# Model Advisor Report – Heli\_inner\_loop.slx

Simulink version: 10.0 Model version: 1.32

System: Heli\_inner\_loop Current run: 13-May-2020 13:55:08

Treat as Referenced Model: on

### **Run Summary**

 Pass
 Fail
 Warning
 Not Run
 Total

 ✓ 182
 3
 0
 ■ 0
 182





-----

Display model version information

Display model configuration and checksum information.

#### Model configuration and checksum information

Attribute	Value
Model Version	1.32
Author	bpotter
Date	Mon May 11 09:09:03 2020
Model Checksum	230689666 257464667 495208736 2705372901





Check usage of Abs blocks

Identify Abs blocks that have unreachable code or produce overflows

#### **Passed**

No Abs blocks found causing unreachable code or produce overflows.

\_\_\_\_\_

Check usage of Math Function blocks (rem and reciprocal functions)

Identify Math Function blocks using rem and reciprocal functions that cause non-finite results

#### **Passed**

All Math Function blocks in the model use operators appropriately.

Check usage of Math Function blocks (log and log10 functions)

Identify Math Function blocks using log and log10 functions that cause non-finite results

#### **Passed**

All Math Function blocks in the model use operators appropriately.

Check usage of While Iterator blocks

Identify While Iterator blocks that do not have a positive value for the maximum number of iterations

#### **Passed**

No While Iterator blocks found that might cause infinite loops

**⊘** Check usage of For and While Iterator subsystems

Identify sample time-dependent blocks in While and For Iterator subsystems.

#### **Passed**

No sample time-dependent blocks in For or While Iterator subsystems.

Check usage of For Iterator blocks

Identify For Iterator blocks that cause variable loops

#### **Passed**

No For Iterator blocks found that cause variable loops.

Check usage of If blocks and If Action Subsystem blocks

Identify If and If Action Subsystem blocks without else conditions

#### **Passed**

No If blocks with questionable configurations or connections were found.

Check usage of Switch Case blocks and Switch Case Action Subsystem blocks

Identify inappropriately used Switch Case blocks and Switch Case Action Subsystem blocks

#### **Passed**

No Switch Case blocks with questionable configurations or connections were found.

**⊘** Check usage of conditionally executed subsystems

Identify inappropriate blocks in conditionally executed subsystems.

#### **Passed**

No blocks with improper sample times or asynchronously executed sample-time dependent blocks were found.

\_\_\_\_

Check usage of Merge blocks

Identify Merge blocks constructs which can lead to ambiguous behavior.

#### **Passed**

No merge blocks found which can lead to ambiguous behavior.

Check Relational Operator blocks equating floating-point types

Identify Relational Operator blocks that equate floating-point types

### **Passed**

No Relational Operator blocks found that equate floating-point types.

Check usage of Relational Operator blocks

Identify Relational Operator blocks that operate on different data types or have a non-boolean output

**Passed** 

No Relational Operator blocks found that operate on different data types or have a non-boolean output.

Check usage of Logical Operator blocks

Identify Logical Operator blocks that operate on non-boolean data types

#### **Passed**

No Logical Operator blocks found that operate on non-boolean data types.

Check usage of bit operation blocks

Identify bit operation blocks with signed data types as inputs

#### **Passed**

No bit operation blocks found with signed data types as inputs.

☑ Check for blocks not recommended for C/C++ production code deployment

Identify blocks not supported by code generation or not recommended for C/C++ production code deployment.

#### **Passed**

No blocks found which are not recommended for C/C++ production code deployment.

Check for inconsistent vector indexing methods

Identify inconsistent usage of vector indexing methods across the model or subsystem

#### **Passed**

No blocks found using inconsistent indexing modes.

Check data types for blocks with index signals

Identify blocks with index signals that have data types other than integers or enums.

#### **Passed**

No blocks or charts found with index signals or variables that have data types other than integer or enums.

Check usage of variant blocks

Check variant block settings that might result in code that doesn't trace back to requirements.

#### **Passed**

There are no variant blocks that have "Generate preprocessor conditionals" active.

Check usage of lookup table blocks

Check for Lookup Table blocks, Prelookup blocks and Interpolation blocks that do not generate out-of-range checking code.

#### **Passed**

No lookup table blocks found to not generate out-of-range checking code.

\_\_\_\_\_

Check usage of Signal Routing blocks

Identify usage of Signal Routing blocks in Simulink that might impact safety

#### **Passed**

No Switch blocks that might generate code with inequality operations (~=) in expressions where at least one side of the expression is a floating-point variable or constant were found.

**⊘** Check for root Inports with missing properties

Identify Inport blocks in the top-level of the model with missing or inherited sample times, data types, or port dimensions. Inport block properties are specified with block parameters or Simulink signal data objects that explicitly resolve to the connected signal lines.

#### **Passed**

There are no Inport blocks in the top-level of the model with missing or inherited sample times, data types, or port dimensions

Check for root Inports with missing range definitions

Identify root-level Inport blocks with missing or erroneous minimum or maximum values. Inport block minimum and maximum values are specified with block parameters or Simulink signal objects that explicitly resolve to the connected signal lines.

#### **Passed**

There are no missing or erroneous Inport range properties at the model root level.

\_\_\_\_\_

Check for root Outports with missing range definitions

Identify root-level Outport blocks with missing or erroneous minimum or maximum values. Outport block minimum and maximum values are specified with block parameters or Simulink signal objects that explicitly resolve to the connected signal lines.

#### **Passed**

There are no missing or erroneous Outport range properties at the model root level. Note: Root Outports with inherited data types are not analyzed by this check.

Check usage of Assignment blocks

Identify Assignment blocks whose array fields are not initialized.

#### Passed

All Assignment blocks are configured with block parameter "Action if any output element is not assigned" set to Warning or Error.

\_\_\_\_\_

**⊘** Check global variables in graphical functions

Identify expressions that both read and write to the same global data.

#### **Passed**

No expressions found that both read and write to the same global data.

Check usage of Gain blocks

Identify Gain blocks with value which resolves to 1

#### **Passed**

No Gain blocks found with value which resolves to 1.

Check for length of user-defined object names

Identify user-defined object names with length greater than threshold

#### **Passed**

No Subsystem blocks found with function name length greater than Maximum identifier length. There are no data objects with names having length greater than Maximum identifier length.

\_\_\_\_\_

Check data type of loop control variables

Identify loop control variables using non-integer data types.

No For Iterator blocks or MATLAB Function blocks found using non-integer data type for loop control counter variable.





Check state machine type of Stateflow charts

Identify Stateflow Charts whose State Machine Type differs from the type set in the Model Advisor Configuration Editor.

#### **Passed**

No Stateflow Charts found that deviate from recommended state machine type.

Check Stateflow charts for ordering of states and transitions

Identify Stateflow charts that do not use explicit ordering of parallel states and transitions.

#### **Passed**

No Stateflow Charts found that deviate from recommended state/transition execution order settings.

Check usage of bitwise operations in Stateflow charts

Identify usage of signed data type operands to bitwise operations in Stateflow charts.

#### **Passed**

No Stateflow objects found that use signed data type operands with bitwise operations.

Check for Strong Data Typing with Simulink I/O

Verify configuration settings for strong data typing on the boundaries between Simulink and Stateflow

#### **Passed**

No Stateflow charts found that set 'Use Strong Data Typing with Simulink I/O' to off.

Check Stateflow debugging options

Identify whether Stateflow debugging options are set appropriately

#### **Passed**

All Stateflow debugging options are set appropriately.

Check Stateflow charts for transition paths that cross parallel state boundaries Identify transition paths that cross parallel state boundaries in Stateflow charts.

#### **Passed**

No transition paths crossing parallel state boundaries were found in Stateflow charts.

Check for inappropriate use of transition paths

Identify transition paths that go into and out of a state without ending on a substate.

#### **Passed**

No transition paths found that go into and out of a state without ending on a substate.

Check Stateflow charts for strong data typing

Identify expressions with variables and parameters of different data types in Stateflow objects.

#### **Passed**

No expressions were found with variables and parameters of different data types.

Check naming of ports in Stateflow charts

Identify mismatches between names of Stateflow ports and associated signals

#### **Passed**

There are no name mismatches between Stateflow ports and associated signals

Check scoping of Stateflow data objects

Identify Stateflow data objects with local scope that are not scoped at the chart level or below

#### **Passed**

All Stateflow data objects are properly scoped.

Check Stateflow charts for uniquely defined data objects

Identify local data identifiers that are defined in multiple scopes within a chart.

#### **Passed**

No Stateflow data identifiers found to be defined in multiple scopes.

Check usage of shift operations for Stateflow data

Identify usage of Stateflow bit-shifting operations that might impact safety.

#### **Passed**

There are no Stateflow bit-shifting operations greater than the bit-width of the input or output type.

Check assignment operations in Stateflow charts

Identify assignment operations in Stateflow objects which cast integer and fixed-point calculations to wider datatype.

#### **Passed**

No assignment operations were found which cast integer and fixed-point calculations to wider datatype.

Check Stateflow charts for unary operators

Identify unary minus operators on unsigned data types in Stateflow objects.

#### **Passed**

No unary minus operations on unsigned data types were found in Stateflow objects.

amatlab ❷9❷0▲0Ⅲ0

Check usage of standardized MATLAB function headers

Identify usage of standardized function headers in MATLAB function.

#### **Passed**

No MATLAB function blocks found without standardized function headers.

Check for MATLAB Function interfaces with inherited properties

Identify MATLAB Functions that have inputs, outputs, or parameters with inherited complexity or data type properties.

#### **Passed**

No MATLAB Function interfaces with inherited complexity or data type properties found.

Check MATLAB Function metrics

Identify MATLAB Functions that violate code and complexity metrics.

#### **Passed**

No MATLAB Function blocks found that violate code and complexity metrics.

✓ Check MATLAB Code Analyzer messages

Check MATLAB functions for %#codegen directive, MATLAB Code Analyzer messages, and justification message IDs.

#### **Passed**

No MATLAB Function blocks found with Code Analyzer messages, missing %#codegen directive or inappropriate usage of justification message IDs.

\_\_\_\_\_

Check if/elseif/else patterns in MATLAB Function blocks

Identify if/elseif/else patterns without appropriate else conditions in embedded MATLAB code

#### **Passed**

No inappropriate if/elseif/else patterns found.

Check switch statements in MATLAB Function blocks

Identify inappropriately used switch statements in embedded MATLAB code

#### **Passed**

No inappropriately used switch statements found.

Check usage of relational operators in MATLAB Function blocks

Identify relational operators operating on operands of different data types in MATLAB Function blocks.

#### **Passed**

No relational operators found operating on operands of different data types.

Check usage of equality operators in MATLAB Function blocks

Identify equality operators used with floating-point operands in MATLAB Function blocks.

#### **Passed**

No equality operators found operating on floating-point operands.

\_\_\_\_\_

Check usage of logical operators and functions in MATLAB Function blocks

Identify logical operators and functions operating on operands with numerical data types.

#### **Passed**

No logical operators or functions found operating on operands with numerical data types.

Check safety-related diagnostic settings for data store memory

Check diagnostic settings in the model configuration that apply to data store memory and might impact safety.

#### **Passed**

	Parameter	Current Value	Recommended
Status			Values
Pass	Detect read before write		EnableAllAsError
	(ReadBeforeWriteMsg)	EnableAllAsError	
Pass	Detect write after read (WriteAfterReadMsg)		EnableAllAsError
		EnableAllAsError	
Pass	Detect write after write (WriteAfterWriteMsg)		EnableAllAsError
		EnableAllAsError	
Pass	Multitask data store (MultiTaskDSMMsg)	error	error

Pass	Duplicate data store names	error	error
	(UniqueDataStoreMsg)		

\_\_\_\_\_

Check safety-related diagnostic settings for saving

Check diagnostic settings in the model configuration that apply to saving model files.

#### **Passed**

All constraints on model configuration parameters have been met.

Status	Parameter	Current Value	Recommended Values
Pass	Block diagram contains disabled library links (SaveWithDisabledLinksMsg)	error	error
Pass	Block diagram contains parameterized library links (SaveWithParameterizedLinksMsg)	error	error

✓ Check safety-related model referencing settings

Check model referencing settings in the model configuration that might impact safety.

#### **Passed**

Stat us	Parameter	Current Value	Recommended Values
Pass	Rebuild (UpdateModelReferenceTargets)	IfOutOfDateOrStructuralC hange	Assume Up To Date, If Out Of Date Or Structural Change
Pass	Pass fixed-size scalar root inputs by value for code generation (ModelReferencePassRootInputsByR eference) *	on	on
Pass	Minimize algebraic loop occurrences (ModelReferenceMinAlgLoopOccurre nces)	off	off

#### **Recommended Action**

Check safety-related code generation settings for comments

Check code generation settings in the model configuration that apply comments and might impact safety.

#### **Passed**

<sup>\*</sup> The Command-Line values provided in the table are reverse of the settings in the Configuration Parameters Dialog. Therefore, 'on' in the Command-Line corresponds to an "Off" setting in the dialog, and 'off' in the Command-Line corresponds to an "On" setting in the dialog.

Status	Parameter	Current Value	Recommended Values	Prerequisites
Status		Value	Values	
Pass	Include comments	on	on	
	(GenerateComments)			
Pass	Simulink block comments	on	on	GenerateComments
	(SimulinkBlockComments)			
Pass	Show eliminated blocks	on	on	GenerateComments
	(ShowEliminatedStatement)			
D -	System target file	ERT	ERT based	
Pass	(SystemTargetFile)	based	target	
		target		
Pass	Verbose comments for 'Model	on	on	GenerateComments
	default' storage class			
	(ForceParamTrailComments)			
D -	Include comments	on	on	
Pass	(GenerateComments)			
Pass	Requirements in block comments	on	on	SystemTargetFile,
	(ReqsInCode)			GenerateComments



**⊘** Check safety-related code generation interface settings

Check code generation interface settings in the model configuration that might impact safety.

#### **Passed**

Status	Parameter	Current Value	Recommended Values	Prerequisites
Pass	non-finite numbers (SupportNonFinite)	off	off	

Pass	absolute time (SupportAbsoluteTime)	off	off	
				SystemTargetFile
Pass	continuous time	off	off	
	(SupportContinuousTime)			SystemTargetFile
D -	System target file (SystemTargetFile)	ERT	ERT based	
Pass		based	target	
		target		
Pass	non-inlined S-functions	off	off	
	(SupportNonInlinedSFcns)			SystemTargetFile
Pass	Classic call interface (GRTInterface)	off	off	
Pass	Single output/update function	on	on	
	(CombineOutputUpdateFcns)			
Pass	Terminate function required	off	off	
	(IncludeMdlTerminateFcn)			SystemTargetFile
Pass	Remove error status field in real-time	on	on	
	model data structure			SystemTargetFile
	(SuppressErrorStatus)			
Pass	MAT-file logging (MatFileLogging)	off	off	

☑ Check safety-related solver settings for simulation time

Identify if the model Start time is set to 0 and Stop time is less than the Application Life Span.

#### **Passed**

No issues found with solver settings for simulation time.

**⊘** Check safety-related solver settings for solver options

Check solver settings in the model configuration that apply to solvers and might impact safety.

#### **Passed**

Status	Parameter	<b>Current Value</b>	Recommended Values
Pass	Type (SolverType)	Fixed-step	Fixed-step
Pass	Solver (SolverName)	FixedStepDiscrete	FixedStepDiscrete

\_\_\_\_\_

Check safety-related solver settings for tasking and sample-time

Check solver settings in the model configuration that apply to tasking and sample-time constraints and might impact safety.

#### **Passed**

All constraints on model configuration parameters have been met.

Status	Parameter	Current Value	Not Recommended Values
Pass	Automatically handle rate transition for data transfer (AutoInsertRateTranBlk)	off	on

\_\_\_\_\_

Check safety-related diagnostic settings for solvers

Check diagnostic settings in the model configuration that apply to solvers and might impact safety.

#### **Passed**

	Parameter	Current	Recommended
Status		Value	Values
Pass	Algebraic loop (AlgebraicLoopMsg)	error	error
Pass	Minimize algebraic loop (ArtificialAlgebraicLoopMsg)	error	error
Pass	Block priority violation (BlockPriorityViolationMsg)	error	error
Pass	Automatic solver parameter selection (SolverPrmCheckMsg)	error	error
Pass	State name clash (StateNameClashWarn)	warning	warning

✓ Check safety-related diagnostic settings for sample time

Check diagnostic settings in the model configuration that apply to sample time and might impact safety.

### **Passed**

	Parameter	Current	Recommended
Status		Value	Values
Pass	Source block specifies -1 sample time (InheritedTsInSrcMsg)	error	error
Pass	Multitask rate transition (MultiTaskRateTransMsg)	error	error
Pass	Multitask conditionally executed subsystem (MultiTaskCondExecSysMsg)	error	error

Pass	Enforce sample times specified by Signal Specification blocks (SigSpecEnsureSampleTimeMsg)	error	error
	blocks (Sigspectificatesample infletwisg)		
Pass	Single task rate transition (SingleTaskRateTransMsg)	error	error
Pass	Tasks with equal priority (TasksWithSamePriorityMsg)	error	error
Pass	Unspecified inheritability of sample time (UnknownTsInhSupMsg)	error	error

\_\_\_\_\_

**⊘** Check safety-related optimization settings for logic signals

Check optimization settings in the model configuration that apply to logic signals and might impact safety.

#### **Passed**

All constraints on model configuration parameters have been met.

Status	Parameter	Current Value	Recommended Values
Pass	Implement logic signals as Boolean data (vs. double) (BooleanDataType)	on	on

**⊘** Check safety-related block reduction optimization settings

Check block reduction optimization settings in the model configuration that might impact safety.

#### **Passed**

All constraints on model configuration parameters have been met.

Status	Parameter	Current Value	Recommended Values
Pass	Block reduction (BlockReduction)	off	off

\_\_\_\_\_

Check safety-related code generation settings for code style

Check code generation settings in the model configuration that apply to code style and might impact safety.

#### **Passed**

All constraints on model configuration parameters have been met.

Status	Parameter	Current Value	Recommended Values	Prerequisites
Pass	Parentheses level (ParenthesesLevel)	Maximum	Maximum	SystemTargetFile
Pass	Preserve operand order in expression (PreserveExpressionOrder)	on	on	SystemTargetFile
D - Pass	System target file (SystemTargetFile)	ERT based target	ERT based target	

\_\_\_\_\_

Check safety-related optimization settings for application lifespan

Check optimization settings in the model configuration that apply to application lifespan and might impact safety.

#### **Passed**

All constraints on model configuration parameters have been met.

Status	Parameter	Current Value	Recommended Values
Pass	Application lifespan (days) (LifeSpan)	inf	Inf

Check safety-related code generation symbols settings

Check code generation identifier settings in the model configuration that might impact safety.

#### **Passed**

Status	Parameter	Current Value	Recommended Values	Not Recommended Values	Prerequisites
D - Pass	System target file (SystemTargetFile)	ERT based target	ERT based target		
Pass	Minimum mangle length (MangleLength)	4		1, 2, 3	SystemTargetFile

✓ Check safety-related optimization settings for loop unrolling threshold

Check optimization settings in the model configuration that apply to loop unrolling threshold and might impact safety.

#### **Passed**

All constraints on model configuration parameters have been met.

Status	Parameter	Current Value	Not Recommended Values
Pass	Loop unrolling threshold (RollThreshold)	5	0, 1

Check safety-related optimization settings for data initialization

Check optimization settings in the model configuration that apply to data initialization and might impact safety.

#### **Passed**

Status	Parameter	Current Value	Recommended Values	Prerequisites
D - Pass	System target file (SystemTargetFile)	ERT based target	ERT based target	

Pass	Remove root level I/O zero initialization (ZeroExternalMemoryAtStartup) *	on	on	SystemTargetFile
Pass	Remove internal data zero initialization (ZeroInternalMemoryAtStartup) *	on	on	SystemTargetFile

#### **Recommended Action**

\* The Command-Line values provided in the table are reverse of the settings in the Configuration Parameters Dialog. Therefore, 'on' in the Command-Line corresponds to an "Off" setting in the dialog, and 'off' in the Command-Line corresponds to an "On" setting in the dialog.

\_\_\_\_\_



Check optimization settings in the model configuration that apply to data type conversions and might impact safety.

#### **Passed**

All constraints on model configuration parameters have been met.

Status	Parameter	Current Value	Recommended Values
Pass	Remove code from floating-point to integer conversions that wraps out-of-range values (EfficientFloat2IntCast)	on	on

\_\_\_\_\_

<b>②</b> C	Check safety	/-related	optimization	settings for	division	arithmetic	exception
------------	--------------	-----------	--------------	--------------	----------	------------	-----------

Check optimization settings in the model configuration that apply to division arithmetic exceptions and might impact safety.

#### **Passed**

All constraints on model configuration parameters have been met.

	Parameter	Current	Recommended	Prerequisites
Status		Value	Values	
D - Pass	System target file (SystemTargetFile)	ERT based target	ERT based target	
Pass	Remove code that protects against division arithmetic exceptions (NoFixptDivByZeroProtection)	off	off	SystemTargetFile

Check safety-related optimization settings for specified minimum and maximum values

Check optimization settings in the model configuration that apply to specified minimum and maximum values and might impact safety.

#### **Passed**

Status	Parameter	Current Value	Recommended Values	Prerequisites
Pass	Optimize using the specified minimum and maximum values (UseSpecifiedMinMax)	off	off	SystemTargetFile
D - Pass	System target file (SystemTargetFile)	ERT based target	ERT based target	

\_\_\_\_\_



Check diagnostic settings in the model configuration that affect compatibility and might impact safety.

#### **Passed**

All constraints on model configuration parameters have been met.

Status	Parameter	Current Value	Recommended Values
Pass	S-function upgrades needed (SFcnCompatibilityMsg)	error	error

\_\_\_\_\_

## **⊘** Check safety-related diagnostic settings for parameters

Check diagnostic settings in the model configuration that apply to parameters and might impact safety.

#### **Passed**

All constraints on model configuration parameters have been met.

Status	Parameter	Current Value	Recommended Values
Pass	Detect downcast (ParameterDowncastMsg)	error	error
Pass	Detect underflow (ParameterUnderflowMsg)	error	error
Pass	Detect overflow (ParameterOverflowMsg)	error	error
Pass	Detect precision loss (ParameterPrecisionLossMsg)	error	error
Pass	Detect loss of tunability (ParameterTunabilityLossMsg)	error	error

Check safety-related diagnostic settings for Merge blocks

Check diagnostic settings in the model configuration that apply to Merge blocks and might impact safety.

#### **Passed**

Status	Parameter	Current Value	Recommended Values
Pass	Detect multiple driving blocks executing at the same time step (MergeDetectMultiDrivingBlocksExec)	error	error

Check safety-related diagnostic settings for model initializa
---

Check diagnostic settings in the model configuration that affect model initialization and might impact safety.

#### **Passed**

All constraints on model configuration parameters have been met.

Status	Parameter	Current Value	Recommended Values
Pass	Underspecified initialization detection (UnderspecifiedInitializationDetection)	Simplified	Simplified

\_\_\_\_\_

## **⊘** Check safety-related diagnostic settings for data used for debugging

Check diagnostic settings in the model configuration that apply to data used for debugging and might impact safety.

#### **Passed**

Status	Parameter	Current Value	Recommended Values
Pass	Model Verification block enabling (AssertControl)	DisableAll	DisableAll

Check safety-related diagnostic settings for signal connectivity

Check diagnostic settings in the model configuration that apply to signal connectivity and might impact safety.

#### **Passed**

All constraints on model configuration parameters have been met.

	Parameter	Current	Recommended
Status		Value	Values
Pass	Signal label mismatch (SignalLabelMismatchMsg)	error	error
Pass	Unconnected block input ports (UnconnectedInputMsg)	error	error
Pass	Unconnected block output ports (UnconnectedOutputMsg)	error	error
Pass	Unconnected line (UnconnectedLineMsg)	error	error

Check safety-related diagnostic settings for bus connectivity

Check diagnostic settings in the model configuration that apply to bus connectivity and might impact safety.

#### **Passed**

	Parameter	Current Value	Recommended Values
Statu			
S			
Pass	Unspecified bus object at root Outport block (RootOutportRequireBusObject )	error	error
Pass	Element name mismatch (BusObjectLabelMismatch)	error	error
Pass	Bus signal treated as vector (StrictBusMsg)	ErrorOnBusTreatedAsVector	ErrorOnBusTreatedAsVecto r
Pass	Non-bus signals treated as bus signals (NonBusSignalsTreatedAsBus)	error	error
Pass	Repair bus selections (BusNameAdapt)	WarnAndRepair	WarnAndRepair

Check safety-related diagnostic settings that apply to function-call connectivity

Check diagnostic settings in the model configuration that apply to function-call connectivity and might impact safety.

#### **Passed**

	Parameter	Current	Recommended
Status		Value	Values

Pass	Invalid function-call connection (InvalidFcnCallConnMsg)	error	error
Pass	Context-dependent inputs (FcnCallInpInsideContextMsg)	error	error

Check safety-related diagnostic settings for type conversions

Check diagnostic settings in the model configuration that apply to type conversions and might impact safety.

#### **Passed**

All constraints on model configuration parameters have been met.

Status	Parameter	Current Value	Recommended Values
Pass	Unnecessary type conversions (UnnecessaryDatatypeConvMsg)	warning	warning
Pass	Vector/matrix block input conversion (VectorMatrixConversionMsg)	error	error
Pass	32-bit integer to single precision float conversion (Int32ToFloatConvMsg)	warning	warning

Check safety-related diagnostic settings for model referencing

Check diagnostic settings in the model configuration that apply to model referencing and might impact safety.

#### **Passed**

All constraints on model configuration parameters have been met.

Status	Parameter	Current Value	Recommended Values
Pass	Model block version mismatch (ModelReferenceVersionMismatchMessage)	none	none
Pass	Port and parameter mismatch (ModelReferenceIOMismatchMessage)	error	error
Pass	Invalid root Inport/Outport block connection (ModelReferenceIOMsg)	error	error
Pass	Unsupported data logging (ModelReferenceDataLoggingMessage)	error	error

☑ Check safety-related diagnostic settings for Stateflow

Check diagnostic settings in the model configuration that apply to Stateflow and might impact safety.

#### **Passed**

Status	Parameter	Current Value	Recommended Values
Pass	Unexpected backtracking (SFUnexpectedBacktrackingDiag)	error	error

Pass	Invalid input data access in chart initialization (SFInvalidInputDataAccessInChartInitDiag)		error
Pass	No unconditional default transitions (SFNoUnconditionalDefaultTransitionDiag)	error	error
Pass	Transition outside natural parent (SFTransitionOutsideNaturalParentDiag)	error	error
Pass	Unreachable execution path (SFUnreachableExecutionPathDiag)	error	error
Pass	Undirected event broadcasts (SFUndirectedBroadcastEventsDiag)	error	error
Pass	Transition action specified before condition action (SFTransitionActionBeforeConditionDiag)	error	error

\_\_\_\_\_



Check diagnostic settings in the model configuration that apply to signal data and might impact safety.

#### **Passed**

	Parameter	<b>Current Value</b>	Recommended
Status			Values
Pass	Signal resolution (SignalResolutionControl)		None,
		UseLocalSettings	UseLocalSettings
Pass	Division by singular matrix	error	error
	(CheckMatrixSingularityMsg)		
Pass	Underspecified data types	error	error
	(UnderSpecifiedDataTypeMsg)		

Pass	Wrap on overflow (IntegerOverflowMsg)	error	error
Pass	Saturate on overflow (IntegerSaturationMsg)	error	error
Pass	Inf or NaN block output (SignalInfNanChecking)	error	error
Pass	"rt" prefix for identifiers (RTPrefix)	error	error
Pass	Simulation range checking (SignalRangeChecking)	error	error







Identify inappropriate characters and length issues in model file name

#### **Passed**

No issues found with model file name.

Check model object names

Identify invalid names of following model objects (first invalid name fragment is highlighted):

- **Blocks**
- Signals
- Parameters
- **Buses**
- Stateflow elements

#### **Passed**

No invalid names of model objects found

Check for model elements that do not link to requirements

Check for model elements that do not link to a requirements document.

No model elements found without links to requirements.



Check for blocks not recommended for MISRA C:2012

Identify blocks that are not recommended for MISRA C:2012 compliant code generation.

#### **Passed**

None of the blocks are defined as "not recommended" for MISRA C:2012 compliant code generation.

Check configuration parameters for MISRA C:2012

Identify configuration parameters that might impact MISRA C:2012 compliant code generation.

#### **Passed**

	Parameter		Recommended Values	Prerequisites
Stat		Current		
us		Value		

Pass	Model Verification block		DisableAll	
	enabling (AssertControl)	Disable		
		All		
D -	Shared code placement	Shared	Shared location	
Pass	(UtilityFuncGeneration)	location	Sharea location	
. 455				
Pass	Generate shared constants	off	off	
	(GenerateSharedConstants)			UtilityFuncGener ation
D -	System target file	ERT	ERT based target	
Pass	(SystemTargetFile)	based	ENT bused target	
1 433	(System algetime)	target		
		_		
Pass	continuous time	off	off	
	(SupportContinuousTime)			SystemTargetFile
Pass	non-inlined S-functions	off	off	
	(SupportNonInlinedSFcns)			SystemTargetFile
Pass	MAT-file logging	off	off	
1 433	(MatFileLogging)		3.1	
Pass	Code replacement library	None	None, AUTOSAR 4.0	
	(CodeReplacementLibrary)			
Pass	Parentheses level		Maximum	
	(ParenthesesLevel)	Maxim		SystemTargetFile
		um		
Pass	Casting modes		Standards	
	(CastingMode)	Standar		SystemTargetFile
	,	ds		,
Pass	System-generated identifiers		Shortened	
Fass	(Internalidentifier)	Shorten	Shortened	SystemTargetFile
	(internandentiner)	ed		System rangetime
Pass	Signed integer division	Zero	Zero, Floor	
	rounds to			
	(ProdIntDivRoundTo)			
Pass	Use division for fixed-point	on	on,	
	net slope computation		UseDivisionForReciprocalsOfInt	
	(UseDivisionForNetSlopeCom		egersOnly	
	putation)			

Pass	Replace multiplications by powers of two with signed bitwise shifts (EnableSignedLeftShifts)	off	off	SystemTargetFile
Pass	Allow right shifts on signed integers (EnableSignedRightShifts)	off	off	SystemTargetFile
Pass	Wrap on overflow (IntegerOverflowMsg)	error	warning, error	
Pass	Preserve static keyword in function declarations (PreserveStaticInFcnDecls)	on	on	
Pass	Inf or NaN block output (SignalInfNanChecking)	error	warning, error	
Pass	Dynamic memory allocation in MATLAB functions (MATLABDynamicMemAlloc)	off	off	
Pass	External mode (ExtMode)	off	off	
Pass	Undirected event broadcasts (SFUndirectedBroadcastEvent sDiag)	error	error	
Pass	Compile-time recursion limit for MATLAB functions (CompileTimeRecursionLimit)	0	0	
Pass	Enable run-time recursion for MATLAB functions (EnableRuntimeRecursion)	off	off	
D - Pass	Include comments (GenerateComments)	on	on	
Pass	MATLAB user comments (MATLABFcnDesc)	on	on	GenerateComme nts, SystemTargetFile

Λ Less



Check for Discrete-Time Integrator blocks with initial condition uncertainty

Passed

\_\_\_\_\_

Check root model Inport block specifications

**Passed** 

-----

Identify unconnected lines, input ports, and output ports

Identify unconnected lines, input ports, and output ports in the model

#### **Passed**

There are no unconnected lines, input ports, and output ports in this model.

\_\_\_\_\_

**⊘** Check usage of tunable parameters in blocks

Identify tunable parameters used to specify expressions, data type conversions, or indexing operations.

#### **Passed**

Tunable parameters are not used in the model.

☑ Check for Strong Data Typing with Simulink I/O

Identify whether Stateflow charts have **Use Strong Data Typing with Simulink I/O** cleared.

#### **Passed**

No Stateflow charts were found.

Check for blocks that have constraints on tunable parameters

Identify Lookup Table blocks that have constraints on tunable parameters.

**Check usage of Lookup Table blocks** 



No Lookup Table blocks with tunable parameters found.

## Check usage of Lookup Table (2-D) blocks

#### **Passed**

No Lookup Table (2-D) blocks with tunable parameters found.

## Check usage of n-D Lookup Table blocks

#### **Passed**

No n-D Lookup Table blocks with tunable parameters found.



Identify questionable subsystem settings

#### **Passed**

Subsystem blocks do not specify their Function packaging option as Nonreusable function.

#### **Passed**

Subsystems that specify the Function packaging option as Reusable function are not checked since the Model Configuration Parameter Shared code placement is set to Shared location.



Check bus signals treated as vectors

## Bus signal treated as vector

Identify bus signals in the model that are treated as vectors by the Simulink software.

#### **Passed**

The model uses bus signals properly. Model is configured to detect future changes that might result in improper bus signal usage.



Check for potentially delayed function-call block return values

**Passed** 



**Check usage of Merge blocks** 

This check finds and reports issues related to merge blocks for migrating to simplified initialization mode.

#### See Also

- Check usage of Merge blocks
- Underspecified initialization detection

#### **Passed**







Identify Stateflow data objects with local scope that are not scoped at the chart level or below.

### **Passed**

No Stateflow charts were found.



Check usage of exclusive and default states in state machines

Identify Stateflow charts and substates that incorrectly use or define exclusive and default states.

## **Check Stateflow charts for exclusive states**

Identify Stateflow charts that have singular exclusive (OR) states.

## **Passed**

The Stateflow charts do not have singular exclusive (OR) states.

## **Check Stateflow charts for undefined default states**

Identify Stateflow charts that do not define default states.

Each Stateflow chart defines a default state.

\_\_\_\_\_

## Check for multiple states assigned as the default state

At the root level in the Stateflow hierarchy only one state should be assigned as the default.

#### **Passed**

The root level of the chart has only one default state assigned.

### Check for substates with singular OR states

States configured as OR should always be part of a group of states.

### **Passed**

No singular OR states were detected.

\_\_\_\_\_

#### Check for substates without default states defined

At every level in the Stateflow hierarchy a default state should be assigned.

#### **Passed**

All substates have default states assigned.

## Check for substates with multiple default states defined

At every level in the Stateflow hierarchy only one state should be assigned as the default.

#### **Passed**

All levels of the chart have only one default state assigned.





#### **Passed**

\_\_\_\_\_

Identify parameterized library links
Passed

✓ Identify unresolved library links
Passed

☐ Model Referencing
Check for model reference configuration mismatch  Passed
Check for parameter tunability information ignored for referenced models
Passed
© Requirements Consistency
Identify requirement links that specify invalid locations within documents  Passed
Identify requirement links with missing documents  Passed
Identify requirement links with path type inconsistent with preferences  Passed
Identify selection-based links having description fields that do not match their requirements document text  Passed
□ Simulink Coder 3 0 0 0 □ 0
Check sample times and tasking mode
Passed

Check solver for code generation

#### **Passed**

\_\_\_\_\_

**⊘** Check the hardware implementation

## Check 'Byte ordering' and 'Signed integer division rounds to' parameters

Identify inconsistencies or underspecification of hardware attributes that can lead to incorrect and inefficient generated code.

#### **Passed**

Target specification is consistent.

\_\_\_\_\_\_

## Check whether 'Production hardware' and 'Test hardware' match

Search for 'Test hardware is the same as production hardware' in the Configuration Parameters dialog box and check if it is selected. If it is cleared, identify whether target specifications match.

#### **Passed**

'Test hardware is the same as production hardware' is selected or is cleared and the target specifications match.

\_\_\_\_\_\_

## Check code generation settings

## Verify 'Code interface packaging' setting

Check whether Code Generation > Interface > Code interface packaging is set to Nonreusable function or Reusable function.

## **Passed**

Code Generation > Interface > Code interface packaging is set to Nonreusable function
or Reusable function.

\_\_\_\_\_

# Verify 'Use dynamic memory allocation for model initialization' setting Check whether Code Generation > Interface > Use dynamic memory allocation for model initialization is cleared.

## **Passed**

**Code Generation > Interface > Use dynamic memory allocation for model initialization** is cleared.

\_\_\_\_\_\_

## Verify 'Shared code placement' setting

Check whether Code Generation > Interface > Shared code placement is set to Shared location.

#### **Passed**

Code Generation > Interface > Shared code placement is set to Shared location.

### Verify 'Source file' setting

Check whether Code Generation > Custom Code > Source file is set to "" (i.e. unspecified).

#### **Passed**

Code Generation > Custom Code > Source file is set to "".

## Verify 'Header file' setting

Check whether Code Generation > Custom Code > Header file is set to "" (i.e. unspecified).

#### **Passed**

Code Generation > Custom Code > Header file is set to "".

\_\_\_\_\_

## **Verify 'Initialize function' setting**

Check whether Code Generation > Custom Code > Initialize function is set to "" (i.e. unspecified).

#### **Passed**

Code Generation > Custom Code > Initialize function is set to "".

\_\_\_\_\_

## Verify 'Terminate function' setting

Check whether Code Generation > Custom Code > Terminate function is set to "" (i.e. unspecified).

#### **Passed**

Code Generation > Custom Code > Terminate function is set to "".

\_\_\_\_\_

## Verify 'Disable incompatible optimizations' setting

Check whether **Code Generation > Optimization > Disable incompatible optimizations** is set to - SLCI.

**Code Generation > Optimization > Disable incompatible optimizations** is set to -SLCI.

\_\_\_\_\_

## Verify 'Combine signal/state structures' setting

Check whether **Code Generation > Interface > Combine signal/state structures** is cleared.

#### **Passed**

**Code Generation > Interface > Combine signal/state structures** is cleared.

\_\_\_\_\_

## **Verify 'Array layout' setting**

Check whether Code Generation > Interface > Array layout is set to Column-major.

#### **Passed**

**Code Generation > Interface > Array layout** is set to Column-major.

\_\_\_\_\_

Verify 'Remove code from floating-point to integer conversions that wraps out-of-range values' setting

Check whether Code Generation > Optimization > Remove code from floating-point to integer conversions that wraps out-of-range values is selected.

#### **Passed**

Code Generation > Optimization > Remove code from floating-point to integer conversions that wraps out-of-range values is selected.

Verify 'Remove code from floating-point to integer conversions with saturation that maps NaN to zero' setting

Check whether Code Generation > Optimization > Remove code from floating-point to integer conversions with saturation that maps NaN to zero is cleared.

#### **Passed**

Code Generation > Optimization > Remove code from floating-point to integer conversions with saturation that maps NaN to zero is cleared.

## Verify 'Include comments' setting

Check whether Code Generation > Comments > Include comments is selected.

### **Passed**

**Code Generation > Comments > Include comments** is selected.

\_\_\_\_\_

## Verify 'Preserve condition expression in if statement' setting

Check whether Code Generation > Code Style > Preserve condition expression in if statement is

selected.

#### **Passed**

Code Generation > Code Style > Preserve condition expression in if statement is selected.

\_\_\_\_\_

## Verify 'Code replacement library' setting

Check whether Code Generation > Interface > Code replacement library is set to None.

#### **Passed**

**Code Generation > Interface > Code replacement library** is set to None.

\_\_\_\_\_

## Verify 'Standard math library' setting

Check whether Code Generation > Interface > Standard math library is set to C89/C90 (ANSI) or C99 (ISO).

#### **Passed**

**Code Generation > Interface > Standard math library** is set to C89/C90 (ANSI) or C99 (ISO).

\_\_\_\_\_

## Verify 'Classic call interface' setting

Check whether **Code Generation > Interface > Classic call interface** is cleared.

## **Passed**

Code Generation > Interface > Classic call interface is cleared.

## Verify 'Terminate function required' setting

Check whether **Code Generation > Interface > Terminate function required** is cleared.

#### **Passed**

**Code Generation > Interface > Terminate function required** is cleared.

\_\_\_\_\_

Verify 'Remove code that protects against division arithmetic exceptions' setting
Check whether Code Generation > Optimization > Remove code that protects against division
arithmetic exceptions is cleared.

#### **Passed**

**Code Generation > Optimization > Remove code that protects against division arithmetic exceptions** is cleared.

9-----

## Verify 'MAT-file logging' setting

Check whether **Code Generation > Interface > MAT-file logging** is cleared.

#### **Passed**

**Code Generation > Interface > MAT-file logging** is cleared.

\_\_\_\_\_

## Verify 'Maximum stack size (bytes)' setting

Check whether Code Generation > Optimization > Maximum stack size (bytes) is set to inf.

#### **Passed**

Code Generation > Optimization > Maximum stack size (bytes) is set to inf.

## Verify 'Pack Boolean data into bitfields' setting

Check whether Code Generation > Optimization > Pack Boolean data into bitfields is cleared.

#### **Passed**

**Code Generation > Optimization > Pack Boolean data into bitfields** is cleared.

\_\_\_\_\_

## Verify 'Use bitsets for storing state configuration' setting

Check whether **Code Generation > Optimization > Use bitsets for storing state configuration** is cleared.

#### **Passed**

Code Generation > Optimization > Use bitsets for storing state configuration is cleared.

\_\_\_\_\_

## Verify 'non-finite numbers' setting

Check whether Code Generation > Interface > non-finite numbers is cleared.

#### **Passed**

**Code Generation > Interface > non-finite numbers** is cleared.

#### Verify 'absolute time' setting

Check whether Code Generation > Interface > absolute time is cleared.

#### **Passed**

**Code Generation > Interface > absolute time** is cleared.

\_\_\_\_\_\_

## Verify 'Default parameter behavior' setting

Check whether Code Generation > Optimization > Default parameter behavior is set to Inlined.

#### **Passed**

**Code Generation > Optimization > Default parameter behavior** is set to Inlined.

## Verify 'Remove error status field in real-time model data structure' setting

Check whether Code Generation > Interface > Remove error status field in real-time model data structure is selected.

#### **Passed**

**Code Generation > Interface > Remove error status field in real-time model data structure** is selected.

### Verify 'Create block' setting

Check whether Code Generation > Verification > Create block is set to none.

#### **Passed**

**Code Generation > Verification > Create block** is set to none.

\_\_\_\_\_

### Verify 'Measure function execution times' setting

Check whether Code Generation > Verification > Measure function execution times is set to off.

#### **Passed**

**Code Generation > Verification > Measure function execution times** is set to off.

\_\_\_\_\_

#### Verify 'Signal naming' setting

Check whether Code Generation > Identifiers > Signal naming is set to None.

#### **Passed**

**Code Generation > Identifiers > Signal naming** is set to None.

## Verify 'Parameter naming' setting

Check whether Code Generation > Identifiers > Parameter naming is set to None.

#### **Passed**

**Code Generation > Identifiers > Parameter naming** is set to None.

## Verify 'TLC options' setting

Check whether Code Generation > TLC options is set to "" (i.e. unspecified).

**Code Generation > TLC options** is set to "".

\_\_\_\_\_

## Verify 'variable-size signals' setting

Check whether **Code Generation > Interface > variable-size signals** is cleared.

#### **Passed**

**Code Generation > Interface > variable-size signals** is cleared.

\_\_\_\_\_\_

## Verify 'Single output/update function' setting

Check whether **Code Generation > Interface > Single output/update function** is selected.

#### **Passed**

**Code Generation > Interface > Single output/update function** is selected.

\_\_\_\_\_

### Verify 'Generate an example main program' setting

Check whether Code Generation > Templates > Generate an example main program is selected.

#### **Passed**

**Code Generation > Templates > Generate an example main program** is selected.

## Verify 'Use bitsets for storing Boolean data' setting

Check whether Code Generation > Optimization > Use bitsets for storing Boolean data is cleared.

## **Passed**

**Code Generation > Optimization > Use bitsets for storing Boolean data** is cleared.

\_\_\_\_\_

#### **Verify 'Casting modes' setting**

Check whether Code Generation > Code Style > Casting modes is set to Nominal or Standards.

#### **Passed**

Code Generation > Code Style > Casting modes is set to Nominal or Standards.

Verify 'Suppress generation of default cases for Stateflow switch statements if unreachable' setting Check whether Code Generation > Code Style > Suppress generation of default cases for Stateflow switch statements if unreachable is cleared.

#### **Passed**

Code Generation > Code Style > Suppress generation of default cases for Stateflow switch

## Verify 'Optimize using the specified minimum and maximum values' setting

Check whether Code Generation > Optimization > Optimize using the specified minimum and maximum values is cleared.

#### **Passed**

**Code Generation > Optimization > Optimize using the specified minimum and maximum values** is cleared

### Verify 'Existing shared code' setting

Check whether Code Generation > Interface > Existing shared code is set to "" (i.e. unspecified).

#### **Passed**

Code Generation > Interface > Existing shared code is set to "".

\_\_\_\_\_

## Verify 'parameters' setting

Check whether **Code Generation > Interface > parameters** is cleared.

#### **Passed**

**Code Generation > Interface > parameters** is cleared.

## Verify 'signals' setting

Check whether **Code Generation > Interface > signals** is cleared.

## Passed

**Code Generation > Interface > signals** is cleared.

\_\_\_\_\_

## **Verify 'states' setting**

Check whether **Code Generation > Interface > states** is cleared.

#### **Passed**

Code Generation > Interface > states is cleared.

\_\_\_\_\_

## Verify 'root-level I/O' setting

Check whether Code Generation > Interface > root-level I/O is cleared.

Code Generation > Interface > root-level I/O is cleared.

\_\_\_\_\_

## Verify 'External mode' setting

Check whether **Code Generation > Interface > External mode** is cleared.

#### **Passed**

Code Generation > Interface > External mode is cleared.

### Verify 'ASAP2 interface' setting

Check whether Code Generation > Interface > ASAP2 interface is cleared.

#### **Passed**

**Code Generation > Interface > ASAP2 interface** is cleared.

\_\_\_\_\_

## Verify 'Rate Transition block code' setting

Check whether Code Generation > Code Placement > Rate Transition block code is set to Inline.

#### **Passed**

**Code Generation > Code Placement > Rate Transition block code** is set to Inline.

## **⊘** Check data import and export settings

## Verify 'Initial state' setting

Check whether **Data Import/Export > Initial state** is cleared.

#### **Passed**

**Data Import/Export > Initial state** is cleared.

## Check diagnostic settings

## Verify 'Multitask data store' setting

Check whether Diagnostics > Data Validity > Multitask data store is set to error.

#### **Passed**

**Diagnostics > Data Validity > Multitask data store** is set to error.

\_\_\_\_\_

## Verify 'Multitask rate transition' setting

Check whether Diagnostics > Sample Time > Multitask rate transition is set to error.

**Diagnostics > Sample Time > Multitask rate transition** is set to error.

\_\_\_\_\_

## Verify 'Multitask conditionally executed subsystem' setting

Check whether Diagnostics > Sample Time > Multitask conditionally executed subsystem is set to error.

#### **Passed**

**Diagnostics > Sample Time > Multitask conditionally executed subsystem** is set to error.

\_\_\_\_\_

## Verify 'Algebraic loop' setting

Check whether **Diagnostics > Algebraic loop** is set to error.

#### **Passed**

**Diagnostics > Algebraic loop** is set to error.

\_\_\_\_\_\_

## Verify 'Detect write after write' setting

Check whether **Diagnostics > Data Validity > Detect write after write** is set to EnableAllAsError.

#### **Passed**

Diagnostics > Data Validity > Detect write after write is set to EnableAllAsError.

\_\_\_\_\_

## Verify 'Underspecified initialization detection' setting

Check whether **Diagnostics > Data Validity > Underspecified initialization detection** is set to Simplified.

#### **Passed**

Diagnostics > Data Validity > Underspecified initialization detection is set to Simplified.

\_\_\_\_\_

## Verify 'Non-bus signals treated as bus signals' setting

Check whether **Diagnostics > Connectivity > Non-bus signals treated as bus signals** is set to error.

#### **Passed**

Diagnostics > Connectivity > Non-bus signals treated as bus signals is set to error.

\_\_\_\_\_

### **Verify 'Detect downcast' setting**

Check whether **Diagnostics > Data Validity > Detect downcast** is set to error.

**Diagnostics > Data Validity > Detect downcast** is set to error.

\_\_\_\_\_

## Verify 'Detect overflow' setting

Check whether **Diagnostics > Data Validity > Detect overflow** is set to error.

#### **Passed**

**Diagnostics > Data Validity > Detect overflow** is set to error.

\_\_\_\_\_

## Verify 'Wrap on overflow' setting

Check whether Diagnostics > Data Validity > Wrap on overflow is set to error.

#### **Passed**

Diagnostics > Data Validity > Wrap on overflow is set to error.

\_\_\_\_\_

## Verify 'Saturate on overflow' setting

Check whether Diagnostics > Data Validity > Saturate on overflow is set to error.

#### **Passed**

**Diagnostics > Data Validity > Saturate on overflow** is set to error.

### Verify 'Detect underflow' setting

Check whether **Diagnostics > Data Validity > Detect underflow** is set to error.

#### **Passed**

**Diagnostics > Data Validity > Detect underflow** is set to error.

### Verify 'Detect loss of tunability' setting

Check whether Diagnostics > Data Validity > Detect loss of tunability is set to error.

#### **Passed**

**Diagnostics > Data Validity > Detect loss of tunability** is set to error.

\_\_\_\_\_

## Verify 'Allow symbolic dimension specification' setting

Check whether **Diagnostics > Allow symbolic dimension specification** is cleared.

#### **Passed**

**Diagnostics > Allow symbolic dimension specification** is cleared.

\_\_\_\_\_

## Verify 'Invalid root Inport/Outport block connection' setting

Check whether **Diagnostics > Model Referencing > Invalid root Inport/Outport block connection** is set to error.

#### **Passed**

Diagnostics > Model Referencing > Invalid root Inport/Outport block connection is set to error.

### Verify 'Unexpected backtracking' setting

Check whether Diagnostics > Stateflow > Unexpected backtracking is set to error.

#### **Passed**

**Diagnostics > Stateflow > Unexpected backtracking** is set to error.

\_\_\_\_\_\_

### Verify 'Invalid input data access in chart initialization' setting

Check whether **Diagnostics > Stateflow > Invalid input data access in chart initialization** is set to error.

#### **Passed**

Diagnostics > Stateflow > Invalid input data access in chart initialization is set to error.

\_\_\_\_\_

## Verify 'No unconditional default transitions' setting

Check whether Diagnostics > Stateflow > No unconditional default transitions is set to error.

#### **Passed**

Diagnostics > Stateflow > No unconditional default transitions is set to error.

\_\_\_\_\_\_

## Verify 'Transition outside natural parent' setting

Check whether Diagnostics > Stateflow > Transition outside natural parent is set to error.

#### **Passed**

**Diagnostics > Stateflow > Transition outside natural parent** is set to error.

\_\_\_\_\_

## Verify 'Unreachable execution path' setting

Check whether Diagnostics > Stateflow > Unreachable execution path is set to error.

#### **Passed**

**Diagnostics > Stateflow > Unreachable execution path** is set to error.

\_\_\_\_\_

## Verify 'Undirected event broadcasts' setting

Check whether Diagnostics > Stateflow > Undirected event broadcasts is set to error.

### **Passed**

**Diagnostics > Stateflow > Undirected event broadcasts** is set to error.

### Verify 'Transition action specified before condition action' setting

Check whether Diagnostics > Stateflow > Transition action specified before condition action is set to error.

#### **Passed**

Diagnostics > Stateflow > Transition action specified before condition action is set to error.

## Verify Bus signal treated as vector setting

Check whether Diagnostics > Connectivity > Bus signal treated as vector is set to 'error'

#### **Passed**

**Diagnostics > Connectivity > Bus signal treated as vector** is set to error.

Check hardware implementation settings

## Verify 'char' setting

Check whether Hardware Implementation > char is set to 8.

#### **Passed**

**Hardware Implementation > char** is set to 8.

## **Verify 'short' setting**

Check whether **Hardware Implementation > short** is set to 16.

#### **Passed**

**Hardware Implementation > short** is set to 16.

## Verify 'int' setting

Check whether **Hardware Implementation > int** is set to 32.

#### **Passed**

**Hardware Implementation > int** is set to 32.

## Verify 'long' setting

Check whether **Hardware Implementation > long** is set to 32.

#### **Passed**

**Hardware Implementation > long** is set to 32.

## Verify 'float' setting

Check whether Hardware Implementation > float is set to 32.

#### **Passed**

**Hardware Implementation > float** is set to 32.

## Verify 'double' setting

Check whether Hardware Implementation > double is set to 64.

#### **Passed**

**Hardware Implementation > double** is set to 64.

\_\_\_\_\_

## Verify 'pointer' setting

Check whether Hardware Implementation > pointer is set to 32.

#### **Passed**

**Hardware Implementation > pointer** is set to 32.

## Verify 'size\_t' setting

Check whether Hardware Implementation > size\_t is set to 32.

#### **Passed**

**Hardware Implementation > size\_t** is set to 32.

.....

## Verify 'ptrdiff\_t' setting

Check whether Hardware Implementation > ptrdiff\_t is set to 32.

#### **Passed**

**Hardware Implementation > ptrdiff\_t** is set to 32.

## Verify 'native' setting

Check whether **Hardware Implementation > native** is set to 32.

#### **Passed**

**Hardware Implementation > native** is set to 32.

\_\_\_\_\_

## Verify 'Signed integer division rounds to' setting

Check whether Hardware Implementation > Signed integer division rounds to is set to Zero.

#### **Passed**

**Hardware Implementation > Signed integer division rounds to** is set to Zero.

\_\_\_\_\_

## Verify 'Shift right on a signed integer as arithmetic shift' setting

Check whether Hardware Implementation > Shift right on a signed integer as arithmetic shift is selected.

#### **Passed**

Hardware Implementation > Shift right on a signed integer as arithmetic shift is selected.

-----

## Verify 'Support long long' setting

Check whether **Hardware Implementation > Support long long** is cleared.

#### **Passed**

**Hardware Implementation > Support long long** is cleared.

\_\_\_\_\_

## Verify 'Test hardware is the same as production hardware' setting

Check whether Hardware Implementation > Test hardware is the same as production hardware is selected.

#### **Passed**

Hardware Implementation > Test hardware is the same as production hardware is selected.

\_\_\_\_\_

## Verify 'Device vendor' setting

Identify whether Hardware Implementation > Device vendor is not set to ASIC/FPGA->ASIC/FPGA .

#### **Passed**

Hardware Implementation > Device vendor is not set to ASIC/FPGA->ASIC/FPGA .

## Check math and data types settings

Verify 'Implement logic signals as Boolean data (vs. double)' setting
Check whether Math and Data Types > Implement logic signals as Boolean data (vs. double) is selected.

Math and Data Types > Implement logic signals as Boolean data (vs. double) is selected.

## Verify 'Use algorithms optimized for row-major array layout' setting

Check whether Math and Data Types > Use algorithms optimized for row-major array layout is cleared.

#### **Passed**

Math and Data Types > Use algorithms optimized for row-major array layout is cleared.

## Check solver settings

## Verify 'Type' setting

Check whether **Solver > Type** is set to Fixed-step.

#### **Passed**

**Solver > Type** is set to Fixed-step.

## **Verify 'Solver' setting**

Check whether Solver > Solver is set to FixedStepDiscrete.

#### **Passed**

**Solver > Solver** is set to FixedStepDiscrete.

## Verify 'Periodic sample time constraint' setting

Check whether Solver > Periodic sample time constraint is set to Unconstrained or STIndependent.

#### **Passed**

Solver > Periodic sample time constraint is set to Unconstrained or STIndependent.

#### Verify 'Automatically handle rate transition for data transfer' setting

Check whether **Solver > Automatically handle rate transition for data transfer** is cleared.

#### **Passed**

**Solver > Automatically handle rate transition for data transfer** is cleared.

#### Verify 'Allow tasks to execute concurrently on target' setting

Check whether **Solver > Allow tasks to execute concurrently on target** is cleared.

#### **Passed**

Solver > Allow tasks to execute concurrently on target is cleared.

Check for unconnected objects in the model

## **Check for unconnected objects**

Identify unconnected lines, input ports, and output ports in the model or subsystem

#### **Passed**

There are no unconnected lines, input ports, or output ports in the model or subsystem.

Check system target file setting

## Verify System target file setting

Check whether Code Generation > System target file is ert.tlc or a System target file derived from ERT

#### **Passed**

The target is ERT or derived from ERT.

Check function specification setting

## **Check model interface settings**

Check whether the Configure arguments for Step function prototype setting in the Configure C Step Function Interface dialog box is cleared.

#### **Passed**

The Configure arguments for Step function prototype setting in the Configure C Step Function Interface dialog box is cleared.

Check for usage of fixed-point instrumentation

## Verify usage of fixed-point instrumentation

Simultaneous usage of fixed-point instrumentation and block reduction can lead to incompatibilities during code inspection

#### **Passed**

Fixed-point instrumentation and block reduction are not used simultaneously.

Check for unsupported blocks

Check for blocks not supported by Simulink Code Inspector

Identify blocks that are not supported by Simulink Code Inspector

### **Passed**

There are no unsupported blocks in this model or subsystem.

Check storage class for workspace variables

## Check storage class for workspace variables referenced by the model

Identify workspace variables using unsupported storage class in the model

#### **Passed**

All of the workspace variables referenced by the model use supported storage classes.

Check GetSet storage class for workspace variables

## Check storage class for workspace variables referenced by the model

Identify GetSet storage class workspace variables in the model that use unsupported specifications

#### **Passed**

All GetSet storage class workspace variables in the model meet a supported specification.

## Check for sample times in the model

## **Check sample times**

Identify continuous, asynchronous, multiple, explicit discrete partition, data driven, union, or variable sample times

#### **Passed**

No continuous, asynchronous, explicit discrete partition, data driven, union, exported discrete or variable sample times were found.

## **Check for explicit partitions**

Identify explicit partitions in the model

#### **Passed**

No explicit partitions were found.

## Check usage of Sources blocks

## **Check Inport blocks**

Identify Inport blocks that might impact compatibility with Simulink Code Inspector

All Inport blocks are compatible with Simulink Code Inspector.

## **Check Inport Shadow blocks**

Identify Inport Shadow blocks that might impact compatibility with Simulink Code Inspector

#### **Passed**

No Inport Shadow blocks were found.

#### **Check Constant blocks**

Identify Constant blocks that might impact compatibility with Simulink Code Inspector

#### **Passed**

No Constant blocks were found.

#### **Check Ground blocks**

Identify Ground blocks that might impact compatibility with Simulink Code Inspector

#### **Passed**

No Ground blocks were found.



Check usage of Signal Routing blocks

#### **Check Bus Creator blocks**

Identify Bus Creator blocks that might impact compatibility with Simulink Code Inspector

#### **Passed**

No Bus Creator blocks were found.

#### **Check Bus Selector blocks**

Identify Bus Selector blocks that might impact compatibility with Simulink Code Inspector

#### **Passed**

No Bus Selector blocks were found.

## **Check Bus Assignment blocks**

Identify Bus Assignment blocks that might impact compatibility with Simulink Code Inspector

#### **Passed**

No Bus Assignment blocks were found.

### **Check Data Store Memory blocks**

Identify Data Store Memory blocks that might impact compatibility with Simulink Code Inspector

#### **Passed**

No Data Store Memory blocks were found.

#### **Check Data Store Read blocks**

Identify Data Store Read blocks that might impact compatibility with Simulink Code Inspector

#### **Passed**

No Data Store Read blocks were found.

#### **Check Data Store Write blocks**

Identify Data Store Write blocks that might impact compatibility with Simulink Code Inspector

#### **Passed**

No Data Store Write blocks were found.

\_\_\_\_\_\_

#### **Check From blocks**

Identify From blocks that might impact compatibility with Simulink Code Inspector

#### **Passed**

No From blocks were found.

#### **Check Goto blocks**

Identify Goto blocks that might impact compatibility with Simulink Code Inspector

#### **Passed**

No Goto blocks were found.

-----

## **Check Merge blocks**

Identify Merge blocks that might impact compatibility with Simulink Code Inspector

#### **Passed**

No Merge blocks were found.

\_\_\_\_\_

## **Check Switch blocks**

Identify Switch blocks that might impact compatibility with Simulink Code Inspector

No Switch blocks were found.

## **Check Multiport Switch blocks**

Identify Multiport Switch blocks that might impact compatibility with Simulink Code Inspector

#### **Passed**

No Multiport Switch blocks were found.

#### **Check Mux blocks**

Identify Mux blocks that might impact compatibility with Simulink Code Inspector

#### **Passed**

All Mux blocks are compatible with Simulink Code Inspector.

#### **Check Demux blocks**

Identify Demux blocks that might impact compatibility with Simulink Code Inspector

#### **Passed**

All Demux blocks are compatible with Simulink Code Inspector.

### **Check Selector blocks**

Identify Selector blocks that might impact compatibility with Simulink Code Inspector

#### **Passed**

No Selector blocks were found.

## **Check Vector Concatenate blocks**

Identify Vector Concatenate blocks that might impact compatibility with Simulink Code Inspector

#### **Passed**

No Vector Concatenate blocks were found.



Check usage of Math Operations blocks

### **Check Absolute blocks**

Identify Absolute blocks that might impact compatibility with Simulink Code Inspector

#### **Passed**

No Absolute blocks were found.

#### **Check Bias blocks**

Identify Bias blocks that might impact compatibility with Simulink Code Inspector

#### **Passed**

No Bias blocks were found.

#### **Check Gain blocks**

Identify Gain blocks that might impact compatibility with Simulink Code Inspector

#### **Passed**

All Gain blocks are compatible with Simulink Code Inspector.

#### **Check Math blocks**

Identify Math blocks that might impact compatibility with Simulink Code Inspector

#### **Passed**

No Math blocks were found.

## **Check Product blocks**

Identify Product blocks that might impact compatibility with Simulink Code Inspector

#### **Passed**

No Product blocks were found.

#### **Check Sum blocks**

Identify Sum blocks that might impact compatibility with Simulink Code Inspector

#### **Passed**

All Sum blocks are compatible with Simulink Code Inspector.

## **Check Trigonometry blocks**

Identify Trigonometry blocks that might impact compatibility with Simulink Code Inspector

#### **Passed**

No Trigonometry blocks were found.

\_\_\_\_\_

## **Check Minmax blocks**

Identify Minmax blocks that might impact compatibility with Simulink Code Inspector

## **Check Rounding Function blocks**

Identify Rounding Function blocks that might impact compatibility with Simulink Code Inspector

#### **Passed**

No Rounding Function blocks were found.

### **Check Reshape blocks**

Identify Reshape blocks that might impact compatibility with Simulink Code Inspector

#### **Passed**

No Reshape blocks were found.

\_\_\_\_\_

## **Check Sign blocks**

Identify Sign blocks that might impact compatibility with Simulink Code Inspector

#### **Passed**

No Sign blocks were found.

## **Check Sqrt blocks**

Identify Sqrt blocks that might impact compatibility with Simulink Code Inspector

#### **Passed**

No Sqrt blocks were found.

## **Check Assignment blocks**

Identify Assignment blocks that might impact compatibility with Simulink Code Inspector

#### **Passed**

No Assignment blocks were found.

#### **Check Polynomial blocks**

Identify Polynomial blocks that might impact compatibility with Simulink Code Inspector

#### **Passed**

No Polynomial blocks were found.

\_\_\_\_\_

## **Check DotProduct blocks**

Identify DotProduct blocks that might impact compatibility with Simulink Code Inspector

### **Passed**

No DotProduct blocks were found.

### **Check UnaryMinus blocks**

Identify UnaryMinus blocks that might impact compatibility with Simulink Code Inspector

#### **Passed**

No UnaryMinus blocks were found.



## Check usage of Signal Attributes blocks

## **Check Data Type Conversion blocks**

Identify Data Type Conversion blocks that might impact compatibility with Simulink Code Inspector

## **Passed**

No Data Type Conversion blocks were found.

### **Check Data Type Duplicate blocks**

Identify Data Type Duplicate blocks that might impact compatibility with Simulink Code Inspector

#### **Passed**

No Data Type Duplicate blocks were found.

## **Check Data Type Propagation blocks**

Identify Data Type Propagation blocks that might impact compatibility with Simulink Code Inspector

## **Passed**

No Data Type Propagation blocks were found.

#### **Check Initial Condition blocks**

Identify Initial Condition blocks that might impact compatibility with Simulink Code Inspector

#### **Passed**

No Initial Condition blocks were found.

## **Check Signal Specification blocks**

Identify Signal Specification blocks that might impact compatibility with Simulink Code Inspector

No Signal Specification blocks were found.

## **Check Signal Conversion blocks**

Identify Signal Conversion blocks that might impact compatibility with Simulink Code Inspector

#### **Passed**

No Signal Conversion blocks were found.

#### **Check Probe blocks**

Identify Probe blocks that might impact compatibility with Simulink Code Inspector

#### **Passed**

No Probe blocks were found.

#### **Check RateTransition blocks**

Identify RateTransition blocks that might impact compatibility with Simulink Code Inspector

## **Passed**

No RateTransition blocks were found.

### **Check Width blocks**

Identify Width blocks that might impact compatibility with Simulink Code Inspector

#### **Passed**

No Width blocks were found.

## **Check Unit Conversion blocks**

Identify Unit Conversion blocks that might impact compatibility with Simulink Code Inspector

#### **Passed**

No Unit Conversion blocks were found.



Check usage of Logical and Bit Operations blocks

## **Check Relational Operator blocks**

Identify Relational Operator blocks that might impact compatibility with Simulink Code Inspector

#### **Passed**

No Relational Operator blocks were found.

## **Check Logic blocks**

Identify Logic blocks that might impact compatibility with Simulink Code Inspector

#### **Passed**

No Logic blocks were found.

## **Check Bitwise Operator blocks**

Identify Bitwise Operator blocks that might impact compatibility with Simulink Code Inspector

#### **Passed**

No Bitwise Operator blocks were found.

#### **Check ArithShift blocks**

Identify ArithShift blocks that might impact compatibility with Simulink Code Inspector

#### **Passed**

No ArithShift blocks were found.

### **Check Combinatorial Logic blocks**

Identify Combinatorial Logic blocks that might impact compatibility with Simulink Code Inspector

#### **Passed**

No Combinatorial Logic blocks were found.



Check usage of Lookup Tables blocks

## Check Lookup Table (n-D) blocks

Identify Lookup Table (n-D) blocks that might impact compatibility with Simulink Code Inspector

#### **Passed**

No Lookup Table (n-D) blocks were found.

## **Check PreLookup blocks**

Identify PreLookup blocks that might impact compatibility with Simulink Code Inspector

## **Passed**

No PreLookup blocks were found.

## **Check Interpolation Using Prelookup (n-D) blocks**

Identify Interpolation Using Prelookup (n-D) blocks that might impact compatibility with Simulink

### **Code Inspector**

#### **Passed**

No Interpolation Using Prelookup (n-D) blocks were found.



## Check usage of User-Defined Function blocks

## **Check S-Function blocks**

Identify S-Function blocks that might impact compatibility with Simulink Code Inspector

#### **Passed**

No S-Function blocks were found.

#### **Check Fcn blocks**

Identify Fcn blocks that might impact compatibility with Simulink Code Inspector

#### **Passed**

No Fcn blocks were found.



## Check usage of Ports and Subsystems blocks

## **Check Enable Port blocks**

Identify Enable Port blocks that might impact compatibility with Simulink Code Inspector

#### **Passed**

No Enable Port blocks were found.

#### **Check Model Reference blocks**

Identify Model Reference blocks that might impact compatibility with Simulink Code Inspector

#### **Passed**

No Model Reference blocks were found.

#### **Check Subsystem blocks**

Identify Subsystem blocks that might impact compatibility with Simulink Code Inspector

#### **Passed**

All Subsystem blocks are compatible with Simulink Code Inspector.

#### **Check Action Subsystem blocks**

Identify Action Subsystem blocks that might impact compatibility with Simulink Code Inspector

### **Passed**

No Action Subsystem blocks were found.

### **Check Trigger Port blocks**

Identify Trigger Port blocks that might impact compatibility with Simulink Code Inspector

#### **Passed**

No Trigger Port blocks were found.

\_\_\_\_\_

#### **Check Action Port blocks**

Identify Action Port blocks that might impact compatibility with Simulink Code Inspector

#### **Passed**

No Action Port blocks were found.

#### **Check If blocks**

Identify If blocks that might impact compatibility with Simulink Code Inspector

#### **Passed**

No If blocks were found.

\_\_\_\_\_

## **Check Function-Call Generator blocks**

Identify Function-Call Generator blocks that might impact compatibility with Simulink Code Inspector

## **Passed**

No Function-Call Generator blocks were found.

\_\_\_\_\_

## **Check Function-Call Split blocks**

Identify Function-Call Split blocks that might impact compatibility with Simulink Code Inspector

## **Passed**

No Function-Call Split blocks were found.

\_\_\_\_\_

## **Check SwitchCase blocks**

Identify SwitchCase blocks that might impact compatibility with Simulink Code Inspector

#### **Passed**

No SwitchCase blocks were found.

\_\_\_\_\_

#### **Check For Iterator blocks**

Identify For Iterator blocks that might impact compatibility with Simulink Code Inspector

#### **Passed**

No For Iterator blocks were found.

#### **Check For Each blocks**

Identify For Each blocks that might impact compatibility with Simulink Code Inspector

#### **Passed**

No For Each blocks were found.

#### **Check State Control blocks**

Identify State Control blocks that might impact compatibility with Simulink Code Inspector

#### **Passed**

No State Control blocks were found.



Check usage of Discontinuities blocks

### **Check Saturate blocks**

Identify Saturate blocks that might impact compatibility with Simulink Code Inspector

#### **Passed**

All Saturate blocks are compatible with Simulink Code Inspector.

## **Check Relay blocks**

Identify Relay blocks that might impact compatibility with Simulink Code Inspector

#### **Passed**

No Relay blocks were found.

## **Check Dead Zone blocks**

Identify Dead Zone blocks that might impact compatibility with Simulink Code Inspector

## **Passed**

No Dead Zone blocks were found.

## Check usage of Sinks blocks

### **Check Outport blocks**

Identify Outport blocks that might impact compatibility with Simulink Code Inspector

#### **Passed**

All Outport blocks are compatible with Simulink Code Inspector.

#### **Check Terminator blocks**

Identify Terminator blocks that might impact compatibility with Simulink Code Inspector

#### **Passed**

No Terminator blocks were found.

## Check usage of Discrete blocks

## **Check Unit Delay blocks**

Identify Unit Delay blocks that might impact compatibility with Simulink Code Inspector

#### **Passed**

All Unit Delay blocks are compatible with Simulink Code Inspector.

## **Check Discrete Integrator blocks**

Identify Discrete Integrator blocks that might impact compatibility with Simulink Code Inspector

## **Passed**

No Discrete Integrator blocks were found.

## **Check Delay blocks**

Identify Delay blocks that might impact compatibility with Simulink Code Inspector

#### **Passed**

No Delay blocks were found.



## Check usage of root Outport blocks

## Verify sample times

Identify root Outport blocks that specify a constant sample time.

#### **Passed**

No root Outport blocks specify a constant sample time.

### Verify root Outports pass buses to parent models as structures

A root Outport block must pass a bus to a parent model as a structure so that Simulink does not introduce a hidden bus conversion block in the parent model.

#### **Passed**

All root outports pass buses as structures.

Check for unsupported Signal Conversion blocks automatically inserted at signals entering block input ports

## Verify no unsupported Signal Conversion blocks are automatically inserted at signals entering block inports

Verification of Signal Conversion blocks that have been automatically inserted is not supported

#### **Passed**

No unsupported Signal Conversion blocks are automatically inserted at signals entering block inports.

## Check usage of buses

#### Check for automatic conversion between virtual to non-virtual buses.

Verify that there is no automatic conversion from virtual to non-virtual buses.

#### **Passed**

There is no automatic conversion between virtual and non-virtual buses.

\_\_\_\_\_

## Verify that blocks in the model do not perform unsupported operations on a bus

Non-virtual blocks must not operate on a virtual bus and Bus Assignment blocks must not operate on a non-virtual bus.

#### **Passed**

No blocks in the model perform an unsupported operation on a bus.

## 🗸 Check for usage of synthesized local data stores

## Verify synthesized local data store usage

Identify signal objects in the model workspace that are referenced as synthesized local data stores by Data Store Read or Data Store Write blocks. If Simulink software finds such a signal object, it creates a hidden Data Store Memory block at the model's root level. This model is not compatible with Simulink Code Inspector.

There are no signal objects in the model workspace referenced as synthesized local data stores by Data Store Read or Data Store Write blocks in this model.

Check usage of global data stores

## Verify global data store usage

Global data store memory blocks may not be used unless parameters are inlined, and their InitialValue is not a tunable parameter

#### **Passed**

There is no unsupported usage of global data stores.

🗸 Check global data stores' name shadow

## Verify global data store's name shadow

Global data store memory blocks may not be used if they are shadowed by local data store memory blocks

#### **Passed**

There is no shadowed usage of global data stores

Check for root Outport blocks being conditionally assigned

## Verify that root outports are not assigned conditionally

Code verification is not supported for submodels for which root outports are assigned by conditionally executed subsystems.

#### **Passed**

The model satisfies the check.

<equation-block>

## Verify conditional input branch execution setting

Check whether 'Signal storage reuse' and 'Enable local block outputs' are set when conditional input branch execution is set

## **Passed**

Conditional input branch execution setting is compatible.

## Verify conditional input branch execution setting when using Multiport Switch blocks

Check whether conditional input branch execution is cleared when using Multiport Switch blocks



Conditional input branch execution setting is compatible.

\_\_\_\_\_

Check usage of Stateflow blocks

## **Check Stateflow blocks**

Identify Stateflow blocks that might impact compatibility with Simulink Code Inspector

#### **Passed**

No Stateflow blocks were found.

Check for Stateflow machine data

## All Stateflow data must be parented by a Stateflow chart

Simulink Code Inspector does not support Stateflow data of machine scope

#### **Passed**

No Stateflow data is of machine scope.

Check for Stateflow machine events

## All Stateflow events must be parented by a Stateflow chart

Simulink Code Inspector does not support Stateflow events of machine scope

#### **Passed**

No Stateflow events are of machine scope.

Check usage of Stateflow charts

No Stateflow charts were found.

\_\_\_\_\_\_

Check usage of Stateflow data

No Stateflow data were found.

\_\_\_\_\_

Check usage of Stateflow events

No Stateflow events were found.

Check usage of Stateflow states

No Stateflow states were found.

<b>②</b>	Check usage of Stateflow junctions  No Stateflow junctions were found.
<b>②</b>	Check usage of Stateflow transitions  No Stateflow transitions were found.
<b>②</b>	Check usage of Stateflow graphical functions  No Stateflow graphical functions were found.

Check usage of Stateflow truth tables No Stateflow truth tables were found.

☑ Check Loop unrolling threshold setting

## Verify Loop unrolling threshold setting

Check whether 'Loop unrolling threshold' is set to a value such that no partially unrolled loops are generated.

#### **Passed**

Loop unrolling threshold setting is compatible.

Check destinations of If and Switchcase blocks

## Check destination Action subsystems of If and Switchcase blocks

Check that Action subsystems connected to same If or Switch Case block uniformly combine or separate their output and update code

#### **Passed**

No Action subsystems that violated the above check were found

Check for root Outport blocks that have non-auto storage class

## Verify that the storage class of root outports is supported

Code verification is not supported for submodels with root outports of non-auto storage class if the parameter 'Pass reusable subsystem outputs as' is set to 'Individual arguments'.



The model satisfies the check.

Check for Terminator blocks connected to Model Reference block outports

## **Check for Model Reference block connectivity**

Identify Model Reference blocks that are connected to Terminator blocks.

#### **Passed**

Not all outports of Model Reference blocks are connected to Terminator blocks.

Check for unsupported propagation of initial condition values

## Check for unsupported propagation of initial condition values

Check if any block propagates initial condition during first time initialization

#### **Passed**

No unsupported propagation of initial condition values detected

Check data type replacement names

Identify replacement names that are not a Simulink Name or a Simulink.AliasType Identify replacement names that are not a Simulink Name or a Simulink.AliasType

#### **Passed**

All replacement names are a Simulink Name or a Simulink. Alias Type

\_\_\_\_\_

Check usage of MATLAB Function Blocks

No MATLAB Function Blocks were found

Check usage of Data in MATLAB Functions

No Data in MATLAB Functions were found

\_\_\_\_\_

\_\_\_\_\_

Check usage of Code in MATLAB Functions

No Code in MATLAB Functions were found

Check MATLAB code used in MATLAB Function blocks

Passed
No MATLAB Function blocks found

Check MATLAB functions defined in Stateflow charts

Passed
No MATLAB functions defined in Stateflow charts

Passed
No MATLAB functions defined in Stateflow charts found

Check called MATLAB functions

Passed
No external MATLAB functions found

Check for multiple sample times in model used as a model reference target

## Check for sample times in model used as model reference target

Identify models used as model reference targets that have multiple sample times

#### **Passed**

Model used as model reference target does not have multiple sample times

Check Treat each discrete rate as a separate task setting

## Verify Treat each discrete rate as a separate task setting

Check whether 'Treat each discrete rate as a separate task' is selected for a model with multiple discrete rates.

## **Passed**

'Treat each discrete rate as a separate task' setting is compatible. This check does not impact a model that has a single rate.

Check model for commented out blocks

Check for commented out blocks in the model Identify blocks in the model that are commented out

#### **Passed**

There are no commented out blocks in the model

Check model for instrumented signals

Check for instrumented signals in the model Identify signals in the model that are instrumented

#### **Passed**

There are no instrumented signals in the model

Check model for void void subsystems that use the same function name

Check function names used by void\_void subsystems in the model Identify void void subsystems in the model that use the same function name

#### **Passed**

Void void subsystems in the model use unique function names

Check n-D Lookup Table blocks for incompatible breakpoint data type

Check n-D Lookup Table blocks for incompatible breakpoint data type

## **Passed**

n-D Lookup Table blocks have compatible breakpoint data type

Check model for reusable subsystems that use the same function interfaces

Check the model for reusable subsystems with the same function interfaces
Identify reusable subsystems with the same function interfaces that are from different library blocks

#### **Passed**

Reusable subsystems with the same function interfaces are from the same library blocks

Check for usage of shared synthesized local data stores

Check for unsupported usage of shared synthesized local Data Store Memory blocks
Identify unsupported hidden data store memory blocks inserted for shared synthesized local data

store memory blocks

#### **Passed**

There are no unsupported hidden data store memory blocks inserted for shared synthesized local data store memory blocks.

Check the code generation folder structure for the model

## Check the code generation folder structure for the model

Check that the code generation folder structure for the model is supported

## **Passed**

The code generation folder structure for the model is supported

Check for unsupported Code Mapping settings

Check for unsupported usage of Storage Class on the Data Defaults tab of the Code Mapping Editor. Identify unsupported Storage Class settings for model element categories on the Data Defaults tab of the Code Mapping Editor.

#### **Passed**

There is no unsupported Storage Class setting for model element categories on the Data Defaults tab of the Code Mapping Editor.

Check model for compiled and graphical block sorted order

## Check for matching compiled and graphical block sorted order in the model

Identify blocks in the compiled block list of the model that do not match graphical block sorted order.

#### **Passed**

All blocks in the compiled block list follow graphical sorted order in the model

Check usage of String blocks

## **Check ASCIIToString blocks**

Identify ASCIIToString blocks that might impact compatibility with Simulink Code Inspector

#### **Passed**

No ASCIIToString blocks were found.

\_\_\_\_\_

## Check usage of shared utilities

## Check code generation symbols settings for shared utils inspection

Code inspection for shared utilities is supported for default values of code generation symbols parameters

#### **Passed**

Values for code generation symbols parameters are supported for shared utilities inspection

## Check standard math library settings for shared utils inspection

Code inspection for shared utilities is supported when standard math library is set to 'C89/C90 (ANSI)'

#### **Passed**

Standard math library settings are supported for shared utilities inspection

## Check code style settings for shared utils inspection

Check that code style settings are supported for shared utils inspection

#### **Passed**

Code style settings are supported for shared utils inspection

Check model arguments for storage classes

## Verify that model arguments do not have a storage class

Check if model arguments have a storage class.

#### **Passed**

The model does not contain any model arguments with a storage class.

Check usage of Stateflow MATLAB action language

No Stateflow MATLAB action language were found.