ERROR_WRITE_PFC_DATATYPE
PFC, [13](#)
- E_PFC_ERROR_WRITE_PODBASE_DATA
PFC, [13](#)
- E_PFC_ERROR_WRITE_PODBASE_OPENERERROR
PFC, [13](#)
- E_PFC_ERROR_WRITE_PODCOEF_DATA
PFC, [13](#)
- E_PFC_ERROR_WRITE_PODCOEF_OPENERERROR
PFC, [13](#)
- E_PFC_ERROR_WRITE_PROCFILE_OPENERERROR
PFC, [13](#)
- E_PFC_ERROR_WRITE_TIMESLICE
PFC, [13](#)
- E_PFC_ERROR_WRITE_UNITLIST
PFC, [13](#)
- E_PFC_ERRORCODE
PFC, [12](#)
- E_PFC_FLOAT32
PFC, [11](#)
- E_PFC_FLOAT64
PFC, [11](#)
- E_PFC_IJKN
PFC, [10](#)
- E_PFC_INT16
PFC, [11](#)
- E_PFC_INT32
PFC, [11](#)
- E_PFC_INT64
PFC, [11](#)
- E_PFC_INT8
PFC, [11](#)
- E_PFC_LITTLE
PFC, [11](#)
- E_PFC_NIJK
PFC, [10](#)
- E_PFC_SUCCESS
PFC, [12](#)
- E_PFC_UINT16
PFC, [11](#)
- E_PFC_UINT32
PFC, [11](#)
- E_PFC_UINT64
PFC, [11](#)
- E_PFC_UINT8
PFC, [11](#)
- endswap
PfcEndianChange.h, [115](#)
- Execute
CPfcCompressCmd, [35](#)
- ExpandData
CPfcRestrstrationRegionPod, [84](#), [85](#)
- ExpandDataOnFile
CPfcRestrstrationRegionPod, [85](#)
- GatherAndCreateDivisionInfo
CPfcCompress, [26](#)
- GatherDataDouble
CPfcMpiCom, [65](#)
- GatherDataInt
CPfcMpiCom, [65](#)
- GatherV_DataDouble
CPfcMpiCom, [66](#)

- get_position
 - CPfcCoeffTree, 21
- GetCalculatedLayer
 - CPfcCompressPod, 45
- GetDfiDomain
 - CPfcCioDfiUtil, 19
- GetDfiInfo
 - CPfcCioDfiUtil, 20
- GetDfiTimeStepList
 - CPfcCioDfiUtil, 20
- GetExpandRowInfo
 - CPfcRestrationRegionPod, 85
- GetGlobalVoxel
 - CPfcRestration, 75
- GetHeadTail
 - CPfcRestration, 77
- GetMaxInt
 - CPfcMpiCom, 66
- GetNodeStr
 - CPfcTextParser, 90
- GetPodBaseIndex
 - CPfcFunction, 63
- GetPodMaxLayer
 - CPfcFunction, 63
- GetPodParallel
 - CPfcFunction, 63
- GetPodRegionID
 - CPfcFunction, 64
- GetPodStepInfo
 - CPfcFunction, 64
- getTPInstance
 - CPfcTextParser, 90
- GetTimeStepList
 - CPfcRestration, 77
- GetValue
 - CPfcTextParser, 90, 91
- GetVector
 - CPfcTextParser, 91
- Init
 - CPfcCompress, 27
 - CPfcCompressCioPod, 32
 - CPfcCompressPod, 45
 - CPfcRestration, 77
 - CPfcRestrationRegion, 81
 - CPfcRestrationRegionPod, 86
- int_pow
 - PfcFunction.h, 117
- is_leaf
 - CPfcCoeffTreeNode, 23
- left_child
 - CPfcCoeffTreeNode, 23
- LoadCompressDataOnMem
 - CPfcRestration, 77
 - CPfcRestrationRegion, 81
 - CPfcRestrationRegionPod, 86
- m_MinMax
 - CPfcDivision, 51
- m_MinMax_data
 - CPfcDivision, 51
- m_arrayShape
 - CPfcCompress, 28
 - CPfcFileInfo, 57
 - CPfcRestrationRegionPod, 87
- m_bLoadCompressData
 - CPfcRestration, 78
 - CPfcRestrationRegion, 82
 - CPfcRestrationRegionPod, 87
- m_bSingle
 - CPfcRestrationRegionPod, 87
- m_calculatedLayer
 - CPfcCompressInfo, 41
 - CPfcCompressPod, 47
- m_cmdList
 - CPfcCompressCmd, 36
- m_comm
 - CPfcCompress, 28
 - CPfcCompressCioPod, 33
 - CPfcCompressPod, 47
- m_compressCntlPath
 - CPfcCompressCmd, 36
- m_compressError
 - CPfcCompress, 28
 - CPfcCompressCioPod, 33
 - CPfcCompressCmdElm, 37
 - CPfcCompressInfo, 41
 - CPfcCompressPod, 47
- m_compressForm
 - CPfcRestrationRegion, 82
- m_compressFormat
 - CPfcCompress, 28
 - CPfcCompressCmdElm, 37
 - CPfcCompressInfo, 41
- m_compressInfo
 - CPfcRestration, 78
- m_curEvaluateError
 - CPfcCompressPod, 47
- m_curNumSize
 - CPfcCompressPod, 47
- m_curPodBaseSize
 - CPfcCompressPod, 47
- m_dataType
 - CPfcCompress, 28
 - CPfcFileInfo, 57
- m_dfiFilePath
 - CPfcCompress, 28
 - CPfcCompressCmdElm, 37
- m_dfiPath
 - CPfcFilePath, 59
- m_dirPath
 - CPfcFileInfo, 57
 - CPfcRestrationRegionPod, 87
- m_division
 - CPfcRestration, 79
- m_domain

- CPfcRestration, 79
- m_domainDivision
 - CPfcCompress, 28
 - CPfcCompressCmd, 36
- m_endStep
 - CPfcCompress, 29
 - CPfcCompressCmdElm, 37
 - CPfcCompressInfo, 41
- m_endianType
 - CPfcCompress, 29
 - CPfcFileInfo, 57
- m_fileFormat
 - CPfcCompress, 29
 - CPfcFileInfo, 57
- m_fileInfo
 - CPfcRestration, 79
- m_filePath
 - CPfcRestration, 79
- m_globalDivision
 - CPfcDomain, 54
- m_globalOrigin
 - CPfcDomain, 54
- m_globalRegion
 - CPfcDomain, 54
- m_globalVoxel
 - CPfcDomain, 55
- m_guideCell
 - CPfcFileInfo, 57
- m_head
 - CPfcRestration, 79
- m_headIndex
 - CPfcRegion, 73
- m_layerNo
 - CPfcCompressPod, 47
- m_maxLayer
 - CPfcCompressPod, 48
- m_myIDinRegion
 - CPfcCompress, 29
 - CPfcCompressPod, 48
- m_myNumStep
 - CPfcCompressPod, 48
- m_myRankID
 - CPfcCompress, 29
 - CPfcCompressCioPod, 33
 - CPfcCompressPod, 48
- m_myStartStepPos
 - CPfcCompressPod, 48
- m_numCalculatedLayer
 - CPfcRestrationRegionPod, 87
- m_numCoef
 - CPfcRestrationRegionPod, 87
- m_numComponent
 - CPfcCompress, 29
 - CPfcFileInfo, 57
 - CPfcRestrationRegionPod, 87
- m_numLayer
 - CPfcCoeffTree, 22
- m_numParallel
 - CPfcCompress, 29
 - CPfcCompressPod, 48
 - CPfcRestrationRegionPod, 88
- m_numRank
 - CPfcCompress, 29
 - CPfcCompressCioPod, 33
 - CPfcCompressPod, 48
- m_numRegion
 - CPfcCompress, 29
 - CPfcCompressCioPod, 34
 - CPfcCompressPod, 48
 - CPfcDivision, 51
 - CPfcRestration, 79
- m_numSize
 - CPfcRestrationRegionPod, 88
- m_numStep
 - CPfcCompress, 29
 - CPfcCompressCioPod, 34
 - CPfcCompressPod, 48
 - CPfcRestrationRegionPod, 88
- m_numTimeStep
 - CPfcCoeffTree, 22
 - CPfcRestration, 79
- m_optFlags
 - CPfcCompress, 29
 - CPfcCompressCioPod, 34
 - CPfcCompressCmdElm, 38
 - CPfcCompressPod, 48
- m_optSave
 - CPfcCompressCmdElm, 38
- m_outDirPath
 - CPfcCompress, 29
 - CPfcCompressCioPod, 34
 - CPfcCompressCmdElm, 38
 - CPfcCompressPod, 48
- m_pBaseData
 - CPfcRestrationRegionPod, 88
- m_pBaseSizes
 - CPfcRestrationRegionPod, 88
- m_pCioPod
 - CPfcCompress, 30
- m_pCoef_a
 - CPfcCompressPod, 48
- m_pCoefData
 - CPfcRestrationRegionPod, 88
- m_pCompressInfo
 - CPfcRestrationRegion, 82
- m_pDfilN
 - CPfcCompress, 30
 - CPfcCompressCioPod, 34
- m_pDomain
 - CPfcRestrationRegion, 83
- m_pFileInfo
 - CPfcRestrationRegion, 83
- m_pFlowData
 - CPfcCompressCioPod, 34
 - CPfcCompressPod, 49
- m_pIndexBase

- CPfcRestrstrationRegionPod, [88](#)
- m_pIndexCoef
 - CPfcRestrstrationRegionPod, [88](#)
- m_pPod
 - CPfcCompressCioPod, [34](#)
- m_pPod_base_r
 - CPfcCompressPod, [49](#)
- m_pRegion
 - CPfcRestrstrationRegion, [83](#)
- m_pStepList
 - CPfcCompressCioPod, [34](#)
- m_pfcFilePath
 - CPfcRestrstration, [79](#)
- m_prefix
 - CPfcCompress, [30](#)
 - CPfcCompressPod, [49](#)
 - CPfcFileInfo, [57](#)
 - CPfcRestrstrationRegionPod, [88](#)
- m_procFileSave
 - CPfcCompressCmdElm, [38](#)
- m_procPath
 - CPfcFilePath, [59](#)
- m_pvTimeStepList
 - CPfcRestrstrationRegion, [83](#)
- m_regionHead
 - CPfcCompress, [30](#)
 - CPfcCompressCioPod, [34](#)
- m_regionID
 - CPfcCompress, [30](#)
 - CPfcCompressPod, [49](#)
 - CPfcRegion, [73](#)
 - CPfcRestrstrationRegionPod, [88](#)
- m_regionIdList
 - CPfcRestrstration, [79](#)
- m_regionList
 - CPfcDivision, [52](#)
 - CPfcRestrstration, [79](#)
- m_regionMasterRankID
 - CPfcCompress, [30](#)
 - CPfcCompressPod, [49](#)
- m_regionMaxStep
 - CPfcCompressPod, [49](#)
- m_regionSize
 - CPfcCompress, [30](#)
 - CPfcCompressPod, [49](#)
- m_regionTail
 - CPfcCompress, [30](#)
 - CPfcCompressCioPod, [34](#)
- m_startStep
 - CPfcCompress, [30](#)
 - CPfcCompressCmdElm, [38](#)
 - CPfcCompressInfo, [41](#)
- m_stepList
 - CPfcCompress, [30](#)
- m_tail
 - CPfcRestrstration, [79](#)
- m_tailIndex
 - CPfcRegion, [73](#)
- m_timeSlice
 - CPfcRestrstration, [79](#)
- m_timeStepList
 - CPfcRestrstration, [80](#)
- m_unitList
 - CPfcRestrstration, [80](#)
- m_version
 - CPfcCompressInfo, [41](#)
- m_voxelSize
 - CPfcRegion, [73](#)
- MAXPATHLEN
 - PfcPathUtil.h, [122](#)
- MPI_Allgather
 - mpi_stubs.h, [98](#)
- MPI_CHAR
 - mpi_stubs.h, [97](#)
- MPI_COMM_WORLD
 - mpi_stubs.h, [97](#)
- MPI_Comm
 - mpi_stubs.h, [98](#)
- MPI_Comm_rank
 - mpi_stubs.h, [98](#)
- MPI_Comm_size
 - mpi_stubs.h, [98](#)
- MPI_Datatype
 - mpi_stubs.h, [98](#)
- MPI_Gather
 - mpi_stubs.h, [98](#)
- MPI_INT
 - mpi_stubs.h, [97](#)
- MPI_Init
 - mpi_stubs.h, [98](#)
- MPI_SUCCESS
 - mpi_stubs.h, [97](#)
- MakeDirectory
 - PFC, [14](#)
- mpi_stubs.h, [97](#)
 - MPI_Allgather, [98](#)
 - MPI_CHAR, [97](#)
 - MPI_COMM_WORLD, [97](#)
 - MPI_Comm, [98](#)
 - MPI_Comm_rank, [98](#)
 - MPI_Comm_size, [98](#)
 - MPI_Datatype, [98](#)
 - MPI_Gather, [98](#)
 - MPI_INT, [97](#)
 - MPI_Init, [98](#)
 - MPI_SUCCESS, [97](#)
- OpenBaseFile
 - CPfcPodFile, [68](#)
- Output
 - CPfcCompressPod, [45](#)
- PFC, [9](#)
 - E_PFC_ARRAYSHAPE, [10](#)
 - E_PFC_ARRAYSHAPE_UNKNOWN, [10](#)
 - E_PFC_BIG, [12](#)
 - E_PFC_COMPRESS_FMT_POD, [11](#)

- E_PFC_COMPRESS_FMT_UNKNOWN, 11
- E_PFC_COMPRESS_FORMAT, 10
- E_PFC_DTYPE, 11
- E_PFC_DTYPE_UNKNOWN, 11
- E_PFC_ENDIANTYPE, 11
- E_PFC_ENDIANTYPE_UNKNOWN, 11
- E_PFC_ERROR, 12
- E_PFC_ERROR_OUT_OF_RANGE, 13
- E_PFC_ERROR_PFC_COMPRESSFORMAT, 13
- E_PFC_ERROR_READ_CNTL_COMPRESS_FMT, 12
- E_PFC_ERROR_READ_CNTL_DFI_PATH, 12
- E_PFC_ERROR_READ_CNTL_DOMAINDIVISION, 12
- E_PFC_ERROR_READ_CNTL_ITEMCNTL, 12
- E_PFC_ERROR_READ_CNTL_NO_ITEM, 12
- E_PFC_ERROR_READ_CNTL_OUTDIR_PATH, 12
- E_PFC_ERROR_READ_CNTL_PROCFILE_SAVE, 12
- E_PFC_ERROR_READ_CNTLFILE_OPENERROR, 12
- E_PFC_ERROR_READ_COMPRESSINFO, 12
- E_PFC_ERROR_READ_DIVISION, 13
- E_PFC_ERROR_READ_DOMAIN, 13
- E_PFC_ERROR_READ_FILEINFO, 12
- E_PFC_ERROR_READ_FILEPATH, 12
- E_PFC_ERROR_READ_INDEXFILE_OPENERROR, 12
- E_PFC_ERROR_READ_PFC_ARRAYSHAPE, 12
- E_PFC_ERROR_READ_PFC_CALCULATEDLAYER, 12
- E_PFC_ERROR_READ_PFC_COMPONENT, 12
- E_PFC_ERROR_READ_PFC_COMPRESSERROR, 12
- E_PFC_ERROR_READ_PFC_COMPRESSFORMAT, 12
- E_PFC_ERROR_READ_PFC_DATATYPE, 12
- E_PFC_ERROR_READ_PFC_DFIPATH, 12
- E_PFC_ERROR_READ_PFC_DIRECTORYPATH, 12
- E_PFC_ERROR_READ_PFC_ENDIAN, 12
- E_PFC_ERROR_READ_PFC_ENDSTEP, 12
- E_PFC_ERROR_READ_PFC_FILEFORMAT, 12
- E_PFC_ERROR_READ_PFC_GLOBALDIVISION, 13
- E_PFC_ERROR_READ_PFC_GLOBALORIGIN, 13
- E_PFC_ERROR_READ_PFC_GLOBALREGION, 13
- E_PFC_ERROR_READ_PFC_GLOBALVOXEL, 13
- E_PFC_ERROR_READ_PFC_GUIDECCELL, 12
- E_PFC_ERROR_READ_PFC_NO_REGION, 13
- E_PFC_ERROR_READ_PFC_PFCPROCESS, 12
- E_PFC_ERROR_READ_PFC_PREFIX, 12
- E_PFC_ERROR_READ_PFC_REGION_HEADER, 13
- E_PFC_ERROR_READ_PFC_REGION_ID, 13
- E_PFC_ERROR_READ_PFC_REGION_TAILINDEX, 13
- E_PFC_ERROR_READ_PFC_REGION_VOXEL_SIZE, 13
- E_PFC_ERROR_READ_PFC_STARTSTEP, 12
- E_PFC_ERROR_READ_PFC_VERSION, 12
- E_PFC_ERROR_READ_PODBASE_DATA, 13
- E_PFC_ERROR_READ_PODBASE_HEADER, 13
- E_PFC_ERROR_READ_PODBASE_OPENERROR, 13
- E_PFC_ERROR_READ_PODCOEF_DATA, 13
- E_PFC_ERROR_READ_PODCOEF_OPENERROR, 13
- E_PFC_ERROR_READ_PROCFILE_OPENERROR, 13
- E_PFC_ERROR_READ_TIMESLICE, 12
- E_PFC_ERROR_READ_UNITLIST, 12
- E_PFC_ERROR_WRITE_COMPRESSINFO, 13
- E_PFC_ERROR_WRITE_DIVISION, 13
- E_PFC_ERROR_WRITE_DOMAIN, 13
- E_PFC_ERROR_WRITE_FILEINFO, 13
- E_PFC_ERROR_WRITE_FILEPATH, 13
- E_PFC_ERROR_WRITE_INDEXFILE_OPENERROR, 13
- E_PFC_ERROR_WRITE_INDEXFILENAME_EMPTY, 12
- E_PFC_ERROR_WRITE_PFC_DATATYPE, 13
- E_PFC_ERROR_WRITE_PODBASE_DATA, 13
- E_PFC_ERROR_WRITE_PODBASE_OPENERROR, 13
- E_PFC_ERROR_WRITE_PODCOEF_DATA, 13
- E_PFC_ERROR_WRITE_PODCOEF_OPENERROR, 13
- E_PFC_ERROR_WRITE_PROCFILE_OPENERROR, 13
- E_PFC_ERROR_WRITE_TIMESLICE, 13
- E_PFC_ERROR_WRITE_UNITLIST, 13
- E_PFC_ERRORCODE, 12
- E_PFC_FLOAT32, 11
- E_PFC_FLOAT64, 11
- E_PFC_IJKN, 10
- E_PFC_INT16, 11
- E_PFC_INT32, 11
- E_PFC_INT64, 11
- E_PFC_INT8, 11
- E_PFC_LITTLE, 11
- E_PFC_NIJK, 10
- E_PFC_SUCCESS, 12
- E_PFC_UINT16, 11
- E_PFC_UINT32, 11
- E_PFC_UINT64, 11
- E_PFC_UINT8, 11
- MakeDirectory, 14
- PfcPath_ConnectPath, 14
- PfcPath_DirName, 15
- PfcPath_FileName, 15
- PfcPath_getDelimChar, 16

- PfcPath_getDelimString, 16
- PfcPath_hasDrive, 17
- PfcPath_isAbsolute, 17
- vfvPath_emitDrive, 17
- PFC_CHECK_OPT_SAVE
 - PfcDefine.h, 112
- PFC_PRINT
 - PfcDefine.h, 112
- PFC_PRINT_AT
 - PfcDefine.h, 112
- PFC_REVISION
 - PfcVersion.h, 130
- PFC_VERSION_NO
 - PfcVersion.h, 130
- PFC_d_sort
 - PfcMatrix.h, 119
- PFC_ddot
 - PfcMatrix.h, 119
- PFC_dgeev
 - PfcMatrix.h, 119
- PFC_dgemm
 - PfcMatrix.h, 119
- PFC_dgemv
 - PfcMatrix.h, 120
- PFC_matrix_Trans_d
 - PfcMatrix.h, 120
- POD_COMPRESS_VERSION
 - PfcVersion.h, 130
- pPod
 - CPfcRestrictionRegion, 83
- PfcCioDfiUtil.h, 99
- PfcCoeffTree.h, 99
- PfcCompress.h, 100
- PfcCompressCioPod.h, 101
- PfcCompressCmd.h, 102
- PfcCompressCmdElm.h, 102
 - _PFC_COMPRESS_CMD_ELM_CNTL_, 103
- PfcCompressInfo.h, 104
- PfcCompressPod.h, 104
- PfcDefine.h, 105
 - _PFC_IDX_IJ, 108
 - _PFC_IDX_IJK, 108
 - _PFC_IDX_IJKN, 109
 - _PFC_IDX_NIJ, 109
 - _PFC_IDX_NIJK, 110
 - _PFC_TAB_STR, 110
 - _PFC_WRITE_TAB, 110
 - D_PFC_BIG, 111
 - D_PFC_COMPRESS_ERROR_DEFAULT, 111
 - D_PFC_EPSILON, 111
 - D_PFC_FLOAT32, 111
 - D_PFC_FLOAT64, 111
 - D_PFC_IJNK, 111
 - D_PFC_LITTLE, 111
 - D_PFC_NIJK, 111
 - D_PFC_OPT_FLAG_SAVE, 111
 - DEBUG_PRINT, 111
 - DEBUG_PRINT_AT, 111
- PFC_CHECK_OPT_SAVE, 112
- PFC_PRINT, 112
- PFC_PRINT_AT, 112
- PfcDivision.h, 112
- PfcDomain.h, 113
- PfcEndianChange.h, 114
 - endswap, 115
- PfcFileInfo.h, 115
- PfcFilePath.h, 116
- PfcFunction.h, 117
 - int_pow, 117
- PfcMatrix.h, 118
 - dgeev_, 118
 - dgemm_, 118
 - dgemv_, 119
 - PFC_d_sort, 119
 - PFC_ddot, 119
 - PFC_dgeev, 119
 - PFC_dgemm, 119
 - PFC_dgemv, 120
 - PFC_matrix_Trans_d, 120
- PfcMpiCom.h, 120
- PfcPath_ConnectPath
 - PFC, 14
- PfcPath_DirName
 - PFC, 15
- PfcPath_FileName
 - PFC, 15
- PfcPath_getDelimChar
 - PFC, 16
- PfcPath_getDelimString
 - PFC, 16
- PfcPath_hasDrive
 - PFC, 17
- PfcPath_isAbsolute
 - PFC, 17
- PfcPathUtil.h, 121
 - MAXPATHLEN, 122
- PfcPodFile.h, 122
- PfcRegion.h, 123
- PfcRestriction.h, 124
- PfcRestrictionRegion.h, 125
- PfcRestrictionRegionPod.h, 126
- PfcTextParser.h, 127
- PfcTimeSlice.h, 128
- PfcUnitList.h, 129
- PfcVersion.h, 130
 - PFC_REVISION, 130
 - PFC_VERSION_NO, 130
 - POD_COMPRESS_VERSION, 130
- position
 - CPfcCoeffTree, 22
- position_interval
 - CPfcCoeffTree, 22
- print_node
 - CPfcCoeffTree, 21
- print_tree
 - CPfcCoeffTree, 21

