

systematictools

Generated by Doxygen 1.8.13

Contents

1	'systematictools'	1
2	ExampleSystProvider	3
3	Event	5
4	Parameter headers	9
5	Tool configuration	13
6	Writing a 'ISystProviderTool'	15
7	Namespace Index	17
7.1	Namespace List	17
8	Hierarchical Index	19
8.1	Class Hierarchy	19
9	Class Index	21
9.1	Class List	21
10	File Index	23
10.1	File List	23

11 Namespace Documentation	25
11.1 cliopts Namespace Reference	25
11.1.1 Variable Documentation	25
11.1.1.1 analyzer_name	25
11.1.1.2 dump_example_config	25
11.1.1.3 envvar	26
11.1.1.4 fclname	26
11.1.1.5 fhicl_key	26
11.1.1.6 lookup_policy	26
11.1.1.7 outputfile	26
11.1.1.8 producer_name	26
11.1.1.9 provider_name	26
11.1.1.10 quiet	26
11.1.1.11 WrapWithPROLOG	27
11.2 fhicl Namespace Reference	27
11.3 systools Namespace Reference	27
11.3.1 Typedef Documentation	30
11.3.1.1 event_unit_response_t	30
11.3.1.2 eventId_t	31
11.3.1.3 EventResponse	31
11.3.1.4 param_header_map_t	31
11.3.1.5 param_list_t	31
11.3.1.6 param_value_list_t	31
11.3.1.7 parameter_throws_list_t	31
11.3.1.8 paramId_t	32
11.3.1.9 provider_list_t	32
11.3.1.10 SystMetaData	32
11.3.2 Function Documentation	32
11.3.2.1 AppendVect()	32
11.3.2.2 BuildParameterHeaders() [1/2]	32

11.3.2.3	BuildParameterHeaders() [2/2]	33
11.3.2.4	ConfigureISystProvidersFromParameterHeaders()	33
11.3.2.5	ConfigureISystProvidersFromToolConfig()	33
11.3.2.6	ContainterHasParam()	34
11.3.2.7	ExtendEventResponse()	34
11.3.2.8	ExtendSystMetaData()	34
11.3.2.9	FHiCLSimpleToolConfigurationParameterExists()	34
11.3.2.10	FHiCLToSystParamHeader()	35
11.3.2.11	FinalizeAndValidateDependentParameters()	35
11.3.2.12	FullOfUnity()	35
11.3.2.13	GetParam() [1/4]	35
11.3.2.14	GetParam() [2/4]	35
11.3.2.15	GetParam() [3/4]	36
11.3.2.16	GetParam() [4/4]	36
11.3.2.17	GetParamContainerIndex()	36
11.3.2.18	GetParamElementFromContainer() [1/2]	36
11.3.2.19	GetParamElementFromContainer() [2/2]	37
11.3.2.20	GetParamId()	37
11.3.2.21	GetParamIndex() [1/2]	37
11.3.2.22	GetParamIndex() [2/2]	37
11.3.2.23	HasAnyParams()	37
11.3.2.24	HasParam() [1/2]	38
11.3.2.25	HasParam() [2/2]	38
11.3.2.26	IndexIsHandled()	38
11.3.2.27	MakeFHiCLDefinedRandomVariations()	38
11.3.2.28	NEW_SYSTTOOLS_EXCEPT() [1/23]	39
11.3.2.29	NEW_SYSTTOOLS_EXCEPT() [2/23]	39
11.3.2.30	NEW_SYSTTOOLS_EXCEPT() [3/23]	39
11.3.2.31	NEW_SYSTTOOLS_EXCEPT() [4/23]	39
11.3.2.32	NEW_SYSTTOOLS_EXCEPT() [5/23]	39

11.3.2.33 NEW_SYSTTOOLS_EXCEPT() [6/23]	39
11.3.2.34 NEW_SYSTTOOLS_EXCEPT() [7/23]	40
11.3.2.35 NEW_SYSTTOOLS_EXCEPT() [8/23]	40
11.3.2.36 NEW_SYSTTOOLS_EXCEPT() [9/23]	40
11.3.2.37 NEW_SYSTTOOLS_EXCEPT() [10/23]	40
11.3.2.38 NEW_SYSTTOOLS_EXCEPT() [11/23]	40
11.3.2.39 NEW_SYSTTOOLS_EXCEPT() [12/23]	40
11.3.2.40 NEW_SYSTTOOLS_EXCEPT() [13/23]	41
11.3.2.41 NEW_SYSTTOOLS_EXCEPT() [14/23]	41
11.3.2.42 NEW_SYSTTOOLS_EXCEPT() [15/23]	41
11.3.2.43 NEW_SYSTTOOLS_EXCEPT() [16/23]	41
11.3.2.44 NEW_SYSTTOOLS_EXCEPT() [17/23]	41
11.3.2.45 NEW_SYSTTOOLS_EXCEPT() [18/23]	41
11.3.2.46 NEW_SYSTTOOLS_EXCEPT() [19/23]	42
11.3.2.47 NEW_SYSTTOOLS_EXCEPT() [20/23]	42
11.3.2.48 NEW_SYSTTOOLS_EXCEPT() [21/23]	42
11.3.2.49 NEW_SYSTTOOLS_EXCEPT() [22/23]	42
11.3.2.50 NEW_SYSTTOOLS_EXCEPT() [23/23]	42
11.3.2.51 ParseFHiCLSimpleToolConfigurationParameter()	42
11.3.2.52 ParseFHiCLVariationDescriptor()	43
11.3.2.53 ParseToVect()	43
11.3.2.54 ScrubUnityEventResponses() [1/2]	44
11.3.2.55 ScrubUnityEventResponses() [2/2]	44
11.3.2.56 str2T()	44
11.3.2.57 str2T< bool >()	44
11.3.2.58 SystGetOptKV()	45
11.3.2.59 SystHasOpt()	45
11.3.2.60 SystHasOptKV()	45
11.3.2.61 SystParamHeaderToFHiCL()	45
11.3.2.62 to_str() [1/2]	46
11.3.2.63 to_str() [2/2]	46
11.3.2.64 Validate() [1/2]	46
11.3.2.65 Validate() [2/2]	47
11.3.3 Variable Documentation	47
11.3.3.1 kDefaultDouble	47
11.3.3.2 kParamUnhandled	47
11.3.3.3 kParamUnhandled< double >	47

12 Class Documentation	49
12.1 CorrelatedMultisimProvider Class Reference	49
12.1.1 Constructor & Destructor Documentation	49
12.1.1.1 CorrelatedMultisimProvider()	49
12.1.2 Member Function Documentation	50
12.1.2.1 AsString()	50
12.1.2.2 Configure()	50
12.1.2.3 ConfigureFromFHICL()	50
12.1.2.4 GetEventResponse()	50
12.1.3 Member Data Documentation	50
12.1.3.1 child_providers	50
12.1.3.2 RNgine	51
12.1.3.3 RNJesus	51
12.2 CovarianceThrower Class Reference	51
12.2.1 Constructor & Destructor Documentation	51
12.2.1.1 CovarianceThrower() [1/5]	52
12.2.1.2 CovarianceThrower() [2/5]	52
12.2.1.3 CovarianceThrower() [3/5]	52
12.2.1.4 CovarianceThrower() [4/5]	52
12.2.1.5 CovarianceThrower() [5/5]	52
12.2.1.6 ~CovarianceThrower()	52
12.2.2 Member Function Documentation	52
12.2.2.1 SetupDecomp()	53
12.2.2.2 Throw()	53
12.2.3 Member Data Documentation	53
12.2.3.1 CVector	53
12.2.3.2 LMatrix	53
12.2.3.3 NRows	53
12.2.3.4 RNgine	53
12.2.3.5 RNJesus	53

12.2.3.6	RVector	54
12.2.3.7	UncertMatrix	54
12.3	systools::EventSplineCache< event_unit_t, CLtight, Enable > Class Template Reference	54
12.4	systools::EventSplineCache< event_unit_t, CLtight, typename std::enable_if< CLtight==Param↵ ValidationAndErrorResponse::kFrog, void >::type > Class Template Reference	54
12.4.1	Member Function Documentation	55
12.4.1.1	GetEventLateralResponse() [1/2]	55
12.4.1.2	GetEventLateralResponse() [2/2]	55
12.4.1.3	GetEventWeightResponse() [1/2]	55
12.4.1.4	GetEventWeightResponse() [2/2]	56
12.4.1.5	GetTotalEventWeightResponse()	56
12.5	systools::EventSplineCache< event_unit_t, CLtight, typename std::enable_if< CLtight==Param↵ ValidationAndErrorResponse::kHare, void >::type > Class Template Reference	56
12.5.1	Member Function Documentation	56
12.5.1.1	GetEventLateralResponse() [1/2]	57
12.5.1.2	GetEventLateralResponse() [2/2]	57
12.5.1.3	GetEventWeightResponse() [1/2]	57
12.5.1.4	GetEventWeightResponse() [2/2]	57
12.5.1.5	GetTotalEventWeightResponse()	57
12.6	systools::EventSplineCache< event_unit_t, CLtight, typename std::enable_if< CLtight==Param↵ ValidationAndErrorResponse::kTortoise, void >::type > Class Template Reference	58
12.6.1	Member Function Documentation	58
12.6.1.1	GetEventLateralResponse() [1/2]	58
12.6.1.2	GetEventLateralResponse() [2/2]	58
12.6.1.3	GetEventWeightResponse() [1/2]	59
12.6.1.4	GetEventWeightResponse() [2/2]	59
12.6.1.5	GetTotalEventWeightResponse()	59
12.7	systools::EventSplineCacheBase< event_unit_t > Class Template Reference	59
12.7.1	Member Typedef Documentation	60
12.7.1.1	event_t	60
12.7.2	Constructor & Destructor Documentation	60
12.7.2.1	EventSplineCacheBase() [1/3]	61

12.7.2.2	EventSplineCacheBase() [2/3]	61
12.7.2.3	EventSplineCacheBase() [3/3]	61
12.7.3	Member Function Documentation	61
12.7.3.1	CacheEvent()	61
12.7.3.2	CacheEvents() [1/2]	61
12.7.3.3	CacheEvents() [2/2]	62
12.7.3.4	DeclareUsingParameter()	62
12.7.3.5	DeclareUsingParameters() [1/2]	62
12.7.3.6	DeclareUsingParameters() [2/2]	62
12.7.3.7	GetEventUnit()	62
12.7.3.8	GetNEventsInCache()	62
12.7.3.9	KnowAboutParameter()	63
12.7.3.10	ParameterAffectsEventLateral()	63
12.7.3.11	ParameterAffectsEventWeight()	63
12.7.3.12	SetChkErr()	63
12.7.3.13	SetHeaders() [1/2]	63
12.7.3.14	SetHeaders() [2/2]	63
12.7.3.15	SetParametersValue()	64
12.7.3.16	SetParameterValue()	64
12.7.4	Member Data Documentation	64
12.7.4.1	currentValues	64
12.7.4.2	fChkErr	64
12.7.4.3	fEvents	64
12.7.4.4	fHeaderHelper	64
12.7.4.5	lateralParams	65
12.7.4.6	weightParams	65
12.8	ExampleISystProvider Class Reference	65
12.8.1	Constructor & Destructor Documentation	66
12.8.1.1	ExampleISystProvider()	66
12.8.2	Member Function Documentation	66

12.8.2.1	AsString()	66
12.8.2.2	BuildSystMetaData()	66
12.8.2.3	GetEventResponse()	66
12.8.2.4	GetExtraToolOptions()	67
12.8.2.5	SetupResponseCalculator()	67
12.8.3	Member Data Documentation	67
12.8.3.1	applyToAll	67
12.8.3.2	RNgine	67
12.8.3.3	RNJesus	67
12.9	systools::ISystProviderTool Class Reference	68
12.9.1	Detailed Description	69
12.9.2	Constructor & Destructor Documentation	69
12.9.2.1	ISystProviderTool()	69
12.9.2.2	~ISystProviderTool()	69
12.9.3	Member Function Documentation	69
12.9.3.1	AsString()	70
12.9.3.2	BuildSystMetaData()	70
12.9.3.3	CheckHaveMetaData()	70
12.9.3.4	ConfigureFromParameterHeaders()	70
12.9.3.5	ConfigureFromToolConfig()	71
12.9.3.6	GetEventResponse()	71
12.9.3.7	GetExampleToolConfiguration()	71
12.9.3.8	GetExtraToolOptions()	71
12.9.3.9	GetFullyQualifiedName()	71
12.9.3.10	GetInstanceName()	72
12.9.3.11	GetNVariations()	72
12.9.3.12	GetParameterHeadersDocument()	72
12.9.3.13	GetParameterId()	72
12.9.3.14	GetSystMetaData()	72
12.9.3.15	GetToolType()	72

12.9.3.16 ParamsHandled()	73
12.9.3.17 SetupResponseCalculator()	73
12.9.3.18 SuggestParameterThrows()	73
12.9.3.19 SuggestSeed()	73
12.9.4 Member Data Documentation	74
12.9.4.1 fQName	74
12.9.4.2 fHaveSystMetaData	74
12.9.4.3 fName	74
12.9.4.4 fIsFullyConfigured	74
12.9.4.5 fSeedSuggestion	74
12.9.4.6 fSystMetaData	75
12.9.4.7 fToolType	75
12.10 MD5 Class Reference	75
12.10.1 Member Typedef Documentation	76
12.10.1.1 size_type	76
12.10.1.2 uint1	76
12.10.1.3 uint4	77
12.10.2 Member Enumeration Documentation	77
12.10.2.1 anonymous enum	77
12.10.3 Constructor & Destructor Documentation	77
12.10.3.1 MD5() [1/2]	77
12.10.3.2 MD5() [2/2]	77
12.10.4 Member Function Documentation	77
12.10.4.1 decode()	77
12.10.4.2 encode()	78
12.10.4.3 F()	78
12.10.4.4 FF()	78
12.10.4.5 finalize()	78
12.10.4.6 G()	78
12.10.4.7 GG()	79

12.10.4.8 <code>H()</code>	79
12.10.4.9 <code>hexdigest()</code>	79
12.10.4.10 <code>HH()</code>	79
12.10.4.11 <code>ll()</code>	79
12.10.4.12 <code>ll()</code>	80
12.10.4.13 <code>nit()</code>	80
12.10.4.14 <code>rotate_left()</code>	80
12.10.4.15 <code>transform()</code>	80
12.10.4.16 <code>update()</code> [1/2]	80
12.10.4.17 <code>update()</code> [2/2]	80
12.10.5 Friends And Related Function Documentation	81
12.10.5.1 <code>operator<<</code>	81
12.10.6 Member Data Documentation	81
12.10.6.1 <code>buffer</code>	81
12.10.6.2 <code>count</code>	81
12.10.6.3 <code>digest</code>	81
12.10.6.4 <code>finalized</code>	81
12.10.6.5 <code>state</code>	81
12.11 <code>systools::ParamHeaderHelper</code> Class Reference	82
12.11.1 Member Typedef Documentation	85
12.11.1.1 <code>discrete_variation_list_t</code>	85
12.11.1.2 <code>param_tspline_map_t</code>	85
12.11.1.3 <code>spline_t</code>	85
12.11.2 Constructor & Destructor Documentation	85
12.11.2.1 <code>ParamHeaderHelper()</code> [1/2]	85
12.11.2.2 <code>ParamHeaderHelper()</code> [2/2]	86
12.11.3 Member Function Documentation	86
12.11.3.1 <code>CheckParamList()</code>	86
12.11.3.2 <code>CheckParamValueList()</code>	86
12.11.3.3 <code>GetAllDiscreteResponses()</code> [1/2]	86

12.11.3.4 GetAllDiscreteResponses() [2/2]	87
12.11.3.5 GetDiscreteResponse() [1/3]	87
12.11.3.6 GetDiscreteResponse() [2/3]	87
12.11.3.7 GetDiscreteResponse() [3/3]	88
12.11.3.8 GetDiscreteResponses() [1/6]	88
12.11.3.9 GetDiscreteResponses() [2/6]	88
12.11.3.10GetDiscreteResponses() [3/6]	89
12.11.3.11GetDiscreteResponses() [4/6]	89
12.11.3.12GetDiscreteResponses() [5/6]	89
12.11.3.13GetDiscreteResponses() [6/6]	89
12.11.3.14GetDiscreteVariationParameterValues()	90
12.11.3.15GetEventResponseInfo()	90
12.11.3.16GetHeader() [1/2]	90
12.11.3.17GetHeader() [2/2]	90
12.11.3.18GetHeaderId()	90
12.11.3.19GetHeaderInfo()	90
12.11.3.20GetHeaders()	91
12.11.3.21GetNDiscreteVariations() [1/2]	91
12.11.3.22GetNDiscreteVariations() [2/2]	91
12.11.3.23GetParameterLowLimit()	91
12.11.3.24GetParameterResponse() [1/3]	91
12.11.3.25GetParameterResponse() [2/3]	92
12.11.3.26GetParameterResponse() [3/3]	92
12.11.3.27GetParameters()	92
12.11.3.28GetParameterUpLimit()	92
12.11.3.29GetPolyResponse()	93
12.11.3.30GetResponseParamId()	93
12.11.3.31GetSpline() [1/4]	93
12.11.3.32GetSpline() [2/4]	93
12.11.3.33GetSpline() [3/4]	94

12.11.3.34	GetSpline() [4/4]	94
12.11.3.35	GetSplines() [1/3]	94
12.11.3.36	GetSplines() [2/3]	94
12.11.3.37	GetSplines() [3/3]	95
12.11.3.38	GetTotalResponse() [1/2]	95
12.11.3.39	GetTotalResponse() [2/2]	95
12.11.3.40	HasParameterLimits()	96
12.11.3.41	HasParameterLowLimit()	96
12.11.3.42	HasParameterUpLimit()	96
12.11.3.43	HaveHeader() [1/2]	96
12.11.3.44	HaveHeader() [2/2]	97
12.11.3.45	IsResponselessParam()	97
12.11.3.46	IsSplineParam()	97
12.11.3.47	IsThrownParam()	97
12.11.3.48	IsWeightResponse()	97
12.11.3.49	SetAllowNegativeWeights()	98
12.11.3.50	SetCareLevel()	98
12.11.3.51	SetChkErr()	98
12.11.3.52	SetErrorResponseLevel()	98
12.11.3.53	SetHeaders() [1/2]	99
12.11.3.54	SetHeaders() [2/2]	99
12.11.3.55	SetLargeWeightBoundary()	99
12.11.3.56	SetPedantLevel()	99
12.11.3.57	SetSmallWeightBoundary()	99
12.11.3.58	ValuesAreInNaturalUnits()	100
12.11.4	Member Data Documentation	100
12.11.4.1	fChkErr	100
12.11.4.2	fHeaders	100
12.11.4.3	nullheader	100
12.11.4.4	scratch_discrete_variation_list_t1	100

12.11.4.5 scratch_spline_t1	100
12.11.4.6 scratch_spline_t2	101
12.12systools::ParamHeaderProviderName Struct Reference	101
12.12.1 Detailed Description	101
12.12.2 Member Data Documentation	101
12.12.2.1 Header	101
12.12.2.2 ProviderFQName	101
12.13systools::PrecalculatedResponseReader< Order >::ParamPolyResponses Struct Reference	102
12.13.1 Member Data Documentation	102
12.13.1.1 pid	102
12.13.1.2 resp	102
12.14systools::ParamResponses Struct Reference	102
12.14.1 Detailed Description	103
12.14.2 Member Data Documentation	103
12.14.2.1 pid	103
12.14.2.2 responses	103
12.15systools::ParamThrows Struct Reference	103
12.15.1 Detailed Description	103
12.15.2 Member Data Documentation	104
12.15.2.1 pid	104
12.15.2.2 thrown_vals	104
12.16ParamValidationAndErrorResponse Struct Reference	104
12.16.1 Member Enumeration Documentation	105
12.16.1.1 CareLevel	105
12.16.1.2 ErrorResponseLevel	105
12.16.1.3 PedantLevel	105
12.16.2 Constructor & Destructor Documentation	106
12.16.2.1 ParamValidationAndErrorResponse()	106
12.16.3 Member Function Documentation	106
12.16.3.1 CheckResponse()	106

12.16.3.2 SetAllowNegativeWeights()	106
12.16.3.3 SetCareLevel()	107
12.16.3.4 SetErrorResponseLevel()	107
12.16.3.5 SetLargeWeightBoundary()	107
12.16.3.6 SetPedantLevel()	107
12.16.3.7 SetSmallWeightBoundary()	107
12.16.4 Member Data Documentation	107
12.16.4.1 fAllowNegativeWeights	107
12.16.4.2 fCare	108
12.16.4.3 fErrorResponse	108
12.16.4.4 fLargeWeight	108
12.16.4.5 fPedantry	109
12.16.4.6 fSmallWeight	109
12.17 systools::ParamValue Struct Reference	109
12.17.1 Detailed Description	109
12.17.2 Member Data Documentation	109
12.17.2.1 pid	110
12.17.2.2 val	110
12.18 systools::PolyResponse< n > Struct Template Reference	110
12.18.1 Constructor & Destructor Documentation	110
12.18.1.1 PolyResponse() [1/3]	110
12.18.1.2 PolyResponse() [2/3]	111
12.18.1.3 PolyResponse() [3/3]	111
12.18.2 Member Function Documentation	111
12.18.2.1 eval()	111
12.19 systools::PrecalculatedResponseReader< Order > Class Template Reference	111
12.19.1 Constructor & Destructor Documentation	112
12.19.1.1 PrecalculatedResponseReader() [1/2]	112
12.19.1.2 PrecalculatedResponseReader() [2/2]	112
12.19.2 Member Function Documentation	113

12.19.2.1 AddEventResponses()	113
12.19.2.2 AllocateVectors()	113
12.19.2.3 GetEntries()	113
12.19.2.4 GetEventResponse()	113
12.19.2.5 MakeTreeWriter()	114
12.19.2.6 NEW_SYSTTOOLS_EXCEPT() [1/4]	114
12.19.2.7 NEW_SYSTTOOLS_EXCEPT() [2/4]	114
12.19.2.8 NEW_SYSTTOOLS_EXCEPT() [3/4]	114
12.19.2.9 NEW_SYSTTOOLS_EXCEPT() [4/4]	114
12.19.2.10 SetBranchAddresses()	115
12.19.3 Member Data Documentation	115
12.19.3.1 coeffs_1D	115
12.19.3.2 fHeaders	115
12.19.3.3 file	115
12.19.3.4 ids	115
12.19.3.5 NCoeffs	116
12.19.3.6 NIds	116
12.19.3.7 tree	116
12.20 systools::systematictools_except Struct Reference	116
12.20.1 Constructor & Destructor Documentation	117
12.20.1.1 systematictools_except() [1/2]	117
12.20.1.2 systematictools_except() [2/2]	117
12.20.2 Member Function Documentation	117
12.20.2.1 operator<<()	117
12.20.2.2 what()	117
12.20.3 Member Data Documentation	117
12.20.3.1 msg	117
12.20.3.2 msgstrm	118
12.21 systools::SystParamHeader Struct Reference	118
12.21.1 Detailed Description	119

12.21.2 Constructor & Destructor Documentation	119
12.21.2.1 SystParamHeader()	119
12.21.3 Member Data Documentation	119
12.21.3.1 centralParamValue	119
12.21.3.2 differsEventByEvent	120
12.21.3.3 isCorrection	120
12.21.3.4 isRandomlyThrown	120
12.21.3.5 isResponselessParam	120
12.21.3.6 isSplineable	120
12.21.3.7 isWeightSystematicVariation	121
12.21.3.8 oneSigmaShifts	121
12.21.3.9 opts	121
12.21.3.10 paramValidityRange	121
12.21.3.11 paramVariations	121
12.21.3.12 prettyName	122
12.21.3.13 responseParamId	122
12.21.3.14 responses	122
12.21.3.15 systParamId	122
12.21.3.16 unitsAreNatural	122
12.22 SystToolsEventResponse Class Reference	123
12.22.1 Constructor & Destructor Documentation	123
12.22.1.1 SystToolsEventResponse() [1/3]	123
12.22.1.2 SystToolsEventResponse() [2/3]	123
12.22.1.3 SystToolsEventResponse() [3/3]	123
12.22.2 Member Function Documentation	124
12.22.2.1 NEW_SYSTTOOLS_EXCEPT() [1/2]	124
12.22.2.2 NEW_SYSTTOOLS_EXCEPT() [2/2]	124
12.22.2.3 operator=() [1/2]	124
12.22.2.4 operator=() [2/2]	124
12.22.2.5 produce()	124

12.22.3 Member Data Documentation	124
12.22.3.1 sp_config_hash	124
12.22.3.2 syst_providers	125
12.23 SystToolsEventResponseTree Class Reference	125
12.23.1 Constructor & Destructor Documentation	125
12.23.1.1 SystToolsEventResponseTree()	125
12.23.2 Member Function Documentation	125
12.23.2.1 Fill()	125
12.23.2.2 MakeBranches()	126
12.23.2.3 SetEvent()	126
12.23.2.4 SetParamResponse()	126
12.23.2.5 SetThrow()	126
12.23.2.6 SetTotalWeight()	126
12.23.2.7 SetTree()	126
12.23.3 Member Data Documentation	126
12.23.3.1 event	127
12.23.3.2 event_responses	127
12.23.3.3 param_values	127
12.23.3.4 t_it	127
12.23.3.5 total_weight	127
12.23.3.6 tree	127
12.24 SystToolsResponseTreeMaker Class Reference	127
12.24.1 Constructor & Destructor Documentation	128
12.24.1.1 SystToolsResponseTreeMaker() [1/3]	128
12.24.1.2 SystToolsResponseTreeMaker() [2/3]	128
12.24.1.3 SystToolsResponseTreeMaker() [3/3]	128
12.24.2 Member Function Documentation	128
12.24.2.1 analyze()	129
12.24.2.2 operator=() [1/2]	129
12.24.2.3 operator=() [2/2]	129
12.24.3 Member Data Documentation	129
12.24.3.1 configuredParameterHeaders	129
12.24.3.2 fEventHelper	129
12.24.3.3 fHeaderHelper	129
12.24.3.4 flnpTag	129
12.24.3.5 fOutputTree	130
12.24.3.6 fSplineMode	130
12.24.3.7 fTweak	130

13 File Documentation	131
13.1 README.md File Reference	131
13.2 systematicstools/app/CheckSystProviderConfigmd5.cc File Reference	131
13.3 systematicstools/app/FindISystProvider.cc File Reference	131
13.3.1 Function Documentation	132
13.3.1.1 HandleOpts()	132
13.3.1.2 main()	132
13.3.1.3 SayUsage()	132
13.4 systematicstools/app/GenerateSystProviderConfig.cc File Reference	132
13.4.1 Function Documentation	133
13.4.1.1 HandleOpts()	133
13.4.1.2 main()	133
13.4.1.3 ReadParameterSet()	133
13.4.1.4 SayUsage()	133
13.5 systematicstools/doc/ExampleSystProvider.md File Reference	134
13.6 systematicstools/doc/MovingParts.md File Reference	134
13.7 systematicstools/doc/ParameterHeaders.md File Reference	134
13.8 systematicstools/doc/ToolConfiguration.md File Reference	134
13.9 systematicstools/doc/WritingAProvider.md File Reference	134
13.10 systematicstools/interface/EventResponse_product.cc File Reference	134
13.11 systematicstools/interface/EventResponse_product.hh File Reference	134
13.12 systematicstools/interface/FHiCLSystParamHeaderConverters.cc File Reference	135
13.13 systematicstools/interface/FHiCLSystParamHeaderConverters.hh File Reference	136
13.14 systematicstools/interface/ISystProviderTool.cc File Reference	136
13.15 systematicstools/interface/ISystProviderTool.hh File Reference	136
13.16 systematicstools/interface/SystMetaData.cc File Reference	137
13.17 systematicstools/interface/SystMetaData.hh File Reference	138
13.18 systematicstools/interface/SystParamHeader.cc File Reference	139
13.19 systematicstools/interface/SystParamHeader.hh File Reference	139
13.20 systematicstools/interface/types.hh File Reference	140

13.21systematicstools/interpreters/EventSplineCacheHelper.hh File Reference	141
13.22systematicstools/interpreters/ParamHeaderHelper.cc File Reference	142
13.23systematicstools/interpreters/ParamHeaderHelper.hh File Reference	142
13.24systematicstools/interpreters/ParamValidationAndErrorResponse.cc File Reference	142
13.25systematicstools/interpreters/ParamValidationAndErrorResponse.hh File Reference	143
13.26systematicstools/interpreters/PolyResponse.hh File Reference	143
13.27systematicstools/interpreters/PrecalculatedResponseReader.hh File Reference	143
13.28systematicstools/module/classes.h File Reference	144
13.29systematicstools/module/SystToolsEventResponse_module.cc File Reference	144
13.30systematicstools/module/SystToolsResponseTreeMaker_module.cc File Reference	144
13.31systematicstools/systproviders/CorrelatedMultisimProvider_tool.cc File Reference	145
13.32systematicstools/systproviders/ExampleISystProvider_tool.cc File Reference	145
13.32.1 Function Documentation	146
13.32.1.1 GetLateralResponse()	146
13.32.1.2 GetNormResponse()	146
13.32.1.3 GetParamShift_nu()	146
13.32.1.4 GetParamValue_nu()	146
13.32.1.5 GetResponse()	147
13.32.1.6 GetResponse_nu()	147
13.32.1.7 GetResponse_shift()	147
13.32.2 Variable Documentation	147
13.32.2.1 default_centralvalue_nu	147
13.32.2.2 default_lowsigmavalue_nu	147
13.32.2.3 default_upsigmavalue_nu	147
13.33systematicstools/systproviders/ExampleISystProvider_tool.hh File Reference	148
13.34systematicstools/utility/CovMatThrower.cc File Reference	148
13.35systematicstools/utility/CovMatThrower.hh File Reference	148
13.36systematicstools/utility/exceptions.hh File Reference	148
13.36.1 Macro Definition Documentation	149
13.36.1.1 NEW_SYSTTOOLS_EXCEPT	149

13.37systematicstools/utility/FHiCLSystParamHeaderUtility.cc File Reference	149
13.38systematicstools/utility/FHiCLSystParamHeaderUtility.hh File Reference	150
13.39systematicstools/utility/md5.cc File Reference	151
13.39.1 Macro Definition Documentation	151
13.39.1.1 S11	151
13.39.1.2 S12	151
13.39.1.3 S13	152
13.39.1.4 S14	152
13.39.1.5 S21	152
13.39.1.6 S22	152
13.39.1.7 S23	152
13.39.1.8 S24	152
13.39.1.9 S31	152
13.39.1.10S32	152
13.39.1.11S33	153
13.39.1.12S34	153
13.39.1.13S41	153
13.39.1.14S42	153
13.39.1.15S43	153
13.39.1.16S44	153
13.39.2 Function Documentation	153
13.39.2.1 md5()	153
13.39.2.2 operator<<()	154
13.40systematicstools/utility/md5.hh File Reference	154
13.40.1 Function Documentation	154
13.40.1.1 md5()	154
13.41systematicstools/utility/ParameterAndProviderConfigurationUtility.cc File Reference	154
13.42systematicstools/utility/ParameterAndProviderConfigurationUtility.hh File Reference	155
13.43systematicstools/utility/printers.hh File Reference	156
13.44systematicstools/utility/ResponselessParamUtility.cc File Reference	156
13.45systematicstools/utility/ResponselessParamUtility.hh File Reference	156
13.46systematicstools/utility/ROOTUtility.hh File Reference	157
13.46.1 Function Documentation	157
13.46.1.1 CheckOpenFile()	157
13.46.1.2 GetHistogram() [1/2]	158
13.46.1.3 GetHistogram() [2/2]	158
13.46.1.4 GetPolyFitCoeffs()	158
13.46.1.5 IsFlowBin()	158
13.46.1.6 IsInHistogramRange()	158
13.46.1.7 NEW_SYSTTOOLS_EXCEPT() [1/2]	158
13.46.1.8 NEW_SYSTTOOLS_EXCEPT() [2/2]	159
13.47systematicstools/utility/string_parsers.hh File Reference	159

Chapter 1

‘systematicstools’

This package provides a framework for writing, using, and interpreting the output of modular tools for propagating systematic uncertainties within an experimental simulation tool chain.

Introduction

Experimental physics is lousy with models. The 'event' signatures recorded in detectors are very often ambiguous and thus extensive simulation is needed to interpret what is observed. If these simulations are incorrect (they are), then our interpretation of the underlying physics will be similarly incorrect. To account for this, we make plausible variations of many components of the simulation and hope (assume) that somewhere within the high dimensional space of variations, is a model that describes nature well enough for our purposes. Because of the computing time involved with running a full simulation, it is often attractive to be able to make such variations to a set of simulated observations at 'analysis-', or interpretation-, time. Sometimes these variations can be exactly parameterized and thus the varied, simulated observations are statistically equivalent to having re-run the entire simulation. Sometimes they are approximate.

This package aims to provide a framework for developing and using tools that provide these variations. The framework itself aims to contain no assumptions on the experiment or physics analysis being performed, these are left up to specific implementations that reside in dependent packages.

Where to start

For a short primer on the main moving parts within `systematicstools`, see [Moving parts](#).

For a description of the two levels of configuration file, with examples, see [Tool Configuration](#) and [Parameter Headers](#).

For tips on how to begin writing a new systematic provider, and a short description of the `ISystProviderTool` abstract base class, see [Writing A Provider](#).

For an example, and recommended documentation structure, of an `ISystProviderTool` implementation, see [ExampleSystProvider](#).

Chapter 2

ExampleSystProvider

This file gives an example of the suggested `ISystProviderTool` documentation layout.

Overview

This dummy provider calculates simple event responses to illustrate how to write an `ISystProviderTool` subclass.

Physics motivation

There isn't any, but if there was, it would go here.

Full Tool Configuration

```
ExampleWeightProvider_multiuniverse_centershift: {
  tool_type: "ExampleISystProvider"
  instance_name: "multiuniverse_centershift"

  ## Correction-like
  # central_value: 1
  ## end Correction-like

  ## Multi-universe-like
  central_value: 1
  variation_descriptor: "{-2,2}" # optional
  rand_dist: "uniform" # optional
  number_of_throws: 10 # optional
  ## end Multi-universe-like

  ## Splineable
  # variation_descriptor: "(-2,2,0.5)"
  ## end Splineable

  # provide_lateral: false # Whether response should be a kinematic shift or an event weight.
  # is_global: false # Whether the response weight differs event-by-event or if the responses themselves
  #                   can be stored in the parameter headers
  # apply_to_all: true # Whether to apply to all events or randomly select some events to produce responses
  #                   for
  # param_name: my_param # Force the prettyName of the generated parameter.
}
```


Chapter 3

Event

For full details, see `art::event`. Will be related to some data-taking unit of the experimental apparatus. For neutrino experiments, it is likely to be equivalent to a proton beam trigger corresponding to a spill or a bucket of protons.

Event unit

This is a logical subdivision of the full `art::event` into 'units' that a given systematic variation applies to. For neutrino interaction systematic uncertainties, this will be an MC truth neutrino interaction that leaves some selected event signature within the detector. However, this is a generalized subdivision, and a different systematic might affect the PID discriminator for all MIP-like tracks within a detector, of which there may be more than one per `art::event`. Often, 'event' and 'event unit' will be used interchangeably.

Responses

Event unit response

The 'response' that some event unit incurs in as a result of a variation of some parameter in the simulation.

Weight response

Often event responses can be encoded as weights, which encapsulate the relative change in probability for an event of that class to have been seen under the assumed simulation parameter change. For neutrino interaction systematic variations, this is almost always the simplest method of propagation. As all independent event weights can be applied multiplicatively, weight responses can be used by analyses with little-to-no knowledge of the parameter being varied, as such, weight responses are logically separate from other classes of responses (collectively called 'lateral responses') so that they can be simply applied at analysis time.

Lateral response

A general response may effect a change in any event property. As the response is fully generalized, each analysis must know a priori how to apply a given lateral event response.

A specific example for of neutrino–nucleus interactions, would be a change in the binding potential of the nucleus causing a shift in the distribution of final state lepton momenta. While such a response is hard to parameterize exactly—it would be most effective to run multiple parallel simulations—these kind of responses can be useful analysis tools. Furthermore, the simulation-time efficiency gained by using such an ad hoc parameterization would be negated if these responses were applied before detector simulation (interaction simulation is often the least CPU-intensive part of the simulation toolchain), as a result, true responses may be applied to fully reconstructed quantities. This is an approximation and should be treated with care and assessed for each such response.

For other classes of lateral response, such as a shift in the reconstructed momenta or PID discriminator value, the response can be less approximate and as such this technique can be a useful tool for propagating systematic variations of the detector model.

Spline or Response function

An analytic interpolation of some discrete set of fully-calculated event responses. This can be implemented as a single N-dimensional polynomial fit to the calculated responses, or a piece-wise 'spline', such as implemented by the `ROOT` class `TSpline3`. This is possible for both weight and lateral responses.

Consumer

Any downstream analysis that uses the calculated event responses. Consumers can interpret event responses by interrogating the 'parameter header' information.

Systematic provider

A 'Systematic provider', 'SystProvider', or 'SystProviderTool', is an implementation of the `systtools::ISystProviderTool` interface. It defines parameters, accepts 'tool configuration', produces 'parameter headers', and calculates event responses to passed `art::events`. An example is included with this package, but the modular design of `systematictools` is based around `art::make_tool` dynamically instantiating instances of `systtools::ISystProviderTool` subclasses—implemented in specific packages—at runtime. For example, systematic providers for neutrino interaction variations exist in the `nusystematics` package, which is neutrino-specific, but not experiment specific. More-specific packages such as a flux uncertainty package for the BNB, or a detector systematics package for NOvA would be in-keeping with this design.

Tool configuration

The tool configuration is a FHiCL document that can be used to fully configure a systematic provider. The structure of the per-provider FHiCL tables is not specified (other than that required by `art::make_tool`), and specific provider documentation should be consulted for the tool configuration layout. This FHiCL is meant to be read and modified by humans in the standard workflow.

Parameter headers

The parameter headers describe all of the systematic parameters that have been considered when calculating the event responses. Given the `systtools::paramId_t` of an `systools::event_unit_response_t`, a consumer can look up all of the meta-data about the relevant systematic parameter and its variations in the parameter headers. The parameter header information can be serialized from the `systtools::SystParamHeader` object interface to and from a structured FHiCL format. The FHiCL format is free to contain extra information that may be needed to configure the systematic provider in a deterministic way across multiple art jobs. This information is not de-serialized to the objectified interface.

Consumers are welcome to directly interrogate the parameter header database in either FHiCL or `systtools::SystMetaData` for (`typedef std::vector<SystParamHeader> SystMetaData`), but a helper class that exposes convenience methods for common work flows is provided by `systtools::ParamHeaderHelper`. More description of the c++ and FHiCL formats, and the use of `systtools::ParamHeaderHelper`, is given in [Parameter headers](#).

While it is useful that the on-disk parameter header information is human-readable, a standard workflow would not require any human modification of the parameter headers. Each systematic provider, configured by a well-formed tool configuration document is required to produce the corresponding parameter header document that can be used to both re-configure a systematic provider for event response calculation and interpret any calculated event responses.

Chapter 4

Parameter headers

Motivation

The two levels of 'configuration' may seem clumsy at first, so the motivation will be described here.

Design principles:

- The calculated responses of each 'event unit' to a configured set of systematic parameter variations should be stored in the `art : : event` object.
- These responses need meta-data to be correctly interpreted, *i.e.*: 'What parameter value does the response at index 3 correspond to?'
- This meta-data does not change event by event, or file by file and it is therefore inefficient to carry it around event by event.
- For the very vast majority of systematic parameters a fixed form of meta-data can be used to fully describe their usage, however, some form of extensibility of the meta-data format should be included.
- As art jobs are designed to be distributed, a per-systematic provider configuration must be able to deterministically produce correctly synchronised event responses: *i.e.* If 100 randomly chosen variations of a parameter should be calculated, the same set of 100 variations must be used on each compute node that processes an art file.
- The structure of configuration of individual systematic providers should not be fully specified to allow extensibility and generality of implementations, it should also be easily human readable, parse-able, and editable.

It was decided that the best fit for this was a staged configuration: firstly, the human-readable, per-`ISystProviderTool` configuration would be read and converted to the second, per-job configuration format, which can also be used to full interpret the calculated event responses. The [Tool Configuration](#) is extensible and easy to read and edit. The second, per-job configuration file, the *Parameter Headers* serves two purposes, it allows concise, per-event vectors of doubles to be correctly interpreted as a wide range of event responses, and can also be used to configure an instance of a `ISystProviderTool` to deterministically calculate the requested responses. Any extra `FHiCL` required for this configuration on top of just the serialized `systtools::SystParamHeaders` is called the `tool_options`.

The on-disk format of the parameter headers could have been anything, but `FHiCL` was chosen because the files in general will not be too large and any `tool_options` must be generally extensible. The human-readability of `FHiCL` is useful even it is not strictly necessary for the standard work flow. These format decisions could be revised in the future but would require updates to the `ISystProviderTool` subclasses that rely on extra `tool_options`.

Format

The example introduced in [Tool Configuration: Description](#) is converted by an instance of `ExampleISystProvider` to :

```
BEGIN_PROLOG
generated_systematic_provider_configuration: {
  ExampleISystProvider_multiuniverse_centershift: {
    ExampleSystToolsEventByEventLateral_all: {
      centralParamValue: 1
      isRandomlyThrown: true
      oneSigmaShifts: [
        -2,
        2
      ]
      paramVariations: [
        -2.42704620044744023843e-1,
        6.72229221295147727488e-1,
        -2.87007100337361986675e-1,
        -5.20900882170851398101e-1,
        4.2953405500459318489e-1,
        -9.3603670495511770433e-1,
        -8.81928152648805196989e-1,
        -6.56415778953291617626e-1,
        2.25181981400627417855,
        2.42714699282725065999
      ]
      prettyName: "ExampleSystToolsEventByEventLateral_all"
      systParamId: 0
    }
  }
  instance_name: "multiuniverse_centershift"
  parameter_headers: [
    "ExampleSystToolsEventByEventLateral_all"
  ]
  tool_options: {
    apply_to_all: true
  }
  tool_type: "ExampleISystProvider"
}
syst_providers: [
  "ExampleISystProvider_multiuniverse_centershift"
]
}
END_PROLOG
```

by an invocation of `GenerateSystProviderConfig()` (c.f. [Tool Configuration: Converting to parameter headers](#)). It should hopefully be more clear why these documents are not intended to be frequently human-modified. The structure of this document is as follows: The outer-most element, `generated_systematic_provider_configuration` is a wrapping FHiCL table that can be passed in it's entirety to helper methods that will instantiate and fully configure all described `ISystProviderTool` subclasses. The `syst_providers` element, much like the equivalent element in the tool configuration document contains a list of keys, each of which correspond to a single `ISystProviderTool` subclass instance. The `tool_type` and `instance_name` atoms are also used in the same way as in the tool configuration. The `parameter_headers` sequence contains a list of keys that correspond to the FHiCL tables describing the systematic parameters exposed by the fully configured systematic provider. In this example, a single parameter, named `ExampleSystToolsEventByEventLateral_all`, is described. This FHiCL table is directly de-serialized to a `systtools::SystParamHeader` instance, which can be used by 'consumers' to interpret any calculated responses. Finally, the `tool_options` table can contain arbitrary FHiCL that should be used by the `ISystProviderTool` subclass instance to perform any extra configuration that cannot be described by the parameter FHiCL tables.

`systtools::SystParamHeader`

In the example above, the FHiCL table:

```
centralParamValue: 1
isRandomlyThrown: true
oneSigmaShifts: [
  -2,
```

```

    2
  ]
  paramVariations: [
    -2.42704620044744023843e-1,
    6.72229221295147727488e-1,
    -2.87007100337361986675e-1,
    -5.20900882170851398101e-1,
    4.2953405500459318489e-1,
    -9.3603670495511770433e-1,
    -8.81928152648805196989e-1,
    -6.56415778953291617626e-1,
    2.25181981400627417855,
    2.42714699282725065999
  ]
  prettyName: "ExampleSystToolsEventByEventLateral_all"
  systParamId: 0

```

describes a systematic parameter named *ExampleSystToolsEventByEventLateral_all*, with unique Id 0, that will respond to relevant event units with ten responses to ten randomly thrown parameter variations between -1 and 3. The serialization to, and de-serialization from, `systtools::SystParamHeader` instances can be performed by helper methods found in `interface/FHiCLSystParamHeaderConverters.hh`. Any members left in their default state are not written to FHiCL during serialization; more meta-data members than those shown in the above example exist and are well-documented in `interface/SystParamHeader.hh`.

The nature of the FHiCL format and c++ bindings mean that reading and writing parameter headers documents is well defined programatically. They are somewhat fragile with respect to manual modification by non-experts, but a number of validity checks are applied to the de-serialized vectors of `systtools::SystParamHeader` objects. These `Validate` methods can be found in `interface/SystMetaData.hh` and `interface/SystParamHeader.hh`.

Interpreting responses

The `ISystProviderTool` interface specifies that subclasses provide event responses in a format described in `interface/EventResponse_product.hh`. Herein inverted commas are used to specify that these words do not have their usual c++ STL meaning. The format is simply a 'list' of 'pairs' of parameter unique Ids and vectors of responses—in the form of double precision floating point numbers. For a given *event unit*, the Id contained in the 'pair' can be matched to the `systtools::SystParamHeader::systParamId` in a vector of `systtools::SystParamHeaders` (which is typedef'd as `systtools::SystMetaData`). The matched `systtools::SystParamHeader` can then be used to interpret the response vector.

A helper class is provided to expose a simple API to a 'list' of `systtools::SystParamHeaders` instance created from the parsing of a parameter headers document by helper methods found in `utility/ParameterAndProviderConfigurationUtility.hh`. The helper class definition is well documented and can be found in `interpreters/ParamHeaderHelper.hh`.

Chapter 5

Tool configuration

Description

As introduced in [Moving Parts](#), the 'tool configuration' is the human-writable FHiCL configuration file for a systematic provider. While the FHiCL table that gets passed to the `systtools::ISystProviderTool` instance is almost unstructured, there are tools provided that can speed development if some structure can be presumed.

Firstly, `art::make_tool` requires that the `fhiicl::ParameterSet` used to instantiate a subclass instance contains a string atom formed like `tool_type: SubclassClassName`, this is used to search for the tool factory method capable of instantiating instances of the named subclass. An optional `instance_name` string atom can be used to disambiguate multiple instances of the same `systtools::ISystProviderTool` subclass. In this example, we will use the only concrete implementation available in this package, [ExampleISystProvider](#). The final necessary component in a tool configuration document is a sequence of keys that correspond to the tool instances to be configured, by default it is expected to be named `syst_providers`, but this default can be overridden if needed. An example of a minimal tool configuration then is:

```
ExampleProvider_toolconfig: {
  tool_type: "ExampleISystProvider"
  instance_name: "myexample"
}
syst_providers: [ExampleProvider_toolconfig]
```

A number of configurations of the example `systtools::ISystProviderTool`, [ExampleISystProvider](#) can be seen in [ExampleISystProvider_tool.Config.fcl](#). For this example, we will use:

```
ExampleWeightProvider_multiuniverse_centersshift: {
  tool_type: "ExampleISystProvider"
  instance_name: "multiuniverse_centersshift"

  central_value: 1
  variation_descriptor: "{-2,2}"
  rand_dist: "uniform"

  number_of_throws: 10
}
```

The hope is that the intent of the tool configuration can be easily parsed by a non-expert human reader who has read the relevant documentation (which hopefully exists). The example here can be used to produce a parameter headers document that can be used to configure an instance of [ExampleISystProvider](#) that exposes a single systematic parameter; Event responses to the 10 randomly thrown parameter values (between -1 and 3) will be calculated for each event unit passed to the configured `ISystProviderTool`.

Converting to parameter headers

The conversion between tool configuration and parameter headers is requested via the `ISysProviderTool` interface; but the implementation is fully tool-specific. See [Writing A Provider](#) for a description of the helper methods available for speeding up development of such an implementation. An application for converting a well-formed tool configuration document (which may reference an arbitrary number of `ISysProviderTool` subclass instances) to parameter headers is provided `GenerateSysProviderConfig`. It uses the tool configuration tables referenced from the specified sequence (named `syst_providers` by default), uses `art::make_tool` to instantiate instances of each of the relevant `ISysProviderTools`, and then compiles the generated parameter header documents into a single FHiCL document. The application usage text reads:

```
[USAGE]: GenerateSysProviderConfig
```

```
-?|--help      : Show this message.
-l <policy_id> : FHiCL_FILE_PATH lookup policy:
                  0 : cet::filepath_maker
                  {1}: cet::filepath_lookup
                  2 : cet::filepath_lookup_nonabsolute
                  3 : cet::filepath_lookup_after1
-p <envvar name> : Environment variable to use when searching for fhiCL.
                  FHiCL_FILE_PATH by default.
-c <config.fcl> : fhiCL file to read.
-o <output.fcl> : fhiCL file to write, stdout by default.
-k <list key>   : fhiCL key to look for list of providers,
                  "syst_providers" by default.
-P             : Wrap output file in {BEGIN,END}_PROLOG.
```

Chapter 6

Writing a ‘ISystProviderTool’

The `ISystProviderTool` interface is well-documented in [interface/ISystProviderTool.hh](#), but some specifics are highlighted here for clarity.

Required

The interface defines three abstract methods that must be implemented by any subclass:

- `SystMetaData ISystProviderTool::BuildSystMetaData(fhicl::ParameterSet const &, paramId_t)`: This method performs the [Tool Configuration](#) to [Parameter Headers](#) conversion. The passed `paramId_t` should be used as the `SystParamHeader::systParamId` for first generated `SystParamHeader`. Any subsequent `SystParamHeader` returned `SystMetaData` should be sequentially numbered—this is checked.
- `bool ISystProviderTool::SetupResponseCalculator(fhicl::ParameterSet const &)`: This method performs the final instance configuration from the [Parameter Headers](#). After this method has been called, any setup required to calculate event responses is expected to have been performed. *N.B.* The parameter headers are read into the instance `SystMetaData` by non-virtual base-class methods and cannot be modified by subclasses, any `SystParamHeader` configuration must be done in `ISystProviderTool::BuildSystMetaData`.
- `std::unique_ptr<EventResponse> ISystProviderTool::GetEventResponse(art::Event const &)`: Once fully configured, subclasses should be able to calculate the relevant event responses to any parameters that they handle. *N.B.* Responses to parameters that do not effect a response for a given event can be omitted. Returning an empty `EventResponse` is fully valid.

Optional

A number of optional virtual methods are available for subclass override:

- `fhicl::ParameterSet GetExampleToolConfiguration()`: Subclasses should provide an example tool configuration document that can be specialized by users when performing physics analyses. This is not an abstract method to reduce the development burden, but it's override by subclasses is very strongly encouraged.
- `std::string AsString()`: Subclasses can provide a string representation of their state.
- `fhicl::ParameterSet GetExtraToolOptions()`: Subclasses that use additional configuration over and above the standard [Parameter Headers](#) format should return it here. It will automatically be built into the output parameter headers document for later initialization.

Utilities

For a concrete example of a systematic provider, see [ExampleSystProvider](#) and [systproviders/ExampleISystProvider_tool.cc](#). The rest of this section highlights utility methods and example process flows to simplify the process of implementing new systematic providers.

Tool Configuration Parsing

As `ISystProviderTool` instances must be configurable from a [Parameter Headers](#) document, the recommended method of configuration is first to translate the [Tool Configuration](#) document to a valid parameter headers document, and then write the initialization/setup routines in terms of this document. Methods to facilitate this process flow for a number of parameter description formats are provided in [utility/FHiCLSystParamHeaderUtility](#). These methods assume a somewhat standardized Tool Configuration FHiCL format that allows the specification of central values and variations—both distributed for interpolation and randomly thrown. An example Tool Config is shown below:

```
ExampleWeightProvider_multiuniverse_centershift: {
  tool_type: "ExampleISystProvider"
  instance_name: "multiuniverse_centershift"

  central_value: 1
  variation_descriptor: "{-2,2}"
  rand_dist: "uniform"

  number_of_throws: 10
}
```

For the example above, the method `ParseFHiCLVariationDescriptor` can be used to extract the `SystParamHeader::centralParamValue` of the parameter being configured as 1, and the `SystParamHeader::oneSigmaShifts` as -2 and 2. Then `MakeFHiCLDefinedRandomVariations` is used to make 10 random throws according to a uniform distribution width $2 - -2 = 4$ about the central value, 1. These thrown values are then set as the `SystParamHeader::paramVariations`. `SystParamHeader::isCorrection`, `SystParamHeader::isSplineable`, and `SystParamHeader::isRandomlyThrown` are also set to their relevant values given the nature of the parameter extracted from the tool configuration. These two helper methods can be called together for a slightly more structured document by the meta-helper: `ParseFHiCLSimpleToolConfigurationParameter`. This assumes that the `<pname>_central_value`, `<pname>_variation_descriptor`, and if relevant, `<pname>_nthrows` and `<pname>_random_distribution` keys are all named correctly for a parameter named `<pname>`. The `variation_descriptor` key can also be used to define a list of points to calculate, e.g. `variation_descriptor: "[-3, -2, -1, 0, 1, 2, 3]"`, for regular lists the shorthand `variation_descriptor: "(<start>, <stop>, <step>)"`, can be used. The form, random, list, regular list is chosen based upon the wrapping brackets, note that the specified list is not a FHiCL list, but a FHiCL atomic string. See the method documentation in [utility/FHiCLSystParamHeaderUtility](#) for more details.

Chapter 7

Namespace Index

7.1 Namespace List

Here is a list of all namespaces with brief descriptions:

cliopts	25
fhicl	27
systools	27

Chapter 8

Hierarchical Index

8.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

array	
systools::PolyResponse< Order >	110
systools::PolyResponse< n >	110
CovarianceThrower	51
EDAnalyzer	
SystToolsResponseTreeMaker	127
EDProducer	
SystToolsEventResponse	123
systools::EventSplineCacheBase< event_unit_t >	59
systools::EventSplineCache< event_unit_t, CLtight, Enable >	54
systools::EventSplineCache< event_unit_t, CLtight, typename std::enable_if< CLtight==Param↵ ValidationAndErrorResponse::kFrog, void >::type >	54
systools::EventSplineCache< event_unit_t, CLtight, typename std::enable_if< CLtight==Param↵ ValidationAndErrorResponse::kHare, void >::type >	56
systools::EventSplineCache< event_unit_t, CLtight, typename std::enable_if< CLtight==Param↵ ValidationAndErrorResponse::kTortoise, void >::type >	58
systools::EventSplineCacheBase< ULong_t >	59
systools::EventSplineCache< ULong_t, ParamValidationAndErrorResponse::kTortoise >	54
exception	
systools::systematictools_except	116
systools::ISystProviderTool	68
CorrelatedMultisimProvider	49
ExampleISystProvider	65
MD5	75
systools::ParamHeaderHelper	82
systools::ParamHeaderProviderName	101
systools::PrecalculatedResponseReader< Order >::ParamPolyResponses	102
systools::ParamResponses	102
systools::ParamThrows	103
ParamValidationAndErrorResponse	104
systools::ParamValue	109
systools::PrecalculatedResponseReader< Order >	111
systools::SystParamHeader	118
SystToolsEventResponseTree	125

Chapter 9

Class Index

9.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

CorrelatedMultisimProvider	49
CovarianceThrower	51
systools::EventSplineCache< event_unit_t, CLtight, Enable >	54
systools::EventSplineCache< event_unit_t, CLtight, typename std::enable_if< CLtight==Param↵ ValidationAndErrorResponse::kFrog, void >::type >	54
systools::EventSplineCache< event_unit_t, CLtight, typename std::enable_if< CLtight==Param↵ ValidationAndErrorResponse::kHare, void >::type >	56
systools::EventSplineCache< event_unit_t, CLtight, typename std::enable_if< CLtight==Param↵ ValidationAndErrorResponse::kTortoise, void >::type >	58
systools::EventSplineCacheBase< event_unit_t >	59
ExampleISystProvider	65
systools::ISystProviderTool	
ABC defining the interface to systematic response syst_providers	68
MD5	75
systools::ParamHeaderHelper	82
systools::ParamHeaderProviderName	
Struct for holding ISystProviderTool unique name–handled parameter header pairs	101
systools::PrecalculatedResponseReader< Order >::ParamPolyResponses	102
systools::ParamResponses	102
systools::ParamThrows	103
ParamValidationAndErrorResponse	104
systools::ParamValue	109
systools::PolyResponse< n >	110
systools::PrecalculatedResponseReader< Order >	111
systools::systematictools_except	116
systools::SystParamHeader	118
SystToolsEventResponse	123
SystToolsEventResponseTree	125
SystToolsResponseTreeMaker	127

Chapter 10

File Index

10.1 File List

Here is a list of all files with brief descriptions:

systematictools/app/CheckSystProviderConfigmd5.cc	131
systematictools/app/FindISystProvider.cc	131
systematictools/app/GenerateSystProviderConfig.cc	132
systematictools/interface/EventResponse_product.cc	134
systematictools/interface/EventResponse_product.hh	134
systematictools/interface/FHiCLSystParamHeaderConverters.cc	135
systematictools/interface/FHiCLSystParamHeaderConverters.hh	136
systematictools/interface/ISystProviderTool.cc	136
systematictools/interface/ISystProviderTool.hh	136
systematictools/interface/SystMetaData.cc	137
systematictools/interface/SystMetaData.hh	138
systematictools/interface/SystParamHeader.cc	139
systematictools/interface/SystParamHeader.hh	139
systematictools/interface/types.hh	140
systematictools/interpreters/EventSplineCacheHelper.hh	141
systematictools/interpreters/ParamHeaderHelper.cc	142
systematictools/interpreters/ParamHeaderHelper.hh	142
systematictools/interpreters/ParamValidationAndErrorResponse.cc	142
systematictools/interpreters/ParamValidationAndErrorResponse.hh	143
systematictools/interpreters/PolyResponse.hh	143
systematictools/interpreters/PrecalculatedResponseReader.hh	143
systematictools/module/classes.h	144
systematictools/module/SystToolsEventResponse_module.cc	144
systematictools/module/SystToolsResponseTreeMaker_module.cc	144
systematictools/systproviders/CorrelatedMultisimProvider_tool.cc	145
systematictools/systproviders/ExampleISystProvider_tool.cc	145
systematictools/systproviders/ExampleISystProvider_tool.hh	148
systematictools/utility/CovMatThrower.cc	148
systematictools/utility/CovMatThrower.hh	148
systematictools/utility/exceptions.hh	148
systematictools/utility/FHiCLSystParamHeaderUtility.cc	149
systematictools/utility/FHiCLSystParamHeaderUtility.hh	150
systematictools/utility/md5.cc	151
systematictools/utility/md5.hh	154
systematictools/utility/ParameterAndProviderConfigurationUtility.cc	154

systematictools/utility/ ParameterAndProviderConfigurationUtility.hh	155
systematictools/utility/ printers.hh	156
systematictools/utility/ ResponselessParamUtility.cc	156
systematictools/utility/ ResponselessParamUtility.hh	156
systematictools/utility/ ROOTUtility.hh	157
systematictools/utility/ string_parsers.hh	159

Chapter 11

Namespace Documentation

11.1 cliopts Namespace Reference

Variables

- std::string `fclname` = ""
- std::string `envvar` = "FHICL_FILE_PATH"
- std::string `producer_name` = ""
- std::string `analyzer_name` = ""
- int `lookup_policy` = 1
- std::string `provider_name` = ""
- bool `quiet` = false
- bool `dump_example_config` = false
- std::string `outputfile` = ""
- std::string `fhicl_key` = "syst_providers"
- bool `WrapWithPROLOG` = false

11.1.1 Variable Documentation

11.1.1.1 analyzer_name

```
std::string cliopts::analyzer_name = ""
```

11.1.1.2 dump_example_config

```
bool cliopts::dump_example_config = false
```

11.1.1.3 envvar

```
std::string cliopts::envvar = "FHICL_FILE_PATH"
```

11.1.1.4 fclname

```
std::string cliopts::fclname = ""
```

11.1.1.5 fhicl_key

```
std::string cliopts::fhicl_key = "syst_providers"
```

11.1.1.6 lookup_policy

```
int cliopts::lookup_policy = 1
```

11.1.1.7 outputfile

```
std::string cliopts::outputfile = ""
```

11.1.1.8 producer_name

```
std::string cliopts::producer_name = ""
```

11.1.1.9 provider_name

```
std::string cliopts::provider_name = ""
```

11.1.1.10 quiet

```
bool cliopts::quiet = false
```

11.1.1.11 WrapWithPROLOG

```
bool cliopts::WrapWithPROLOG = false
```

11.2 fhicl Namespace Reference

11.3 systtools Namespace Reference

Classes

- class [EventSplineCache](#)
- class [EventSplineCache< event_unit_t, CLtight, typename std::enable_if< CLtight==ParamValidationAnd↳ ErrorResponse::kFrog, void >::type >](#)
- class [EventSplineCache< event_unit_t, CLtight, typename std::enable_if< CLtight==ParamValidationAnd↳ ErrorResponse::kHare, void >::type >](#)
- class [EventSplineCache< event_unit_t, CLtight, typename std::enable_if< CLtight==ParamValidationAnd↳ ErrorResponse::kTortoise, void >::type >](#)
- class [EventSplineCacheBase](#)
- class [ISystProviderTool](#)
ABC defining the interface to systematic response syst_providers.
- class [ParamHeaderHelper](#)
- struct [ParamHeaderProviderName](#)
Struct for holding [ISystProviderTool](#) unique name–handled parameter header pairs.
- struct [ParamResponses](#)
- struct [ParamThrows](#)
- struct [ParamValue](#)
- struct [PolyResponse](#)
- class [PrecalculatedResponseReader](#)
- struct [systematictools_except](#)
- struct [SystParamHeader](#)

Typedefs

- typedef std::vector< [ParamResponses](#) > [event_unit_response_t](#)
- typedef std::vector< [event_unit_response_t](#) > [EventResponse](#)
The systematic parameter responses calculated for an ART event.
- typedef std::vector< [SystParamHeader](#) > [SystMetaData](#)
A list of Parameter Headers.
- typedef unsigned [paramId_t](#)
- typedef std::vector< [ParamValue](#) > [param_value_list_t](#)
- typedef std::vector< [paramId_t](#) > [param_list_t](#)
- typedef std::vector< [ParamThrows](#) > [parameter_throws_list_t](#)
- typedef std::vector< std::unique_ptr< [ISystProviderTool](#) > > [provider_list_t](#)
- typedef std::map< [paramId_t](#), [ParamHeaderProviderName](#) > [param_header_map_t](#)
Map of parameter Identifiers to the relevant metadata and the unique name of the [ISystProviderTool](#) responsible for generating them.
- typedef size_t [eventId_t](#)

Functions

- void [ExtendEventResponse](#) (std::unique_ptr< [EventResponse](#) > &e1, std::unique_ptr< [EventResponse](#) > &&e2)
Extends one EventResponse with the event_unit_response_ts of another.
- bool [FullOfUnity](#) (std::vector< double > const &vec, double tolerance=std::numeric_limits< double >::epsilon())
- void [ScrubUnityEventResponses](#) (std::unique_ptr< [EventResponse](#) > &er)
Removes [systools::ParamResponses](#) from event_unit_response_ts contained within an EventResponse that contain only unity responses.
- void [ScrubUnityEventResponses](#) ([event_unit_response_t](#) &er)
Removes [systools::ParamResponses](#) from event_unit_response_t that contain only unity responses.
- [NEW_SYSTTOOLS_EXCEPT](#) (incompatible_number_of_event_units)
Exception raised when attempting to merge two event responses with differing number of event units.
- [SystParamHeader FHiCLToSystParamHeader](#) (fhicl::ParameterSet const ¶mset)
Deserializes a [SystParamHeader](#) instance from a passed FHiCL parameter set.
- fhicl::ParameterSet [SystParamHeaderToFHiCL](#) ([SystParamHeader](#) const &sph)
Serializes a [SystParamHeader](#) instance to a FHiCL table.
- [NEW_SYSTTOOLS_EXCEPT](#) (invalid_SystParamHeader_key)
- [NEW_SYSTTOOLS_EXCEPT](#) (ISystProviderTool_method_unimplemented)
- [NEW_SYSTTOOLS_EXCEPT](#) (ISystProviderTool_seed_suggestion_post_configure)
- [NEW_SYSTTOOLS_EXCEPT](#) (ISystProviderTool_noncontiguous_parameter_ids)
- [NEW_SYSTTOOLS_EXCEPT](#) (ISystProviderTool_metadata_not_generated)
- [NEW_SYSTTOOLS_EXCEPT](#) (invalid_ToolConfigurationFHiCL)
- [NEW_SYSTTOOLS_EXCEPT](#) (invalid_ToolOptions)
- [paramId_t GetParamId](#) ([SystMetaData](#) const &md, std::string const &name)
Get parameter Id from a SystMetaData and pretty name.
- [size_t GetParamIndex](#) ([SystMetaData](#) const &md, [paramId_t](#) pid)
Get parameter index in header list for supplied parameter Id.
- bool [IndexIsHandled](#) ([SystMetaData](#) const &md, [size_t](#) index)
Whether a given index is handled by the Syst meta data headers.
- [size_t GetParamIndex](#) ([SystMetaData](#) const &md, std::string const &name)
Get parameter index in header list for supplied parameter pretty name.
- bool [HasParam](#) ([SystMetaData](#) const &md, std::string const &name)
Checks if named parameter exists in header list.
- bool [HasAnyParams](#) ([SystMetaData](#) const &md, std::vector< std::string > const &names)
Checks if any of the named parameters exists in header list.
- bool [HasParam](#) ([SystMetaData](#) const &md, [paramId_t](#) pid)
Checks if parameter with given Id exists in header list.
- [SystParamHeader](#) const & [GetParam](#) ([SystMetaData](#) const &md, std::string const &name)
Gets a const reference to a parameter header given a header list and a parameter pretty name.
- [SystParamHeader](#) & [GetParam](#) ([SystMetaData](#) &md, std::string const &name)
Gets a non-const reference to a parameter header given a header list and a parameter pretty name.
- [SystParamHeader](#) const & [GetParam](#) ([SystMetaData](#) const &md, [paramId_t](#) pid)
Gets a const reference to a parameter header given a header list and a parameter Id.
- [SystParamHeader](#) & [GetParam](#) ([SystMetaData](#) &md, [paramId_t](#) pid)
Gets a const reference to a parameter header given a header list and a parameter Id.
- bool [Validate](#) ([SystMetaData](#) const &sh, bool quiet=true)
Checks for declared and mis-used interdependency between parameters in a list of parameter headers.
- void [ExtendSystMetaData](#) ([SystMetaData](#) &md1, [SystMetaData](#) const &md2)
Merges two SystMetaData instances.
- [NEW_SYSTTOOLS_EXCEPT](#) (no_such_opt_kv)

- Exception raised when no key-value pair with a given key can be found in a given [SystParamHeader](#).*

 - [NEW_SYSTTOOLS_EXCEPT](#) (invalid_SystMetaData)

Exception raised if a SystMetaData fails basic interface validation.
 - template<typename T >

bool [SystHasOpt](#) ([SystMetaData](#) const &md, T const &ident, std::string const &opt)

Returns true if the Parameter Header specified by ident has a matching opts entry.
 - template<typename T >

bool [SystHasOptKV](#) ([SystMetaData](#) const &md, T const &ident, std::string const &key)

Returns true if the Parameter Header specified by ident has a matching opts key-value entry.
 - template<typename T >

std::string [SystGetOptKV](#) ([SystMetaData](#) const &md, T const &ident, std::string const &key)

Returns the option value corresponding to key on the Param Header specified by ident.
 - bool [Validate](#) ([SystParamHeader](#) const &hdr, bool quiet=true)

Checks interface validity of a [SystParamHeader](#).
 - [NEW_SYSTTOOLS_EXCEPT](#) (invalid_SystParamHeader)
 - template<typename T >

size_t [GetParamContainerIndex](#) (std::vector< T > const &container, paramId_t &pid)

Gets the index of a parameter-X association with a given paramId_t.
 - template<typename T >

bool [ContainterHasParam](#) (std::vector< T > const &container, paramId_t pid)
 - template<typename T >

T & [GetParamElementFromContainer](#) (std::vector< T > &container, paramId_t pid)
 - template<typename T >

T const & [GetParamElementFromContainer](#) (std::vector< T > const &container, paramId_t pid)
 - [NEW_SYSTTOOLS_EXCEPT](#) (invalid_tfile_name)
 - [NEW_SYSTTOOLS_EXCEPT](#) (invalid_ttree_name)
 - [NEW_SYSTTOOLS_EXCEPT](#) (invalid_parameter_name)
 - [NEW_SYSTTOOLS_EXCEPT](#) (invalid_parameter_Id)
 - [NEW_SYSTTOOLS_EXCEPT](#) (invalid_parameter_value)
 - [NEW_SYSTTOOLS_EXCEPT](#) (incorrectly_configured)
 - [NEW_SYSTTOOLS_EXCEPT](#) (parameter_Id_not_handled)
 - [NEW_SYSTTOOLS_EXCEPT](#) (parameter_name_not_handled)
 - [NEW_SYSTTOOLS_EXCEPT](#) (systParamId_collision)
 - bool [ParseFHiCLVariationDescriptor](#) (fhiCL::ParameterSet const ¶mset, std::string const &CV_key, std::string const &vardescriptor_key, [SystParamHeader](#) &hdr)

Set up [SystParamHeader](#) variation definitions from common format.
 - bool [MakeFHiCLDefinedRandomVariations](#) (fhiCL::ParameterSet const ¶mset, std::string const &nthrows_key, [SystParamHeader](#) &hdr, std::string const &distribution_key="", uint64_t seed=0, size_t NThrows=0)

Throws random parameter variations.
 - bool [FHiCLSimpleToolConfigurationParameterExists](#) (fhiCL::ParameterSet const ¶mset, std::string const ¶meter_name)

Checks if paramset appears to provide standardized Tool Configuration for a named parameter.
 - bool [ParseFHiCLSimpleToolConfigurationParameter](#) (fhiCL::ParameterSet const ¶mset, std::string const ¶meter_name, [SystParamHeader](#) &hdr, uint64_t seed=0, size_t NThrows=0)

Builds [SystParamHeader](#) from standardized FHiCL that can be used to write Tool Configuration files.
 - [NEW_SYSTTOOLS_EXCEPT](#) (invalid_FHiCL_variation_descriptor)
 - [NEW_SYSTTOOLS_EXCEPT](#) (invalid_FHiCL_random_distribution_descriptor)
 - param_header_map_t [BuildParameterHeaders](#) (fhiCL::ParameterSet const ¶mset, std::string const &key="syst_providers")

Builds map of SystProvider instances and handled parameters from a ParameterHeaders FHiCL document.
 - [NEW_SYSTTOOLS_EXCEPT](#) (ISystProvider_FQName_collision)

Exception thrown when two ISystProviderTools have identical fully qualified (tool_name + instance_name) names.

- `template<typename T = systools::ISystProviderTool>`
`param_header_map_t BuildParameterHeaders` (`std::vector< std::unique_ptr< T >> const &ConfiguredProviders`)
Builds map of SystProvider instances and handled parameters from a set of pre-configured providers.
- `template<typename T = systools::ISystProviderTool>`
`std::vector< std::unique_ptr< T >> ConfigureISystProvidersFromToolConfig` (`fhicl::ParameterSet const ¶mset, std::function< std::unique_ptr< T >(fhicl::ParameterSet const &)> InstanceBuilder=[](fhicl::ParameterSet const ¶mset) -> std::unique_ptr< T > { return art::make_tool< T >(paramset);}, std::string const &key="syst_providers", paramId_t syst_param_id=0`)
Configures the set of ISystProviders from a Tool Configuration document.
- `template<typename T = systools::ISystProviderTool>`
`std::vector< std::unique_ptr< T >> ConfigureISystProvidersFromParameterHeaders` (`fhicl::ParameterSet const ¶mset, std::function< std::unique_ptr< T >(fhicl::ParameterSet const &)> InstanceBuilder=[](fhicl::ParameterSet const ¶mset) -> std::unique_ptr< T > { return art::make_tool< T >(paramset);}, std::string const &key="syst_providers"`)
Configures the set of ISystProviders from a Parameter Headers document.
- `std::string to_str` (`SystParamHeader const &sph, bool indent=true`)
- `std::string to_str` (`EventResponse const &er`)
- `void FinalizeAndValidateDependentParameters` (`SystMetaData &, std::string const &response_parameter_name, std::vector< std::string > const &dependent_parameter_names`)
Checks that all responseless parameters in the set have the same number of configured variations and adds dummy variations for the response parameter.
- `template<typename T >`
`T str2T` (`std::string const &str`)
- `template<>`
`bool str2T< bool >` (`std::string const &str`)
- `template<typename T >`
`void AppendVect` (`std::vector< T > &target, std::vector< T > const &toApp`)
- `template<typename T >`
`std::vector< T > ParseToVect` (`std::string const &inp, std::string const &delim, bool PushEmpty=false, bool trimInput=true`)

Variables

- `constexpr double kDefaultDouble = 0xdeadbeef`
Magic values for signalling that a value is defaulted.
- `template<typename T >`
`constexpr T kParamUnhandled = std::numeric_limits<T>::max()`
- `template<>`
`constexpr double kParamUnhandled< double > = kDefaultDouble`

11.3.1 Typedef Documentation

11.3.1.1 event_unit_response_t

```
typedef std::vector<ParamResponses> systools::event_unit_response_t
```

11.3.1.2 eventId_t

```
typedef size_t systools::eventId_t
```

11.3.1.3 EventResponse

```
typedef std::vector<event_unit_response_t> systools::EventResponse
```

The systematic parameter responses calculated for an ART event.

For each 'object of interest' (e.g. neutrino interaction, muon track, ...) within an event, the relevant responses to parameter variations are stored.

Note

Use [systools::SystParamHeader](#) and [systools::ParamHeaderHelper](#) for response interpretation tools

11.3.1.4 param_header_map_t

```
typedef std::map<paramId_t, ParamHeaderProviderName> systools::param_header_map_t
```

Map of parameter Identifiers to the relevant metadata and the unique name of the [ISystProviderTool](#) responsible for generating them.

11.3.1.5 param_list_t

```
typedef std::vector<paramId_t> systools::param_list_t
```

11.3.1.6 param_value_list_t

```
typedef std::vector<ParamValue> systools::param_value_list_t
```

List of parameter–value associations

Useful for handling 'state' in an analysis that is sampling a parameter-space.

11.3.1.7 parameter_throws_list_t

```
typedef std::vector<ParamThrows> systools::parameter_throws_list_t
```

List of parameter–thrown value associations

Useful for tracking parameter values of multi-universe error propagation approaches.

11.3.1.8 paramId_t

```
typedef unsigned systtools::paramId_t
```

Parameter identifier.

Unique for a given parameter set configuration, but value-parameter associations should never be hard coded by consumers as they may change for different sets of systematic parameters.

11.3.1.9 provider_list_t

```
typedef std::vector<std::unique_ptr<ISystProviderTool> > systtools::provider_list_t
```

11.3.1.10 SystMetaData

```
typedef std::vector<SystParamHeader> systtools::SystMetaData
```

A list of Parameter Headers.

Used throughout the interface and interpreters as a 'database' of currently handled systematic parameters.

11.3.2 Function Documentation

11.3.2.1 AppendVect()

```
template<typename T >
void systtools::AppendVect (
    std::vector< T > & target,
    std::vector< T > const & toApp ) [inline]
```

11.3.2.2 BuildParameterHeaders() [1/2]

```
param_header_map_t systtools::BuildParameterHeaders (
    fhicl::ParameterSet const & paramset,
    std::string const & key = "syst_providers" )
```

Builds map of SystProvider instances and handled parameters from a ParameterHeaders FHiCL document.

Used by standalone interpreters to read response interpretation metadata from input FHiCL

11.3.2.3 BuildParameterHeaders() [2/2]

```
template<typename T = systtools::ISystProviderTool>
param_header_map_t systtools::BuildParameterHeaders (
    std::vector< std::unique_ptr< T >> const & ConfiguredProviders )
```

Builds map of SystProvider instances and handled parameters from a set of pre-configured providers.

Avoids reading the same FHiCL twice!

11.3.2.4 ConfigureISystProvidersFromParameterHeaders()

```
template<typename T = systtools::ISystProviderTool>
std::vector<std::unique_ptr<T> > systtools::ConfigureISystProvidersFromParameterHeaders (
    fhicl::ParameterSet const & paramset,
    std::function< std::unique_ptr< T >(fhicl::ParameterSet const &)> InstanceBuilder)
Builder = [] (fhicl::ParameterSet const &paramset) -> std::unique_ptr<T> { return art::make_tool<T>(paramset); },
    std::string const & key = "syst_providers" )
```

Configures the set of ISystProviders from a Parameter Headers document.

The structure of paramset must adhere to the Parameter Headers structure described in [systematictools/doc/ParameterHeaders.md](#)

The InstanceBuilder function argument is used to instantiate [ISystProviderTool](#) instances held by std::unique_ptrs. When running with ART support this will default to art::make_tool<systtools::ISystProviderTool>, but when running outside of art, other instantiators must be used.

11.3.2.5 ConfigureISystProvidersFromToolConfig()

```
template<typename T = systtools::ISystProviderTool>
std::vector<std::unique_ptr<T> > systtools::ConfigureISystProvidersFromToolConfig (
    fhicl::ParameterSet const & paramset,
    std::function< std::unique_ptr< T >(fhicl::ParameterSet const &)> InstanceBuilder)
Builder = [] (fhicl::ParameterSet const &paramset) -> std::unique_ptr<T> { return art::make_tool<T>(paramset); },
    std::string const & key = "syst_providers",
    paramId_t syst_param_id = 0 )
```

Configures the set of ISystProviders from a Tool Configuration document.

Some structure over the paramset is necessary (and is described in [systematictools/doc/ToolConfiguration.md](#)), but the FHiCL document passed to InstanceBuild is tool sub-class-specific. This is as opposed to ConfigureISystProvidersFromParameterHeaders which requires a rigidly structure document.

The InstanceBuilder function argument is used to instantiate [ISystProviderTool](#) instances held by std::unique_ptrs. When running with ART support this will default to art::make_tool<systtools::ISystProviderTool>, but when running outside of art, other instantiators must be used.

11.3.2.6 ContainterHasParam()

```
template<typename T >
bool systtools::ContainterHasParam (
    std::vector< T > const & container,
    paramId_t pid ) [inline]
```

Checks whether a parameter with paramId_t == pid is contained

Uses GetParamContainerIndex and so will work with any types that that method does.

11.3.2.7 ExtendEventResponse()

```
void systtools::ExtendEventResponse (
    std::unique_ptr< EventResponse > & e1,
    std::unique_ptr< EventResponse > && e2 )
```

Extends one EventResponse with the event_unit_response_ts of another.

The sizes of each EventResponse must be the same or a incompatible_number_of_event_units exception will be raised.

The event_unit_response_ts from e2 are moved into e1.

11.3.2.8 ExtendSystMetaData()

```
void systtools::ExtendSystMetaData (
    SystMetaData & md1,
    SystMetaData const & md2 )
```

Merges two SystMetaData instances.

The elements of the second parameter list are copied into the first parameter list.

Note

If either fail validation (systtools::invalid_SystMetaData), or if the second set uses a systParamId that is already used by the first set (systtools::systParamId_collision), then an exception is raised.

11.3.2.9 FHiCLSimpleToolConfigurationParameterExists()

```
bool systtools::FHiCLSimpleToolConfigurationParameterExists (
    fhiCL::ParameterSet const & paramset,
    std::string const & parameter_name )
```

Checks if paramset appears to provide standardized Tool Configuration for a named parameter.

If either "<parameter_name>_central_value" or "<parameter_name>_variation_descriptor" exist, the parameter named <parameter_name> is considered to exist in the configuration.

11.3.2.10 FHiCLToSystParamHeader()

```
SystParamHeader systools::FHiCLToSystParamHeader (
    fhiCL::ParameterSet const & paramset )
```

Deserializes a [SystParamHeader](#) instance from a passed FHiCL parameter set.

11.3.2.11 FinalizeAndValidateDependentParameters()

```
void systools::FinalizeAndValidateDependentParameters (
    SystMetaData & metadata,
    std::string const & response_parameter_name,
    std::vector< std::string > const & dependent_parameter_names )
```

Checks that all responseless parameters in the set have the same number of configured variations and adds dummy variations for the response parameter.

11.3.2.12 FullOfUnity()

```
bool systools::FullOfUnity (
    std::vector< double > const & vec,
    double tolerance = std::numeric_limits<double>::epsilon() )
```

11.3.2.13 GetParam() [1/4]

```
SystParamHeader const & systools::GetParam (
    SystMetaData const & md,
    std::string const & name )
```

Gets a const reference to a parameter header given a header list and a parameter pretty name.

Note

Throws on failure, look before you leap (or prepare a safety net).

11.3.2.14 GetParam() [2/4]

```
SystParamHeader & systools::GetParam (
    SystMetaData & md,
    std::string const & name )
```

Gets a non-const reference to a parameter header given a header list and a parameter pretty name.

Note

Throws on failure, look before you leap (or prepare a safety net).

11.3.2.15 GetParam() [3/4]

```
SystParamHeader const & systtools::GetParam (
    SystMetaData const & md,
    paramId_t pid )
```

Gets a const reference to a parameter header given a header list and a parameter Id.

Note

Throws on failure, look before you leap (or prepare a safety net).

11.3.2.16 GetParam() [4/4]

```
SystParamHeader & systtools::GetParam (
    SystMetaData & md,
    paramId_t pid )
```

Gets a const reference to a parameter header given a header list and a parameter Id.

Note

Throws on failure, look before you leap (or prepare a safety net).

11.3.2.17 GetParamContainerIndex()

```
template<typename T >
size_t systtools::GetParamContainerIndex (
    std::vector< T > const & container,
    paramId_t & pid )
```

Gets the index of a parameter-X association with a given paramId_t.

Returns kParamUnhandled<size_t> if parameter does not exist in the list

Useful for interacting with: param_value_list_t, parameter_throws_list_t, and

11.3.2.18 GetParamElementFromContainer() [1/2]

```
template<typename T >
T& systtools::GetParamElementFromContainer (
    std::vector< T > & container,
    paramId_t pid ) [inline]
```

Gets a reference to a contained element with paramId_t == pid.

Note

throws for non-contained elements. Look before you leap.

11.3.2.19 GetParamElementFromContainer() [2/2]

```
template<typename T >
T const& systools::GetParamElementFromContainer (
    std::vector< T > const & container,
    paramId_t pid ) [inline]
```

Gets a const reference to a contained element with paramId_t == pid.

Note

throws for non-contained elements. Look before you leap.

11.3.2.20 GetParamId()

```
paramId_t systools::GetParamId (
    SystMetaData const & md,
    std::string const & name )
```

Get parameter Id from a SystMetaData and pretty name.

Returns kParamUnhandled<paramId_t> on failure.

11.3.2.21 GetParamIndex() [1/2]

```
size_t systools::GetParamIndex (
    SystMetaData const & md,
    paramId_t pid )
```

Get parameter index in header list for supplied parameter Id.

Returns kParamUnhandled<size_t> on failure.

11.3.2.22 GetParamIndex() [2/2]

```
size_t systools::GetParamIndex (
    SystMetaData const & md,
    std::string const & name )
```

Get parameter index in header list for supplied parameter pretty name.

Returns kParamUnhandled<size_t> on failure.

11.3.2.23 HasAnyParams()

```
bool systools::HasAnyParams (
    SystMetaData const & md,
    std::vector< std::string > const & names )
```

Checks if any of the named parameters exists in header list.

11.3.2.24 HasParam() [1/2]

```
bool systtools::HasParam (
    SystMetaData const & md,
    std::string const & name )
```

Checks if named parameter exists in header list.

11.3.2.25 HasParam() [2/2]

```
bool systtools::HasParam (
    SystMetaData const & md,
    paramId_t pid )
```

Checks if parameter with given Id exists in header list.

11.3.2.26 IndexIsHandled()

```
bool systtools::IndexIsHandled (
    SystMetaData const & md,
    size_t index )
```

Whether a given index is handled by the Syst meta data headers.

11.3.2.27 MakeFHiCLDefinedRandomVariations()

```
bool systtools::MakeFHiCLDefinedRandomVariations (
    fhicl::ParameterSet const & paramset,
    std::string const & nthrows_key,
    SystParamHeader & hdr,
    std::string const & distribution_key = "",
    uint64_t seed = 0,
    size_t NThrows = 0 )
```

Throws random parameter variations.

Returns whether any throws were made. If `distribution_key` is not found, a gaussian distribution will be used. Currently handles "normal", "gaussian", and "uniform" distributions, other values will cause a `invalid_FHiCL_random_distribution_descriptor` exception to be thrown.

If `SystParamHeader::isRandomlyThrown` is not true, or `nthrows_key` cannot be found in `paramset` and the `NThrows` argument is 0, `hdr` is not modified.

If no seed is passed, the current time will be used.

11.3.2.28 NEW_SYSTTOOLS_EXCEPT() [1/23]

```
systtools::NEW_SYSTTOOLS_EXCEPT (
    invalid_FHiCL_variation_descriptor )
```

11.3.2.29 NEW_SYSTTOOLS_EXCEPT() [2/23]

```
systtools::NEW_SYSTTOOLS_EXCEPT (
    invalid_FHiCL_random_distribution_descriptor )
```

11.3.2.30 NEW_SYSTTOOLS_EXCEPT() [3/23]

```
systtools::NEW_SYSTTOOLS_EXCEPT (
    invalid_SystParamHeader_key )
```

Exception thrown when an unexpected key is found in a `fhiCL::ParameterSet` being parsed as a [SystParamHeader](#)

11.3.2.31 NEW_SYSTTOOLS_EXCEPT() [4/23]

```
systtools::NEW_SYSTTOOLS_EXCEPT (
    invalid_tfile_name )
```

11.3.2.32 NEW_SYSTTOOLS_EXCEPT() [5/23]

```
systtools::NEW_SYSTTOOLS_EXCEPT (
    no_such_opt_kv )
```

Exception raised when no key-value pair with a given key can be found in a given [SystParamHeader](#).

11.3.2.33 NEW_SYSTTOOLS_EXCEPT() [6/23]

```
systtools::NEW_SYSTTOOLS_EXCEPT (
    invalid_tttree_name )
```

11.3.2.34 NEW_SYSTTOOLS_EXCEPT() [7/23]

```
systtools::NEW_SYSTTOOLS_EXCEPT (
    invalid_SystMetaData )
```

Exception raised if a SystMetaData fails basic interface validation.

11.3.2.35 NEW_SYSTTOOLS_EXCEPT() [8/23]

```
systtools::NEW_SYSTTOOLS_EXCEPT (
    ISystProviderTool_method_unimplemented )
```

11.3.2.36 NEW_SYSTTOOLS_EXCEPT() [9/23]

```
systtools::NEW_SYSTTOOLS_EXCEPT (
    ISystProviderTool_seed_suggestion_post_configure )
```

11.3.2.37 NEW_SYSTTOOLS_EXCEPT() [10/23]

```
systtools::NEW_SYSTTOOLS_EXCEPT (
    ISystProviderTool_noncontiguous_parameter_Ids )
```

11.3.2.38 NEW_SYSTTOOLS_EXCEPT() [11/23]

```
systtools::NEW_SYSTTOOLS_EXCEPT (
    ISystProviderTool_metadata_not_generated )
```

11.3.2.39 NEW_SYSTTOOLS_EXCEPT() [12/23]

```
systtools::NEW_SYSTTOOLS_EXCEPT (
    ISystProvider_FQName_collision )
```

Exception thrown when two ISystProviderTools have identical fully qualified (tool_name + instance_name) names.

11.3.2.40 NEW_SYSTTOOLS_EXCEPT() [13/23]

```
systools::NEW_SYSTTOOLS_EXCEPT (
    invalid_ToolConfigurationFHiCL )
```

11.3.2.41 NEW_SYSTTOOLS_EXCEPT() [14/23]

```
systools::NEW_SYSTTOOLS_EXCEPT (
    invalid_ToolOptions )
```

11.3.2.42 NEW_SYSTTOOLS_EXCEPT() [15/23]

```
systools::NEW_SYSTTOOLS_EXCEPT (
    invalid_SystParamHeader )
```

Exception to be thrown when a [SystParamHeader](#) fails Validate N.B. It is not thrown by the validate method upon failure, but should be thrown by calling methods that cannot handle invalid SystParamHeaders.

11.3.2.43 NEW_SYSTTOOLS_EXCEPT() [16/23]

```
systools::NEW_SYSTTOOLS_EXCEPT (
    incompatible_number_of_event_units )
```

Exception raised when attempting to merge two event responses with differing number of event units.

11.3.2.44 NEW_SYSTTOOLS_EXCEPT() [17/23]

```
systools::NEW_SYSTTOOLS_EXCEPT (
    invalid_parameter_name )
```

11.3.2.45 NEW_SYSTTOOLS_EXCEPT() [18/23]

```
systools::NEW_SYSTTOOLS_EXCEPT (
    invalid_parameter_Id )
```

11.3.2.46 NEW_SYSTTOOLS_EXCEPT() [19/23]

```
systtools::NEW_SYSTTOOLS_EXCEPT (
    invalid_parameter_value )
```

11.3.2.47 NEW_SYSTTOOLS_EXCEPT() [20/23]

```
systtools::NEW_SYSTTOOLS_EXCEPT (
    incorrectly_configured )
```

11.3.2.48 NEW_SYSTTOOLS_EXCEPT() [21/23]

```
systtools::NEW_SYSTTOOLS_EXCEPT (
    parameter_Id_not_handled )
```

11.3.2.49 NEW_SYSTTOOLS_EXCEPT() [22/23]

```
systtools::NEW_SYSTTOOLS_EXCEPT (
    parameter_name_not_handled )
```

11.3.2.50 NEW_SYSTTOOLS_EXCEPT() [23/23]

```
systtools::NEW_SYSTTOOLS_EXCEPT (
    systParamId_collision )
```

11.3.2.51 ParseFHiCLSimpleToolConfigurationParameter()

```
bool systtools::ParseFHiCLSimpleToolConfigurationParameter (
    fhicl::ParameterSet const & paramset,
    std::string const & parameter_name,
    SystParamHeader & hdr,
    uint64_t seed = 0,
    size_t NThrows = 0 )
```

Builds [SystParamHeader](#) from standardized FHiCL that can be used to write Tool Configuration files.

Returns if parameter configuration keys were found.

Looks for the following keys in paramset:

- <parameter_name>_central_value
- <parameter_name>_variation_descriptor

e.g { MyParam_central_value: 0.5 MyParam_tweak_definition: (-3,3,1) } Will build the [SystParamHeader](#): { pretty↔
Name = MyParam centralParamValue = 0.5 paramVariations = [-3, -2, -1, 0, 1, 2, 3] }

Uses [ParseFHiCLVariationDescriptor](#) and [MakeFHiCLDefinedRandomVariations](#)

11.3.2.52 ParseFHiCLVariationDescriptor()

```
bool systools::ParseFHiCLVariationDescriptor (
    fhicl::ParameterSet const & paramset,
    std::string const & CV_key,
    std::string const & vardescrptor_key,
    SystParamHeader & hdr )
```

Set up [SystParamHeader](#) variation definitions from common format.

Returns whether any setup occurred

Uses string value of key in paramset to initialize [SystParamHeader](#) variation datamembers for a few common uses:

- New central value: { CV_key:
- ```
}
```
- [SystParamHeader::isCorrection](#) = true
  - [SystParamHeader::centralParamValue](#) =
- 
- One sigma shifts (sym): { vardescrptor\_key: "{<shift>}" }
    - [SystParamHeader::isRandomlyThrown](#) = true
    - [SystParamHeader::oneSigmaShifts](#) = {-<shift>, <shift>}
  - One sigma shifts (asym): { vardescrptor\_key: "{<low>, <high>}" }
    - [SystParamHeader::isRandomlyThrown](#) = true
    - [SystParamHeader::oneSigmaShifts](#) = {<low>, <high>}
  - List descriptor: { vardescrptor\_key: "(1,2,0.5)" }
    - [SystParamHeader::isSplineable](#) = true
    - [SystParamHeader::paramVariations](#) = {1, 1.5, 2}
  - Discrete variations: { vardescrptor\_key: "[5, 3, 1, 4]" }
    - [SystParamHeader::paramVariations](#) = {5, 3, 1, 4}

throws `invalid_FHiCL_variation_descriptor` on error

## 11.3.2.53 ParseToVect()

```
template<typename T >
std::vector<T> systools::ParseToVect (
 std::string const & inp,
 std::string const & delim,
 bool PushEmpty = false,
 bool trimInput = true) [inline]
```

**11.3.2.54 ScrubUnityEventResponses()** [1/2]

```
void systtools::ScrubUnityEventResponses (
 std::unique_ptr< EventResponse > & er)
```

Removes [systtools::ParamResponses](#) from `event_unit_response_ts` contained within an `EventResponse` that contain only unity responses.

**Note**

that this is intended to be applied to weight systematics that do not affect a given event

**11.3.2.55 ScrubUnityEventResponses()** [2/2]

```
void systtools::ScrubUnityEventResponses (
 event_unit_response_t & er)
```

Removes [systtools::ParamResponses](#) from `event_unit_response_t` that contain only unity responses.

**Note**

that this is intended to be applied to weight systematics that do not affect a given event

**11.3.2.56 str2T()**

```
template<typename T >
T systtools::str2T (
 std::string const & str) [inline]
```

**11.3.2.57 str2T< bool >()**

```
template<>
bool systtools::str2T< bool > (
 std::string const & str) [inline]
```

**11.3.2.58 SystGetOptKV()**

```
template<typename T >
std::string systtools::SystGetOptKV (
 SystMetaData const & md,
 T const & ident,
 std::string const & key) [inline]
```

Returns the option value corresponding to `key` on the Param Header specified by `ident`.

**Note**

Looks for an entry in [SystParamHeader::opts](#) that begins with `<key>=` and returns the rest of the string.

**11.3.2.59 SystHasOpt()**

```
template<typename T >
bool systtools::SystHasOpt (
 SystMetaData const & md,
 T const & ident,
 std::string const & opt) [inline]
```

Returns true if the Parameter Header specified by `ident` has a matching `opts` entry.

**11.3.2.60 SystHasOptKV()**

```
template<typename T >
bool systtools::SystHasOptKV (
 SystMetaData const & md,
 T const & ident,
 std::string const & key) [inline]
```

Returns true if the Parameter Header specified by `ident` has a matching `opts` key-value entry.

**Note**

Looks for an entry in [SystParamHeader::opts](#) that begins with `<key>=`

**11.3.2.61 SystParamHeaderToFHiCL()**

```
fhiCL::ParameterSet systtools::SystParamHeaderToFHiCL (
 SystParamHeader const & sph)
```

Serializes a `SystParamHeader` instance to a FHiCL table.

**11.3.2.62 to\_str()** [1/2]

```
std::string systtools::to_str (
 SystParamHeader const & sph,
 bool indent = true) [inline]
```

**11.3.2.63 to\_str()** [2/2]

```
std::string systtools::to_str (
 EventResponse const & er) [inline]
```

**11.3.2.64 Validate()** [1/2]

```
bool systtools::Validate (
 SystParamHeader const & hdr,
 bool quiet = true)
```

Checks interface validity of a [SystParamHeader](#).

Checks performed:

- Has valid Id
- Has non-empty pretty name
- If it is a correction:
  - Does it have a specified central value? (should)
  - Does it have any responses or parameter variations defined? (shouldn't)
- If it is not a correction, does it have at least one parameter variation specified?
- If it is marked as splineable:
  - Is it also marked as randomly thrown? (shouldn't)
  - Is it also marked as responseless? (shouldn't)
- If it is marked as responseless:
  - Does it have a corresponding response parameter? (should)
  - Does it have any responses defined? (shouldn't)
- If it is marked as not differing event-by-event:
  - Does it have header-level responses defined? (should)
  - Does it have parameter variations specified? (should unless marked as a correction)
- If it is marked as differing event-by-event, does it have header-level responses defined? (shouldn't)

### 11.3.2.65 Validate() [2/2]

```
bool systools::Validate (
 SystMetaData const & sh,
 bool quiet = true)
```

Checks for declared and mis-used interdependency between parameters in a list of parameter headers.

Checks performed:

- Are all header parameter Ids unique within the parameter set? (should)
- Do the declared response parameter of responless parameters exist within the set? (should)
- Do all associated responseless, and the response parameter itself, have the same number of parameter variations? (should, N.B. Can be 0 for corrections.)

## 11.3.3 Variable Documentation

### 11.3.3.1 kDefaultDouble

```
constexpr double systools::kDefaultDouble = 0xdeadbeef
```

Magic values for signalling that a value is defaulted.

### 11.3.3.2 kParamUnhandled

```
template<typename T >
constexpr T systools::kParamUnhandled = std::numeric_limits<T>::max()
```

Magic value for signalling that a parameter is not configured.

Often specialized with paramId\_t when requesting the Id of a named parameter, or with size\_t when requesting the index of a parameter.

### 11.3.3.3 kParamUnhandled< double >

```
template<>
constexpr double systools::kParamUnhandled< double > = kDefaultDouble
```





# Chapter 12

## Class Documentation

### 12.1 CorrelatedMultisimProvider Class Reference

Inheritance diagram for CorrelatedMultisimProvider:

Collaboration diagram for CorrelatedMultisimProvider:

#### Public Member Functions

- [CorrelatedMultisimProvider](#) (ParameterSet const &)
  - [SystMetaData ConfigureFromFHICL](#) (ParameterSet const &, [paramId\\_t](#))
  - bool [Configure](#) ()
  - std::unique\_ptr< [EventResponse](#) > [GetEventResponse](#) (art::Event &)
  - std::string [AsString](#) ()
- Sub-classes may override this method to provide string-representations of their state.*

#### Private Attributes

- std::unique\_ptr< CLHEP::HepRandomEngine > [RNgine](#)
- std::unique\_ptr< CLHEP::RandGaussQ > [RNJesus](#)
- provider\_map\_t [child\\_providers](#)

#### Additional Inherited Members

#### 12.1.1 Constructor & Destructor Documentation

##### 12.1.1.1 CorrelatedMultisimProvider()

```
CorrelatedMultisimProvider::CorrelatedMultisimProvider (
 ParameterSet const & params) [explicit]
```

## 12.1.2 Member Function Documentation

### 12.1.2.1 AsString()

```
std::string CorrelatedMultisimProvider::AsString () [virtual]
```

Sub-classes may override this method to provide string-representations of their state.

Reimplemented from [systools::ISystProviderTool](#).

### 12.1.2.2 Configure()

```
bool CorrelatedMultisimProvider::Configure ()
```

### 12.1.2.3 ConfigureFromFHICL()

```
SystMetaData CorrelatedMultisimProvider::ConfigureFromFHICL (
 ParameterSet const & params,
 paramId_t firstParamId)
```

### 12.1.2.4 GetEventResponse()

```
std::unique_ptr< EventResponse > CorrelatedMultisimProvider::GetEventResponse (
 art::Event & e)
```

## 12.1.3 Member Data Documentation

### 12.1.3.1 child\_providers

```
provider_map_t CorrelatedMultisimProvider::child_providers [private]
```

## 12.1.3.2 RNengine

```
std::unique_ptr<CLHEP::HepRandomEngine> CorrelatedMultisimProvider::RNengine [private]
```

## 12.1.3.3 RNJesus

```
std::unique_ptr<CLHEP::RandGaussQ> CorrelatedMultisimProvider::RNJesus [private]
```

The documentation for this class was generated from the following file:

- systematicstools/systproviders/[CorrelatedMultisimProvider\\_tool.cc](#)

## 12.2 CovarianceThrower Class Reference

```
#include <CovMatThrower.hh>
```

## Public Member Functions

- void [SetupDecomp](#) ()
- [CovarianceThrower](#) (TMatrixD &covmat, uint64\_t Seed=0)
- [CovarianceThrower](#) (TMatrixDSym &covmat, uint64\_t Seed=0)
- [CovarianceThrower](#) (TMatrixD &covmat, std::unique\_ptr< CLHEP::HepRandomEngine > &)
- [CovarianceThrower](#) (TMatrixDSym &covmat, std::unique\_ptr< CLHEP::HepRandomEngine > &)
- TMatrixD const \* [Throw](#) ()
- [~CovarianceThrower](#) ()

## Private Member Functions

- [CovarianceThrower](#) (int [NRows](#))

## Private Attributes

- TMatrixD \* [UncertMatrix](#)
- TMatrixD \* [LMatrix](#)
- TMatrixD \* [RVector](#)
- TMatrixD \* [CVector](#)
- std::unique\_ptr< CLHEP::HepRandomEngine > [RNengine](#)
- std::unique\_ptr< CLHEP::RandGaussQ > [RNJesus](#)
- int [NRows](#)

## 12.2.1 Constructor &amp; Destructor Documentation

**12.2.1.1 CovarianceThrower()** [1/5]

```
CovarianceThrower::CovarianceThrower (
 int NRows) [private]
```

**12.2.1.2 CovarianceThrower()** [2/5]

```
CovarianceThrower::CovarianceThrower (
 TMatrixD & covmat,
 uint64_t Seed = 0)
```

**12.2.1.3 CovarianceThrower()** [3/5]

```
CovarianceThrower::CovarianceThrower (
 TMatrixDSym & covmat,
 uint64_t Seed = 0)
```

**12.2.1.4 CovarianceThrower()** [4/5]

```
CovarianceThrower::CovarianceThrower (
 TMatrixD & covmat,
 std::unique_ptr< CLHEP::HepRandomEngine > & RNgine)
```

**12.2.1.5 CovarianceThrower()** [5/5]

```
CovarianceThrower::CovarianceThrower (
 TMatrixDSym & covmat,
 std::unique_ptr< CLHEP::HepRandomEngine > & RNgine)
```

**12.2.1.6 ~CovarianceThrower()**

```
CovarianceThrower::~CovarianceThrower () [inline]
```

**12.2.2 Member Function Documentation**

### 12.2.2.1 SetupDecomp()

```
void CovarianceThrower::SetupDecomp ()
```

### 12.2.2.2 Throw()

```
TMatrixD const * CovarianceThrower::Throw ()
```

## 12.2.3 Member Data Documentation

### 12.2.3.1 CVector

```
TMatrixD* CovarianceThrower::CVector [private]
```

### 12.2.3.2 LMatrix

```
TMatrixD* CovarianceThrower::LMatrix [private]
```

### 12.2.3.3 NRows

```
int CovarianceThrower::NRows [private]
```

### 12.2.3.4 RNgine

```
std::unique_ptr<CLHEP::HepRandomEngine> CovarianceThrower::RNgine [private]
```

### 12.2.3.5 RNJesus

```
std::unique_ptr<CLHEP::RandGaussQ> CovarianceThrower::RNJesus [private]
```

### 12.2.3.6 RVector

```
TMatrixD* CovarianceThrower::RVector [private]
```

### 12.2.3.7 UncertMatrix

```
TMatrixD* CovarianceThrower::UncertMatrix [private]
```

The documentation for this class was generated from the following files:

- [systematictools/utility/CovMatThrower.hh](#)
- [systematictools/utility/CovMatThrower.cc](#)

## 12.3 systools::EventSplineCache< event\_unit\_t, CLtight, Enable > Class Template Reference

```
#include <EventSplineCacheHelper.hh>
```

Inheritance diagram for systools::EventSplineCache< event\_unit\_t, CLtight, Enable >:

Collaboration diagram for systools::EventSplineCache< event\_unit\_t, CLtight, Enable >:

### Additional Inherited Members

The documentation for this class was generated from the following file:

- [systematictools/interpreters/EventSplineCacheHelper.hh](#)

## 12.4 systools::EventSplineCache< event\_unit\_t, CLtight, typename std::enable\_if< C↵ Ltight==ParamValidationAndErrorResponse::kFrog, void >::type > Class Template Reference

```
#include <EventSplineCacheHelper.hh>
```

Inheritance diagram for systools::EventSplineCache< event\_unit\_t, CLtight, typename std::enable\_if< C↵ Ltight==ParamValidationAndErrorResponse::kFrog, void >::type >:

Collaboration diagram for systools::EventSplineCache< event\_unit\_t, CLtight, typename std::enable\_if< C↵ Ltight==ParamValidationAndErrorResponse::kFrog, void >::type >:

## Public Member Functions

- double [GetEventWeightResponse](#) (paramId\_t i, eventId\_t eid, double v)
- double [GetEventWeightResponse](#) (paramId\_t i, eventId\_t eid)
- double [GetTotalEventWeightResponse](#) (eventId\_t eid)
- double [GetEventLateralResponse](#) (paramId\_t i, eventId\_t eid, double v)
- double [GetEventLateralResponse](#) (paramId\_t i, eventId\_t eid)

## Additional Inherited Members

### 12.4.1 Member Function Documentation

#### 12.4.1.1 GetEventLateralResponse() [1/2]

```
template<typename event_unit_t , ParamValidationAndErrorResponse::CareLevel CLtight>
double systools::EventSplineCache< event_unit_t, CLtight, typename std::enable_if< CLtight==Param↵
ValidationAndErrorResponse::kFrog, void >::type >::GetEventLateralResponse (
 paramId_t i,
 eventId_t eid,
 double v) [inline]
```

#### 12.4.1.2 GetEventLateralResponse() [2/2]

```
template<typename event_unit_t , ParamValidationAndErrorResponse::CareLevel CLtight>
double systools::EventSplineCache< event_unit_t, CLtight, typename std::enable_if< CLtight==Param↵
ValidationAndErrorResponse::kFrog, void >::type >::GetEventLateralResponse (
 paramId_t i,
 eventId_t eid) [inline]
```

#### 12.4.1.3 GetEventWeightResponse() [1/2]

```
template<typename event_unit_t , ParamValidationAndErrorResponse::CareLevel CLtight>
double systools::EventSplineCache< event_unit_t, CLtight, typename std::enable_if< CLtight==Param↵
ValidationAndErrorResponse::kFrog, void >::type >::GetEventWeightResponse (
 paramId_t i,
 eventId_t eid,
 double v) [inline]
```

#### 12.4.1.4 GetEventWeightResponse() [2/2]

```
template<typename event_unit_t , ParamValidationAndErrorResponse::CareLevel CLtight>
double systtools::EventSplineCache< event_unit_t, CLtight, typename std::enable_if< CLtight==Param↵
ValidationAndErrorResponse::kFrog, void >::type >::GetEventWeightResponse (
 paramId_t i,
 eventId_t eid) [inline]
```

#### 12.4.1.5 GetTotalEventWeightResponse()

```
template<typename event_unit_t , ParamValidationAndErrorResponse::CareLevel CLtight>
double systtools::EventSplineCache< event_unit_t, CLtight, typename std::enable_if< CLtight==Param↵
ValidationAndErrorResponse::kFrog, void >::type >::GetTotalEventWeightResponse (
 eventId_t eid) [inline]
```

The documentation for this class was generated from the following file:

- systematicstools/interpreters/[EventSplineCacheHelper.hh](#)

## 12.5 systtools::EventSplineCache< event\_unit\_t, CLtight, typename std::enable\_if< C↵ Ltight==ParamValidationAndErrorResponse::kHare, void >::type > Class Template Reference

```
#include <EventSplineCacheHelper.hh>
```

Inheritance diagram for systtools::EventSplineCache< event\_unit\_t, CLtight, typename std::enable\_if< C↵  
Ltight==ParamValidationAndErrorResponse::kHare, void >::type >:

Collaboration diagram for systtools::EventSplineCache< event\_unit\_t, CLtight, typename std::enable\_if< C↵  
Ltight==ParamValidationAndErrorResponse::kHare, void >::type >:

### Public Member Functions

- double [GetEventWeightResponse](#) (paramId\_t i, eventId\_t eid, double v)
- double [GetEventWeightResponse](#) (paramId\_t i, eventId\_t eid)
- double [GetTotalEventWeightResponse](#) (eventId\_t eid)
- double [GetEventLateralResponse](#) (paramId\_t i, eventId\_t eid, double v)
- double [GetEventLateralResponse](#) (paramId\_t i, eventId\_t eid)

### Additional Inherited Members

#### 12.5.1 Member Function Documentation



## 12.5.1.1 GetEventLateralResponse() [1/2]

```
template<typename event_unit_t , ParamValidationAndErrorResponse::CareLevel CLtight>
double systtools::EventSplineCache< event_unit_t, CLtight, typename std::enable_if< CLtight==Param↵
ValidationAndErrorResponse::kHare, void >::type >::GetEventLateralResponse (
 paramId_t i,
 eventId_t eid,
 double v) [inline]
```

## 12.5.1.2 GetEventLateralResponse() [2/2]

```
template<typename event_unit_t , ParamValidationAndErrorResponse::CareLevel CLtight>
double systtools::EventSplineCache< event_unit_t, CLtight, typename std::enable_if< CLtight==Param↵
ValidationAndErrorResponse::kHare, void >::type >::GetEventLateralResponse (
 paramId_t i,
 eventId_t eid) [inline]
```

## 12.5.1.3 GetEventWeightResponse() [1/2]

```
template<typename event_unit_t , ParamValidationAndErrorResponse::CareLevel CLtight>
double systtools::EventSplineCache< event_unit_t, CLtight, typename std::enable_if< CLtight==Param↵
ValidationAndErrorResponse::kHare, void >::type >::GetEventWeightResponse (
 paramId_t i,
 eventId_t eid,
 double v) [inline]
```

## 12.5.1.4 GetEventWeightResponse() [2/2]

```
template<typename event_unit_t , ParamValidationAndErrorResponse::CareLevel CLtight>
double systtools::EventSplineCache< event_unit_t, CLtight, typename std::enable_if< CLtight==Param↵
ValidationAndErrorResponse::kHare, void >::type >::GetEventWeightResponse (
 paramId_t i,
 eventId_t eid) [inline]
```

## 12.5.1.5 GetTotalEventWeightResponse()

```
template<typename event_unit_t , ParamValidationAndErrorResponse::CareLevel CLtight>
double systtools::EventSplineCache< event_unit_t, CLtight, typename std::enable_if< CLtight==Param↵
ValidationAndErrorResponse::kHare, void >::type >::GetTotalEventWeightResponse (
 eventId_t eid) [inline]
```

The documentation for this class was generated from the following file:

- systematictools/interpreters/[EventSplineCacheHelper.hh](#)

## 12.6 systools::EventSplineCache< event\_unit\_t, CLtight, typename std::enable\_if< C↵ Ltight==ParamValidationAndErrorResponse::kTortoise, void >::type > Class Tem- plate Reference

```
#include <EventSplineCacheHelper.hh>
```

Inheritance diagram for systools::EventSplineCache< event\_unit\_t, CLtight, typename std::enable\_if< C↵  
Ltight==ParamValidationAndErrorResponse::kTortoise, void >::type >:

Collaboration diagram for systools::EventSplineCache< event\_unit\_t, CLtight, typename std::enable\_if< C↵  
Ltight==ParamValidationAndErrorResponse::kTortoise, void >::type >:

### Public Member Functions

- double [GetEventWeightResponse](#) (paramId\_t i, eventId\_t eid, double v)
- double [GetEventWeightResponse](#) (paramId\_t i, eventId\_t eid)
- double [GetTotalEventWeightResponse](#) (eventId\_t eid)
- double [GetEventLateralResponse](#) (paramId\_t i, eventId\_t eid, double v)
- double [GetEventLateralResponse](#) (paramId\_t i, eventId\_t eid)

### Additional Inherited Members

#### 12.6.1 Member Function Documentation

##### 12.6.1.1 GetEventLateralResponse() [1/2]

```
template<typename event_unit_t , ParamValidationAndErrorResponse::CareLevel CLtight>
double systools::EventSplineCache< event_unit_t, CLtight, typename std::enable_if< CLtight==Param↵
ValidationAndErrorResponse::kTortoise, void >::type >::GetEventLateralResponse (
 paramId_t i,
 eventId_t eid,
 double v) [inline]
```

##### 12.6.1.2 GetEventLateralResponse() [2/2]

```
template<typename event_unit_t , ParamValidationAndErrorResponse::CareLevel CLtight>
double systools::EventSplineCache< event_unit_t, CLtight, typename std::enable_if< CLtight==Param↵
ValidationAndErrorResponse::kTortoise, void >::type >::GetEventLateralResponse (
 paramId_t i,
 eventId_t eid) [inline]
```

## 12.6.1.3 GetEventWeightResponse() [1/2]

```
template<typename event_unit_t , ParamValidationAndErrorResponse::CareLevel CLtight>
double systools::EventSplineCache< event_unit_t, CLtight, typename std::enable_if< CLtight==Param↵
ValidationAndErrorResponse::kTortoise, void >::type >::GetEventWeightResponse (
 paramId_t i,
 eventId_t eid,
 double v) [inline]
```

## 12.6.1.4 GetEventWeightResponse() [2/2]

```
template<typename event_unit_t , ParamValidationAndErrorResponse::CareLevel CLtight>
double systools::EventSplineCache< event_unit_t, CLtight, typename std::enable_if< CLtight==Param↵
ValidationAndErrorResponse::kTortoise, void >::type >::GetEventWeightResponse (
 paramId_t i,
 eventId_t eid) [inline]
```

## 12.6.1.5 GetTotalEventWeightResponse()

```
template<typename event_unit_t , ParamValidationAndErrorResponse::CareLevel CLtight>
double systools::EventSplineCache< event_unit_t, CLtight, typename std::enable_if< CLtight==Param↵
ValidationAndErrorResponse::kTortoise, void >::type >::GetTotalEventWeightResponse (
 eventId_t eid) [inline]
```

The documentation for this class was generated from the following file:

- systematicstools/interpreters/[EventSplineCacheHelper.hh](#)

## 12.7 systools::EventSplineCacheBase&lt; event\_unit\_t &gt; Class Template Reference

```
#include <EventSplineCacheHelper.hh>
```

Inheritance diagram for systools::EventSplineCacheBase< event\_unit\_t >:

Collaboration diagram for systools::EventSplineCacheBase< event\_unit\_t >:

## Public Types

- typedef std::vector< event\_unit\_t > [event\\_t](#)

## Public Member Functions

- [EventSplineCacheBase](#) ()
- [EventSplineCacheBase](#) ([param\\_header\\_map\\_t](#) headers)
- [EventSplineCacheBase](#) ([param\\_header\\_map\\_t](#) &&headers)
- void [SetHeaders](#) ([param\\_header\\_map\\_t](#) const &headers)
- void [SetHeaders](#) ([param\\_header\\_map\\_t](#) &&headers)
- void [SetChkErr](#) ([ParamValidationAndErrorResponse](#) const &ChkErr)
- [eventId\\_t](#) [CacheEvent](#) ([event\\_unit\\_t](#) const &eu, [event\\_unit\\_response\\_t](#) const &eur)  
*Take a copy of the event and build the internal splines from the supplied event information.*
- [std::vector< eventId\\_t >](#) [CacheEvents](#) ([event\\_t](#) const &e, [EventResponse](#) const &er)
- [std::vector< eventId\\_t >](#) [CacheEvents](#) ([event\\_t](#) &&e, [EventResponse](#) &&er)
- [size\\_t](#) [GetNEventsInCache](#) ()
- void [DeclareUsingParameter](#) ([paramId\\_t](#) i, double v=[kDefaultDouble](#))
- void [DeclareUsingParameters](#) ([param\\_value\\_map\\_t](#) const &ivmap)
- void [DeclareUsingParameters](#) ([param\\_list\\_t](#) const &ilist)
- void [SetParameterValue](#) ([paramId\\_t](#) i, double v)
- bool [KnowAboutParameter](#) ([paramId\\_t](#) i)
- void [SetParametersValue](#) ([param\\_value\\_map\\_t](#) const &ivmap)
- bool [ParameterAffectsEventWeight](#) ([paramId\\_t](#) i, [eventId\\_t](#) eid)
- bool [ParameterAffectsEventLateral](#) ([paramId\\_t](#) i, [eventId\\_t](#) eid)
- [event\\_unit\\_t](#) const & [GetEventUnit](#) ([eventId\\_t](#) eid)

## Protected Attributes

- [param\\_value\\_map\\_t](#) [currentValues](#)
- [param\\_list\\_t](#) [weightParams](#)
- [param\\_list\\_t](#) [lateralParams](#)
- [std::vector< std::pair< event\\_unit\\_t, std::pair< ParamHeaderHelper::param\\_tspline\\_map\\_t, ParamHeaderHelper::param\\_tspline\\_map\\_t > > >](#) [fEvents](#)
- [ParamHeaderHelper](#) [fHeaderHelper](#)
- [ParamValidationAndErrorResponse](#) [fChkErr](#)

## 12.7.1 Member Typedef Documentation

### 12.7.1.1 event\_t

```
template<typename event_unit_t>
typedef std::vector<event_unit_t> systtools::EventSplineCacheBase< event_unit_t >::event_t
```

## 12.7.2 Constructor & Destructor Documentation

## 12.7.2.1 EventSplineCacheBase() [1/3]

```
template<typename event_unit_t>
systools::EventSplineCacheBase< event_unit_t >::EventSplineCacheBase () [inline]
```

## 12.7.2.2 EventSplineCacheBase() [2/3]

```
template<typename event_unit_t>
systools::EventSplineCacheBase< event_unit_t >::EventSplineCacheBase (
 param_header_map_t headers) [inline]
```

## 12.7.2.3 EventSplineCacheBase() [3/3]

```
template<typename event_unit_t>
systools::EventSplineCacheBase< event_unit_t >::EventSplineCacheBase (
 param_header_map_t && headers) [inline]
```

## 12.7.3 Member Function Documentation

## 12.7.3.1 CacheEvent()

```
template<typename event_unit_t>
eventId_t systools::EventSplineCacheBase< event_unit_t >::CacheEvent (
 event_unit_t const & eu,
 event_unit_response_t const & eur) [inline]
```

Take a copy of the event and build the internal splines from the supplied event information.

## 12.7.3.2 CacheEvents() [1/2]

```
template<typename event_unit_t>
std::vector<eventId_t> systools::EventSplineCacheBase< event_unit_t >::CacheEvents (
 event_t const & e,
 EventResponse const & er) [inline]
```

**12.7.3.3 CacheEvents()** [2/2]

```
template<typename event_unit_t>
std::vector<eventId_t> systtools::EventSplineCacheBase< event_unit_t >::CacheEvents (
 event_t && e,
 EventResponse && er) [inline]
```

**12.7.3.4 DeclareUsingParameter()**

```
template<typename event_unit_t>
void systtools::EventSplineCacheBase< event_unit_t >::DeclareUsingParameter (
 paramId_t i,
 double v = kDefaultDouble) [inline]
```

**12.7.3.5 DeclareUsingParameters()** [1/2]

```
template<typename event_unit_t>
void systtools::EventSplineCacheBase< event_unit_t >::DeclareUsingParameters (
 param_value_map_t const & ivmap) [inline]
```

**12.7.3.6 DeclareUsingParameters()** [2/2]

```
template<typename event_unit_t>
void systtools::EventSplineCacheBase< event_unit_t >::DeclareUsingParameters (
 param_list_t const & ilist) [inline]
```

**12.7.3.7 GetEventUnit()**

```
template<typename event_unit_t>
event_unit_t const& systtools::EventSplineCacheBase< event_unit_t >::GetEventUnit (
 eventId_t eid) [inline]
```

**12.7.3.8 GetNEventsInCache()**

```
template<typename event_unit_t>
size_t systtools::EventSplineCacheBase< event_unit_t >::GetNEventsInCache () [inline]
```

## 12.7.3.9 KnowAboutParameter()

```
template<typename event_unit_t>
bool systools::EventSplineCacheBase< event_unit_t >::KnowAboutParameter (
 paramId_t i) [inline]
```

## 12.7.3.10 ParameterAffectsEventLateral()

```
template<typename event_unit_t>
bool systools::EventSplineCacheBase< event_unit_t >::ParameterAffectsEventLateral (
 paramId_t i,
 eventId_t eid) [inline]
```

## 12.7.3.11 ParameterAffectsEventWeight()

```
template<typename event_unit_t>
bool systools::EventSplineCacheBase< event_unit_t >::ParameterAffectsEventWeight (
 paramId_t i,
 eventId_t eid) [inline]
```

## 12.7.3.12 SetChkErr()

```
template<typename event_unit_t>
void systools::EventSplineCacheBase< event_unit_t >::SetChkErr (
 ParamValidationAndErrorResponse const & ChkErr) [inline]
```

## 12.7.3.13 SetHeaders() [1/2]

```
template<typename event_unit_t>
void systools::EventSplineCacheBase< event_unit_t >::SetHeaders (
 param_header_map_t const & headers) [inline]
```

## 12.7.3.14 SetHeaders() [2/2]

```
template<typename event_unit_t>
void systools::EventSplineCacheBase< event_unit_t >::SetHeaders (
 param_header_map_t && headers) [inline]
```

### 12.7.3.15 SetParametersValue()

```
template<typename event_unit_t>
void systtools::EventSplineCacheBase< event_unit_t >::SetParametersValue (
 param_value_map_t const & ivmap) [inline]
```

### 12.7.3.16 SetParameterValue()

```
template<typename event_unit_t>
void systtools::EventSplineCacheBase< event_unit_t >::SetParameterValue (
 paramId_t i,
 double v) [inline]
```

## 12.7.4 Member Data Documentation

### 12.7.4.1 currentValues

```
template<typename event_unit_t>
param_value_map_t systtools::EventSplineCacheBase< event_unit_t >::currentValues [protected]
```

### 12.7.4.2 fChkErr

```
template<typename event_unit_t>
ParamValidationAndErrorResponse systtools::EventSplineCacheBase< event_unit_t >::fChkErr
[protected]
```

### 12.7.4.3 fEvents

```
template<typename event_unit_t>
std::vector<std::pair<event_unit_t, std::pair<ParamHeaderHelper::param_tspline_map_t, Param↵
HeaderHelper::param_tspline_map_t> > > systtools::EventSplineCacheBase< event_unit_t >::f↵
Events [protected]
```

### 12.7.4.4 fHeaderHelper

```
template<typename event_unit_t>
ParamHeaderHelper systtools::EventSplineCacheBase< event_unit_t >::fHeaderHelper [protected]
```



#### 12.7.4.5 lateralParams

```
template<typename event_unit_t>
param_list_t systtools::EventSplineCacheBase< event_unit_t >::lateralParams [protected]
```

#### 12.7.4.6 weightParams

```
template<typename event_unit_t>
param_list_t systtools::EventSplineCacheBase< event_unit_t >::weightParams [protected]
```

The documentation for this class was generated from the following file:

- [systematictools/interpreters/EventSplineCacheHelper.hh](#)

## 12.8 ExampleISystProvider Class Reference

```
#include <ExampleISystProvider_tool.hh>
```

Inheritance diagram for ExampleISystProvider:

Collaboration diagram for ExampleISystProvider:

### Public Member Functions

- [ExampleISystProvider](#) (fhicl::ParameterSet const &)
- [systtools::SystMetaData BuildSystMetaData](#) (fhicl::ParameterSet const &, [systtools::paramId\\_t](#))  
*Convert tool-specific configuration fhicl parameter set into generic SystParamHeaders.*
- fhicl::ParameterSet [GetExtraToolOptions](#) ()  
*Gets any extra tool options generated during ConfigureFromToolConfig that aren't de-serializable to the SystParamHeader format.*
- bool [SetupResponseCalculator](#) (fhicl::ParameterSet const &)  
*Any further configuration required by a subclass before GetEventResponse can be called.*
- std::unique\_ptr< [systtools::EventResponse](#) > [GetEventResponse](#) (art::Event const &)
- std::string [AsString](#) ()  
*Sub-classes may override this method to provide string-representations of their state.*

### Private Attributes

- bool [applyToAll](#)
- std::unique\_ptr< std::mt19937\_64 > [RNGine](#)
- std::unique\_ptr< std::normal\_distribution< double > > [RNJesus](#)

## Additional Inherited Members

### 12.8.1 Constructor & Destructor Documentation

#### 12.8.1.1 ExampleISystProvider()

```
ExampleISystProvider::ExampleISystProvider (
 fhicl::ParameterSet const &) [explicit]
```

### 12.8.2 Member Function Documentation

#### 12.8.2.1 AsString()

```
std::string ExampleISystProvider::AsString () [virtual]
```

Sub-classes may override this method to provide string-representations of their state.

Reimplemented from [systtools::ISystProviderTool](#).

#### 12.8.2.2 BuildSystMetaData()

```
SystMetaData ExampleISystProvider::BuildSystMetaData (
 fhicl::ParameterSet const & ,
 systtools::paramId_t) [virtual]
```

Convert tool-specific configuration fhicl parameter set into generic SystParamHeaders.

Implements [systtools::ISystProviderTool](#).

#### 12.8.2.3 GetEventResponse()

```
std::unique_ptr< EventResponse > ExampleISystProvider::GetEventResponse (
 art::Event const & e) [virtual]
```

Implements [systtools::ISystProviderTool](#).

#### 12.8.2.4 GetExtraToolOptions()

```
ParameterSet ExampleISystProvider::GetExtraToolOptions () [virtual]
```

Gets any extra tool options generated during `ConfigureFromToolConfig` that aren't de-serializable to the `SystParamHeader` format.

Reimplemented from [systtools::ISystProviderTool](#).

#### 12.8.2.5 SetupResponseCalculator()

```
bool ExampleISystProvider::SetupResponseCalculator (
 fhicl::ParameterSet const &) [virtual]
```

Any further configuration required by a subclass before `GetEventResponse` can be called.

This is meant for setting up slave weight calculators that are needed to calculate event responses but not for parameter variation re-interpretation.

Configuration returned by `GetExtraToolOptions` after initial Tool Configuration will be passed into here during `ConfigureFromParameterHeaders`

Implements [systtools::ISystProviderTool](#).

### 12.8.3 Member Data Documentation

#### 12.8.3.1 applyToAll

```
bool ExampleISystProvider::applyToAll [private]
```

#### 12.8.3.2 RNgine

```
std::unique_ptr<std::mt19937_64> ExampleISystProvider::RNgine [private]
```

#### 12.8.3.3 RNJesus

```
std::unique_ptr<std::normal_distribution<double> > ExampleISystProvider::RNJesus [private]
```

The documentation for this class was generated from the following files:

- [systematictools/systproviders/ExampleISystProvider\\_tool.hh](#)
- [systematictools/systproviders/ExampleISystProvider\\_tool.cc](#)

## 12.9 systools::ISystProviderTool Class Reference

ABC defining the interface to systematic response syst\_providers.

```
#include <ISystProviderTool.hh>
```

Inheritance diagram for systools::ISystProviderTool:

### Public Member Functions

- [ISystProviderTool](#) (fhicl::ParameterSet const &ps)  
*ABC constructor required for art::make\_tool.*
- template<typename T >  
bool [ParamsHandled](#) (T ident)  
*Check if instance handles parameter.*
- [paramId\\_t](#) [GetParameterId](#) (std::string const &prettyName)  
*Get paramId\_t for handled, named parameter.*
- template<typename T >  
size\_t [GetNVariations](#) (T ident)  
*Get the number of variations to be calculated for parameter i.*
- void [SuggestSeed](#) (uint64\_t seed)  
*Allows RNG seeds to be suggested to tool instances.*
- virtual void [SuggestParameterThrows](#) ([parameter\\_throws\\_list\\_t](#) &&throws, bool Check=false)  
*Suggest a list of parameter throws to an instance.*
- virtual fhicl::ParameterSet [GetExampleToolConfiguration](#) ()  
*Sub-classes may override this method to provide an example Tool Configuration FHiCL document.*
- void [ConfigureFromToolConfig](#) (fhicl::ParameterSet const &ps, [paramId\\_t](#) firstId)  
*Configure an ISystProvider instance with tool-specific FHiCL.*
- [SystMetaData](#) const & [GetSystMetaData](#) ()  
*Gets the currently configured set of systematic parameter headers.*
- fhicl::ParameterSet [GetParameterHeadersDocument](#) ()  
*Build the Parameter Headers FHiCL document that can be used to re-configure an instance of this tool via [ConfigureFromParameterHeaders](#).*
- bool [ConfigureFromParameterHeaders](#) (fhicl::ParameterSet const &ps)  
*Try and read parameter configuration from input fhicl file.*
- virtual std::unique\_ptr< [EventResponse](#) > [GetEventResponse](#) (art::Event const &)=0
- std::string const & [GetToolType](#) () const
- std::string const & [GetFullyQualifiedName](#) () const
- std::string const & [GetInstanceName](#) () const
- virtual std::string [AsString](#) ()  
*Sub-classes may override this method to provide string-representations of their state.*
- virtual [~ISystProviderTool](#) ()

### Protected Member Functions

- virtual [SystMetaData](#) [BuildSystMetaData](#) (fhicl::ParameterSet const &, [paramId\\_t](#))=0  
*Convert tool-specific configuration fhicl parameter set into generic SystParamHeaders.*
- virtual fhicl::ParameterSet [GetExtraToolOptions](#) ()  
*Gets any extra tool options generated during [ConfigureFromToolConfig](#) that aren't de-serializable to the [SystParamHeader](#) format.*
- virtual bool [SetupResponseCalculator](#) (fhicl::ParameterSet const &)=0  
*Any further configuration required by a subclass before [GetEventResponse](#) can be called.*
- void [CheckHaveMetaData](#) ([paramId\\_t](#) i=kParamUnhandled< [paramId\\_t](#) >)  
*Checks if internal parameter metadata has been generated or loaded from a Parameter Headers file.*

## Protected Attributes

- `std::string fToolType`  
*Class name of the tool implementation.*
- `std::string fInstanceName`
- `std::string fQName`  
*The unique name of the tool instance: <fToolType>\_<fInstanceName>*
- `uint64_t fSeedSuggestion`  
*A suggested seed.*
- `bool flsFullyConfigured`  
*Whether this instance successfully configured itself.*

## Private Attributes

- `bool fHaveSystMetaData`  
*Whether this instance has generated/loaded its parameter set.*
- `SystMetaData fSystMetaData`  
*The SystMetaData describing the parameters handled by this tool.*

### 12.9.1 Detailed Description

ABC defining the interface to systematic response syst\_providers.

### 12.9.2 Constructor & Destructor Documentation

#### 12.9.2.1 ISystProviderTool()

```
systools::ISystProviderTool::ISystProviderTool (
 fhicl::ParameterSet const & ps)
```

ABC constructor required for art::make\_tool.

#### 12.9.2.2 ~ISystProviderTool()

```
virtual systools::ISystProviderTool::~~ISystProviderTool () [inline], [virtual]
```

### 12.9.3 Member Function Documentation

### 12.9.3.1 AsString()

```
virtual std::string systtools::ISystProviderTool::AsString () [inline], [virtual]
```

Sub-classes may override this method to provide string-representations of their state.

Reimplemented in [CorrelatedMultisimProvider](#), and [ExampleISystProvider](#).

### 12.9.3.2 BuildSystMetaData()

```
virtual SystMetaData systtools::ISystProviderTool::BuildSystMetaData (
 fhicl::ParameterSet const & ,
 paramId_t) [protected], [pure virtual]
```

Convert tool-specific configuration fhicl parameter set into generic SystParamHeaders.

Implemented in [ExampleISystProvider](#).

### 12.9.3.3 CheckHaveMetaData()

```
void systtools::ISystProviderTool::CheckHaveMetaData (
 paramId_t i = kParamUnhandled<paramId_t>) [protected]
```

Checks if internal parameter metadata has been generated or loaded from a Parameter Headers file.

If i is passed then it only checks for that specific paramId\_t.

Throws if no such metadata can be found.

### 12.9.3.4 ConfigureFromParameterHeaders()

```
bool systtools::ISystProviderTool::ConfigureFromParameterHeaders (
 fhicl::ParameterSet const & ps)
```

Try and read parameter configuration from input fhicl file.

After reading parameters, the pure virtual SetupResponseCalculator method is called for any final subclass configuration.

#### Note

Sub-classes may not alter fSystMetaData during the configure call. This is checked for by md5-ing the stringified fhicl representation of the parameters before and after the call.

### 12.9.3.5 ConfigureFromToolConfig()

```
void systools::ISystProviderTool::ConfigureFromToolConfig (
 fhicl::ParameterSet const & ps,
 paramId_t firstId)
```

Configure an ISystProvider instance with tool-specific FHiCL.

Takes the tool-specific FHiCL configuration and the paramId\_t of the first unused paramId\_t (closest to 0) and builds the parameter metadata that can be used to configure the ISystProvider for response calculation and also interpret the calculated responses.

Validates that the SystParamHeaders created by the subclass in BuildSystMetaData are contiguous.

### 12.9.3.6 GetEventResponse()

```
virtual std::unique_ptr<EventResponse> systools::ISystProviderTool::GetEventResponse (
 art::Event const &) [pure virtual]
```

Implemented in [ExampleISystProvider](#).

### 12.9.3.7 GetExampleToolConfiguration()

```
virtual fhicl::ParameterSet systools::ISystProviderTool::GetExampleToolConfiguration ()
[inline], [virtual]
```

Sub-classes may override this method to provide an example Tool Configuration FHiCL document.

### 12.9.3.8 GetExtraToolOptions()

```
virtual fhicl::ParameterSet systools::ISystProviderTool::GetExtraToolOptions () [inline],
[protected], [virtual]
```

Gets any extra tool options generated during ConfigureFromToolConfig that aren't de-serializable to the [Syst↔ParamHeader](#) format.

Reimplemented in [ExampleISystProvider](#).

### 12.9.3.9 GetFullyQualifiedName()

```
std::string const& systools::ISystProviderTool::GetFullyQualifiedName () const [inline]
```

**12.9.3.10 GetInstanceName()**

```
std::string const& systools::ISystProviderTool::GetInstanceName () const [inline]
```

**12.9.3.11 GetNVariations()**

```
template<typename T >
size_t systools::ISystProviderTool::GetNVariations (
 T ident) [inline]
```

Get the number of variations to be calculated for parameter i.

**12.9.3.12 GetParameterHeadersDocument()**

```
fhicl::ParameterSet systools::ISystProviderTool::GetParameterHeadersDocument ()
```

Build the Parameter Headers FHiCL document that can be used to re-configure an instance of this tool via `ConfigureFromParameterHeaders`.

If a sub-class requires extra configuration options they should be exposed through `GetExtraToolOptions`

**12.9.3.13 GetParameterId()**

```
paramId_t systools::ISystProviderTool::GetParameterId (
 std::string const & prettyName)
```

Get `paramId_t` for handled, named parameter.

**12.9.3.14 GetSystMetaData()**

```
SystMetaData const & systools::ISystProviderTool::GetSystMetaData ()
```

Gets the currently configured set of systematic parameter headers.

Checks that the headers have been built/loaded with `CheckHaveMetaData`, which throws if they haven't.

**12.9.3.15 GetToolType()**

```
std::string const& systools::ISystProviderTool::GetToolType () const [inline]
```



**12.9.3.16 ParamIsHandled()**

```
template<typename T >
bool systools::ISystProviderTool::ParamIsHandled (
 T ident) [inline]
```

Check if instance handles parameter.

Uses helper methods in [systematictools/interface/SystMetaData.hh](#) to check for parameters identified by param↵  
Id\_t or std::string

**12.9.3.17 SetupResponseCalculator()**

```
virtual bool systools::ISystProviderTool::SetupResponseCalculator (
 fhicl::ParameterSet const &) [protected], [pure virtual]
```

Any further configuration required by a subclass before GetEventResponse can be called.

This is meant for setting up slave weight calculators that are needed to calculate event responses but not for parameter variation re-interpretation.

Configuration returned by GetExtraToolOptions after initial Tool Configuration will be passed into here during ConfigureFromParameterHeaders

Implemented in [ExampleISystProvider](#).

**12.9.3.18 SuggestParameterThrows()**

```
void systools::ISystProviderTool::SuggestParameterThrows (
 parameter_throws_list_t && throws,
 bool Check = false) [virtual]
```

Suggest a list of parameter throws to an instance.

Allows a meta provider to be written that delegates well-correlated throws to multiple child providers.

**Note**

Unfortunately must be public as sub-classes do not get access to protected member functions via a base class pointer (as they may actually call a protected member of another subclass).

**12.9.3.19 SuggestSeed()**

```
void systools::ISystProviderTool::SuggestSeed (
 uint64_t seed)
```

Allows RNG seeds to be suggested to tool instances.

Instances should use this seed to deterministically generate random numbers.

This also stops many syst providers being set up in quick succession all using similar seeds.

## 12.9.4 Member Data Documentation

### 12.9.4.1 fQName

```
std::string systools::ISystProviderTool::fQName [protected]
```

The unique name of the tool instance: <fToolType>\_<fInstanceName>

### 12.9.4.2 fHaveSystMetaData

```
bool systools::ISystProviderTool::fHaveSystMetaData [private]
```

Whether this instance has generated/loaded its parameter set.

### 12.9.4.3 fInstanceName

```
std::string systools::ISystProviderTool::fInstanceName [protected]
```

A name for an instance of a tool used to disambiguate multiple instances of the same tool.

### 12.9.4.4 fIsFullyConfigured

```
bool systools::ISystProviderTool::fIsFullyConfigured [protected]
```

Whether this instance successfully configured itself.

Tools may be configured either from a "tool configuration" file, or from a "parameter headers" file.

### 12.9.4.5 fSeedSuggestion

```
uint64_t systools::ISystProviderTool::fSeedSuggestion [protected]
```

A suggested seed.

Should be used by tool implementations to seed any RNGs to allow deterministic random numbers across separate executions.

## 12.9.4.6 fSystMetaData

```
SystMetaData systtools::ISystProviderTool::fSystMetaData [private]
```

The SystMetaData describing the parameters handled by this tool.

## Note

Only the base class is allowed to alter the SystMetaData after the original generation. Subclasses and external callers may use GetSystMetaData to inspect it.

## 12.9.4.7 fToolType

```
std::string systtools::ISystProviderTool::fToolType [protected]
```

Class name of the tool implementation.

The documentation for this class was generated from the following files:

- systematictools/interface/[ISystProviderTool.hh](#)
- systematictools/interface/[ISystProviderTool.cc](#)

## 12.10 MD5 Class Reference

```
#include <md5.hh>
```

## Public Types

- typedef unsigned int [size\\_type](#)

## Public Member Functions

- [MD5](#) ()
- [MD5](#) (const std::string &text)
- void [update](#) (const unsigned char \*buf, [size\\_type](#) length)
- void [update](#) (const char \*buf, [size\\_type](#) length)
- [MD5](#) & [finalize](#) ()
- std::string [hexdigest](#) () const

## Private Types

- enum { [blocksize](#) = 64 }
- typedef uint8\_t [uint1](#)
- typedef uint32\_t [uint4](#)

## Private Member Functions

- void `init` ()
- void `transform` (const `uint1` block[`blocksize`])

## Static Private Member Functions

- static void `decode` (`uint4` output[], const `uint1` input[], `size_type` len)
- static void `encode` (`uint1` output[], const `uint4` input[], `size_type` len)
- static `uint4` `F` (`uint4` x, `uint4` y, `uint4` z)
- static `uint4` `G` (`uint4` x, `uint4` y, `uint4` z)
- static `uint4` `H` (`uint4` x, `uint4` y, `uint4` z)
- static `uint4` `I` (`uint4` x, `uint4` y, `uint4` z)
- static `uint4` `rotate_left` (`uint4` x, int n)
- static void `FF` (`uint4` &a, `uint4` b, `uint4` c, `uint4` d, `uint4` x, `uint4` s, `uint4` ac)
- static void `GG` (`uint4` &a, `uint4` b, `uint4` c, `uint4` d, `uint4` x, `uint4` s, `uint4` ac)
- static void `HH` (`uint4` &a, `uint4` b, `uint4` c, `uint4` d, `uint4` x, `uint4` s, `uint4` ac)
- static void `II` (`uint4` &a, `uint4` b, `uint4` c, `uint4` d, `uint4` x, `uint4` s, `uint4` ac)

## Private Attributes

- bool `finalized`
- `uint1` `buffer` [`blocksize`]
- `uint4` `count` [2]
- `uint4` `state` [4]
- `uint1` `digest` [16]

## Friends

- `std::ostream` & `operator<<` (`std::ostream` &, `MD5` md5)

## 12.10.1 Member Typedef Documentation

### 12.10.1.1 `size_type`

```
typedef unsigned int MD5::size_type
```

### 12.10.1.2 `uint1`

```
typedef uint8_t MD5::uint1 [private]
```

### 12.10.1.3 uint4

```
typedef uint32_t MD5::uint4 [private]
```

## 12.10.2 Member Enumeration Documentation

### 12.10.2.1 anonymous enum

```
anonymous enum [private]
```

#### Enumerator

|           |  |
|-----------|--|
| blocksize |  |
|-----------|--|

## 12.10.3 Constructor & Destructor Documentation

### 12.10.3.1 MD5() [1/2]

```
MD5::MD5 ()
```

### 12.10.3.2 MD5() [2/2]

```
MD5::MD5 (
 const std::string & text)
```

## 12.10.4 Member Function Documentation

### 12.10.4.1 decode()

```
void MD5::decode (
 uint4 output[],
 const uint1 input[],
 size_type len) [static], [private]
```

#### 12.10.4.2 encode()

```
void MD5::encode (
 uint1 output[],
 const uint4 input[],
 size_type len) [static], [private]
```

#### 12.10.4.3 F()

```
MD5::uint4 MD5::F (
 uint4 x,
 uint4 y,
 uint4 z) [inline], [static], [private]
```

#### 12.10.4.4 FF()

```
void MD5::FF (
 uint4 & a,
 uint4 b,
 uint4 c,
 uint4 d,
 uint4 x,
 uint4 s,
 uint4 ac) [inline], [static], [private]
```

#### 12.10.4.5 finalize()

```
MD5 & MD5::finalize ()
```

#### 12.10.4.6 G()

```
MD5::uint4 MD5::G (
 uint4 x,
 uint4 y,
 uint4 z) [inline], [static], [private]
```

#### 12.10.4.7 GG()

```
void MD5::GG (
 uint4 & a,
 uint4 b,
 uint4 c,
 uint4 d,
 uint4 x,
 uint4 s,
 uint4 ac) [inline], [static], [private]
```

#### 12.10.4.8 H()

```
MD5::uint4 MD5::H (
 uint4 x,
 uint4 y,
 uint4 z) [inline], [static], [private]
```

#### 12.10.4.9 hexdigest()

```
std::string MD5::hexdigest () const
```

#### 12.10.4.10 HH()

```
void MD5::HH (
 uint4 & a,
 uint4 b,
 uint4 c,
 uint4 d,
 uint4 x,
 uint4 s,
 uint4 ac) [inline], [static], [private]
```

#### 12.10.4.11 I()

```
MD5::uint4 MD5::I (
 uint4 x,
 uint4 y,
 uint4 z) [inline], [static], [private]
```

**12.10.4.12 II()**

```
void MD5::II (
 uint4 & a,
 uint4 b,
 uint4 c,
 uint4 d,
 uint4 x,
 uint4 s,
 uint4 ac) [inline], [static], [private]
```

**12.10.4.13 init()**

```
void MD5::init () [private]
```

**12.10.4.14 rotate\_left()**

```
MD5::uint4 MD5::rotate_left (
 uint4 x,
 int n) [inline], [static], [private]
```

**12.10.4.15 transform()**

```
void MD5::transform (
 const uint1 block[blocksize]) [private]
```

**12.10.4.16 update()** [1/2]

```
void MD5::update (
 const unsigned char * buf,
 size_type length)
```

**12.10.4.17 update()** [2/2]

```
void MD5::update (
 const char * buf,
 size_type length)
```



## 12.10.5 Friends And Related Function Documentation

### 12.10.5.1 operator<<

```
std::ostream& operator<< (
 std::ostream & out,
 MD5 md5) [friend]
```

## 12.10.6 Member Data Documentation

### 12.10.6.1 buffer

```
uint1 MD5::buffer[blocksize] [private]
```

### 12.10.6.2 count

```
uint4 MD5::count[2] [private]
```

### 12.10.6.3 digest

```
uint1 MD5::digest[16] [private]
```

### 12.10.6.4 finalized

```
bool MD5::finalized [private]
```

### 12.10.6.5 state

```
uint4 MD5::state[4] [private]
```

The documentation for this class was generated from the following files:

- systematictools/utility/[md5.hh](#)
- systematictools/utility/[md5.cc](#)

## 12.11 systools::ParamHeaderHelper Class Reference

```
#include <ParamHeaderHelper.hh>
```

Collaboration diagram for systools::ParamHeaderHelper:

### Public Types

- typedef std::vector< double > [spline\\_t](#)
- typedef std::map< [paramId\\_t](#), TSpline3 > [param\\_tspline\\_map\\_t](#)
- typedef std::vector< double > [discrete\\_variation\\_list\\_t](#)

### Public Member Functions

- [ParamHeaderHelper](#) ([param\\_header\\_map\\_t](#) headers={}, [ParamValidationAndErrorResponse](#) chkerrs=[ParamValidationAndErrorResponse](#)())  
*Constructor for parameter header meta-data helper class.*
- [ParamHeaderHelper](#) ([param\\_header\\_map\\_t](#) &&headers, [ParamValidationAndErrorResponse](#) chkerrs=[ParamValidationAndErrorResponse](#)())
- void [SetHeaders](#) ([param\\_header\\_map\\_t](#) const &headers)
- void [SetHeaders](#) ([param\\_header\\_map\\_t](#) &&headers)
- [param\\_header\\_map\\_t](#) const & [GetHeaders](#) ()
- void [SetChkErr](#) ([ParamValidationAndErrorResponse](#) const &ChkErr)
- [SystParamHeader](#) const & [GetHeader](#) ([paramId\\_t](#) i) const  
*Get the header object for parameter i.*
- bool [HaveHeader](#) ([paramId\\_t](#)) const  
*Whether parameter i is handled by this helper.*
- [SystParamHeader](#) const & [GetHeader](#) (std::string const &name) const  
*Get the header object for parameter named, name.*
- bool [HaveHeader](#) (std::string const &) const  
*Whether parameter named, name, is handled by this helper.*
- [paramId\\_t](#) [GetHeaderId](#) (std::string const &name) const  
*Get the paramId\_t for parameter named, name, if it doesn't exist, kParamUnhandled<paramId\_t> is returned.*
- [param\\_list\\_t](#) [GetParameters](#) () const  
*Get list of all handled parameter ids.*
- bool [IsThrownParam](#) ([paramId\\_t](#)) const  
*Whether the values of parameter i have been randomly thrown.*
- bool [IsResponselessParam](#) ([paramId\\_t](#)) const  
*Whether responses to parameter i are retrieved directly, or induce response in another parameter.*
- [paramId\\_t](#) [GetResponseParamId](#) ([paramId\\_t](#)) const  
*Get parameter id that variations in parameter i induce a response on.*
- bool [IsSplineParam](#) ([paramId\\_t](#)) const  
*Whether parameter i is a spline-style parameter.*
- bool [ValuesAreInNaturalUnits](#) ([paramId\\_t](#)) const  
*Whether values of parameter i are considered to be in units of sigma or in 'natural' units.*
- bool [IsWeightResponse](#) ([paramId\\_t](#)) const  
*Whether responses to variations in parameter i are characterised by an event weight or some lateral shift in observables.*
- bool [HasParameterLimits](#) ([paramId\\_t](#)) const  
*Whether parameter i has a bounded range of validity.*

- bool [HasParameterLowLimit](#) (paramId\_t) const  
*Whether parameter i has a lower bound on the range of validity.*
- bool [HasParameterUpLimit](#) (paramId\_t) const  
*Whether parameter i has an upper bound on the range of validity.*
- double [GetParameterLowLimit](#) (paramId\_t) const  
*Get the lower bound for on parameter i values.*
- double [GetParameterUpLimit](#) (paramId\_t) const  
*Get the upper bound for on parameter i values.*
- TSpline3 [GetSpline](#) (paramId\_t, spline\_t const &event\_responses={}) const  
*Get a TSpline object for a given parameter for a given event from the passed vector of responses.*
- TSpline3 [GetSpline](#) (paramId\_t, event\_unit\_response\_t const &) const  
*Get a TSpline object for a given parameter for a given event from the passed event unit response.*
- template<size\_t n>  
[PolyResponse](#)< n > [GetPolyResponse](#) (paramId\_t i, event\_unit\_response\_t const &eur) const  
*Get a PolyResponse object for a given parameter, for a given event from the passed event unit response.*
- std::vector< TSpline3 > [GetSplines](#) (paramId\_t, EventResponse const &) const  
*Get all of the splines for parameter i from the passed event responses.*
- param\_tspline\_map\_t [GetSplines](#) (param\_list\_t const &, event\_unit\_response\_t const &) const  
*Get a map of the parameter-spline responses for all parameters in passed list from the passed event unit response.*
- std::vector< param\_tspline\_map\_t > [GetSplines](#) (param\_list\_t const &, EventResponse const &) const  
*Get the splined parameter responses for each event unit in the passed event response.*
- double [GetParameterResponse](#) (paramId\_t, double, spline\_t const &event\_responses={}) const  
*Gets the splined response for parameter i, set to value v, given the passed spline information.*
- double [GetParameterResponse](#) (paramId\_t i, double v, event\_unit\_response\_t const &) const  
*Gets the splined response for parameter i, set to value v, given the passed event unit information.*
- double [GetTotalResponse](#) (param\_value\_list\_t const &, event\_unit\_response\_t const &) const  
*Gets the multiplicatively combined, splined response for all passed parameter-value pairs given the passed event unit information.*
- std::vector< double > [GetParameterResponse](#) (paramId\_t, double, EventResponse const &) const  
*Gets the splined response for parameter i, set to value v, for each event unit in the passed event response.*
- std::vector< double > [GetTotalResponse](#) (param\_value\_list\_t const &, EventResponse const &) const  
*Gets the multiplicatively combined, splined response for all passed parameter-value pairs separately for each event unit in the passed event response.*
- size\_t [GetNDiscreteVariations](#) (paramId\_t) const  
*Gets the number of variations for parameter i.*
- std::vector< size\_t > [GetNDiscreteVariations](#) (param\_list\_t const &) const  
*Gets the number of variations for each parameter in the passed parameter list.*
- discrete\_variation\_list\_t [GetDiscreteResponses](#) (paramId\_t, discrete\_variation\_list\_t const &) const  
*Gets the list of responses of parameter i from the passed mutlisim response information.*
- discrete\_variation\_list\_t [GetDiscreteResponses](#) (paramId\_t, event\_unit\_response\_t const &eur={}) const  
*Gets the list of responses of parameter i from the passed mutlisim response information.*
- double [GetDiscreteResponse](#) (paramId\_t, size\_t j, discrete\_variation\_list\_t const &event\_responses={}) const  
*Gets the response at variation j of parameter i from the passed mutlisim response information.*
- double [GetDiscreteResponse](#) (paramId\_t i, size\_t j, event\_unit\_response\_t const &) const  
*Gets the response at variation j of parameter i from the passed event unit response information.*
- double [GetDiscreteResponse](#) (param\_list\_t const &, size\_t j, event\_unit\_response\_t const &) const  
*Gets the multiplicatively combined response at variation j of each parameter in the passed parameter list from the passed event unit response information.*
- std::vector< double > [GetDiscreteResponses](#) (paramId\_t, size\_t j, EventResponse const &) const  
*Gets the response at variation j of parameter i for each event unit from the event response information.*
- std::vector< double > [GetDiscreteResponses](#) (param\_list\_t const &, size\_t j, EventResponse const &) const

*Gets the multiplicatively combined responses at variation  $j$  of the passed parameter set for each event unit from the event response information.*

- `std::vector< discrete_variation_list_t > GetAllDiscreteResponses (paramId_t, EventResponse const &) const`

*Gets the response to all variations, for all events in the passed event response information.*

- `std::vector< discrete_variation_list_t > GetAllDiscreteResponses (param_list_t const &, EventResponse const &) const`

*Gets the multiplicatively combined responses to all variations, for all passed parameters, for all events in the passed event response information.*

- `std::map< paramId_t, discrete_variation_list_t > GetDiscreteVariationParameterValues (param_list_t const &) const`

*Gets the thrown parameter values for all parameters specified in the passed parameter list.*

- `void SetCareLevel (ParamValidationAndErrorResponse::CareLevel c)`

*How carefully to check parameter usage.*

- `void SetPedantLevel (ParamValidationAndErrorResponse::PedantLevel p)`

*How to react to the result of usage checks.*

- `void SetErrorResponseLevel (ParamValidationAndErrorResponse::ErrorResponseLevel e)`

*How responses to failed checks are handled for  $fPedantry \neq kNotOnMyWatch$ .*

- `void SetAllowNegativeWeights (bool a)`

*Whether negative weights are allowed, ignored for non-weight systematics.*

- `void SetLargeWeightBoundary (double l)`

*Weights further from 0 than this will be considered 'too large' for error checking purposes.*

- `void SetSmallWeightBoundary (double s)`

*Weights closer to 0 than this will be considered 'too small' for error checking purposes.*

- `std::string GetHeaderInfo () const`

- `std::string GetEventResponseInfo (event_unit_response_t) const`

## Private Member Functions

- `TSpline3 GetSpline (paramId_t, spline_t const &event_responses, SystParamHeader const &) const`

*Used internally to skip getting a header that we have already got.*

- `TSpline3 GetSpline (paramId_t, event_unit_response_t const &, SystParamHeader const &) const`

*Used internally to skip getting a header that we have already got.*

- `discrete_variation_list_t GetDiscreteResponses (paramId_t, discrete_variation_list_t const &, SystParamHeader const &) const`

*Used internally to skip getting a header that we have already got.*

- `discrete_variation_list_t GetDiscreteResponses (paramId_t i, event_unit_response_t const &eur, SystParamHeader const &hdr) const`

*Used internally to skip getting a header that we have already got.*

- `param_value_list_t CheckParamValueList (param_value_list_t) const`

*Checks parameter-value map for parameter mis-use.*

- `param_list_t CheckParamList (param_list_t, bool ExpectSpline, bool RequireWeightResponse) const`

*Checks parameter list of parameter mis-use.*

## Private Attributes

- `param_header_map_t fHeaders`
- `ParamValidationAndErrorResponse fChkErr`
- `spline_t scratch_spline_t1`
- `spline_t scratch_spline_t2`
- `discrete_variation_list_t scratch_discrete_variation_list_t1`

## Static Private Attributes

- static [SystParamHeader](#) nullheader = [SystParamHeader\(\)](#)  
*Empty header.*

### 12.11.1 Member Typedef Documentation

#### 12.11.1.1 discrete\_variation\_list\_t

```
typedef std::vector<double> systools::ParamHeaderHelper::discrete_variation_list_t
```

#### 12.11.1.2 param\_tspline\_map\_t

```
typedef std::map<paramId_t, TSpline3> systools::ParamHeaderHelper::param_tspline_map_t
```

#### 12.11.1.3 spline\_t

```
typedef std::vector<double> systools::ParamHeaderHelper::spline_t
```

### 12.11.2 Constructor & Destructor Documentation

#### 12.11.2.1 ParamHeaderHelper() [1/2]

```
systools::ParamHeaderHelper::ParamHeaderHelper (
 param_header_map_t headers = {},
 ParamValidationAndErrorResponse chkerrs = ParamValidationAndErrorResponse())
[inline]
```

Constructor for parameter header meta-data helper class.

#### Note

a `param_header_map_t` instance can be retrieved from a parameter headers FHiCL document by [systools::BuildParameterHeaders](#), found in [utility/ParameterAndProviderConfigurationUtility.hh](#)

Headers can be set/overridden after construction by [ParamHeaderHelper::SetHeaders](#).

### 12.11.2.2 ParamHeaderHelper() [2/2]

```
systtools::ParamHeaderHelper::ParamHeaderHelper (
 param_header_map_t && headers,
 ParamValidationAndErrorResponse chkerrs = ParamValidationAndErrorResponse())
[inline]
```

## 12.11.3 Member Function Documentation

### 12.11.3.1 CheckParamList()

```
param_list_t systtools::ParamHeaderHelper::CheckParamList (
 param_list_t ilist,
 bool ExpectSpline,
 bool RequireWeightResponse) const [private]
```

Checks parameter list of parameter mis-use.

Checks for bad parameters in a parameter list and acts accordingly.

Optionally checks for splineable parameters in the list.

Optionally checks for weight-only responses, useful when returning a total weight

### 12.11.3.2 CheckParamValueList()

```
param_value_list_t systtools::ParamHeaderHelper::CheckParamValueList (
 param_value_list_t ivlist) const [private]
```

Checks parameter-value map for parameter mis-use.

Checks for bad parameters in a parameter map and acts accordingly.

### 12.11.3.3 GetAllDiscreteResponses() [1/2]

```
std::vector< ParamHeaderHelper::discrete_variation_list_t > systtools::ParamHeaderHelper::↔
GetAllDiscreteResponses (
 paramId_t i,
 EventResponse const & er) const
```

Gets the response to all variations, for all events in the passed event response information.

## 12.11.3.4 GetAllDiscreteResponses() [2/2]

```
std::vector< ParamHeaderHelper::discrete_variation_list_t > systools::ParamHeaderHelper::↵
GetAllDiscreteResponses (
 param_list_t const & ilist,
 EventResponse const & er) const
```

Gets the multiplicatively combined responses to all variations, for all passed parameters, for all events in the passed event response information.

**Note**

This shouldn't be used on non-weight parameters. For higher care levels this will be enforced.

## 12.11.3.5 GetDiscreteResponse() [1/3]

```
double systools::ParamHeaderHelper::GetDiscreteResponse (
 paramId_t i,
 size_t j,
 discrete_variation_list_t const & event_responses = {}) const
```

Gets the response at variation j of parameter i from the passed mutlisim response information.

**Note**

For parameters that produce a different response for each event, this method essentially returns event\_↵ responses[j]

At higher care levels, the passing of spline parameters will checked for.

## 12.11.3.6 GetDiscreteResponse() [2/3]

```
double systools::ParamHeaderHelper::GetDiscreteResponse (
 paramId_t i,
 size_t j,
 event_unit_response_t const & eur) const
```

Gets the response at variation j of parameter i from the passed event unit response information.

**Note**

For parameters that produce a different response for each event, this method essentially returns event\_↵ responses[j]

**12.11.3.7 GetDiscreteResponse()** [3/3]

```
double systtools::ParamHeaderHelper::GetDiscreteResponse (
 param_list_t const & ilist,
 size_t j,
 event_unit_response_t const & eur) const
```

Gets the multiplicatively combined response at variation j of each parameter in the passed parameter list from the passed event unit response information.

**12.11.3.8 GetDiscreteResponses()** [1/6]

```
ParamHeaderHelper::discrete_variation_list_t systtools::ParamHeaderHelper::GetDiscreteResponses
(
 paramId_t i,
 discrete_variation_list_t const & event_responses) const
```

Gets the list of responses of parameter i from the passed mutlisim response information.

**Note**

For parameters that produce a different response for each event, this method essentially returns the input event\_responses. It will apply validity checks based on the current values of fCare and fPedantry and possibly truncate bad responses based on fErrorResponse.

This method can be of practical use for parameters which do not effect a response that differs event by event.

**12.11.3.9 GetDiscreteResponses()** [2/6]

```
ParamHeaderHelper::discrete_variation_list_t systtools::ParamHeaderHelper::GetDiscreteResponses
(
 paramId_t i,
 event_unit_response_t const & eur = {}) const
```

Gets the list of responses of parameter i from the passed mutlisim response information.

**Note**

For parameters that produce a different response for each event, this method essentially returns the input event\_responses. It will apply validity checks based on the current values of fCare and fPedantry and possibly truncate bad responses based on fErrorResponse.

This method can be of practical use for parameters which do not effect a response that differs event by event.

At higher care levels the existance of parameter in event unit response will be checked for.



**12.11.3.10 GetDiscreteResponses()** [3/6]

```
std::vector< double > systools::ParamHeaderHelper::GetDiscreteResponses (
 paramId_t i,
 size_t j,
 EventResponse const & er) const
```

Gets the response at variation j of parameter i for each event unit from the event response information.

**12.11.3.11 GetDiscreteResponses()** [4/6]

```
std::vector< double > systools::ParamHeaderHelper::GetDiscreteResponses (
 param_list_t const & ilist,
 size_t j,
 EventResponse const & er) const
```

Gets the multiplicatively combined responses at variation j of the passed parameter set for each event unit from the event response information.

**Note**

This shouldn't be used on non-weight parameters. For higher care levels this will be enforced.

**12.11.3.12 GetDiscreteResponses()** [5/6]

```
ParamHeaderHelper::discrete_variation_list_t systools::ParamHeaderHelper::GetDiscreteResponses
(
 paramId_t i,
 discrete_variation_list_t const & event_responses,
 SystParamHeader const & hdr) const [private]
```

Used internally to skip getting a header that we have already got.

Probably reeks of premature optimization.

**12.11.3.13 GetDiscreteResponses()** [6/6]

```
ParamHeaderHelper::discrete_variation_list_t systools::ParamHeaderHelper::GetDiscreteResponses
(
 paramId_t i,
 event_unit_response_t const & eur,
 SystParamHeader const & hdr) const [private]
```

Used internally to skip getting a header that we have already got.

Probably reeks of premature optimization.

**Note**

At higher care levels checks before assuming parameter is in event unit response.

**12.11.3.14 GetDiscreteVariationParameterValues()**

```
std::map< paramId_t, ParamHeaderHelper::discrete_variation_list_t > systools::ParamHeaderHelper::GetDiscreteVariationParameterValues (
 param_list_t const & ilist) const
```

Gets the thrown parameter values for all parameters specified in the passed parameter list.

**12.11.3.15 GetEventResponseInfo()**

```
std::string systools::ParamHeaderHelper::GetEventResponseInfo (
 event_unit_response_t eur) const
```

**12.11.3.16 GetHeader()** [1/2]

```
SystParamHeader const & systools::ParamHeaderHelper::GetHeader (
 paramId_t i) const
```

Get the header object for parameter i.

**12.11.3.17 GetHeader()** [2/2]

```
SystParamHeader const & systools::ParamHeaderHelper::GetHeader (
 std::string const & name) const
```

Get the header object for parameter named, name.

**12.11.3.18 GetHeaderId()**

```
paramId_t systools::ParamHeaderHelper::GetHeaderId (
 std::string const & name) const
```

Get the paramId\_t for parameter named, name, if it doesn't exist, kParamUnhandled<paramId\_t> is returned.

**12.11.3.19 GetHeaderInfo()**

```
std::string systools::ParamHeaderHelper::GetHeaderInfo () const
```

**12.11.3.20 GetHeaders()**

```
param_header_map_t const& systools::ParamHeaderHelper::GetHeaders () [inline]
```

**12.11.3.21 GetNDiscreteVariations()** [1/2]

```
size_t systools::ParamHeaderHelper::GetNDiscreteVariations (
 paramId_t i) const
```

Gets the number of variations for parameter i.

**12.11.3.22 GetNDiscreteVariations()** [2/2]

```
std::vector< size_t > systools::ParamHeaderHelper::GetNDiscreteVariations (
 param_list_t const & paramlist) const
```

Gets the number of variations for each parameter in the passed parameter list.

**12.11.3.23 GetParameterLowLimit()**

```
double systools::ParamHeaderHelper::GetParameterLowLimit (
 paramId_t i) const
```

Get the lower bound for on parameter i values.

**Note**

For higher pedantry levels, requesting this for a non-spline parameter will constitute an error.  
For higher pedantry levels, requesting this for a spline-type parameter that isn't bounded from below will constitute an error.

**12.11.3.24 GetParameterResponse()** [1/3]

```
double systools::ParamHeaderHelper::GetParameterResponse (
 paramId_t i,
 double v,
 spline_t const & event_responses = {}) const
```

Gets the splined response for parameter i, set to value v, given the passed spline information.

For events where the response is fully characterised in the header, the event\_responses vector can be empty.

**Note**

At higher care levels, the passing of non-spline parameters will be checked for.  
This method is very inefficient for anything but the simplest tests, if you want to vary and fit event-splines, please use GetSpline{,s} and cache the splines for each event.

Uses GetSpline internally

**12.11.3.25 GetParameterResponse()** [2/3]

```
double systtools::ParamHeaderHelper::GetParameterResponse (
 paramId_t i,
 double v,
 event_unit_response_t const & eur) const
```

Gets the splined response for parameter i, set to value v, given the passed event unit information.

**Note**

This shouldn't be used on non-weight parameters. For higher care levels this will be enforced.  
 At higher care levels, the passing non-spline parameters will be checked for.  
 This method is very inefficient for anything but the simplest tests, if you want to vary and fit event-splines, please use GetSpline{,s} and cache the splines for each event.

Uses GetSpline internally

**12.11.3.26 GetParameterResponse()** [3/3]

```
std::vector< double > systtools::ParamHeaderHelper::GetParameterResponse (
 paramId_t i,
 double v,
 EventResponse const & er) const
```

Gets the splined response for parameter i, set to value v, for each event unit in the passed event response.

**Note**

At higher care levels, the passing of non-spline parameters will be checked for.  
 This method is very inefficient for anything but the simplest tests, if you want to vary and fit event-splines, please use GetSpline{,s} and cache the splines for each event.

Uses GetSpline internally

**12.11.3.27 GetParameters()**

```
param_list_t systtools::ParamHeaderHelper::GetParameters () const
```

Get list of all handled parameter Ids.

**12.11.3.28 GetParameterUpLimit()**

```
double systtools::ParamHeaderHelper::GetParameterUpLimit (
 paramId_t i) const
```

Get the upper bound for on parameter i values.

**Note**

For higher pedantry levels, requesting this for a non-spline parameter will constitute an error.  
 For higher pedantry levels, requesting this for a spline-type parameter that isn't bounded from above will constitute an error.

## 12.11.3.29 GetPolyResponse()

```
template<size_t n>
PolyResponse<n> systools::ParamHeaderHelper::GetPolyResponse (
 paramId_t i,
 event_unit_response_t const & eur) const [inline]
```

Get a [PolyResponse](#) object for a given parameter, for a given event from the passed event unit response.

**Note**

Performs very few checks.

## 12.11.3.30 GetResponseParamId()

```
paramId_t systools::ParamHeaderHelper::GetResponseParamId (
 paramId_t i) const
```

Get parameter id that variations in parameter i induce a response on.

**Note**

At higher care levels, will check if parameter i is a responseless parameter.

## 12.11.3.31 GetSpline() [1/4]

```
TSpline3 systools::ParamHeaderHelper::GetSpline (
 paramId_t i,
 spline_t const & event_responses = {}) const
```

Get a TSpline object for a given parameter for a given event from the passed vector of responses.

**Note**

At higher care levels, the passing of non-spline parameters will be checked for.

## 12.11.3.32 GetSpline() [2/4]

```
TSpline3 systools::ParamHeaderHelper::GetSpline (
 paramId_t i,
 event_unit_response_t const & eur) const
```

Get a TSpline object for a given parameter for a given event from the passed event unit response.

**Note**

At higher care levels, the passing of non-spline parameters will be checked for.

**12.11.3.33 GetSpline()** [3/4]

```
TSpline3 systtools::ParamHeaderHelper::GetSpline (
 paramId_t i,
 spline_t const & event_responses,
 SystParamHeader const & hdr) const [private]
```

Used internally to skip getting a header that we have already got.

Probably reeks of premature optimization. No TSpline3 constructor that takes const arrays...

**12.11.3.34 GetSpline()** [4/4]

```
TSpline3 systtools::ParamHeaderHelper::GetSpline (
 paramId_t i,
 event_unit_response_t const & eur,
 SystParamHeader const & hdr) const [private]
```

Used internally to skip getting a header that we have already got.

Probably reeks of premature optimization.

**Note**

At higher care levels checks before assuming parameter is in event unit response.

**12.11.3.35 GetSplines()** [1/3]

```
std::vector< TSpline3 > systtools::ParamHeaderHelper::GetSplines (
 paramId_t i,
 EventResponse const & er) const
```

Get all of the splines for parameter i from the passed event responses.

**Note**

At higher care levels, the passing of non-spline parameters will checked for.

**12.11.3.36 GetSplines()** [2/3]

```
ParamHeaderHelper::param_tspline_map_t systtools::ParamHeaderHelper::GetSplines (
 param_list_t const & ilist,
 event_unit_response_t const & eur) const
```

Get a map of the parameter-spline responses for all parameters in passed list from the passed event unit response.

**Note**

At higher care levels, the passing of non-spline parameters will checked for.

**12.11.3.37 GetSplines()** [3/3]

```
std::vector< ParamHeaderHelper::param_tspline_map_t > systools::ParamHeaderHelper::GetSplines
(
 param_list_t const & ilist,
 EventResponse const & er) const
```

Get the splined parameter responses for each event unit in the passed event response.

**Note**

At higher care levels, the passing of non-spline parameters will be checked for.

**12.11.3.38 GetTotalResponse()** [1/2]

```
double systools::ParamHeaderHelper::GetTotalResponse (
 param_value_list_t const & ivlist,
 event_unit_response_t const & eur) const
```

Gets the multiplicatively combined, splined response for all passed parameter-value pairs given the passed event unit information.

**Note**

This shouldn't be used on non-weight parameters. For higher care levels this will be enforced.

At higher care levels, the passing non-spline parameters will be checked for.

This method is very inefficient for anything but the simplest tests, if you want to vary and fit event-splines, please use GetSpline{,s} and cache the splines for each event.

Uses GetSpline internally

**12.11.3.39 GetTotalResponse()** [2/2]

```
std::vector< double > systools::ParamHeaderHelper::GetTotalResponse (
 param_value_list_t const & ivlist,
 EventResponse const & er) const
```

Gets the multiplicatively combined, splined response for all passed parameter-value pairs separately for each event unit in the passed event response.

**Note**

This shouldn't be used on non-weight parameters. For higher care levels this will be enforced.

At higher care levels, the passing of non-spline parameters will be checked for.

This method is very inefficient for anything but the simplest tests, if you want to vary and fit event-splines, please use GetSpline{,s} and cache the splines for each event.

Uses GetSpline internally

#### 12.11.3.40 HasParameterLimits()

```
bool systools::ParamHeaderHelper::HasParameterLimits (
 paramId_t i) const
```

Whether parameter i has a bounded range of validity.

##### Note

For higher pedantry levels, requesting this for a non spline parameter will constiute an error.

#### 12.11.3.41 HasParameterLowLimit()

```
bool systools::ParamHeaderHelper::HasParameterLowLimit (
 paramId_t i) const
```

Whether parameter i has a lower bound on the range of validity.

##### Note

For higher pedantry levels, requesting this for a non spline parameter will constiute an error.

#### 12.11.3.42 HasParameterUpLimit()

```
bool systools::ParamHeaderHelper::HasParameterUpLimit (
 paramId_t i) const
```

Whether parameter i has an upper bound on the range of validity.

##### Note

For higher pedantry levels, requesting this for a non spline parameter will constiute an error.

#### 12.11.3.43 HaveHeader() [1/2]

```
bool systools::ParamHeaderHelper::HaveHeader (
 paramId_t i) const
```

Whether parameter i is handled by this helper.



#### 12.11.3.44 HaveHeader() [2/2]

```
bool systools::ParamHeaderHelper::HaveHeader (
 std::string const & name) const
```

Whether parameter named, name, is handled by this helper.

#### 12.11.3.45 IsResponselessParam()

```
bool systools::ParamHeaderHelper::IsResponselessParam (
 paramId_t i) const
```

Whether responses to parameter i are retrieved directly, or induce response in another parameter.

If false, use GetResponseParamId to determine which parameter variations of i are included in.

#### 12.11.3.46 IsSplineParam()

```
bool systools::ParamHeaderHelper::IsSplineParam (
 paramId_t i) const
```

Whether parameter i is a spline-style parameter.

For spline-style parameters, the GetSpline, GetParameterResponse, and GetTotalResponse methods should be used to get parameter response information for a given value or set of parameter-value pairs.

#### 12.11.3.47 IsThrownParam()

```
bool systools::ParamHeaderHelper::IsThrownParam (
 paramId_t i) const
```

Whether the values of parameter i have been randomly thrown.

For non-spline style parameters for which this returns false, the variations are still discrete and accessed through the GetDiscreteResponse{s} interface, but it should be known by a consumer that they are not randomly distributed.

#### 12.11.3.48 IsWeightResponse()

```
bool systools::ParamHeaderHelper::IsWeightResponse (
 paramId_t i) const
```

Whether responses to variations in parameter i are characterised by an event weight or some lateral shift in observables.

#### 12.11.3.49 SetAllowNegativeWeights()

```
void systtools::ParamHeaderHelper::SetAllowNegativeWeights (
 bool a) [inline]
```

Whether negative weights are allowed, ignored for non-weight systematics.

#### 12.11.3.50 SetCareLevel()

```
void systtools::ParamHeaderHelper::SetCareLevel (
 ParamValidationAndErrorResponse::CareLevel c) [inline]
```

How carefully to check parameter usage.

#### Note

This defines what usage checks should be run, as opposed to fPedanty, which defines the reaction to failed checks.

- kTortoise: Check that parameters exist, check spline and variation number ranges are within bounds, for weight systematics check whether weights are too large or too small or negative, check that when getting a multiplicatively combined response, all of the responses are weight systematics.
- kFrog: Check that parameters exist and are used correctly (spline type).
- kHare: Assume everything is correct.

#### 12.11.3.51 SetChkErr()

```
void systtools::ParamHeaderHelper::SetChkErr (
 ParamValidationAndErrorResponse const & ChkErr) [inline]
```

#### 12.11.3.52 SetErrorResponseLevel()

```
void systtools::ParamHeaderHelper::SetErrorResponseLevel (
 ParamValidationAndErrorResponse::ErrorResponseLevel e) [inline]
```

How responses to failed checks are handled for fPedantry != kNotOnMyWatch.

- kZeroResponse: Return 0 response
- kUnityWeight: Return 0 response for non-weight systematics and a response weight of 1 for weight systematics.
- kBoundaryResponse: If the error is out of bounds, return the response of the closest boundary.

**12.11.3.53 SetHeaders()** [1/2]

```
void systools::ParamHeaderHelper::SetHeaders (
 param_header_map_t const & headers) [inline]
```

**12.11.3.54 SetHeaders()** [2/2]

```
void systools::ParamHeaderHelper::SetHeaders (
 param_header_map_t && headers) [inline]
```

**12.11.3.55 SetLargeWeightBoundary()**

```
void systools::ParamHeaderHelper::SetLargeWeightBoundary (
 double l) [inline]
```

Weights further from 0 than this will be considered 'too large' for error checking purposes.

**12.11.3.56 SetPedantLevel()**

```
void systools::ParamHeaderHelper::SetPedantLevel (
 ParamValidationAndErrorResponse::PedantLevel p) [inline]
```

How to react to the result of usage checks.

**Note**

This defines how to react to failed checks, as opposed to fCare, which defines which checks to run.

- kAnythingGoes: For bad parameter usage or response, return the default response.
- kMeh: For bad parameter usage or response, post a warning, and return the default response.
- kNotOnMyWatch: For bad parameter usage or response, post an error and throw an exception detailing the bad usage or response.

**12.11.3.57 SetSmallWeightBoundary()**

```
void systools::ParamHeaderHelper::SetSmallWeightBoundary (
 double s) [inline]
```

Weights closer to 0 than this will be considered 'too small' for error checking purposes.

### 12.11.3.58 ValuesAreInNaturalUnits()

```
bool systools::ParamHeaderHelper::ValuesAreInNaturalUnits (
 paramId_t i) const
```

Whether values of parameter i are considered to be in units of sigma or in 'natural' units.

## 12.11.4 Member Data Documentation

### 12.11.4.1 fChkErr

```
ParamValidationAndErrorResponse systools::ParamHeaderHelper::fChkErr [private]
```

### 12.11.4.2 fHeaders

```
param_header_map_t systools::ParamHeaderHelper::fHeaders [private]
```

### 12.11.4.3 nullheader

```
SystParamHeader systools::ParamHeaderHelper::nullheader = SystParamHeader() [static], [private]
```

Empty header.

Used when a reference to a unhandled header is expected but fPedantry is set low.

### 12.11.4.4 scratch\_discrete\_variation\_list\_t1

```
discrete_variation_list_t systools::ParamHeaderHelper::scratch_discrete_variation_list_t1
[mutable], [private]
```

### 12.11.4.5 scratch\_spline\_t1

```
spline_t systools::ParamHeaderHelper::scratch_spline_t1 [mutable], [private]
```

#### 12.11.4.6 scratch\_spline\_t2

```
spline_t systools::ParamHeaderHelper::scratch_spline_t2 [mutable], [private]
```

The documentation for this class was generated from the following files:

- systematictools/interpreters/[ParamHeaderHelper.hh](#)
- systematictools/interpreters/[ParamHeaderHelper.cc](#)

## 12.12 systools::ParamHeaderProviderName Struct Reference

Struct for holding [ISystProviderTool](#) unique name–handled parameter header pairs.

```
#include <types.hh>
```

Collaboration diagram for systools::ParamHeaderProviderName:

### Public Attributes

- std::string [ProviderFQName](#)
- [SystParamHeader](#) Header

### 12.12.1 Detailed Description

Struct for holding [ISystProviderTool](#) unique name–handled parameter header pairs.

Gives semantic meaning to what might otherwise be implemented as a std::pair.

### 12.12.2 Member Data Documentation

#### 12.12.2.1 Header

```
SystParamHeader systools::ParamHeaderProviderName::Header
```

#### 12.12.2.2 ProviderFQName

```
std::string systools::ParamHeaderProviderName::ProviderFQName
```

The documentation for this struct was generated from the following file:

- systematictools/interface/[types.hh](#)

## 12.13 systools::PrecalculatedResponseReader< Order >::ParamPolyResponses Struct Reference

```
#include <PrecalculatedResponseReader.hh>
```

Collaboration diagram for systools::PrecalculatedResponseReader< Order >::ParamPolyResponses:

### Public Attributes

- [systools::paramId\\_t pid](#)
- [systools::PolyResponse< Order > resp](#)

### 12.13.1 Member Data Documentation

#### 12.13.1.1 pid

```
template<size_t Order>
systools::paramId_t systools::PrecalculatedResponseReader< Order >::ParamPolyResponses::pid
```

#### 12.13.1.2 resp

```
template<size_t Order>
systools::PolyResponse<Order> systools::PrecalculatedResponseReader< Order >::ParamPolyResponses::resp
```

The documentation for this struct was generated from the following file:

- [systematictools/interpreters/PrecalculatedResponseReader.hh](#)

## 12.14 systools::ParamResponses Struct Reference

```
#include <EventResponse_product.hh>
```

### Public Attributes

- [paramId\\_t pid](#)
- [std::vector< double > responses](#)

### 12.14.1 Detailed Description

Struct for holding parameter–response associations

#### Note

Kept distinct from [ParamThrows](#) to preserve the important semantic difference of the two, even if the data structures are identical.

Gives semantic meaning to what might otherwise be implemented as a `std::pair`.

### 12.14.2 Member Data Documentation

#### 12.14.2.1 pid

`paramId_t` `systools::ParamResponses::pid`

#### 12.14.2.2 responses

`std::vector<double>` `systools::ParamResponses::responses`

The documentation for this struct was generated from the following file:

- `systematictools/interface/EventResponse_product.hh`

## 12.15 systools::ParamThrows Struct Reference

```
#include <types.hh>
```

### Public Attributes

- `paramId_t` `pid`
- `std::vector< double >` `thrown_vals`

### 12.15.1 Detailed Description

Struct for holding parameter–thrown value associations

Gives semantic meaning to what might otherwise be implemented as a `std::pair`.

## 12.15.2 Member Data Documentation

### 12.15.2.1 pid

`paramId_t` `systtools::ParamThrows::pid`

### 12.15.2.2 thrown\_vals

`std::vector<double>` `systtools::ParamThrows::thrown_vals`

The documentation for this struct was generated from the following file:

- [systematictools/interface/types.hh](#)

## 12.16 ParamValidationAndErrorResponse Struct Reference

```
#include <ParamValidationAndErrorResponse.hh>
```

### Public Types

- enum [CareLevel](#) { [kTortoise](#) = -1, [kFrog](#) = 0, [kHare](#) = 1 }
- enum [PedantLevel](#) { [kNotOnMyWatch](#) = -1, [kMeh](#) = 0, [kAnythingGoes](#) = 1 }
- enum [ErrorResponseLevel](#) { [kZeroResponse](#) = 0, [kUnityWeight](#), [kBoundaryResponse](#) }

### Public Member Functions

- [ParamValidationAndErrorResponse](#) ()
  - void [SetCareLevel](#) ([CareLevel](#) c)
  - void [SetPedantLevel](#) ([PedantLevel](#) p)
  - void [SetErrorResponseLevel](#) ([ErrorResponseLevel](#) e)
  - void [SetAllowNegativeWeights](#) (bool a)
  - void [SetLargeWeightBoundary](#) (double l)
  - void [SetSmallWeightBoundary](#) (double s)
  - double [CheckResponse](#) (double, [systtools::SystParamHeader](#) const &, `size_t idx=std::numeric_limits<size_t>::max()`) const
- Checks a response for validity.*



## Public Attributes

- [CareLevel fCare](#)  
*How carefully to check parameter usage.*
- [PedantLevel fPedantry](#)  
*How to react to the result of usage checks.*
- [ErrorResponseLevel fErrorResponse](#)  
*How responses to failed checks are handled for fPedantry != kNotOnMyWatch.*
- bool [fAllowNegativeWeights](#)  
*Whether negative weights are allowed, ignored for non-weight systematics.*
- double [fLargeWeight](#)  
*Weights further from 0 than this will be considered 'too large' for error checking purposes.*
- double [fSmallWeight](#)  
*Weights closer to 0 than this will be considered 'too small' for error checking purposes.*

## 12.16.1 Member Enumeration Documentation

### 12.16.1.1 CareLevel

```
enum ParamValidationAndErrorResponse::CareLevel
```

#### Enumerator

|           |  |
|-----------|--|
| kTortoise |  |
| kFrog     |  |
| kHare     |  |

### 12.16.1.2 ErrorResponseLevel

```
enum ParamValidationAndErrorResponse::ErrorResponseLevel
```

#### Enumerator

|                   |  |
|-------------------|--|
| kZeroResponse     |  |
| kUnityWeight      |  |
| kBoundaryResponse |  |

### 12.16.1.3 PedantLevel

```
enum ParamValidationAndErrorResponse::PedantLevel
```

## Enumerator

|               |  |
|---------------|--|
| kNotOnMyWatch |  |
| kMeh          |  |
| kAnythingGoes |  |

## 12.16.2 Constructor & Destructor Documentation

### 12.16.2.1 ParamValidationAndErrorResponse()

```
ParamValidationAndErrorResponse::ParamValidationAndErrorResponse () [inline]
```

## 12.16.3 Member Function Documentation

### 12.16.3.1 CheckResponse()

```
double ParamValidationAndErrorResponse::CheckResponse (
 double r,
 systools::SystParamHeader const & hdr,
 size_t idx = std::numeric_limits<size_t>::max()) const
```

Checks a response for validity.

Checks with the current settings of fSmallWeight, fLargeWeight, fAllowNegativeWeights, fCare and responds according to fPedanty.

#### Note

If and index is passed, then the response is determined to be from a spline or a multisim parameter, and any error messages tailored.

### 12.16.3.2 SetAllowNegativeWeights()

```
void ParamValidationAndErrorResponse::SetAllowNegativeWeights (
 bool a) [inline]
```

### 12.16.3.3 SetCareLevel()

```
void ParamValidationAndErrorResponse::SetCareLevel (
 CareLevel c) [inline]
```

### 12.16.3.4 SetErrorResponseLevel()

```
void ParamValidationAndErrorResponse::SetErrorResponseLevel (
 ErrorResponseLevel e) [inline]
```

### 12.16.3.5 SetLargeWeightBoundary()

```
void ParamValidationAndErrorResponse::SetLargeWeightBoundary (
 double l) [inline]
```

### 12.16.3.6 SetPedantLevel()

```
void ParamValidationAndErrorResponse::SetPedantLevel (
 PedantLevel p) [inline]
```

### 12.16.3.7 SetSmallWeightBoundary()

```
void ParamValidationAndErrorResponse::SetSmallWeightBoundary (
 double s) [inline]
```

## 12.16.4 Member Data Documentation

### 12.16.4.1 fAllowNegativeWeights

```
bool ParamValidationAndErrorResponse::fAllowNegativeWeights
```

Whether negative weights are allowed, ignored for non-weight systematics.

#### 12.16.4.2 fCare

`CareLevel` `ParamValidationAndErrorResponse::fCare`

How carefully to check parameter usage.

##### Note

This defines what usage checks should be run, as opposed to `fPedanty`, which defines the reaction to failed checks.

- `kTortoise`: Check that parameters exist, check spline and number of throw ranges are within bounds, for weight systematics check whether weights are too large or too small or negative, check that when getting a multiplicatively combined response, all of the responses are weight systematics.
- `kFrog`: Check that parameters exist and are used correctly (spline vs. thrown).
- `kHare`: Assume everything is correct.

#### 12.16.4.3 fErrorResponse

`ErrorResponseLevel` `ParamValidationAndErrorResponse::fErrorResponse`

How responses to failed checks are handled for `fPedantry` != `kNotOnMyWatch`.

- `kZeroResponse`: Return 0 response
- `kUnityWeight`: Return 0 response for non-weight systematics and a response weight of 1 for weight systematics.
- `kBoundaryResponse`: If the error is out of bounds, return the response of the closest boundary.

#### 12.16.4.4 fLargeWeight

`double` `ParamValidationAndErrorResponse::fLargeWeight`

Weights further from 0 than this will be considered 'too large' for error checking purposes.

#### 12.16.4.5 fPedantry

`PedantLevel` `ParamValidationAndErrorResponse::fPedantry`

How to react to the result of usage checks.

##### Note

This defines how to react to failed checks, as opposed to `fCare`, which defines which checks to run.

- `kAnythingGoes`: For bad parameter usage or response, return the default response.
- `kMeh`: For bad parameter usage or response, post a warning, and return the default response.
- `kNotOnMyWatch`: For bad parameter usage or response, post an error and throw an exception detailing the bad usage or response.

#### 12.16.4.6 fSmallWeight

`double` `ParamValidationAndErrorResponse::fSmallWeight`

Weights closer to 0 than this will be considered 'too small' for error checking purposes.

The documentation for this struct was generated from the following files:

- `systematicstools/interpreters/ParamValidationAndErrorResponse.hh`
- `systematicstools/interpreters/ParamValidationAndErrorResponse.cc`

## 12.17 systools::ParamValue Struct Reference

```
#include <types.hh>
```

### Public Attributes

- `paramId_t` `pid`
- `double` `val`

#### 12.17.1 Detailed Description

Struct for holding parameter–value associations

Gives semantic meaning to what might otherwise be implemented as a `std::pair`.

#### 12.17.2 Member Data Documentation

### 12.17.2.1 pid

```
paramId_t systtools::ParamValue::pid
```

### 12.17.2.2 val

```
double systtools::ParamValue::val
```

The documentation for this struct was generated from the following file:

- [systematictools/interface/types.hh](#)

## 12.18 systtools::PolyResponse< n > Struct Template Reference

```
#include <PolyResponse.hh>
```

Inheritance diagram for systtools::PolyResponse< n >:

Collaboration diagram for systtools::PolyResponse< n >:

### Public Member Functions

- [PolyResponse](#) (std::vector< double > const &xvals, std::vector< double > const &yvals)
- [PolyResponse](#) (std::array< double, n+1 > const &coeffs)
- [PolyResponse](#) (double const \*coeffs)
- double [eval](#) (double x) const

### 12.18.1 Constructor & Destructor Documentation

#### 12.18.1.1 PolyResponse() [1/3]

```
template<size_t n>
systtools::PolyResponse< n >::PolyResponse (
 std::vector< double > const & xvals,
 std::vector< double > const & yvals) [inline]
```

## 12.18.1.2 PolyResponse() [2/3]

```
template<size_t n>
systools::PolyResponse< n >::PolyResponse (
 std::array< double, n+1 > const & coeffs) [inline]
```

## 12.18.1.3 PolyResponse() [3/3]

```
template<size_t n>
systools::PolyResponse< n >::PolyResponse (
 double const * coeffs) [inline]
```

## 12.18.2 Member Function Documentation

## 12.18.2.1 eval()

```
template<size_t n>
double systools::PolyResponse< n >::eval (
 double x) const [inline]
```

The documentation for this struct was generated from the following file:

- [systematicstools/interpreters/PolyResponse.hh](#)

## 12.19 systools::PrecalculatedResponseReader&lt; Order &gt; Class Template Reference

```
#include <PrecalculatedResponseReader.hh>
```

## Classes

- struct [ParamPolyResponses](#)

## Public Member Functions

- [NEW\\_SYSTTOOLS\\_EXCEPT](#) (in\_wrong\_mode)
- [NEW\\_SYSTTOOLS\\_EXCEPT](#) (entry\_overflow)
- [NEW\\_SYSTTOOLS\\_EXCEPT](#) (missing\_TBranches)
- [NEW\\_SYSTTOOLS\\_EXCEPT](#) (too\_many\_headers)
- [PrecalculatedResponseReader](#) ()
- [PrecalculatedResponseReader](#) (std::string const &file\_name, std::string const &tree\_name, size\_t NHeaders)  
*Constructor for instantiating a [PrecalculatedResponseReader](#) in read mode.*
- size\_t [GetEntries](#) ()  
*Gets the number of entries in an input tree when in read mode.*
- std::vector< [ParamPolyResponses](#) > [GetEventResponse](#) (size\_t entry)  
*Gets the parameterized, precalculated event responses for all relevant parameters for event number entry.*
- void [AddEventResponses](#) (event\_unit\_response\_t eur)  
*Converts discrete, splineable event responses to parameterized response functions and fills them to the tree.*

## Static Public Member Functions

- static `std::unique_ptr< PrecalculatedResponseReader< Order > > MakeTreeWriter (param_header_map_t headers, TTree *tree)`

*Instantiator for a [PrecalculatedResponseReader](#) in write mode.*

## Private Member Functions

- void [AllocateVectors](#) (size\_t NHeaders)
- void [SetBranchAddresses](#) (TTree \*tree)

## Private Attributes

- TFile \* [file](#)
- TTree \* [tree](#)
- [param\\_header\\_map\\_t](#) [fHeaders](#)
- [Int\\_t](#) [NIds](#)
- `std::vector< Int\_t >` [ids](#)
- `std::vector< Double\_t >` [coeffs\\_1D](#)

## Static Private Attributes

- static const size\_t [NCoeffs](#) = (Order + 1)

## 12.19.1 Constructor & Destructor Documentation

### 12.19.1.1 [PrecalculatedResponseReader](#)() [1/2]

```
template<size_t Order>
systools::PrecalculatedResponseReader< Order >::PrecalculatedResponseReader () [inline]
```

### 12.19.1.2 [PrecalculatedResponseReader](#)() [2/2]

```
template<size_t Order>
systools::PrecalculatedResponseReader< Order >::PrecalculatedResponseReader (
 std::string const & file_name,
 std::string const & tree_name,
 size_t NHeaders) [inline]
```

Constructor for instantiating a [PrecalculatedResponseReader](#) in read mode.



## 12.19.2 Member Function Documentation

### 12.19.2.1 AddEventResponses()

```
template<size_t Order>
void systools::PrecalculatedResponseReader< Order >::AddEventResponses (
 event_unit_response_t eur) [inline]
```

Converts discrete, splineable event responses to parameterized response functions and fills them to the tree.

### 12.19.2.2 AllocateVectors()

```
template<size_t Order>
void systools::PrecalculatedResponseReader< Order >::AllocateVectors (
 size_t NHeaders) [inline], [private]
```

### 12.19.2.3 GetEntries()

```
template<size_t Order>
size_t systools::PrecalculatedResponseReader< Order >::GetEntries () [inline]
```

Gets the number of entries in an input tree when in read mode.

### 12.19.2.4 GetEventResponse()

```
template<size_t Order>
std::vector<ParamPolyResponses> systools::PrecalculatedResponseReader< Order >::GetEventResponse (
 size_t entry) [inline]
```

Gets the parameterized, precalculated event responses for all relevant parameters for event number entry.

**12.19.2.5 MakeTreeWriter()**

```
template<size_t Order>
static std::unique_ptr<PrecalculatedResponseReader<Order> > systools::PrecalculatedResponseReader< Order >::MakeTreeWriter (
 param_header_map_t headers,
 TTree * tree) [inline], [static]
```

Instantiator for a [PrecalculatedResponseReader](#) in write mode.

**Note**

The tree ownership is not passed. The caller is responsible for proper storage and writing of the TTree.

**12.19.2.6 NEW\_SYSTTOOLS\_EXCEPT()** [1/4]

```
template<size_t Order>
systools::PrecalculatedResponseReader< Order >::NEW_SYSTTOOLS_EXCEPT (
 in_wrong_mode)
```

**12.19.2.7 NEW\_SYSTTOOLS\_EXCEPT()** [2/4]

```
template<size_t Order>
systools::PrecalculatedResponseReader< Order >::NEW_SYSTTOOLS_EXCEPT (
 entry_overflow)
```

**12.19.2.8 NEW\_SYSTTOOLS\_EXCEPT()** [3/4]

```
template<size_t Order>
systools::PrecalculatedResponseReader< Order >::NEW_SYSTTOOLS_EXCEPT (
 missing_TBranches)
```

**12.19.2.9 NEW\_SYSTTOOLS\_EXCEPT()** [4/4]

```
template<size_t Order>
systools::PrecalculatedResponseReader< Order >::NEW_SYSTTOOLS_EXCEPT (
 too_many_headers)
```

#### 12.19.2.10 SetBranchAddresses()

```
template<size_t Order>
void systools::PrecalculatedResponseReader< Order >::SetBranchAddresses (
 TTree * tree) [inline], [private]
```

### 12.19.3 Member Data Documentation

#### 12.19.3.1 coeffs\_1D

```
template<size_t Order>
std::vector<Double_t> systools::PrecalculatedResponseReader< Order >::coeffs_1D [private]
```

Tree variable to hold responses.

#### Note

This is a 1D vector that is passed to the TTree as a 2D object, array stacking follows C standard for stack-allocated two dimensional arrays.

#### 12.19.3.2 fHeaders

```
template<size_t Order>
param_header_map_t systools::PrecalculatedResponseReader< Order >::fHeaders [private]
```

#### 12.19.3.3 file

```
template<size_t Order>
TFile* systools::PrecalculatedResponseReader< Order >::file [private]
```

#### 12.19.3.4 ids

```
template<size_t Order>
std::vector<Int_t> systools::PrecalculatedResponseReader< Order >::ids [private]
```

### 12.19.3.5 NCoeffs

```
template<size_t Order>
const size_t systtools::PrecalculatedResponseReader< Order >::NCoeffs = (Order + 1) [static],
[private]
```

### 12.19.3.6 NIds

```
template<size_t Order>
Int_t systtools::PrecalculatedResponseReader< Order >::NIds [private]
```

### 12.19.3.7 tree

```
template<size_t Order>
TTree* systtools::PrecalculatedResponseReader< Order >::tree [private]
```

The documentation for this class was generated from the following file:

- systematictools/interpreters/[PrecalculatedResponseReader.hh](#)

## 12.20 systtools::systematictools\_except Struct Reference

```
#include <exceptions.hh>
```

Inheritance diagram for systtools::systematictools\_except:

Collaboration diagram for systtools::systematictools\_except:

### Public Member Functions

- [systematictools\\_except](#) ()
- [systematictools\\_except](#) ([systematictools\\_except](#) const &other)
- const char \* [what](#) () const noexcept
- template<typename T >  
  [systematictools\\_except](#) & [operator<<](#) (T const &obj)

### Public Attributes

- std::stringstream [msgstrm](#)
- std::string [msg](#)

## 12.20.1 Constructor & Destructor Documentation

### 12.20.1.1 systematictools\_except() [1/2]

```
systools::systematictools_except::systematictools_except () [inline]
```

### 12.20.1.2 systematictools\_except() [2/2]

```
systools::systematictools_except::systematictools_except (
 systematictools_except const & other) [inline]
```

## 12.20.2 Member Function Documentation

### 12.20.2.1 operator<<()

```
template<typename T >
systematictools_except& systools::systematictools_except::operator<< (
 T const & obj) [inline]
```

### 12.20.2.2 what()

```
const char* systools::systematictools_except::what () const [inline], [noexcept]
```

## 12.20.3 Member Data Documentation

### 12.20.3.1 msg

```
std::string systools::systematictools_except::msg
```

### 12.20.3.2 msgstrm

`std::stringstream systools::systematicstools_except::msgstrm`

The documentation for this struct was generated from the following file:

- [systematicstools/utility/exceptions.hh](#)

## 12.21 systools::SystParamHeader Struct Reference

```
#include <SystParamHeader.hh>
```

### Public Member Functions

- [SystParamHeader](#) ()

### Public Attributes

- `std::string` [prettyName](#)  
*Human readable systematic parameter name.*
- `paramId_t` [systParamId](#)  
*Unique identifier for this systematic parameter.*
- `bool` [isWeightSystematicVariation](#)  
*Whether this systematic corresponds to a weight or property shift.*
- `bool` [unitsAreNatural](#)  
*Whether the quantities stored in `paramVariations` and `centralParamValue` are in 'natural' units or units of 'one sigma' uncertainty.*
- `bool` [differsEventByEvent](#)  
*Whether the the response of this parameter is fully described by this meta-data.*
- `double` [centralParamValue](#)  
*The parameter value to be considered as the central value when evaluating variations of this parameter.*
- `bool` [isCorrection](#)  
*Whether to only expect a single response that should always be applied by consumers.*
- `std::array< double, 2 >` [oneSigmaShifts](#)  
*The 'one sigma' shifts of this parameter, if present, always defined in natural units.*
- `std::array< double, 2 >` [paramValidityRange](#)  
*The range of valid parameter values.*
- `bool` [isSplineable](#)  
*Whether the `paramVariations` were chosen to facilitate a downstream consumer to interpolate between the calculated responses.*
- `bool` [isRandomlyThrown](#)  
*Whether the non-splineable variations have been hand-picked to randomly distributed according to some prior (like gaussian).*
- `std::vector< double >` [paramVariations](#)  
*The shifted values that were calculated for this parameter.*
- `bool` [isResponselessParam](#)  
*Whether variations of this parameter produce responses via this header.*
- `paramId_t` [responseParamId](#)  
*The parameter Id of where responses to parameters with `isResponselessParam == true` can be found.*
- `std::vector< double >` [responses](#)  
*The parameter responses for 'parameter-level' systematics.*
- `std::vector< std::string >` [opts](#)  
*Arbitrary string options stored in the metadata for further [ISystProviderTool](#) configuration.*

### 12.21.1 Detailed Description

Systematic parameter metadata class

Instances are used to inform systematic response consumers how to interpret responses. A number of the most commonly used features are explicitly exposed as data members, but extensibility is provided by the `opts` data member which can hold an arbitrary vector of strings.

This class is currently only serialized to and from FHiCL and as such, adding new/removing/altering members will break usage, but it is fairly easy to fix in pre-generated parameter sets.

#### Note

Changes to this class *must* be reflected in `systematictools/interpreters/load_parameter_headers.xx` and `systematictools/utility/build_parameter_set_from_header.xx`

Usually analyzers/users will interact with instances through `systematictools/interpreters/ParamHeaderHelper.xx`

### 12.21.2 Constructor & Destructor Documentation

#### 12.21.2.1 SystParamHeader()

```
systools::SystParamHeader::SystParamHeader () [inline]
```

### 12.21.3 Member Data Documentation

#### 12.21.3.1 centralParamValue

```
double systools::SystParamHeader::centralParamValue
```

The parameter value to be considered as the central value when evaluating variations of this parameter.

#### Note

This may not be the value generated with in the case of `isCorrection == true` or when the CV tune changes post-MC production.

Respects `unitsAreNatural` value.

### 12.21.3.2 differsEventByEvent

```
bool systtools::SystParamHeader::differsEventByEvent
```

Whether the the response of this parameter is fully described by this meta-data.

Equivalent to `bool (Responses.size())`;

This is used for variations that do not depend on the event properties of events that variations of this parameter effects, such as normalization weights for classes of events.

### 12.21.3.3 isCorrection

```
bool systtools::SystParamHeader::isCorrection
```

Whether to only expect a single response that should always be applied by consumers.

Uses `centralParamValue` to generate a single response, respects `differsEventByEvent`.

### 12.21.3.4 isRandomlyThrown

```
bool systtools::SystParamHeader::isRandomlyThrown
```

Whether the non-splineable variations have been hand-picked to randomly distributed according to some prior (like gaussian).

### 12.21.3.5 isResponselessParam

```
bool systtools::SystParamHeader::isResponselessParam
```

Whether variations of this parameter produce responses via this header.

This is used for multi-dimensional responses, e.g.  $R(p_1, p_2)$ , where  $R(p_1, \text{nominal}_2) * R(\text{nominal}_1, p_2) \neq R(p_1, p_2)$ . In this instance, two parameter headers would be used, one describing variations in  $p_1$  and one in  $p_2$ . All of the response to variations in both will be included on  $p_1$

#### Note

`responseParamId` holds the parameter Id that contains  $R(p_1, p_2, \dots)$ .

### 12.21.3.6 isSplineable

```
bool systtools::SystParamHeader::isSplineable
```

Whether the `paramVariations` were chosen to facilitate a downstream consumer to interpolate between the calculated responses.

When `isSplineable == false`, this parameter has likely been run in 'multi-universe' mode.



### 12.21.3.7 isWeightSystematicVariation

```
bool systools::SystParamHeader::isWeightSystematicVariation
```

Whether this systematic corresponds to a weight or property shift.

#### Note

Non-weight systematics will always need custom code on the part of a downstream consumer.

### 12.21.3.8 oneSigmaShifts

```
std::array<double, 2> systools::SystParamHeader::oneSigmaShifts
```

The 'one sigma' shifts of this parameter, if present, always defined in natural units.

Can be used by a downstream consumer to convert centralParamValue and paramVariations to and from natural units.

### 12.21.3.9 opts

```
std::vector<std::string> systools::SystParamHeader::opts
```

Arbitrary string options stored in the metadata for further [ISystProviderTool](#) configuration.

### 12.21.3.10 paramValidityRange

```
std::array<double, 2> systools::SystParamHeader::paramValidityRange
```

The range of valid parameter values.

If either end of the range is set to kDefaultDouble, that 'side' is unbounded.

Respects unitsAreNatural

### 12.21.3.11 paramVariations

```
std::vector<double> systools::SystParamHeader::paramVariations
```

The shifted values that were calculated for this parameter.

Contains the parameter values (either in sigma-shift units or natural units, see [oneSigmaShifts](#)) that were used to determine responses. The responses can either be event-level or parameter-level, parameter-level responses are stored in [responses](#).

### 12.21.3.12 prettyName

```
std::string systtools::SystParamHeader::prettyName
```

Human readable systematic parameter name.

### 12.21.3.13 responseParamId

```
paramId_t systtools::SystParamHeader::responseParamId
```

The parameter Id of where responses to parameters with `isResponselessParam == true` can be found.

### 12.21.3.14 responses

```
std::vector<double> systtools::SystParamHeader::responses
```

The parameter responses for 'parameter-level' systematics.

Empty for event-by-event parameters, contains universe or spline knot responses for dials that affect all events in the same way.

These will most often be used for overall event-class re-normalisations, which do not need to be stored event-by-event.

### 12.21.3.15 systParamId

```
paramId_t systtools::SystParamHeader::systParamId
```

Unique identifier for this systematic parameter.

Used to `key` the per-event systematic response data product.

#### Note

Not guaranteed to persist between different configurations. i.e. `systParamId == 0` might be used for some physics model parameter in one data product and a calibration parameter in another.

### 12.21.3.16 unitsAreNatural

```
bool systtools::SystParamHeader::unitsAreNatural
```

Whether the quantities stored in `paramVariations` and `centralParamValue` are in 'natural' units or units of 'one sigma' uncertainty.

The documentation for this struct was generated from the following file:

- [systematictools/interface/SystParamHeader.hh](#)

## 12.22 SystToolsEventResponse Class Reference

Inheritance diagram for SystToolsEventResponse:

Collaboration diagram for SystToolsEventResponse:

### Public Member Functions

- [SystToolsEventResponse](#) (fhicl::ParameterSet const &p)
- [SystToolsEventResponse](#) ([SystToolsEventResponse](#) const &)=delete
- [SystToolsEventResponse](#) ([SystToolsEventResponse](#) &&)=delete
- [SystToolsEventResponse](#) & operator= ([SystToolsEventResponse](#) const &)=delete
- [SystToolsEventResponse](#) & operator= ([SystToolsEventResponse](#) &&)=delete
- void [produce](#) (art::Event &e) override

### Private Member Functions

- [NEW\\_SYSTTOOLS\\_EXCEPT](#) (no\_systprovider\_key)
- [NEW\\_SYSTTOOLS\\_EXCEPT](#) (nullptr\_event\_response)

### Private Attributes

- [systtools::provider\\_list\\_t](#) [syst\\_providers](#)
- std::string [sp\\_config\\_hash](#)

## 12.22.1 Constructor & Destructor Documentation

### 12.22.1.1 SystToolsEventResponse() [1/3]

```
SystToolsEventResponse::SystToolsEventResponse (
 fhicl::ParameterSet const & p) [explicit]
```

### 12.22.1.2 SystToolsEventResponse() [2/3]

```
SystToolsEventResponse::SystToolsEventResponse (
 SystToolsEventResponse const &) [delete]
```

### 12.22.1.3 SystToolsEventResponse() [3/3]

```
SystToolsEventResponse::SystToolsEventResponse (
 SystToolsEventResponse &&) [delete]
```

## 12.22.2 Member Function Documentation

### 12.22.2.1 NEW\_SYSTTOOLS\_EXCEPT() [1/2]

```
SystToolsEventResponse::NEW_SYSTTOOLS_EXCEPT (
 no_systprovider_key) [private]
```

### 12.22.2.2 NEW\_SYSTTOOLS\_EXCEPT() [2/2]

```
SystToolsEventResponse::NEW_SYSTTOOLS_EXCEPT (
 nullptr_event_response) [private]
```

### 12.22.2.3 operator=() [1/2]

```
SystToolsEventResponse& SystToolsEventResponse::operator= (
 SystToolsEventResponse const &) [delete]
```

### 12.22.2.4 operator=() [2/2]

```
SystToolsEventResponse& SystToolsEventResponse::operator= (
 SystToolsEventResponse &&) [delete]
```

### 12.22.2.5 produce()

```
void SystToolsEventResponse::produce (
 art::Event & e) [override]
```

## 12.22.3 Member Data Documentation

### 12.22.3.1 sp\_config\_hash

```
std::string SystToolsEventResponse::sp_config_hash [private]
```

## 12.22.3.2 syst\_providers

```
systtools::provider_list_t SystToolsEventResponse::syst_providers [private]
```

The documentation for this class was generated from the following file:

- systematicstools/module/[SystToolsEventResponse\\_module.cc](#)

## 12.23 SystToolsEventResponseTree Class Reference

### Public Member Functions

- [SystToolsEventResponseTree](#) ()
- void [SetTree](#) (TTree \*t)
- void [Fill](#) ()
- void [SetEvent](#) (ULong\_t ev)
- void [SetThrow](#) (ULong\_t t)
- void [SetParamResponse](#) (paramId\_t i, double v, double r)
- void [SetTotalWeight](#) (double w)
- void [MakeBranches](#) (param\_header\_map\_t const &param\_map={}, bool isThrows=false)

### Private Attributes

- TTree \* [tree](#)
- ULong\_t [event](#)
- ULong\_t [t\\_it](#)
- std::map< [paramId\\_t](#), double > [param\\_values](#)
- std::map< [paramId\\_t](#), double > [event\\_responses](#)
- double [total\\_weight](#)

### 12.23.1 Constructor & Destructor Documentation

#### 12.23.1.1 SystToolsEventResponseTree()

```
SystToolsEventResponseTree::SystToolsEventResponseTree () [inline]
```

### 12.23.2 Member Function Documentation

#### 12.23.2.1 Fill()

```
void SystToolsEventResponseTree::Fill () [inline]
```

#### 12.23.2.2 MakeBranches()

```
void SystToolsEventResponseTree::MakeBranches (
 param_header_map_t const & param_map = {},
 bool isThrows = false) [inline]
```

#### 12.23.2.3 SetEvent()

```
void SystToolsEventResponseTree::SetEvent (
 ULong_t ev) [inline]
```

#### 12.23.2.4 SetParamResponse()

```
void SystToolsEventResponseTree::SetParamResponse (
 paramId_t i,
 double v,
 double r) [inline]
```

#### 12.23.2.5 SetThrow()

```
void SystToolsEventResponseTree::SetThrow (
 ULong_t t) [inline]
```

#### 12.23.2.6 SetTotalWeight()

```
void SystToolsEventResponseTree::SetTotalWeight (
 double w) [inline]
```

#### 12.23.2.7 SetTree()

```
void SystToolsEventResponseTree::SetTree (
 TTree * t) [inline]
```

### 12.23.3 Member Data Documentation

### 12.23.3.1 event

```
ULong_t SystToolsEventResponseTree::event [private]
```

### 12.23.3.2 event\_responses

```
std::map<paramId_t, double> SystToolsEventResponseTree::event_responses [private]
```

### 12.23.3.3 param\_values

```
std::map<paramId_t, double> SystToolsEventResponseTree::param_values [private]
```

### 12.23.3.4 t\_it

```
ULong_t SystToolsEventResponseTree::t_it [private]
```

### 12.23.3.5 total\_weight

```
double SystToolsEventResponseTree::total_weight [private]
```

### 12.23.3.6 tree

```
TTree* SystToolsEventResponseTree::tree [private]
```

The documentation for this class was generated from the following file:

- systematicstools/module/[SystToolsResponseTreeMaker\\_module.cc](#)

## 12.24 SystToolsResponseTreeMaker Class Reference

Inheritance diagram for SystToolsResponseTreeMaker:

Collaboration diagram for SystToolsResponseTreeMaker:

## Public Member Functions

- [SystToolsResponseTreeMaker](#) (fhicl::ParameterSet const &p)
- [SystToolsResponseTreeMaker](#) ([SystToolsResponseTreeMaker](#) const &)=delete
- [SystToolsResponseTreeMaker](#) ([SystToolsResponseTreeMaker](#) &&)=delete
- [SystToolsResponseTreeMaker](#) & operator= ([SystToolsResponseTreeMaker](#) const &)=delete
- [SystToolsResponseTreeMaker](#) & operator= ([SystToolsResponseTreeMaker](#) &&)=delete
- void [analyze](#) (art::Event const &e) override

## Private Attributes

- art::InputTag [fInpTag](#)
- [SystToolsEventResponseTree](#) [fOutputTree](#)
- double [fTweak](#)
- bool [fSplineMode](#)
- [param\\_header\\_map\\_t](#) [configuredParameterHeaders](#)
- [EventSplineCache](#)< ULong\_t, [ParamValidationAndErrorResponse::kTortoise](#) > [fEventHelper](#)
- [ParamHeaderHelper](#) [fHeaderHelper](#)

## 12.24.1 Constructor & Destructor Documentation

### 12.24.1.1 [SystToolsResponseTreeMaker\(\)](#) [1/3]

```
SystToolsResponseTreeMaker::SystToolsResponseTreeMaker (
 fhicl::ParameterSet const & p) [explicit]
```

### 12.24.1.2 [SystToolsResponseTreeMaker\(\)](#) [2/3]

```
SystToolsResponseTreeMaker::SystToolsResponseTreeMaker (
 SystToolsResponseTreeMaker const &) [delete]
```

### 12.24.1.3 [SystToolsResponseTreeMaker\(\)](#) [3/3]

```
SystToolsResponseTreeMaker::SystToolsResponseTreeMaker (
 SystToolsResponseTreeMaker &&) [delete]
```

## 12.24.2 Member Function Documentation



**12.24.2.1 analyze()**

```
void SystToolsResponseTreeMaker::analyze (
 art::Event const & e) [override]
```

**12.24.2.2 operator=()** [1/2]

```
SystToolsResponseTreeMaker& SystToolsResponseTreeMaker::operator= (
 SystToolsResponseTreeMaker const &) [delete]
```

**12.24.2.3 operator=()** [2/2]

```
SystToolsResponseTreeMaker& SystToolsResponseTreeMaker::operator= (
 SystToolsResponseTreeMaker &&) [delete]
```

**12.24.3 Member Data Documentation****12.24.3.1 configuredParameterHeaders**

```
param_header_map_t SystToolsResponseTreeMaker::configuredParameterHeaders [private]
```

**12.24.3.2 fEventHelper**

```
EventSplineCache<ULong_t, ParamValidationAndErrorResponse::kTortoise> SystToolsResponseTree↔
Maker::fEventHelper [private]
```

**12.24.3.3 fHeaderHelper**

```
ParamHeaderHelper SystToolsResponseTreeMaker::fHeaderHelper [private]
```

**12.24.3.4 fInpTag**

```
art::InputTag SystToolsResponseTreeMaker::fInpTag [private]
```

#### 12.24.3.5 fOutputTree

```
SystToolsEventResponseTree SystToolsResponseTreeMaker::fOutputTree [private]
```

#### 12.24.3.6 fSplineMode

```
bool SystToolsResponseTreeMaker::fSplineMode [private]
```

#### 12.24.3.7 fTweak

```
double SystToolsResponseTreeMaker::fTweak [private]
```

The documentation for this class was generated from the following file:

- [systematicstools/module/SystToolsResponseTreeMaker\\_module.cc](#)

## Chapter 13

# File Documentation

### 13.1 README.md File Reference

### 13.2 systematicstools/app/CheckSystProviderConfigmd5.cc File Reference

```
#include "systematicstools/utility/md5.hh"
#include "systematicstools/utility/string_parsers.hh"
#include "fhiclcpp/ParameterSet.h"
#include "fhiclcpp/make_ParameterSet.h"
#include "cetlib/filepath_maker.h"
#include <fstream>
#include <iomanip>
#include <iostream>
```

Include dependency graph for CheckSystProviderConfigmd5.cc:

### 13.3 systematicstools/app/FindISystProvider.cc File Reference

```
#include "systematicstools/interface/ISystProviderTool.hh"
#include "fhiclcpp/ParameterSet.h"
#include "art/Utilities/make_tool.h"
#include <iomanip>
#include <iostream>
```

Include dependency graph for FindISystProvider.cc:

#### Namespaces

- [cliopts](#)

#### Functions

- void [SayUsage](#) (char const \*argv[])
- void [HandleOpts](#) (int argc, char const \*argv[])
- int [main](#) (int argc, char const \*argv[])

## Variables

- `std::string cliopts::provider_name = ""`
- `bool cliopts::quiet = false`
- `bool cliopts::dump_example_config = false`

### 13.3.1 Function Documentation

#### 13.3.1.1 HandleOpts()

```
void HandleOpts (
 int argc,
 char const * argv[])
```

#### 13.3.1.2 main()

```
int main (
 int argc,
 char const * argv[])
```

#### 13.3.1.3 SayUsage()

```
void SayUsage (
 char const * argv[])
```

## 13.4 systematicstools/app/GenerateSystProviderConfig.cc File Reference

```
#include "systematicstools/interface/ISystProviderTool.hh"
#include "systematicstools/interface/SystMetaData.hh"
#include "systematicstools/interface/types.hh"
#include "systematicstools/utility/ParameterAndProviderConfigurationUtility.↵
hh"
#include "systematicstools/utility/md5.hh"
#include "systematicstools/utility/printers.hh"
#include "systematicstools/utility/string_parsers.hh"
#include "fhiclcpp/ParameterSet.h"
#include "fhiclcpp/make_ParameterSet.h"
#include "cetlib/filepath_maker.h"
#include <fstream>
#include <iomanip>
#include <iostream>
Include dependency graph for GenerateSystProviderConfig.cc:
```

## Namespaces

- [cliopts](#)

## Functions

- void [SayUsage](#) (char const \*argv[ ])
- void [HandleOpts](#) (int argc, char const \*argv[ ])
- fhicl::ParameterSet [ReadParameterSet](#) (char const \*argv[ ])
- int [main](#) (int argc, char const \*argv[ ])

## Variables

- std::string [cliopts::outputfile](#) = ""
- std::string [cliopts::fhicl\\_key](#) = "syst\_providers"
- bool [cliopts::WrapWithPROLOG](#) = false

### 13.4.1 Function Documentation

#### 13.4.1.1 HandleOpts()

```
void HandleOpts (
 int argc,
 char const * argv[])
```

#### 13.4.1.2 main()

```
int main (
 int argc,
 char const * argv[])
```

#### 13.4.1.3 ReadParameterSet()

```
fhicl::ParameterSet ReadParameterSet (
 char const * argv[])
```

#### 13.4.1.4 SayUsage()

```
void SayUsage (
 char const * argv[])
```

### 13.5 systematicstools/doc/ExampleSystProvider.md File Reference

### 13.6 systematicstools/doc/MovingParts.md File Reference

### 13.7 systematicstools/doc/ParameterHeaders.md File Reference

### 13.8 systematicstools/doc/ToolConfiguration.md File Reference

### 13.9 systematicstools/doc/WritingAProvider.md File Reference

### 13.10 systematicstools/interface/EventResponse\_product.cc File Reference

```
#include "systematicstools/interface/EventResponse_product.hh"
#include <cmath>
Include dependency graph for EventResponse_product.cc:
```

#### Namespaces

- [systtools](#)

#### Functions

- void [systtools::ExtendEventResponse](#) (std::unique\_ptr< EventResponse > &e1, std::unique\_ptr< EventResponse > &e2)  
*Extends one EventResponse with the event\_unit\_response\_ts of another.*
- bool [systtools::FullOfUnity](#) (std::vector< double > const &vec, double tolerance=std::numeric\_limits< double >::epsilon())
- void [systtools::ScrubUnityEventResponses](#) (std::unique\_ptr< EventResponse > &er)  
*Removes [systtools::ParamResponses](#) from event\_unit\_response\_ts contained within an EventResponse that contain only unity responses.*
- void [systtools::ScrubUnityEventResponses](#) (event\_unit\_response\_t &er)  
*Removes [systtools::ParamResponses](#) from event\_unit\_response\_t that contain only unity responses.*

### 13.11 systematicstools/interface/EventResponse\_product.hh File Reference

```
#include "systematicstools/interface/SystParamHeader.hh"
#include "systematicstools/interface/types.hh"
#include "systematicstools/utility/exceptions.hh"
#include <memory>
#include <vector>
```

Include dependency graph for EventResponse\_product.hh: This graph shows which files directly or indirectly include this file:

## Classes

- struct [systools::ParamResponses](#)

## Namespaces

- [systools](#)

## Typedefs

- typedef std::vector< ParamResponses > [systools::event\\_unit\\_response\\_t](#)
- typedef std::vector< event\_unit\_response\_t > [systools::EventResponse](#)

*The systematic parameter responses calculated for an ART event.*

## Functions

- [systools::NEW\\_SYSTTOOLS\\_EXCEPT](#) (incompatible\_number\_of\_event\_units)  
*Exception raised when attempting to merge two event responses with differing number of event units.*
- void [systools::ExtendEventResponse](#) (std::unique\_ptr< EventResponse > &e1, std::unique\_ptr< EventResponse > &e2)  
*Extends one EventResponse with the event\_unit\_response\_ts of another.*
- void [systools::ScrubUnityEventResponses](#) (std::unique\_ptr< EventResponse > &er)  
*Removes [systools::ParamResponses](#) from event\_unit\_response\_ts contained within an EventResponse that contain only unity responses.*
- void [systools::ScrubUnityEventResponses](#) (event\_unit\_response\_t &er)  
*Removes [systools::ParamResponses](#) from event\_unit\_response\_t that contain only unity responses.*

## 13.12 systematicstools/interface/FHiCLSystParamHeaderConverters.cc File Reference

```
#include "systematicstools/interface/FHiCLSystParamHeaderConverters.hh"
#include "systematicstools/utility/string_parsers.hh"
#include "systematicstools/interface/types.hh"
#include "fhiclcpp/ParameterSet.h"
#include <vector>
#include <iomanip>
```

Include dependency graph for FHiCLSystParamHeaderConverters.cc:

## Namespaces

- [systools](#)

## Functions

- SystParamHeader [systools::FHiCLToSystParamHeader](#) (fhicl::ParameterSet const &paramset)  
*Deserializes a [SystParamHeader](#) instance from a passed FHiCL parameter set.*
- fhicl::ParameterSet [systools::SystParamHeaderToFHiCL](#) (SystParamHeader const &sph)  
*Serializes a [SystParamHeader](#) instance to a FHiCL table.*

### 13.13 systematicstools/interface/FHiCLSystParamHeaderConverters.hh File Reference

```
#include "systematicstools/utility/exceptions.hh"
#include <string>
```

Include dependency graph for FHiCLSystParamHeaderConverters.hh: This graph shows which files directly or indirectly include this file:

#### Namespaces

- [fhicl](#)
- [systtools](#)

#### Functions

- [systtools::NEW\\_SYSTTOOLS\\_EXCEPT](#) (invalid\_SystParamHeader\_key)
- SystParamHeader [systtools::FHiCLToSystParamHeader](#) (fhicl::ParameterSet const &paramset)  
Deserializes a [SystParamHeader](#) instance from a passed FHiCL parameter set.
- fhicl::ParameterSet [systtools::SystParamHeaderToFHiCL](#) (SystParamHeader const &sph)  
Serializes a [SystParamHeader](#) instance to a FHiCL table.

### 13.14 systematicstools/interface/ISystProviderTool.cc File Reference

```
#include "systematicstools/interface/ISystProviderTool.hh"
Include dependency graph for ISystProviderTool.cc:
```

#### Namespaces

- [systtools](#)

### 13.15 systematicstools/interface/ISystProviderTool.hh File Reference

```
#include "systematicstools/interface/EventResponse_product.hh"
#include "systematicstools/interface/SystMetaData.hh"
#include "systematicstools/interface/FHiCLSystParamHeaderConverters.hh"
#include "systematicstools/utility/exceptions.hh"
#include "art/Framework/Principal/Event.h"
#include "fhiclcpp/ParameterSet.h"
#include <iomanip>
#include <iostream>
#include <map>
#include <string>
```

Include dependency graph for ISystProviderTool.hh: This graph shows which files directly or indirectly include this file:



## Classes

- class [systools::ISystProviderTool](#)  
*ABC defining the interface to systematic response syst\_providers.*

## Namespaces

- [systools](#)

## Functions

- [systools::NEW\\_SYSTTOOLS\\_EXCEPT](#) (ISystProviderTool\_method\_unimplemented)
- [systools::NEW\\_SYSTTOOLS\\_EXCEPT](#) (ISystProviderTool\_seed\_suggestion\_post\_configure)
- [systools::NEW\\_SYSTTOOLS\\_EXCEPT](#) (ISystProviderTool\_noncontiguous\_parameter\_ids)
- [systools::NEW\\_SYSTTOOLS\\_EXCEPT](#) (ISystProviderTool\_metadata\_not\_generated)
- [systools::NEW\\_SYSTTOOLS\\_EXCEPT](#) (invalid\_ToolConfigurationFHiCL)
- [systools::NEW\\_SYSTTOOLS\\_EXCEPT](#) (invalid\_ToolOptions)

## 13.16 systematicstools/interface/SystMetaData.cc File Reference

```
#include "systematicstools/interface/SystMetaData.hh"
#include <set>
```

Include dependency graph for SystMetaData.cc:

## Namespaces

- [systools](#)

## Functions

- paramId\_t [systools::GetParamId](#) (SystMetaData const &md, std::string const &name)  
*Get parameter Id from a SystMetaData and pretty name.*
- size\_t [systools::GetParamIndex](#) (SystMetaData const &md, paramId\_t pid)  
*Get parameter index in header list for supplied parameter Id.*
- bool [systools::IndexIsHandled](#) (SystMetaData const &md, size\_t index)  
*Whether a given index is handled by the Syst meta data headers.*
- size\_t [systools::GetParamIndex](#) (SystMetaData const &md, std::string const &name)  
*Get parameter index in header list for supplied parameter pretty name.*
- bool [systools::HasParam](#) (SystMetaData const &md, std::string const &name)  
*Checks if named parameter exists in header list.*
- bool [systools::HasAnyParams](#) (SystMetaData const &md, std::vector< std::string > const &names)  
*Checks if any of the named parameters exists in header list.*
- bool [systools::HasParam](#) (SystMetaData const &md, paramId\_t pid)  
*Checks if parameter with given Id exists in header list.*
- SystParamHeader const & [systools::GetParam](#) (SystMetaData const &md, std::string const &name)  
*Gets a const reference to a parameter header given a header list and a parameter pretty name.*
- SystParamHeader & [systools::GetParam](#) (SystMetaData &md, std::string const &name)

- Gets a non-const reference to a parameter header given a header list and a parameter pretty name.*
  - `SystParamHeader const & systools::GetParam (SystMetaData const &md, paramId_t pid)`
    - Gets a const reference to a parameter header given a header list and a parameter Id.*
  - `SystParamHeader & systools::GetParam (SystMetaData &md, paramId_t pid)`
    - Gets a const reference to a parameter header given a header list and a parameter Id.*
  - `bool systools::Validate (SystMetaData const &sh, bool quiet=true)`
    - Checks for declared and mis-used interdependency between parameters in a list of parameter headers.*
  - `void systools::ExtendSystMetaData (SystMetaData &md1, SystMetaData const &md2)`
    - Merges two SystMetaData instances.*

## 13.17 systematicstools/interface/SystMetaData.hh File Reference

```
#include "systematicstools/interface/SystParamHeader.hh"
#include "systematicstools/interface/types.hh"
#include "systematicstools/utility/exceptions.hh"
#include <iomanip>
#include <iostream>
#include <string>
#include <vector>
#include <algorithm>
```

Include dependency graph for SystMetaData.hh: This graph shows which files directly or indirectly include this file:

### Namespaces

- [systools](#)

### Typedefs

- `typedef std::vector< SystParamHeader > systools::SystMetaData`
  - A list of Parameter Headers.*

### Functions

- `systools::NEW\_SYSTTOOLS\_EXCEPT (no_such_opt_kv)`
  - Exception raised when no key-value pair with a given key can be found in a given [SystParamHeader](#).*
- `systools::NEW\_SYSTTOOLS\_EXCEPT (invalid_SystMetaData)`
  - Exception raised if a SystMetaData fails basic interface validation.*
- `paramId_t systools::GetParamId (SystMetaData const &md, std::string const &name)`
  - Get parameter Id from a SystMetaData and pretty name.*
- `size_t systools::GetParamIndex (SystMetaData const &md, paramId_t pid)`
  - Get parameter index in header list for supplied parameter Id.*
- `bool systools::IndexIsHandled (SystMetaData const &md, size_t index)`
  - Whether a given index is handled by the Syst meta data headers.*
- `size_t systools::GetParamIndex (SystMetaData const &md, std::string const &name)`
  - Get parameter index in header list for supplied parameter pretty name.*
- `bool systools::HasParam (SystMetaData const &md, std::string const &name)`
  - Checks if named parameter exists in header list.*
- `bool systools::HasAnyParams (SystMetaData const &md, std::vector< std::string > const &names)`

- Checks if any of the named parameters exists in header list.*
- `bool systools::HasParam (SystMetaData const &md, paramId_t pid)`  
*Checks if parameter with given Id exists in header list.*
- `SystParamHeader const & systools::GetParam (SystMetaData const &md, std::string const &name)`  
*Gets a const reference to a parameter header given a header list and a parameter pretty name.*
- `SystParamHeader & systools::GetParam (SystMetaData &md, std::string const &name)`  
*Gets a non-const reference to a parameter header given a header list and a parameter pretty name.*
- `SystParamHeader const & systools::GetParam (SystMetaData const &md, paramId_t pid)`  
*Gets a const reference to a parameter header given a header list and a parameter Id.*
- `SystParamHeader & systools::GetParam (SystMetaData &md, paramId_t pid)`  
*Gets a const reference to a parameter header given a header list and a parameter Id.*
- `template<typename T >`  
`bool systools::SystHasOpt (SystMetaData const &md, T const &ident, std::string const &opt)`  
*Returns true if the Parameter Header specified by ident has a matching opts entry.*
- `template<typename T >`  
`bool systools::SystHasOptKV (SystMetaData const &md, T const &ident, std::string const &key)`  
*Returns true if the Parameter Header specified by ident has a matching opts key-value entry.*
- `template<typename T >`  
`std::string systools::SystGetOptKV (SystMetaData const &md, T const &ident, std::string const &key)`  
*Returns the option value corresponding to key on the Param Header specified by ident.*
- `bool systools::Validate (SystMetaData const &sh, bool quiet=true)`  
*Checks for declared and mis-used interdependency between parameters in a list of parameter headers.*
- `void systools::ExtendSystMetaData (SystMetaData &md1, SystMetaData const &md2)`  
*Merges two SystMetaData instances.*

## 13.18 systematicstools/interface/SystParamHeader.cc File Reference

```
#include "systematicstools/interface/SystParamHeader.hh"
#include <iomanip>
#include <iostream>
Include dependency graph for SystParamHeader.cc:
```

### Namespaces

- `systools`

### Functions

- `bool systools::Validate (SystParamHeader const &hdr, bool quiet=true)`  
*Checks interface validity of a [SystParamHeader](#).*

## 13.19 systematicstools/interface/SystParamHeader.hh File Reference

```
#include "systematicstools/utility/exceptions.hh"
#include <array>
#include <limits>
#include <string>
#include <vector>
```

Include dependency graph for SystParamHeader.hh: This graph shows which files directly or indirectly include this file:

## Classes

- struct [systools::SystParamHeader](#)

## Namespaces

- [systools](#)

## Typedefs

- typedef unsigned [systools::paramId\\_t](#)

## Functions

- [systools::NEW\\_SYSTTOOLS\\_EXCEPT](#) (invalid\_SystParamHeader)
- bool [systools::Validate](#) (SystParamHeader const &hdr, bool quiet=true)  
*Checks interface validity of a [SystParamHeader](#).*

## Variables

- constexpr double [systools::kDefaultDouble](#) = 0xdeadbeef  
*Magic values for signalling that a value is defaulted.*
- template<typename T >  
constexpr T [systools::kParamUnhandled](#) = std::numeric\_limits<T>::max()
- template<>  
constexpr double [systools::kParamUnhandled< double >](#) = kDefaultDouble

## 13.20 systematicstools/interface/types.hh File Reference

```
#include "systematicstools/interface/EventResponse_product.hh"
#include "systematicstools/interface/SystMetaData.hh"
#include "systematicstools/interface/SystParamHeader.hh"
#include "systematicstools/utility/exceptions.hh"
#include <map>
#include <memory>
#include <vector>
```

Include dependency graph for types.hh: This graph shows which files directly or indirectly include this file:

## Classes

- struct [systools::ParamValue](#)
- struct [systools::ParamThrows](#)
- struct [systools::ParamHeaderProviderName](#)

*Struct for holding [ISystProviderTool](#) unique name–handled parameter header pairs.*

## Namespaces

- [systtools](#)

## Typedefs

- typedef std::vector< ParamValue > [systtools::param\\_value\\_list\\_t](#)
- typedef std::vector< paramId\_t > [systtools::param\\_list\\_t](#)
- typedef std::vector< ParamThrows > [systtools::parameter\\_throws\\_list\\_t](#)
- typedef std::vector< std::unique\_ptr< ISystProviderTool > > [systtools::provider\\_list\\_t](#)
- typedef std::map< paramId\_t, ParamHeaderProviderName > [systtools::param\\_header\\_map\\_t](#)

*Map of parameter Identifiers to the relevant metadata and the unique name of the [ISystProviderTool](#) responsible for generating them.*

## Functions

- template<typename T >  
size\_t [systtools::GetParamContainerIndex](#) (std::vector< T > const &container, paramId\_t &pid)  
*Gets the index of a parameter-X association with a given paramId\_t.*
- template<typename T >  
bool [systtools::ContainterHasParam](#) (std::vector< T > const &container, paramId\_t pid)
- template<typename T >  
T & [systtools::GetParamElementFromContainer](#) (std::vector< T > &container, paramId\_t pid)
- template<typename T >  
T const & [systtools::GetParamElementFromContainer](#) (std::vector< T > const &container, paramId\_t pid)

## 13.21 systematicstools/interpreters/EventSplineCacheHelper.hh File Reference

```
#include "ParamHeaderHelper.hh"
#include "ParamValidationAndErrorResponse.hh"
#include <iostream>
```

Include dependency graph for EventSplineCacheHelper.hh: This graph shows which files directly or indirectly include this file:

## Classes

- class [systtools::EventSplineCacheBase< event\\_unit\\_t >](#)
- class [systtools::EventSplineCache< event\\_unit\\_t, CLtight, Enable >](#)
- class [systtools::EventSplineCache< event\\_unit\\_t, CLtight, typename std::enable\\_if< CLtight==ParamValidationAndErrorResponse::kHare, void >::type >](#)
- class [systtools::EventSplineCache< event\\_unit\\_t, CLtight, typename std::enable\\_if< CLtight==ParamValidationAndErrorResponse::kFrog, void >::type >](#)
- class [systtools::EventSplineCache< event\\_unit\\_t, CLtight, typename std::enable\\_if< CLtight==ParamValidationAndErrorResponse::kTortoise, void >::type >](#)

## Namespaces

- [systtools](#)

## Typedefs

- typedef size\_t [systools::eventId\\_t](#)

## 13.22 systematicstools/interpreters/ParamHeaderHelper.cc File Reference

```
#include "ParamHeaderHelper.hh"
#include "systematicstools/utility/printers.hh"
#include <iostream>
#include <utility>
Include dependency graph for ParamHeaderHelper.cc:
```

## Namespaces

- [systools](#)

## 13.23 systematicstools/interpreters/ParamHeaderHelper.hh File Reference

```
#include "systematicstools/interface/SystMetaData.hh"
#include "systematicstools/interface/types.hh"
#include "systematicstools/interpreters/PolyResponse.hh"
#include "systematicstools/interpreters/ParamValidationAndErrorResponse.hh"
#include "TSpline.h"
Include dependency graph for ParamHeaderHelper.hh: This graph shows which files directly or indirectly include this file:
```

## Classes

- class [systools::ParamHeaderHelper](#)

## Namespaces

- [systools](#)

## 13.24 systematicstools/interpreters/ParamValidationAndErrorResponse.cc File Reference

```
#include "ParamValidationAndErrorResponse.hh"
#include <cmath>
#include <iostream>
Include dependency graph for ParamValidationAndErrorResponse.cc:
```

## 13.25 systematicstools/interpreters/ParamValidationAndErrorResponse.hh File Reference

```
#include "systematicstools/interface/SystMetaData.hh"
#include <limits>
```

Include dependency graph for ParamValidationAndErrorResponse.hh: This graph shows which files directly or indirectly include this file:

### Classes

- struct [ParamValidationAndErrorResponse](#)

## 13.26 systematicstools/interpreters/PolyResponse.hh File Reference

```
#include "systematicstools/utility/ROOTUtility.hh"
#include <array>
```

Include dependency graph for PolyResponse.hh: This graph shows which files directly or indirectly include this file:

### Classes

- struct [systtools::PolyResponse< n >](#)

### Namespaces

- [systtools](#)

## 13.27 systematicstools/interpreters/PrecalculatedResponseReader.hh File Reference

```
#include "systematicstools/interface/EventResponse_product.hh"
#include "systematicstools/interface/types.hh"
#include "systematicstools/interpreters/PolyResponse.hh"
#include "systematicstools/utility/exceptions.hh"
#include "TFile.h"
#include "TTree.h"
#include <iomanip>
#include <vector>
```

Include dependency graph for PrecalculatedResponseReader.hh:

### Classes

- class [systtools::PrecalculatedResponseReader< Order >](#)
- struct [systtools::PrecalculatedResponseReader< Order >::ParamPolyResponses](#)

### Namespaces

- [systtools](#)

## Functions

- [systools::NEW\\_SYSTTOOLS\\_EXCEPT](#) (invalid\_tfile\_name)
- [systools::NEW\\_SYSTTOOLS\\_EXCEPT](#) (invalid\_ttree\_name)

## 13.28 systematictools/module/classes.h File Reference

```
#include "systematictools/interface/EventResponse_product.hh"
#include "canvas/Persistency/Common/Wrapper.h"
#include <vector>
Include dependency graph for classes.h:
```

## 13.29 systematictools/module/SystToolsEventResponse\_module.cc File Reference

```
#include "systematictools/interface/ISystProviderTool.hh"
#include "systematictools/utility/ParameterAndProviderConfigurationUtility.↵
hh"
#include "systematictools/utility/exceptions.hh"
#include "systematictools/utility/md5.hh"
#include "art/Framework/Core/EDProducer.h"
#include "art/Framework/Core/ModuleMacros.h"
#include "art/Framework/Principal/Event.h"
#include "art/Framework/Principal/Handle.h"
#include "art/Framework/Principal/Run.h"
#include "art/Framework/Principal/SubRun.h"
#include "art/Utilities/make_tool.h"
#include "canvas/Utilities/InputTag.h"
#include "fhiclcpp/ParameterSet.h"
#include "fhiclcpp/types/Sequence.h"
#include "fhiclcpp/types/Table.h"
#include "messagefacility/MessageLogger/MessageLogger.h"
#include <memory>
Include dependency graph for SystToolsEventResponse_module.cc:
```

## Classes

- class [SystToolsEventResponse](#)

## 13.30 systematictools/module/SystToolsResponseTreeMaker\_module.cc File Reference

```
#include "systematictools/interface/EventResponse_product.hh"
#include "systematictools/interface/types.hh"
#include "systematictools/interpreters/EventSplineCacheHelper.hh"
#include "systematictools/interpreters/ParamHeaderHelper.hh"
#include "systematictools/interpreters/load_parameter_headers.hh"
#include "systematictools/utility/md5.hh"
#include "art/Framework/Core/EDAnalyzer.h"
```



```
#include "art/Framework/Core/ModuleMacros.h"
#include "art/Framework/Principal/Event.h"
#include "art/Framework/Principal/Handle.h"
#include "art/Framework/Principal/Run.h"
#include "art/Framework/Principal/SubRun.h"
#include "art/Framework/Services/Optional/TFileService.h"
#include "canvas/Utilities/InputTag.h"
#include "fhiclcpp/ParameterSet.h"
#include "messagefacility/MessageLogger/MessageLogger.h"
#include "TTree.h"
```

Include dependency graph for SystToolsResponseTreeMaker\_module.cc:

## Classes

- class [SystToolsEventResponseTree](#)
- class [SystToolsResponseTreeMaker](#)

## 13.31 systematictools/systproviders/CorrelatedMultisimProvider\_tool.cc File Reference

```
#include "systematictools/interface/ISystProviderTool.hh"
#include "systematictools/utility/CovMatThrower.hh"
#include "systematictools/utility/append_event_response.hh"
#include "systematictools/utility/configure_syst_providers.hh"
#include "systematictools/utility/generate_provider_parameter_set.hh"
#include "systematictools/utility/printers.hh"
#include "systematictools/utility/string_parsers.hh"
#include "art/Framework/Services/Optional/RandomNumberGenerator.h"
#include "art/Framework/Services/Registry/ServiceHandle.h"
#include "art/Utilities/ToolMacros.h"
#include "fhiclcpp/types/Atom.h"
#include "fhiclcpp/types/Sequence.h"
#include "fhiclcpp/types/Table.h"
#include "fhiclcpp/types/Comment.h"
#include "fhiclcpp/types/Name.h"
#include "CLHEP/Random/MTwistEngine.h"
#include "CLHEP/Random/RandGaussQ.h"
#include "TFile.h"
#include "TMatrixD.h"
#include <chrono>
#include <sstream>
```

Include dependency graph for CorrelatedMultisimProvider\_tool.cc:

## Classes

- class [CorrelatedMultisimProvider](#)

## 13.32 systematictools/systproviders/ExampleSystProvider\_tool.cc File Reference

```
#include "systematictools/systproviders/ExampleISystProvider_tool.hh"
#include "systematictools/utility/FHiCLSystParamHeaderUtility.hh"
#include "systematictools/utility/printers.hh"
#include "systematictools/utility/string_parsers.hh"
#include "art/Utilities/ToolMacros.h"
```

Include dependency graph for ExampleSystProvider\_tool.cc:

## Functions

- double [GetNormResponse](#) (double param\_val\_nu)
- double [GetLateralResponse](#) (double param\_val\_nu)
- double [GetResponse\\_nu](#) (double param\_val\_nu, [SystParamHeader](#) const &sph)
- double [GetParamShift\\_nu](#) (double shift\_sigma, [SystParamHeader](#) const &sph)
- double [GetParamValue\\_nu](#) (double shift\_sigma, [SystParamHeader](#) const &sph)
- double [GetResponse\\_shift](#) (double shift\_sigma, [SystParamHeader](#) const &sph)
- double [GetResponse](#) (double val, [SystParamHeader](#) const &sph)

## Variables

- double [default\\_centralvalue\\_nu](#) = 1
- double [default\\_lowsigmavalue\\_nu](#) = 5
- double [default\\_upsigmavalue\\_nu](#) = 5

### 13.32.1 Function Documentation

#### 13.32.1.1 GetLateralResponse()

```
double GetLateralResponse (
 double param_val_nu)
```

#### 13.32.1.2 GetNormResponse()

```
double GetNormResponse (
 double param_val_nu)
```

#### 13.32.1.3 GetParamShift\_nu()

```
double GetParamShift_nu (
 double shift_sigma,
 SystParamHeader const & sph)
```

#### 13.32.1.4 GetParamValue\_nu()

```
double GetParamValue_nu (
 double shift_sigma,
 SystParamHeader const & sph)
```

#### 13.32.1.5 GetResponse()

```
double GetResponse (
 double val,
 SystParamHeader const & sph)
```

#### 13.32.1.6 GetResponse\_nu()

```
double GetResponse_nu (
 double param_val_nu,
 SystParamHeader const & sph)
```

#### 13.32.1.7 GetResponse\_shift()

```
double GetResponse_shift (
 double shift_sigma,
 SystParamHeader const & sph)
```

### 13.32.2 Variable Documentation

#### 13.32.2.1 default\_centralvalue\_nu

```
double default_centralvalue_nu = 1
```

#### 13.32.2.2 default\_lowsigmavalue\_nu

```
double default_lowsigmavalue_nu = 5
```

#### 13.32.2.3 default\_upsigmavalue\_nu

```
double default_upsigmavalue_nu = 5
```

### 13.33 systematictools/systproviders/ExampleSystProvider\_tool.hh File Reference

```
#include "systematictools/interface/ISystProviderTool.hh"
#include <memory>
#include <random>
#include <string>
```

Include dependency graph for ExampleSystProvider\_tool.hh: This graph shows which files directly or indirectly include this file:

#### Classes

- class [ExampleSystProvider](#)

### 13.34 systematictools/utility/CovMatThrower.cc File Reference

```
#include "CovMatThrower.hh"
#include "TDecompChol.h"
#include <iostream>
```

Include dependency graph for CovMatThrower.cc:

### 13.35 systematictools/utility/CovMatThrower.hh File Reference

```
#include "CLHEP/Random/MTwistEngine.h"
#include "CLHEP/Random/RandGaussQ.h"
#include "TMatrixDSym.h"
#include "TMatrixD.h"
#include <memory>
```

Include dependency graph for CovMatThrower.hh: This graph shows which files directly or indirectly include this file:

#### Classes

- class [CovarianceThrower](#)

### 13.36 systematictools/utility/exceptions.hh File Reference

```
#include <sstream>
#include <stdexcept>
#include <string>
```

Include dependency graph for exceptions.hh: This graph shows which files directly or indirectly include this file:

#### Classes

- struct [systtools::systematictools\\_except](#)

## Namespaces

- [systtools](#)

## Macros

- `#define NEW_SYSTTOOLS_EXCEPT(EXCEPT_NAME)`

## Functions

- [systtools::NEW\\_SYSTTOOLS\\_EXCEPT](#) (invalid\_parameter\_name)
- [systtools::NEW\\_SYSTTOOLS\\_EXCEPT](#) (invalid\_parameter\_id)
- [systtools::NEW\\_SYSTTOOLS\\_EXCEPT](#) (invalid\_parameter\_value)
- [systtools::NEW\\_SYSTTOOLS\\_EXCEPT](#) (incorrectly\_configured)
- [systtools::NEW\\_SYSTTOOLS\\_EXCEPT](#) (parameter\_id\_not\_handled)
- [systtools::NEW\\_SYSTTOOLS\\_EXCEPT](#) (parameter\_name\_not\_handled)
- [systtools::NEW\\_SYSTTOOLS\\_EXCEPT](#) (systParamId\_collision)

### 13.36.1 Macro Definition Documentation

#### 13.36.1.1 NEW\_SYSTTOOLS\_EXCEPT

```
#define NEW_SYSTTOOLS_EXCEPT(
 EXCEPT_NAME)
```

#### Value:

```
struct EXCEPT_NAME : public systtools::systematicstools_except {
 EXCEPT_NAME() : systtools::systematicstools_except() {}
 EXCEPT_NAME(EXCEPT_NAME const &other) : systematicstools_except(other) {}
 template <typename T> EXCEPT_NAME &operator<<(T const &obj) {
 msgstrm << obj;
 msg = msgstrm.str();
 return (*this);
 }
}
```

## 13.37 systematicstools/utility/FHiCLSystParamHeaderUtility.cc File Reference

```
#include "systematicstools/utility/FHiCLSystParamHeaderUtility.hh"
#include "systematicstools/utility/string_parsers.hh"
#include "systematicstools/interface/SystMetaData.hh"
#include "systematicstools/interface/types.hh"
#include "fhiclcpp/ParameterSet.h"
#include <chrono>
#include <functional>
#include <iomanip>
#include <iostream>
#include <random>
#include <vector>
```

Include dependency graph for FHiCLSystParamHeaderUtility.cc:

## Namespaces

- [systtools](#)

## Functions

- bool [systtools::ParseFHiCLVariationDescriptor](#) (fhiCL::ParameterSet const &paramset, std::string const &C↵ V\_key, std::string const &vardescriptor\_key, SystParamHeader &hdr)  
*Set up [SystParamHeader](#) variation definitions from common format.*
- bool [systtools::MakeFHiCLDefinedRandomVariations](#) (fhiCL::ParameterSet const &paramset, std::string const &nthrows\_key, SystParamHeader &hdr, std::string const &distribution\_key="", uint64\_t seed=0, size\_t N↵ Throws=0)  
*Throws random parameter variations.*
- bool [systtools::FHiCLSimpleToolConfigurationParameterExists](#) (fhiCL::ParameterSet const &paramset, std↵ ::string const &parameter\_name)  
*Checks if paramset appears to provide standardized Tool Configuration for a named parameter.*
- bool [systtools::ParseFHiCLSimpleToolConfigurationParameter](#) (fhiCL::ParameterSet const &paramset, std↵ ::string const &parameter\_name, SystParamHeader &hdr, uint64\_t seed=0, size\_t NThrows=0)  
*Builds [SystParamHeader](#) from standardized FHiCL that can be used to write Tool Configuration files.*

## 13.38 systematictools/utility/FHiCLSystParamHeaderUtility.hh File Reference

```
#include "systematictools/utility/exceptions.hh"
#include <string>
```

Include dependency graph for FHiCLSystParamHeaderUtility.hh: This graph shows which files directly or indirectly include this file:

## Namespaces

- [fhiCL](#)
- [systtools](#)

## Functions

- [systtools::NEW\\_SYSTTOOLS\\_EXCEPT](#) (invalid\_FHiCL\_variation\_descriptor)
- [systtools::NEW\\_SYSTTOOLS\\_EXCEPT](#) (invalid\_FHiCL\_random\_distribution\_descriptor)
- bool [systtools::ParseFHiCLVariationDescriptor](#) (fhiCL::ParameterSet const &paramset, std::string const &C↵ V\_key, std::string const &vardescriptor\_key, SystParamHeader &hdr)  
*Set up [SystParamHeader](#) variation definitions from common format.*
- bool [systtools::MakeFHiCLDefinedRandomVariations](#) (fhiCL::ParameterSet const &paramset, std::string const &nthrows\_key, SystParamHeader &hdr, std::string const &distribution\_key="", uint64\_t seed=0, size\_t N↵ Throws=0)  
*Throws random parameter variations.*
- bool [systtools::FHiCLSimpleToolConfigurationParameterExists](#) (fhiCL::ParameterSet const &paramset, std↵ ::string const &parameter\_name)  
*Checks if paramset appears to provide standardized Tool Configuration for a named parameter.*
- bool [systtools::ParseFHiCLSimpleToolConfigurationParameter](#) (fhiCL::ParameterSet const &paramset, std↵ ::string const &parameter\_name, SystParamHeader &hdr, uint64\_t seed=0, size\_t NThrows=0)  
*Builds [SystParamHeader](#) from standardized FHiCL that can be used to write Tool Configuration files.*

## 13.39 systematicstools/utility/md5.cc File Reference

```
#include "systematicstools/utility/md5.hh"
#include <cstdio>
Include dependency graph for md5.cc:
```

### Macros

- `#define S11` 7
- `#define S12` 12
- `#define S13` 17
- `#define S14` 22
- `#define S21` 5
- `#define S22` 9
- `#define S23` 14
- `#define S24` 20
- `#define S31` 4
- `#define S32` 11
- `#define S33` 16
- `#define S34` 23
- `#define S41` 6
- `#define S42` 10
- `#define S43` 15
- `#define S44` 21

### Functions

- `std::ostream & operator<<` (`std::ostream &out`, `MD5 md5`)
- `std::string md5` (`const std::string str`)

#### 13.39.1 Macro Definition Documentation

##### 13.39.1.1 S11

```
#define S11 7
```

##### 13.39.1.2 S12

```
#define S12 12
```

**13.39.1.3 S13**

```
#define S13 17
```

**13.39.1.4 S14**

```
#define S14 22
```

**13.39.1.5 S21**

```
#define S21 5
```

**13.39.1.6 S22**

```
#define S22 9
```

**13.39.1.7 S23**

```
#define S23 14
```

**13.39.1.8 S24**

```
#define S24 20
```

**13.39.1.9 S31**

```
#define S31 4
```

**13.39.1.10 S32**

```
#define S32 11
```



#### 13.39.1.11 S33

```
#define S33 16
```

#### 13.39.1.12 S34

```
#define S34 23
```

#### 13.39.1.13 S41

```
#define S41 6
```

#### 13.39.1.14 S42

```
#define S42 10
```

#### 13.39.1.15 S43

```
#define S43 15
```

#### 13.39.1.16 S44

```
#define S44 21
```

### 13.39.2 Function Documentation

#### 13.39.2.1 md5()

```
std::string md5 (
 const std::string str)
```

### 13.39.2.2 operator<<()

```
std::ostream& operator<< (
 std::ostream & out,
 MD5 md5)
```

## 13.40 systematicstools/utility/md5.hh File Reference

```
#include <cstring>
#include <iostream>
#include <stdint>
```

Include dependency graph for md5.hh: This graph shows which files directly or indirectly include this file:

### Classes

- class [MD5](#)

### Functions

- std::string [md5](#) (const std::string str)

### 13.40.1 Function Documentation

#### 13.40.1.1 md5()

```
std::string md5 (
 const std::string str)
```

## 13.41 systematicstools/utility/ParameterAndProviderConfigurationUtility.cc File Reference

```
#include "systematicstools/utility/ParameterAndProviderConfigurationUtility.h"
#include "systematicstools/interface/FHiCLSysParamHeaderConverters.hh"
#include <iomanip>
#include <iostream>
#include <map>
#include <vector>
```

Include dependency graph for ParameterAndProviderConfigurationUtility.cc:

### Namespaces

- [systools](#)

## Functions

- param\_header\_map\_t [systools::BuildParameterHeaders](#) (fhicl::ParameterSet const &paramset, std::string const &key="syst\_providers")

*Builds map of SystProvider instances and handled parameters from a ParameterHeaders FHiCL document.*

## 13.42 systematicstools/utility/ParameterAndProviderConfigurationUtility.hh File Reference

```
#include "systematicstools/interface/ISystProviderTool.hh"
#include "systematicstools/interface/SystParamHeader.hh"
#include "systematicstools/interface/types.hh"
#include "systematicstools/utility/exceptions.hh"
#include "art/Utilities/make_tool.h"
#include "fhiclcpp/ParameterSet.h"
#include <chrono>
#include <functional>
#include <memory>
#include <random>
#include <string>
```

Include dependency graph for ParameterAndProviderConfigurationUtility.hh: This graph shows which files directly or indirectly include this file:

## Namespaces

- [systools](#)

## Functions

- [systools::NEW\\_SYSTTOOLS\\_EXCEPT](#) (ISystProvider\_FQName\_collision)
 

*Exception thrown when two ISystProviderTools have identical fully qualified (tool\_name + instance\_name) names.*
- param\_header\_map\_t [systools::BuildParameterHeaders](#) (fhicl::ParameterSet const &paramset, std::string const &key="syst\_providers")
 

*Builds map of SystProvider instances and handled parameters from a ParameterHeaders FHiCL document.*
- template<typename T = systools::ISystProviderTool>
 param\_header\_map\_t [systools::BuildParameterHeaders](#) (std::vector< std::unique\_ptr< T >> const &ConfiguredProviders)
 

*Builds map of SystProvider instances and handled parameters from a set of pre-configured providers.*
- template<typename T = systools::ISystProviderTool>
 std::vector< std::unique\_ptr< T >> [systools::ConfigureISystProvidersFromToolConfig](#) (fhicl::ParameterSet const &paramset, std::function< std::unique\_ptr< T >(fhicl::ParameterSet const &)> InstanceBuilder= [](fhicl::ParameterSet const &paramset) -> std::unique\_ptr< T > { return art::make\_tool< T >(paramset);}, std::string const &key="syst\_providers", paramId\_t syst\_param\_id=0)
 

*Configures the set of ISystProviders from a Tool Configuration document.*
- template<typename T = systools::ISystProviderTool>
 std::vector< std::unique\_ptr< T >> [systools::ConfigureISystProvidersFromParameterHeaders](#) (fhicl::ParameterSet const &paramset, std::function< std::unique\_ptr< T >(fhicl::ParameterSet const &)> InstanceBuilder= [](fhicl::ParameterSet const &paramset) -> std::unique\_ptr< T > { return art::make\_tool< T >(paramset);}, std::string const &key="syst\_providers")
 

*Configures the set of ISystProviders from a Parameter Headers document.*

### 13.43 systematictools/utility/printers.hh File Reference

```
#include "systematictools/interface/EventResponse_product.hh"
#include "systematictools/interface/FHiCLSysParamHeaderConverters.hh"
#include "systematictools/interface/SystParamHeader.hh"
#include "fhiclcpp/ParameterSet.h"
#include <iomanip>
#include <sstream>
#include <string>
```

Include dependency graph for printers.hh: This graph shows which files directly or indirectly include this file:

#### Namespaces

- [systtools](#)

#### Functions

- `std::string systtools::to_str` (SystParamHeader const &sph, bool indent=true)
- `std::string systtools::to_str` (EventResponse const &er)

### 13.44 systematictools/utility/ResponselessParamUtility.cc File Reference

```
#include "systematictools/utility/ResponselessParamUtility.hh"
#include "systematictools/interface/ISystProviderTool.hh"
#include <iomanip>
```

Include dependency graph for ResponselessParamUtility.cc:

#### Namespaces

- [systtools](#)

#### Functions

- void `systtools::FinalizeAndValidateDependentParameters` (SystMetaData &, std::string const &response\_<sub>↔</sub> parameter\_name, std::vector< std::string > const &dependent\_parameter\_names)  
*Checks that all responseless parameters in the set have the same number of configured variations and adds dummy variations for the response parameter.*

### 13.45 systematictools/utility/ResponselessParamUtility.hh File Reference

```
#include "systematictools/interface/SystMetaData.hh"
#include <vector>
#include <string>
```

Include dependency graph for ResponselessParamUtility.hh: This graph shows which files directly or indirectly include this file:

## Namespaces

- [systools](#)

## Functions

- void [systools::FinalizeAndValidateDependentParameters](#) (SystMetaData &, std::string const &response\_  
parameter\_name, std::vector< std::string > const &dependent\_parameter\_names)  
*Checks that all responseless parameters in the set have the same number of configured variations and adds dummy variations for the response parameter.*

## 13.46 systematicstools/utility/ROOTUtility.hh File Reference

```
#include "systematicstools/utility/exceptions.hh"
#include "TAxis.h"
#include "TF1.h"
#include "TFile.h"
#include "TGraph.h"
#include <string>
#include <memory>
```

Include dependency graph for ROOTUtility.hh: This graph shows which files directly or indirectly include this file:

## Functions

- [NEW\\_SYSTTOOLS\\_EXCEPT](#) (invalid\_tfile)
- [NEW\\_SYSTTOOLS\\_EXCEPT](#) (invalid\_hist\_name)
- TFile \* [CheckOpenFile](#) (std::string const &fname, char const \*opts="")
- template<class TH >  
TH \* [GetHistogram](#) (TFile \*f, std::string const &fname)
- template<class TH >  
TH \* [GetHistogram](#) (std::string const &fname, std::string const &hname)
- bool [IsFlowBin](#) (TAxis \*ax, Int\_t bin\_it)
- bool [IsInHistogramRange](#) (TAxis \*ax, double v)
- template<size\_t n>  
std::array< double, n+1 > [GetPolyFitCoeffs](#) (std::vector< double > const &xvals, std::vector< double >  
const &yvals)

### 13.46.1 Function Documentation

#### 13.46.1.1 CheckOpenFile()

```
TFile* CheckOpenFile (
 std::string const & fname,
 char const * opts = "") [inline]
```

**13.46.1.2 GetHistogram()** [1/2]

```
template<class TH >
TH* GetHistogram (
 TFile * f,
 std::string const & fname) [inline]
```

**13.46.1.3 GetHistogram()** [2/2]

```
template<class TH >
TH* GetHistogram (
 std::string const & fname,
 std::string const & hname) [inline]
```

**13.46.1.4 GetPolyFitCoeffs()**

```
template<size_t n>
std::array<double, n + 1> GetPolyFitCoeffs (
 std::vector< double > const & xvals,
 std::vector< double > const & yvals) [inline]
```

**13.46.1.5 IsFlowBin()**

```
bool IsFlowBin (
 TAxis * ax,
 Int_t bin_it) [inline]
```

**13.46.1.6 IsInHistogramRange()**

```
bool IsInHistogramRange (
 TAxis * ax,
 double v) [inline]
```

**13.46.1.7 NEW\_SYSTTOOLS\_EXCEPT()** [1/2]

```
NEW_SYSTTOOLS_EXCEPT (
 invalid_tfile)
```

## 13.46.1.8 NEW\_SYSTTOOLS\_EXCEPT() [2/2]

```
NEW_SYSTTOOLS_EXCEPT (
 invalid_hist_name)
```

## 13.47 systematicstools/utility/string\_parsers.hh File Reference

```
#include <algorithm>
#include <cctype>
#include <iostream>
#include <locale>
#include <sstream>
#include <string>
#include <vector>
```

Include dependency graph for string\_parsers.hh: This graph shows which files directly or indirectly include this file:

## Namespaces

- [systtools](#)

## Functions

- `template<typename T >`  
`T systtools::str2T (std::string const &str)`
- `template<>`  
`bool systtools::str2T< bool > (std::string const &str)`
- `template<typename T >`  
`void systtools::AppendVect (std::vector< T > &target, std::vector< T > const &toApp)`
- `template<typename T >`  
`std::vector< T > systtools::ParseToVect (std::string const &inp, std::string const &delim, bool Push←  
Empty=false, bool trimInput=true)`





# Index

- ~CovarianceThrower
  - CovarianceThrower, [52](#)
- ~ISystProviderTool
  - systtools::ISystProviderTool, [69](#)
- AddEventResponses
  - systtools::PrecalculatedResponseReader, [113](#)
- AllocateVectors
  - systtools::PrecalculatedResponseReader, [113](#)
- analyze
  - SystToolsResponseTreeMaker, [128](#)
- analyzer\_name
  - cliopts, [25](#)
- AppendVect
  - systtools, [32](#)
- applyToAll
  - ExampleISystProvider, [67](#)
- AsString
  - CorrelatedMultisimProvider, [50](#)
  - ExampleISystProvider, [66](#)
  - systtools::ISystProviderTool, [69](#)
- buffer
  - MD5, [81](#)
- BuildParameterHeaders
  - systtools, [32](#)
- BuildSystMetaData
  - ExampleISystProvider, [66](#)
  - systtools::ISystProviderTool, [70](#)
- CVector
  - CovarianceThrower, [53](#)
- CacheEvent
  - systtools::EventSplineCacheBase, [61](#)
- CacheEvents
  - systtools::EventSplineCacheBase, [61](#)
- CareLevel
  - ParamValidationAndErrorResponse, [105](#)
- centralParamValue
  - systtools::SystParamHeader, [119](#)
- CheckHaveMetaData
  - systtools::ISystProviderTool, [70](#)
- CheckOpenFile
  - ROOTUtility.hh, [157](#)
- CheckParamList
  - systtools::ParamHeaderHelper, [86](#)
- CheckParamValueList
  - systtools::ParamHeaderHelper, [86](#)
- CheckResponse
  - ParamValidationAndErrorResponse, [106](#)
- child\_providers
  - CorrelatedMultisimProvider, [50](#)
- cliopts, [25](#)
  - analyzer\_name, [25](#)
  - dump\_example\_config, [25](#)
  - envvar, [25](#)
  - fname, [26](#)
  - fhicl\_key, [26](#)
  - lookup\_policy, [26](#)
  - outputfile, [26](#)
  - producer\_name, [26](#)
  - provider\_name, [26](#)
  - quiet, [26](#)
  - WrapWithPROLOG, [26](#)
- coeffs\_1D
  - systtools::PrecalculatedResponseReader, [115](#)
- Configure
  - CorrelatedMultisimProvider, [50](#)
- ConfigureFromFHICL
  - CorrelatedMultisimProvider, [50](#)
- ConfigureFromParameterHeaders
  - systtools::ISystProviderTool, [70](#)
- ConfigureFromToolConfig
  - systtools::ISystProviderTool, [70](#)
- ConfigureISystProvidersFromParameterHeaders
  - systtools, [33](#)
- ConfigureISystProvidersFromToolConfig
  - systtools, [33](#)
- configuredParameterHeaders
  - SystToolsResponseTreeMaker, [129](#)
- ContainterHasParam
  - systtools, [33](#)
- CorrelatedMultisimProvider, [49](#)
  - AsString, [50](#)
  - child\_providers, [50](#)
  - Configure, [50](#)
  - ConfigureFromFHICL, [50](#)
  - CorrelatedMultisimProvider, [49](#)
  - GetEventResponse, [50](#)
  - RNJesus, [51](#)
  - RNgin, [50](#)
- count
  - MD5, [81](#)
- CovarianceThrower, [51](#)
  - ~CovarianceThrower, [52](#)
  - CVector, [53](#)
  - CovarianceThrower, [51](#), [52](#)
  - LMatrix, [53](#)
  - NRows, [53](#)

- RNJesus, [53](#)
- RNgine, [53](#)
- RVector, [53](#)
- SetupDecomp, [52](#)
- Throw, [53](#)
- UncertMatrix, [54](#)
- currentValues
  - systtools::EventSplineCacheBase, [64](#)
- DeclareUsingParameter
  - systtools::EventSplineCacheBase, [62](#)
- DeclareUsingParameters
  - systtools::EventSplineCacheBase, [62](#)
- decode
  - MD5, [77](#)
- default\_centralvalue\_nu
  - ExampleISystProvider\_tool.cc, [147](#)
- default\_lowsigmavalue\_nu
  - ExampleISystProvider\_tool.cc, [147](#)
- default\_upsigmavalue\_nu
  - ExampleISystProvider\_tool.cc, [147](#)
- differsEventByEvent
  - systtools::SystParamHeader, [119](#)
- digest
  - MD5, [81](#)
- discrete\_variation\_list\_t
  - systtools::ParamHeaderHelper, [85](#)
- dump\_example\_config
  - cliopts, [25](#)
- encode
  - MD5, [77](#)
- envvar
  - cliopts, [25](#)
- ErrorResponseLevel
  - ParamValidationAndErrorResponse, [105](#)
- eval
  - systtools::PolyResponse, [111](#)
- event
  - SystToolsEventResponseTree, [126](#)
- event\_responses
  - SystToolsEventResponseTree, [127](#)
- event\_t
  - systtools::EventSplineCacheBase, [60](#)
- event\_unit\_response\_t
  - systtools, [30](#)
- eventId\_t
  - systtools, [30](#)
- EventResponse
  - systtools, [31](#)
- EventSplineCacheBase
  - systtools::EventSplineCacheBase, [60](#), [61](#)
- ExampleISystProvider, [65](#)
  - applyToAll, [67](#)
  - AsString, [66](#)
  - BuildSystMetaData, [66](#)
  - ExampleISystProvider, [66](#)
  - GetEventResponse, [66](#)
  - GetExtraToolOptions, [66](#)
  - RNJesus, [67](#)
  - RNgine, [67](#)
  - SetupResponseCalculator, [67](#)
- ExampleISystProvider\_tool.cc
  - default\_centralvalue\_nu, [147](#)
  - default\_lowsigmavalue\_nu, [147](#)
  - default\_upsigmavalue\_nu, [147](#)
  - GetLateralResponse, [146](#)
  - GetNormResponse, [146](#)
  - GetParamShift\_nu, [146](#)
  - GetParamValue\_nu, [146](#)
  - GetResponse, [146](#)
  - GetResponse\_nu, [147](#)
  - GetResponse\_shift, [147](#)
- exceptions.hh
  - NEW\_SYSTTOOLS\_EXCEPT, [149](#)
- ExtendEventResponse
  - systtools, [34](#)
- ExtendSystMetaData
  - systtools, [34](#)
- F
  - MD5, [78](#)
- fAllowNegativeWeights
  - ParamValidationAndErrorResponse, [107](#)
- fCare
  - ParamValidationAndErrorResponse, [107](#)
- fChkErr
  - systtools::EventSplineCacheBase, [64](#)
  - systtools::ParamHeaderHelper, [100](#)
- fErrorResponse
  - ParamValidationAndErrorResponse, [108](#)
- fEventHelper
  - SystToolsResponseTreeMaker, [129](#)
- fEvents
  - systtools::EventSplineCacheBase, [64](#)
- fFQName
  - systtools::ISystProviderTool, [74](#)
- fHaveSystMetaData
  - systtools::ISystProviderTool, [74](#)
- fHeaderHelper
  - SystToolsResponseTreeMaker, [129](#)
  - systtools::EventSplineCacheBase, [64](#)
- fHeaders
  - systtools::ParamHeaderHelper, [100](#)
  - systtools::PrecalculatedResponseReader, [115](#)
- FHiCLSimpleToolConfigurationParameterExists
  - systtools, [34](#)
- FHiCLToSystParamHeader
  - systtools, [34](#)
- fInpTag
  - SystToolsResponseTreeMaker, [129](#)
- fInstanceName
  - systtools::ISystProviderTool, [74](#)
- fIsFullyConfigured
  - systtools::ISystProviderTool, [74](#)
- fLargeWeight
  - ParamValidationAndErrorResponse, [108](#)
- fOutputTree

- SystToolsResponseTreeMaker, [129](#)
- fPedantry
  - ParamValidationAndErrorResponse, [108](#)
- fSeedSuggestion
  - systtools::ISystProviderTool, [74](#)
- fSmallWeight
  - ParamValidationAndErrorResponse, [109](#)
- fSplineMode
  - SystToolsResponseTreeMaker, [130](#)
- fSystMetaData
  - systtools::ISystProviderTool, [74](#)
- fToolType
  - systtools::ISystProviderTool, [75](#)
- fTweak
  - SystToolsResponseTreeMaker, [130](#)
- fcName
  - cliopts, [26](#)
- FF
  - MD5, [78](#)
- fhicl, [27](#)
- fhicl\_key
  - cliopts, [26](#)
- file
  - systtools::PrecalculatedResponseReader, [115](#)
- Fill
  - SystToolsEventResponseTree, [125](#)
- finalize
  - MD5, [78](#)
- FinalizeAndValidateDependentParameters
  - systtools, [35](#)
- finalized
  - MD5, [81](#)
- FindISystProvider.cc
  - HandleOpts, [132](#)
  - main, [132](#)
  - SayUsage, [132](#)
- FullOfUnity
  - systtools, [35](#)
- G
  - MD5, [78](#)
- GenerateSystProviderConfig.cc
  - HandleOpts, [133](#)
  - main, [133](#)
  - ReadParameterSet, [133](#)
  - SayUsage, [133](#)
- GetAllDiscreteResponses
  - systtools::ParamHeaderHelper, [86](#)
- GetDiscreteResponse
  - systtools::ParamHeaderHelper, [87](#)
- GetDiscreteResponses
  - systtools::ParamHeaderHelper, [88](#), [89](#)
- GetDiscreteVariationParameterValues
  - systtools::ParamHeaderHelper, [89](#)
- GetEntries
  - systtools::PrecalculatedResponseReader, [113](#)
- GetEventLateralResponse
  - systtools::EventSplineCache< event\_unit\_t, C↵  
Ltight, typename std::enable\_if< CLtight==↵
- ParamValidationAndErrorResponse::kFrog, void >::type >, [55](#)
- systtools::EventSplineCache< event\_unit\_t, C↵  
Ltight, typename std::enable\_if< CLtight==↵  
ParamValidationAndErrorResponse::kHare, void >::type >, [56](#), [57](#)
- systtools::EventSplineCache< event\_unit\_t, C↵  
Ltight, typename std::enable\_if< CLtight==↵  
ParamValidationAndErrorResponse::k↵  
Tortoise, void >::type >, [58](#)
- GetEventResponse
  - CorrelatedMultisimProvider, [50](#)
  - ExampleISystProvider, [66](#)
  - systtools::ISystProviderTool, [71](#)
  - systtools::PrecalculatedResponseReader, [113](#)
- GetEventResponseInfo
  - systtools::ParamHeaderHelper, [90](#)
- GetEventUnit
  - systtools::EventSplineCacheBase, [62](#)
- GetEventWeightResponse
  - systtools::EventSplineCache< event\_unit\_t, C↵  
Ltight, typename std::enable\_if< CLtight==↵  
ParamValidationAndErrorResponse::kFrog, void >::type >, [55](#)
  - systtools::EventSplineCache< event\_unit\_t, C↵  
Ltight, typename std::enable\_if< CLtight==↵  
ParamValidationAndErrorResponse::kHare, void >::type >, [57](#)
  - systtools::EventSplineCache< event\_unit\_t, C↵  
Ltight, typename std::enable\_if< CLtight==↵  
ParamValidationAndErrorResponse::k↵  
Tortoise, void >::type >, [58](#), [59](#)
- GetExampleToolConfiguration
  - systtools::ISystProviderTool, [71](#)
- GetExtraToolOptions
  - ExampleISystProvider, [66](#)
  - systtools::ISystProviderTool, [71](#)
- GetFullyQualifiedName
  - systtools::ISystProviderTool, [71](#)
- GetHeader
  - systtools::ParamHeaderHelper, [90](#)
- GetHeaderId
  - systtools::ParamHeaderHelper, [90](#)
- GetHeaderInfo
  - systtools::ParamHeaderHelper, [90](#)
- GetHeaders
  - systtools::ParamHeaderHelper, [90](#)
- GetHistogram
  - ROOTUtility.hh, [157](#), [158](#)
- GetInstanceName
  - systtools::ISystProviderTool, [71](#)
- GetLateralResponse
  - ExampleISystProvider\_tool.cc, [146](#)
- GetNDiscreteVariations
  - systtools::ParamHeaderHelper, [91](#)
- GetNEventsInCache
  - systtools::EventSplineCacheBase, [62](#)
- GetNVariations

- systools::ISystProviderTool, [72](#)
- GetNormResponse
  - ExampleISystProvider\_tool.cc, [146](#)
- GetParam
  - systools, [35](#), [36](#)
- GetParamContainerIndex
  - systools, [36](#)
- GetParamElementFromContainer
  - systools, [36](#)
- GetParamId
  - systools, [37](#)
- GetParamIndex
  - systools, [37](#)
- GetParamShift\_nu
  - ExampleISystProvider\_tool.cc, [146](#)
- GetParamValue\_nu
  - ExampleISystProvider\_tool.cc, [146](#)
- GetParameterHeadersDocument
  - systools::ISystProviderTool, [72](#)
- GetParameterId
  - systools::ISystProviderTool, [72](#)
- GetParameterLowLimit
  - systools::ParamHeaderHelper, [91](#)
- GetParameterResponse
  - systools::ParamHeaderHelper, [91](#), [92](#)
- GetParameterUpLimit
  - systools::ParamHeaderHelper, [92](#)
- GetParameters
  - systools::ParamHeaderHelper, [92](#)
- GetPolyFitCoeffs
  - ROOTUtility.hh, [158](#)
- GetPolyResponse
  - systools::ParamHeaderHelper, [92](#)
- GetResponse
  - ExampleISystProvider\_tool.cc, [146](#)
- GetResponse\_nu
  - ExampleISystProvider\_tool.cc, [147](#)
- GetResponse\_shift
  - ExampleISystProvider\_tool.cc, [147](#)
- GetResponseParamId
  - systools::ParamHeaderHelper, [93](#)
- GetSpline
  - systools::ParamHeaderHelper, [93](#), [94](#)
- GetSplines
  - systools::ParamHeaderHelper, [94](#)
- GetSystMetaData
  - systools::ISystProviderTool, [72](#)
- GetToolType
  - systools::ISystProviderTool, [72](#)
- GetTotalEventWeightResponse
  - systools::EventSplineCache< event\_unit\_t, C↔  
Ltight, typename std::enable\_if< CLtight==↔  
ParamValidationAndErrorResponse::kFrog,  
void >::type >, [56](#)
  - systools::EventSplineCache< event\_unit\_t, C↔  
Ltight, typename std::enable\_if< CLtight==↔  
ParamValidationAndErrorResponse::kHare,  
void >::type >, [57](#)
- systools::EventSplineCache< event\_unit\_t, C↔  
Ltight, typename std::enable\_if< CLtight==↔  
ParamValidationAndErrorResponse::k↔  
Tortoise, void >::type >, [59](#)
- GetTotalResponse
  - systools::ParamHeaderHelper, [95](#)
- GG
  - MD5, [78](#)
- H
  - MD5, [79](#)
- HandleOpts
  - FindISystProvider.cc, [132](#)
  - GenerateSystProviderConfig.cc, [133](#)
- HasAnyParams
  - systools, [37](#)
- HasParam
  - systools, [37](#), [38](#)
- HasParameterLimits
  - systools::ParamHeaderHelper, [95](#)
- HasParameterLowLimit
  - systools::ParamHeaderHelper, [96](#)
- HasParameterUpLimit
  - systools::ParamHeaderHelper, [96](#)
- HaveHeader
  - systools::ParamHeaderHelper, [96](#)
- Header
  - systools::ParamHeaderProviderName, [101](#)
- hexdigest
  - MD5, [79](#)
- HH
  - MD5, [79](#)
- I
  - MD5, [79](#)
- ISystProviderTool
  - systools::ISystProviderTool, [69](#)
- ids
  - systools::PrecalculatedResponseReader, [115](#)
- II
  - MD5, [79](#)
- IndexIsHandled
  - systools, [38](#)
- init
  - MD5, [80](#)
- isCorrection
  - systools::SystParamHeader, [120](#)
- IsFlowBin
  - ROOTUtility.hh, [158](#)
- IsInHistogramRange
  - ROOTUtility.hh, [158](#)
- isRandomlyThrown
  - systools::SystParamHeader, [120](#)
- IsResponselessParam
  - systools::ParamHeaderHelper, [97](#)
- isResponselessParam
  - systools::SystParamHeader, [120](#)
- IsSplineParam
  - systools::ParamHeaderHelper, [97](#)

- isSplineable
  - systtools::SystParamHeader, 120
- IsThrownParam
  - systtools::ParamHeaderHelper, 97
- IsWeightResponse
  - systtools::ParamHeaderHelper, 97
- isWeightSystematicVariation
  - systtools::SystParamHeader, 120
- kDefaultDouble
  - systtools, 47
- kParamUnhandled
  - systtools, 47
- kParamUnhandled< double >
  - systtools, 47
- KnowAboutParameter
  - systtools::EventSplineCacheBase, 62
- LMatrix
  - CovarianceThrower, 53
- lateralParams
  - systtools::EventSplineCacheBase, 64
- lookup\_policy
  - cliopts, 26
- MD5, 75
  - buffer, 81
  - count, 81
  - decode, 77
  - digest, 81
  - encode, 77
  - F, 78
  - FF, 78
  - finalize, 78
  - finalized, 81
  - G, 78
  - GG, 78
  - H, 79
  - hexdigest, 79
  - HH, 79
  - I, 79
  - II, 79
  - init, 80
  - MD5, 77
  - operator<<, 81
  - rotate\_left, 80
  - size\_type, 76
  - state, 81
  - transform, 80
  - uint1, 76
  - uint4, 76
  - update, 80
- main
  - FindISystProvider.cc, 132
  - GenerateSystProviderConfig.cc, 133
- MakeBranches
  - SystToolsEventResponseTree, 125
- MakeFHiCLDefinedRandomVariations
  - systtools, 38
- MakeTreeWriter
  - systtools::PrecalculatedResponseReader, 113
- md5
  - md5.cc, 153
  - md5.hh, 154
- md5.cc
  - md5, 153
  - operator<<, 153
  - S11, 151
  - S12, 151
  - S13, 151
  - S14, 152
  - S21, 152
  - S22, 152
  - S23, 152
  - S24, 152
  - S31, 152
  - S32, 152
  - S33, 152
  - S34, 153
  - S41, 153
  - S42, 153
  - S43, 153
  - S44, 153
- md5.hh
  - md5, 154
- msg
  - systtools::systematicstools\_except, 117
- msgstrm
  - systtools::systematicstools\_except, 117
- NCoeffs
  - systtools::PrecalculatedResponseReader, 115
- NEW\_SYSTTOOLS\_EXCEPT
  - exceptions.hh, 149
  - ROOTUtility.hh, 158
  - SystToolsEventResponse, 124
  - systtools, 38–42
  - systtools::PrecalculatedResponseReader, 114
- NIds
  - systtools::PrecalculatedResponseReader, 116
- NRows
  - CovarianceThrower, 53
- nullheader
  - systtools::ParamHeaderHelper, 100
- oneSigmaShifts
  - systtools::SystParamHeader, 121
- operator<<
  - MD5, 81
  - md5.cc, 153
  - systtools::systematicstools\_except, 117
- operator=
  - SystToolsEventResponse, 124
  - SystToolsResponseTreeMaker, 129
- opts
  - systtools::SystParamHeader, 121
- outputfile
  - cliopts, 26

- param\_header\_map\_t
  - systtools, 31
- param\_list\_t
  - systtools, 31
- param\_tspline\_map\_t
  - systtools::ParamHeaderHelper, 85
- param\_value\_list\_t
  - systtools, 31
- param\_values
  - SystToolsEventResponseTree, 127
- ParamHeaderHelper
  - systtools::ParamHeaderHelper, 85
- paramId\_t
  - systtools, 31
- ParamsHandled
  - systtools::ISystProviderTool, 72
- ParamValidationAndErrorResponse, 104
  - CareLevel, 105
  - CheckResponse, 106
  - ErrorResponseLevel, 105
  - fAllowNegativeWeights, 107
  - fCare, 107
  - fErrorResponse, 108
  - fLargeWeight, 108
  - fPedantry, 108
  - fSmallWeight, 109
  - ParamValidationAndErrorResponse, 106
  - PedantLevel, 105
  - SetAllowNegativeWeights, 106
  - SetCareLevel, 106
  - SetErrorResponseLevel, 107
  - SetLargeWeightBoundary, 107
  - SetPedantLevel, 107
  - SetSmallWeightBoundary, 107
- paramValidityRange
  - systtools::SystParamHeader, 121
- paramVariations
  - systtools::SystParamHeader, 121
- parameter\_throws\_list\_t
  - systtools, 31
- ParameterAffectsEventLateral
  - systtools::EventSplineCacheBase, 63
- ParameterAffectsEventWeight
  - systtools::EventSplineCacheBase, 63
- ParseFHiCLSimpleToolConfigurationParameter
  - systtools, 42
- ParseFHiCLVariationDescriptor
  - systtools, 42
- ParseToVect
  - systtools, 43
- PedantLevel
  - ParamValidationAndErrorResponse, 105
- pid
  - systtools::ParamResponses, 103
  - systtools::ParamThrows, 104
  - systtools::ParamValue, 109
  - systtools::PrecalculatedResponseReader::↔
    - ParamPolyResponses, 102
- PolyResponse
  - systtools::PolyResponse, 110, 111
- PrecalculatedResponseReader
  - systtools::PrecalculatedResponseReader, 112
- prettyName
  - systtools::SystParamHeader, 121
- produce
  - SystToolsEventResponse, 124
- producer\_name
  - cliopts, 26
- provider\_list\_t
  - systtools, 32
- provider\_name
  - cliopts, 26
- ProviderFQName
  - systtools::ParamHeaderProviderName, 101
- quiet
  - cliopts, 26
- README.md, 131
- RNJesus
  - CorrelatedMultisimProvider, 51
  - CovarianceThrower, 53
  - ExampleISystProvider, 67
- RNgin
  - CorrelatedMultisimProvider, 50
  - CovarianceThrower, 53
  - ExampleISystProvider, 67
- ROOTUtility.hh
  - CheckOpenFile, 157
  - GetHistogram, 157, 158
  - GetPolyFitCoeffs, 158
  - IsFlowBin, 158
  - IsInHistogramRange, 158
  - NEW\_SYSTTOOLS\_EXCEPT, 158
- RVector
  - CovarianceThrower, 53
- ReadParameterSet
  - GenerateSystProviderConfig.cc, 133
- resp
  - systtools::PrecalculatedResponseReader::↔
    - ParamPolyResponses, 102
- responseParamId
  - systtools::SystParamHeader, 122
- responses
  - systtools::ParamResponses, 103
  - systtools::SystParamHeader, 122
- rotate\_left
  - MD5, 80
- S11
  - md5.cc, 151
- S12
  - md5.cc, 151
- S13
  - md5.cc, 151
- S14
  - md5.cc, 152

- S21
  - md5.cc, [152](#)
- S22
  - md5.cc, [152](#)
- S23
  - md5.cc, [152](#)
- S24
  - md5.cc, [152](#)
- S31
  - md5.cc, [152](#)
- S32
  - md5.cc, [152](#)
- S33
  - md5.cc, [152](#)
- S34
  - md5.cc, [153](#)
- S41
  - md5.cc, [153](#)
- S42
  - md5.cc, [153](#)
- S43
  - md5.cc, [153](#)
- S44
  - md5.cc, [153](#)
- SayUsage
  - FindISystProvider.cc, [132](#)
  - GenerateSystProviderConfig.cc, [133](#)
- scratch\_discrete\_variation\_list\_t1
  - systtools::ParamHeaderHelper, [100](#)
- scratch\_spline\_t1
  - systtools::ParamHeaderHelper, [100](#)
- scratch\_spline\_t2
  - systtools::ParamHeaderHelper, [100](#)
- ScrubUnityEventResponses
  - systtools, [43, 44](#)
- SetAllowNegativeWeights
  - ParamValidationAndErrorResponse, [106](#)
  - systtools::ParamHeaderHelper, [97](#)
- SetBranchAddresses
  - systtools::PrecalculatedResponseReader, [114](#)
- SetCareLevel
  - ParamValidationAndErrorResponse, [106](#)
  - systtools::ParamHeaderHelper, [98](#)
- SetChkErr
  - systtools::EventSplineCacheBase, [63](#)
  - systtools::ParamHeaderHelper, [98](#)
- SetErrorResponseLevel
  - ParamValidationAndErrorResponse, [107](#)
  - systtools::ParamHeaderHelper, [98](#)
- SetEvent
  - SystToolsEventResponseTree, [126](#)
- SetHeaders
  - systtools::EventSplineCacheBase, [63](#)
  - systtools::ParamHeaderHelper, [98, 99](#)
- SetLargeWeightBoundary
  - ParamValidationAndErrorResponse, [107](#)
  - systtools::ParamHeaderHelper, [99](#)
- SetParamResponse
  - SystToolsEventResponseTree, [126](#)
- SetParameterValue
  - systtools::EventSplineCacheBase, [64](#)
- SetParametersValue
  - systtools::EventSplineCacheBase, [63](#)
- SetPedantLevel
  - ParamValidationAndErrorResponse, [107](#)
  - systtools::ParamHeaderHelper, [99](#)
- SetSmallWeightBoundary
  - ParamValidationAndErrorResponse, [107](#)
  - systtools::ParamHeaderHelper, [99](#)
- SetThrow
  - SystToolsEventResponseTree, [126](#)
- SetTotalWeight
  - SystToolsEventResponseTree, [126](#)
- SetTree
  - SystToolsEventResponseTree, [126](#)
- SetupDecomp
  - CovarianceThrower, [52](#)
- SetupResponseCalculator
  - ExampleISystProvider, [67](#)
  - systtools::ISystProviderTool, [73](#)
- size\_type
  - MD5, [76](#)
- sp\_config\_hash
  - SystToolsEventResponse, [124](#)
- spline\_t
  - systtools::ParamHeaderHelper, [85](#)
- state
  - MD5, [81](#)
- str2T< bool >
  - systtools, [44](#)
- str2T
  - systtools, [44](#)
- SuggestParameterThrows
  - systtools::ISystProviderTool, [73](#)
- SuggestSeed
  - systtools::ISystProviderTool, [73](#)
- syst\_providers
  - SystToolsEventResponse, [124](#)
- SystGetOptKV
  - systtools, [44](#)
- SystHasOpt
  - systtools, [45](#)
- SystHasOptKV
  - systtools, [45](#)
- SystMetaData
  - systtools, [32](#)
- SystParamHeader
  - systtools::SystParamHeader, [119](#)
- SystParamHeaderToFHiCL
  - systtools, [45](#)
- systParamId
  - systtools::SystParamHeader, [122](#)
- SystToolsEventResponse, [123](#)
- NEW\_SYSTTOOLS\_EXCEPT, [124](#)
- operator=, [124](#)
- produce, [124](#)



- sp\_config\_hash, 124
- syst\_providers, 124
- SystToolsEventResponse, 123
- SystToolsEventResponseTree, 125
  - event, 126
  - event\_responses, 127
  - Fill, 125
  - MakeBranches, 125
  - param\_values, 127
  - SetEvent, 126
  - SetParamResponse, 126
  - SetThrow, 126
  - SetTotalWeight, 126
  - SetTree, 126
  - SystToolsEventResponseTree, 125
  - t\_it, 127
  - total\_weight, 127
  - tree, 127
- SystToolsResponseTreeMaker, 127
  - analyze, 128
  - configuredParameterHeaders, 129
  - fEventHelper, 129
  - fHeaderHelper, 129
  - fInpTag, 129
  - fOutputTree, 129
  - fSplineMode, 130
  - fTweak, 130
  - operator=, 129
  - SystToolsResponseTreeMaker, 128
- systematicstools/app/CheckSystProviderConfigmd5.cc, 131
- systematicstools/app/FindISystProvider.cc, 131
- systematicstools/app/GenerateSystProviderConfig.cc, 132
- systematicstools/doc/ExampleSystProvider.md, 134
- systematicstools/doc/MovingParts.md, 134
- systematicstools/doc/ParameterHeaders.md, 134
- systematicstools/doc/ToolConfiguration.md, 134
- systematicstools/doc/WritingAProvider.md, 134
- systematicstools/interface/EventResponse\_product.cc, 134
- systematicstools/interface/EventResponse\_product.hh, 134
- systematicstools/interface/FHiCLSystParamHeader↔
  - Converters.cc, 135
- systematicstools/interface/FHiCLSystParamHeader↔
  - Converters.hh, 136
- systematicstools/interface/ISystProviderTool.cc, 136
- systematicstools/interface/ISystProviderTool.hh, 136
- systematicstools/interface/SystMetaData.cc, 137
- systematicstools/interface/SystMetaData.hh, 138
- systematicstools/interface/SystParamHeader.cc, 139
- systematicstools/interface/SystParamHeader.hh, 139
- systematicstools/interface/types.hh, 140
- systematicstools/interpreters/EventSplineCache↔
  - Helper.hh, 141
- systematicstools/interpreters/ParamHeaderHelper.cc, 142
- systematicstools/interpreters/ParamHeaderHelper.hh, 142
- systematicstools/interpreters/ParamValidationAnd↔
  - ErrorResponse.cc, 142
- systematicstools/interpreters/ParamValidationAnd↔
  - ErrorResponse.hh, 143
- systematicstools/interpreters/PolyResponse.hh, 143
- systematicstools/interpreters/PrecalculatedResponse↔
  - Reader.hh, 143
- systematicstools/module/SystToolsEventResponse↔
  - module.cc, 144
- systematicstools/module/SystToolsResponseTree↔
  - Maker\_module.cc, 144
- systematicstools/module/classes.h, 144
- systematicstools/systproviders/CorrelatedMultisim↔
  - Provider\_tool.cc, 145
- systematicstools/systproviders/ExampleISystProvider↔
  - \_tool.cc, 145
- systematicstools/systproviders/ExampleISystProvider↔
  - \_tool.hh, 148
- systematicstools/utility/CovMatThrower.cc, 148
- systematicstools/utility/CovMatThrower.hh, 148
- systematicstools/utility/FHiCLSystParamHeader↔
  - Utility.cc, 149
- systematicstools/utility/FHiCLSystParamHeader↔
  - Utility.hh, 150
- systematicstools/utility/ParameterAndProviderConfiguration↔
  - Utility.cc, 154
- systematicstools/utility/ParameterAndProviderConfiguration↔
  - Utility.hh, 155
- systematicstools/utility/ROOTUtility.hh, 157
- systematicstools/utility/ResponselessParamUtility.cc, 156
- systematicstools/utility/ResponselessParamUtility.hh, 156
- systematicstools/utility/exceptions.hh, 148
- systematicstools/utility/md5.cc, 151
- systematicstools/utility/md5.hh, 154
- systematicstools/utility/printers.hh, 156
- systematicstools/utility/string\_parsers.hh, 159
- systematicstools\_except
  - systools::systematicstools\_except, 117
- systools, 27
  - AppendVect, 32
  - BuildParameterHeaders, 32
  - ConfigureISystProvidersFromParameterHeaders, 33
  - ConfigureISystProvidersFromToolConfig, 33
  - ContainerHasParam, 33
  - event\_unit\_response\_t, 30
  - eventId\_t, 30
  - EventResponse, 31
  - ExtendEventResponse, 34
  - ExtendSystMetaData, 34
  - FHiCLSimpleToolConfigurationParameterExists, 34
  - FHiCLToSystParamHeader, 34
  - FinalizeAndValidateDependentParameters, 35



- FullOfUnity, 35
- GetParam, 35, 36
- GetParamContainerIndex, 36
- GetParamElementFromContainer, 36
- GetParamId, 37
- GetParamIndex, 37
- HasAnyParams, 37
- HasParam, 37, 38
- IndexIsHandled, 38
- kDefaultDouble, 47
- kParamUnhandled, 47
- kParamUnhandled< double >, 47
- MakeFHiCLDefinedRandomVariations, 38
- NEW\_SYSTTOOLS\_EXCEPT, 38–42
- param\_header\_map\_t, 31
- param\_list\_t, 31
- param\_value\_list\_t, 31
- paramId\_t, 31
- parameter\_throws\_list\_t, 31
- ParseFHiCLSimpleToolConfigurationParameter, 42
- ParseFHiCLVariationDescriptor, 42
- ParseToVect, 43
- provider\_list\_t, 32
- ScrubUnityEventResponses, 43, 44
- str2T< bool >, 44
- str2T, 44
- SystGetOptKV, 44
- SystHasOpt, 45
- SystHasOptKV, 45
- SystMetaData, 32
- SystParamHeaderToFHiCL, 45
- to\_str, 45, 46
- Validate, 46
- systools::EventSplineCache< event\_unit\_t, CLtight, Enable >, 54
- systools::EventSplineCache< event\_unit\_t, CLtight, typename std::enable\_if< CLtight==Param↵ ValidationAndErrorResponse::kFrog, void >↵ ::type >, 54
- GetEventLateralResponse, 55
- GetEventWeightResponse, 55
- GetTotalEventWeightResponse, 56
- systools::EventSplineCache< event\_unit\_t, CLtight, typename std::enable\_if< CLtight==Param↵ ValidationAndErrorResponse::kHare, void >↵ ::type >, 56
- GetEventLateralResponse, 56, 57
- GetEventWeightResponse, 57
- GetTotalEventWeightResponse, 57
- systools::EventSplineCache< event\_unit\_t, CLtight, typename std::enable\_if< CLtight==Param↵ ValidationAndErrorResponse::kTortoise, void >↵ ::type >, 58
- GetEventLateralResponse, 58
- GetEventWeightResponse, 58, 59
- GetTotalEventWeightResponse, 59
- systools::EventSplineCacheBase
- CacheEvent, 61
- CacheEvents, 61
- currentValues, 64
- DeclareUsingParameter, 62
- DeclareUsingParameters, 62
- event\_t, 60
- EventSplineCacheBase, 60, 61
- fChkErr, 64
- fEvents, 64
- fHeaderHelper, 64
- GetEventUnit, 62
- GetNEventsInCache, 62
- KnowAboutParameter, 62
- lateralParams, 64
- ParameterAffectsEventLateral, 63
- ParameterAffectsEventWeight, 63
- SetChkErr, 63
- SetHeaders, 63
- SetParameterValue, 64
- SetParametersValue, 63
- weightParams, 65
- systools::EventSplineCacheBase< event\_unit\_t >, 59
- systools::ISystProviderTool, 68
- ~ISystProviderTool, 69
- AsString, 69
- BuildSystMetaData, 70
- CheckHaveMetaData, 70
- ConfigureFromParameterHeaders, 70
- ConfigureFromToolConfig, 70
- fQName, 74
- fHaveSystMetaData, 74
- fInstanceName, 74
- fIsFullyConfigured, 74
- fSeedSuggestion, 74
- fSystMetaData, 74
- fToolType, 75
- GetEventResponse, 71
- GetExampleToolConfiguration, 71
- GetExtraToolOptions, 71
- GetFullyQualifiedName, 71
- GetInstanceName, 71
- GetNVariations, 72
- GetParameterHeadersDocument, 72
- GetParameterId, 72
- GetSystMetaData, 72
- GetToolType, 72
- ISystProviderTool, 69
- ParamsIsHandled, 72
- SetupResponseCalculator, 73
- SuggestParameterThrows, 73
- SuggestSeed, 73
- systools::ParamHeaderHelper, 82
- CheckParamList, 86
- CheckParamValueList, 86
- discrete\_variation\_list\_t, 85
- fChkErr, 100
- fHeaders, 100
- GetAllDiscreteResponses, 86
- GetDiscreteResponse, 87

- GetDiscreteResponses, [88, 89](#)
- GetDiscreteVariationParameterValues, [89](#)
- GetEventResponseInfo, [90](#)
- GetHeader, [90](#)
- GetHeaderId, [90](#)
- GetHeaderInfo, [90](#)
- GetHeaders, [90](#)
- GetNDiscreteVariations, [91](#)
- GetParameterLowLimit, [91](#)
- GetParameterResponse, [91, 92](#)
- GetParameterUpLimit, [92](#)
- GetParameters, [92](#)
- GetPolyResponse, [92](#)
- GetResponseParamId, [93](#)
- GetSpline, [93, 94](#)
- GetSplines, [94](#)
- GetTotalResponse, [95](#)
- HasParameterLimits, [95](#)
- HasParameterLowLimit, [96](#)
- HasParameterUpLimit, [96](#)
- HaveHeader, [96](#)
- IsResponselessParam, [97](#)
- IsSplineParam, [97](#)
- IsThrownParam, [97](#)
- IsWeightResponse, [97](#)
- nullheader, [100](#)
- param\_tspline\_map\_t, [85](#)
- ParamHeaderHelper, [85](#)
- scratch\_discrete\_variation\_list\_t1, [100](#)
- scratch\_spline\_t1, [100](#)
- scratch\_spline\_t2, [100](#)
- SetAllowNegativeWeights, [97](#)
- SetCareLevel, [98](#)
- SetChkErr, [98](#)
- SetErrorResponseLevel, [98](#)
- SetHeaders, [98, 99](#)
- SetLargeWeightBoundary, [99](#)
- SetPedantLevel, [99](#)
- SetSmallWeightBoundary, [99](#)
- spline\_t, [85](#)
- ValuesAreInNaturalUnits, [99](#)
- systools::ParamHeaderProviderName, [101](#)
  - Header, [101](#)
  - ProviderFQName, [101](#)
- systools::ParamResponses, [102](#)
  - pid, [103](#)
  - responses, [103](#)
- systools::ParamThrows, [103](#)
  - pid, [104](#)
  - thrown\_vals, [104](#)
- systools::ParamValue, [109](#)
  - pid, [109](#)
  - val, [110](#)
- systools::PolyResponse
  - eval, [111](#)
  - PolyResponse, [110, 111](#)
- systools::PolyResponse< n >, [110](#)
- systools::PrecalculatedResponseReader
  - AddEventResponses, [113](#)
  - AllocateVectors, [113](#)
  - coeffs\_1D, [115](#)
  - fHeaders, [115](#)
  - file, [115](#)
  - GetEntries, [113](#)
  - GetEventResponse, [113](#)
  - ids, [115](#)
  - MakeTreeWriter, [113](#)
  - NCoeffs, [115](#)
  - NEW\_SYSTTOOLS\_EXCEPT, [114](#)
  - NIds, [116](#)
  - PrecalculatedResponseReader, [112](#)
  - SetBranchAddresses, [114](#)
  - tree, [116](#)
- systools::PrecalculatedResponseReader< Order >, [111](#)
- systools::PrecalculatedResponseReader< Order >::ParamPolyResponses, [102](#)
- systools::PrecalculatedResponseReader::ParamPolyResponses
  - pid, [102](#)
  - resp, [102](#)
- systools::SystParamHeader, [118](#)
  - centralParamValue, [119](#)
  - differsEventByEvent, [119](#)
  - isCorrection, [120](#)
  - isRandomlyThrown, [120](#)
  - isResponselessParam, [120](#)
  - isSplineable, [120](#)
  - isWeightSystematicVariation, [120](#)
  - oneSigmaShifts, [121](#)
  - opts, [121](#)
  - paramValidityRange, [121](#)
  - paramVariations, [121](#)
  - prettyName, [121](#)
  - responseParamId, [122](#)
  - responses, [122](#)
  - SystParamHeader, [119](#)
  - systParamId, [122](#)
  - unitsAreNatural, [122](#)
- systools::systematictools\_except, [116](#)
  - msg, [117](#)
  - msgstrm, [117](#)
  - operator<<, [117](#)
  - systematictools\_except, [117](#)
  - what, [117](#)
- t\_it
  - SystToolsEventResponseTree, [127](#)
- Throw
  - CovarianceThrower, [53](#)
- thrown\_vals
  - systools::ParamThrows, [104](#)
- to\_str
  - systools, [45, 46](#)
- total\_weight
  - SystToolsEventResponseTree, [127](#)
- transform

- MD5, [80](#)
- tree
  - SystToolsEventResponseTree, [127](#)
  - systtools::PrecalculatedResponseReader, [116](#)
- uint1
  - MD5, [76](#)
- uint4
  - MD5, [76](#)
- UncertMatrix
  - CovarianceThrower, [54](#)
- unitsAreNatural
  - systtools::SystParamHeader, [122](#)
- update
  - MD5, [80](#)
- val
  - systtools::ParamValue, [110](#)
- Validate
  - systtools, [46](#)
- ValuesAreInNaturalUnits
  - systtools::ParamHeaderHelper, [99](#)
- weightParams
  - systtools::EventSplineCacheBase, [65](#)
- what
  - systtools::systematicstools\_except, [117](#)
- WrapWithPROLOG
  - cliopts, [26](#)