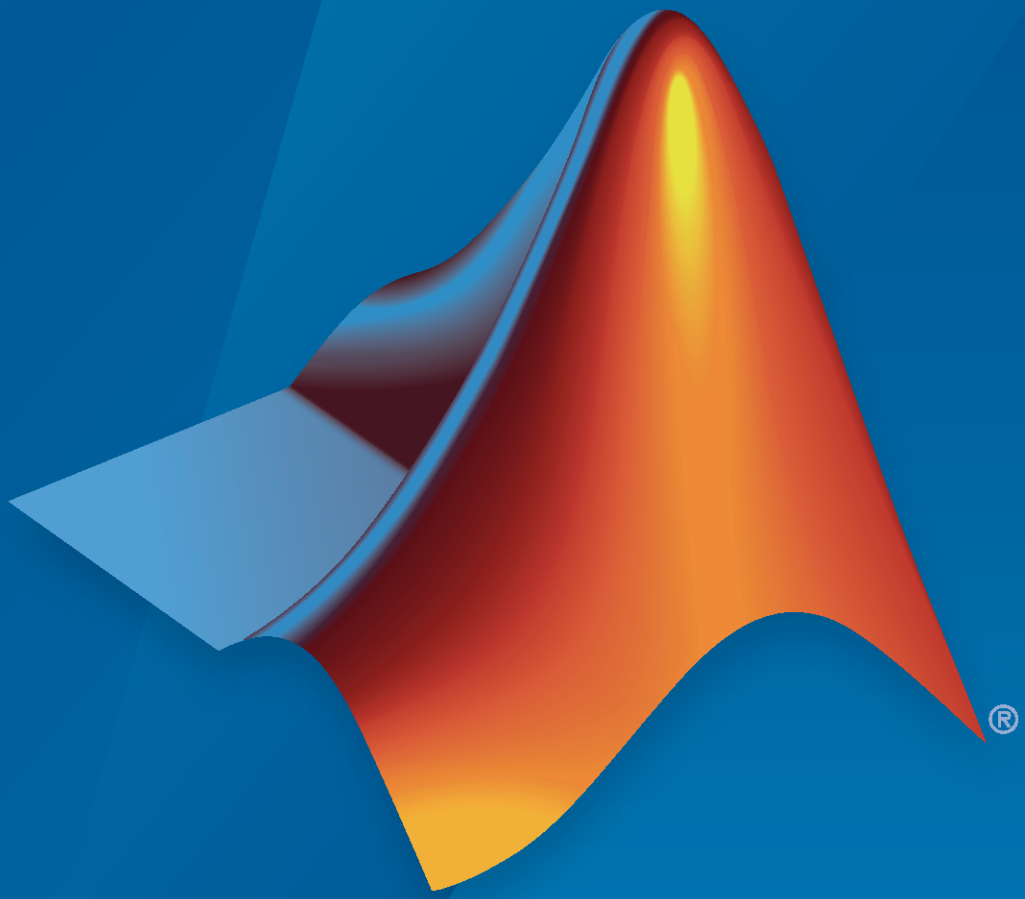
Simulink® PLC Coder™ Release Notes



MATLAB® & SIMULINK®

MathWorks®

How to Contact MathWorks

Latest news: `www.mathworks.com`

Sales and services: `www.mathworks.com/sales_and_services`

User community: `www.mathworks.com/matlabcentral`

Technical support: `www.mathworks.com/support/contact_us`

Phone: 508-647-7000

The MathWorks, Inc.
1 Apple Hill Drive
Natick, MA 01760-2098

Simulink® PLC Coder™ Release Notes

Contents

R2022a

R2021b

R2021a

R2020b

R2020a

R2019b

R2019a

R2018b

R2018a

R2017b

R2017a

R2016b

R2016a

R2015b

R2015a

R2014b

R2014a

R2013b

R2013a

R2012b

R2012a

R2011b

R2011a

R2010b

R2022a

Call back customization for targets

In R2022a, use Simulink PLC Coder to generate custom code for supported target integrated development environments (IDEs). To generate custom code create pre-code and post-code generation custom callback functions. For more information, see “Create Custom Target IDE for Code Generation”.

Externally Defined Blocks: Exclude block definitions as Function blocks

In R2022a, suppress code generation for externally defined blocks by defining the blocks to be eliminated. Include your custom code for the suppressed blocks in the generated code as function block calls. See, “Exclude block definitions as Function blocks”.

Complex data type for sinh, cosh, and tanh functions

In R2022a, Simulink PLC Coder supports complex data types for `sinh`, `cosh`, and `tanh` functions.

Include Descriptions for subsystem input ports, output ports, and bus elements in generated structured text code

In R2022a, the generated structured text code includes descriptions for the subsystem input ports, output ports, and bus elements. For descriptions to be included in the generated code, the input and output ports must be inside the subsystem. Descriptions for bus elements appear as `struct` type descriptions in the generated code. See “Add Subsystem Port and Bus Descriptions in Generated Code”.

Code generation for Signal Editor block

In R2022a, Simulink PLC Coder supports the Signal Editor block for test bench code generation. Starting in R2022a, use the Signal Editor block instead of the Signal Builder block to generate test bench code. Test bench code generation for the Signal Builder block will be removed in a future release.

Functionality being removed or changed

Externally Defined Blocks: Replace externally defined blocks during code generation
Behavior change

In R2022a, the `Replace externally defined blocks during code generation` has been renamed as `Exclude block definitions as functions`.

R2021b

Version: 3.5

New Features

Bug Fixes

Version History

Variable-Size Data Code Generation

In R2021b, you can use Simulink PLC Coder to generate structured text code for:

- Simulink Blocks
- Stateflow® Charts
- MATLAB® Function Blocks

To generate variable-size code, set the subsystem `Function Packaging` to `Inline`. Verify the generated code by generating a testbench for the variable-size data. If the top-level subsystem has variable-size inputs or outputs, the generated testbench code includes test variable-size signal data. See Variable-Size Signal Code Generation.

Simulink PLC Coder Model Advisor Checks: Check your Simulink model for model configuration and block settings

In R2021b, you can use the Simulink Model Advisor PLC Coder checks to check your Simulink models for blocks that are not supported, Stateflow messages, `Inf` data, and so on. For more information, see Model configuration checks overview and Check for blocks and block settings overview.

Externally Defined Blocks: Replace externally defined blocks during code generation

Suppress code generation for externally defined blocks by defining the blocks to be eliminated. In R2021b, you can suppress code generation for:

- Stateflow charts
- MATLAB Function Blocks
- Subsystem blocks

For more information, see Externally Defined Blocks.

Pure function generation

In R2021b, Simulink PLC Coder generates functions instead of function blocks for non-top-level simple subsystems. A simple subsystem is a subsystem that does not have any states. This feature is not supported for Rockwell Automation® IDEs. When you do not select the **Generate functions instead of function block** option for the MULTIPROG® and Phoenix Contact® PC WORX™ IDEs, pure function generation is not supported. See Generate Functions Instead of Function Block.

Functionality being removed or changed

Ladder diagram code generation not supported for 3S-Smart Software Solutions CODESYS Version 2.3 or 3.3 or 3.5 (SP4 or later) and PLCOpen XML

Errors

Starting in R2021b, you cannot generate ladder diagram code for these IDEs:

- 3S-Smart Software Solutions CODESYS Version 2.3 or 3.3 or 3.5 (SP4 or later)

- PLCOpen XML

R2021a

PLC Model Advisor Checks: Check and update your Simulink model for PLC Coder code generation compatibility

In R2021a, Simulink PLC Coder has integrated checks into the Simulink Model Advisor. When you open the Model Advisor in Simulink, you see the checks in the **Simulink PLC Coder** subfolder of the **By Product** or **By Task** folder. With the Model Advisor support in Simulink PLC Coder, you can run industry standard checks on your model or subsystem prior to code generation. For more information, see Run Simulink PLC Coder Model Advisor Checks.

Hook file for custom target keyword list

In R2021a, you can use MATLAB to create a callback hook file. Use the hook file to create a custom target keyword list. Compare the names in your model or subsystem to the keywords in the hook file. Remove matching names from your model or subsystem before code generation. For more information, see Create Custom Target-Based Keyword List.

Signal Builder block time range to generate multi-testbench

In R2021a, you can generate multiple variable-size testbenches for your model or subsystem. The generated testbench size depends on the time duration of the respective signal group in the Signal Builder block. In the Configuration Parameters dialog box, **PLC Code Generation** pane, select the **Use Signal Builder block time range to generate multi-testbench** option. Or, at the command line, enter `PLC_MultiTBSigbuilderTimeRange`. For more information, see Signal Builder Block Time Range to Generate Multi Testbench.

Inline enum cast function

In R2021a, you can inline the generated enum-to-integer or integer-to-enum function. In the Configuration Parameters dialog box, **Identifiers** pane, select the **Inline enum cast function** option. Or, at the command line, enter `PLC_InlineEnumCastFunction`. For more information, see Inline Enum Cast Function.

Custom plugin-based target IDE

In R2021a, use the built-in plugins in Simulink PLC Coder to generate code for your plugin-based custom target IDEs. See Create Custom Target IDE for Code Generation.

Code verification using cosimulation

Learn how to use Open Platform Communication Unified Architecture (OPC UA) to verify your generated code by comparing simulation results to results from code running on a Codesys-based soft PLC. For more information, see Verify Generated Code by Using Cosimulation.

Functionality being removed or changed

Auto import support removed for Siemens SIMATIC Step 7, Rockwell Automation RSLogix 5000, KW-Software MULTIPROG, and Phoenix Contact PC WORX
Behavior change

Starting in R2021a, you cannot import the generated code into these IDEs:

- Siemens SIMATIC Step 7
- Rockwell Automation RSLogix™ 5000
- KW-Software MULTIPROG
- Phoenix Contact PC WORX

R2020b

Version: 3.3

New Features

Bug Fixes

Target-independent absolute-time temporal logic

In R2020b, you can generate code that has a counter-based target-independent implementation for Stateflow absolute-time temporal logic semantics. The counter-based target independent implementation enables testbench verifications, allowing you to generate testbench code to use in your verification workflow. For more information, see Absolute-Time Temporal Logic.

Code generation for IEC integer data types for tiaportal_double target

In R2020b, Simulink PLC Coder generates code for models containing IEC integer data types, except for 64-bit integer data types, when using the TIA Portal: Double Precision as the target IDE.

Tunable parameters that have fixed-point data type

In R2020b, Simulink PLC Coder supports code generation for tunable parameters that have fixed-point data types and generates an error during code generation if there is a mismatch between the data type and value type.

Suppress initial value for externally defined variables in generated code

Perform code generation for distributed models that have externally defined variables, by suppressing their initialization in the generated code. For more information, see Remove Initialization Statements for Externally Defined State Variables.

Code generation by using InOut variable for Siemens targets

In R2020b, Simulink PLC Coder generates code by using InOut variables for Siemens® STEP® 7, and TIA Portal target IDEs.

R2020a

PLC Data dictionary support

The Simulink data dictionary (SLDD) is a persistent repository that stores user model data. SLDD provides data separation, logical partitioning, change detection, dependency analysis, and traceability. For large Model-Based Design (MBD) projects, users can share and reuse model data from multiple models by using SLDD. In R2020a, you can you use Simulink PLC Coder to generate code from SLDD data.

Structured text code generation support for Selectron IDE

In R2020a, you can use Simulink PLC Coder to generate Structured Text (ST) for the Selectron CAP1131 IDE version 12.

Variable enum to integer (int) conversion support

In R2020a Simulink PLC Coder autogenerates the enum type conversion code, that includes enum to int and int to enum conversion. In the Configuration Parameters dialog box, **Identifiers** pane, select the **Generate enum cast function** option. Or, at the command line enter `PLC_GenerateEnumCastFunction`. For more information, see Generate enum Cast Function.

Generate PLC code from library-linked subsystems

You can now generate code for library-linked subsystems by using Simulink PLC Coder.

Display status messages for PLC code generation

When you generate code by using Simulink PLC Coder, you can now see status messages related to the code generation process.

Distributed code generation: Same ssMethod type regardless of SS depth

When you generate a model with an `ssMethod` type that is marked as external in another top-level model, you can now prevent the renaming of `ssMethod` from the name `SS_OUTPUT` to `SS_STEP`. In the Configuration Parameters dialog box, **PLC Code Generation** pane **Identifiers** tab, select the **Keep top level ssMethod name same as non-top level** option. Or, at the command line, enter `PLC_RemoveSSStep`. For more information, see Keep Top-Level ssmethod Name the Same As Non-Top Level.

Improved workflow for unsupported data types for Target IDEs

Simulink PLC Coder now displays either a warning if an unsupported data type for a target IDE is present and is auto converted, or an error if the data type for a target IDE is not auto converted.

Subsystems reference block support

Simulink PLC Coder now supports code generation for subsystem reference blocks.

Virtual bus support

Simulink PLC Coder supports code generation for virtual buses at the top-level subsystem of a model.

Pointer data type elimination for code generation

In R2020a, Simulink PLC Coder eliminates pointer data types as part of the code generation process.

Mixed-order support for Add-On Instruction (AOI) argument list in PLC Ladder Diagram models

In R2020a, the **Function Block Variable** table now preserves the argument order of Add-On Instruction during import and export.

Custom instructions in PLC Ladder Diagram models

In R2020a, support for the Custom Instruction block is added to the PLC library (`plcladderlib`). You can use these blocks to create customized instructions for your ladder models and store these instructions in a user-defined library named `plcuserlib.slx`. You can also import, simulate, and export your ladder instructions by using these custom blocks. For more information, see Create Custom Instruction in PLC Ladder Diagram Models.

Import comments for Ladder Diagram

In R2020a, you can create, import, and export rung comments in Simulink PLC Coder models. For more information, see Import L5X Ladder Files into Simulink and Modeling and Simulation of Ladder Diagrams in Simulink.

Coder.inliner directive for structured text code generation

In R2020a, the `coder.inliner` directive of `always` or `never` now works in the generated structured text code. Use `coder.inliner('never')` in your M file to avoid inlining your code.

Functionality being removed or changed

plccheckforladder is not recommended
Errors

This function has been removed.

plcgenerateladder is not recommended
Errors

This function has been removed.

plcprepareforladder is not recommended
Errors

This function has been removed.

R2019b

Version: 3.1

New Features

Bug Fixes

Testbench Diagnostics: Identify failed output variables when running testbench code

In R2019b, you can generate test bench code with additional diagnostic information that will help you identify output variables causing test bench failures. You can enable this feature by using **Include testbench diagnostic code** GUI option available on the **PLC Code Generation** pane in the Configuration Parameters dialog box or the `PLC_GenerateTestbenchDiagCode` command-line property. To enable this parameter, you must select the **Generate testbench for subsystem** option

The software adds `testVarName` to the generated test bench code and assigns to it, the name of the output variable causing the test failure. If the target IDE supports IEC-61131 `STRING` data type, `testVarName` will also be of `STRING` type. For Rockwell Automation targets, `testVarName` is of Rockwell Structure `STRING` type with `LEN` and `DATA` fields.

Test bench diagnostics works for Structured Text and Ladder Diagram test bench generation.

Support for Mac and Linux platforms

In R2019b, you can use Simulink PLC Coder on both the Mac and Linux® platforms. You can use the software to generate Structured Text code from Simulink models as well as import, model, simulate and generate Ladder diagrams. If the target IDE is not supported on the Mac and Linux platforms, certain functionalities such as `Generate and Import Code for Subsystem` and `Generate, Import, and Verify Code for Subsystem` are not supported.

Ladder blocks enhancements

This release adds new Simulink blocks to the Ladder Diagram library that enable you to model the `CPT`, instructions. In addition, this release also contains improvements to the type transformation functionality of the `DIV` block.

Simulink PLC Coder contextual tab on Simulink Toolstrip

In R2019b, the Simulink Toolstrip replaces the Simulink menu bar. See “Simulink Toolstrip: Access and discover Simulink capabilities when you need them” in the Simulink release notes for more details.

Several Simulink PLC Coder features and buttons are located in contextual tabs. The Simulink Toolstrip contextual tabs appear only when you need to access them.

- To access the **PLC Code** tab, open the **PLC Coder** app from the **Apps** tab on the Simulink Toolstrip.
- To access options in the **PLC Code** tab, such as to **Generate PLC Code** for a Subsystem in your model, select that Subsystem. The options change contextually depending on what you select in the model.
- To change report settings, in the **Settings** tab, select **Report Options**. To access the reports, click the **Open Report** button.

Suppress auto generated type names for MULTIPROG and PC WORX targets

In this release, you can use the **Suppress auto-generated data types** option available on the **PLC Code Generation** pane in the Configuration Parameters dialog box or through the `PLC_SuppressAutoGenType` command-line property. You must set the **Target IDE** drop-down to MULTIPROG or PC WORX targets to see this **Target specific option**.

In previous releases, the software automatically generated named types for array types in your Simulink model.

Report generation enhancements

In R2019b, the software automatically generates a code generation report that shows a mapping between Simulink model objects and locations in the generated code. The report also shows static code metrics about files, global variables, and function blocks. You can still control this behavior through the **Generate traceability report** option available on the **PLC Code Generation > Report** pane in the Configuration Parameters dialog box or through the `PLC_GenerateReport` command-line property.

Improved name handling in the generated code

In this release, the software implements improved name handling functionality for PLC target IDEs such as Rockwell Automation Studio 5000 that have a maximum limit on the character length of operand tags, variable and function names, and AOI names. As a result, the software automatically adds a prefix and truncates names that exceed the maximum limit of the target IDE. If the truncated version of the name already exists, the software automatically increments the prefix.

R2019a

Ladder diagram import

You can now import a Rockwell Automation Ladder Diagram Controller or an AOI organizational unit into Simulink. When you import the L5X file, the model contains ladder diagram blocks that are mapped from the ladder diagram contents. You can simulate and edit the imported models, and generate code for the Rockwell Automation IDE.

To import a valid L5X file into Simulink use the `plcimportladder` function. For more information, see Import L5X Ladder Files into Simulink.

Modeling and simulation of ladder diagram in Simulink

You can create and simulate a ladder diagram in Simulink by using blocks in the PLC Ladder Diagram library. You can simulate the ladder diagram model and generate ladder diagram code for the Rockwell Automation IDE. You can also generate a test bench for the Rockwell Automation IDE. For more information, see Modelling and Simulation of Ladder Diagrams in Simulink.

Ladder diagram code generation for Simulink model

You can generate ladder diagram code from:

- A Simulink model that results from importing the L5X ladder logic into Simulink.
- A Simulink model that you build by using ladder diagram blocks in the PLC Ladder Diagram library.

For more information, see Generating Ladder Code from Simulink.

Version History

Previously, you used a Stateflow chart to model ladder logic and generate code. Now, you can build a ladder logic model in Simulink by using the Simulink PLC Coder Ladder Diagram library and generate code from the model. These functions associated with Stateflow ladder code generation are being removed in a future release:

- `plccheckforladder`
- `plcprepareforladder`
- `plcgenerateladder`

To use these functions, you must add the path `plccoder/old_ladder_commands`. Those functions are no longer on the default path.

To view Simulink PLC Coder documentation for ladder modeling using Stateflow charts, refer to these links:

- Ladder Diagram Generation for PLC Controllers
- Prepare Chart for Ladder Diagram Generation
- Generate Ladder Diagram Code from Stateflow Chart
- Import Ladder Diagram Code to CODESYS 3.5 IDE and Validate Diagram
- Restrictions on Stateflow Chart for Ladder Diagram Generation

- Import L5X Ladder Files into Simulink
- `plcimportladder`

Ladder diagram testbench generation

In R2019a, you can now generate PLC ladder test benches for the Rockwell Automation IDE. You can use a test bench to verify that the generated PLC ladder diagram code produces accurate results by using the Rockwell Automation IDE. To generate a test bench, the top organizational unit must be an AOI Runner block. For more information, see Verify Generated Ladder Code.

To verify the generated code against the Simulink model, create a test bench using `plcgeneratecode`.

If you import an L5X file into Simulink and want to use the test bench to verify the L5X file, use `plcgenerateunnertb`.

Functionality being removed or changed

plccheckforladder is not recommended
Still runs

This function will be removed in a future release. Avoid using this function. To use the function, you must add the path `plccoder/old_ladder_commands`.

plcprepareforladder is not recommended
Still runs

This function will be removed in a future release. Avoid using this function. To use the function, you must add the path `plccoder/old_ladder_commands`.

plcladdergenerate is not recommended
Still runs

This function will be removed in a future release. Avoid using this function. To use the function, you must add the path `plccoder/old_ladder_commands`.

R2018b

Version: 2.6

New Features

Bug Fixes

Simulink PLC Coder Qualification: Qualify Simulink PLC Coder for use with ISO 26262

The TÜV SÜD has certified Simulink PLC Coder for use in development processes that must comply with ISO 26262 or derivative standards.

The Model-Based Design for ISO 26262 document provides suggestions for leveraging the Simulink PLC Coder for Model-Based Design when applying the ISO 26262 standard.

For more information about the IEC Certification Kit support for ISO 26262, see:

- Access Supporting Artifacts for ISO 26262 (IEC Certification Kit)
- Access Certification Artifacts for Simulink PLC Coder (IEC Certification Kit)

Ladder logic import enhancements

Ladder logic import now supports these ladder diagram elements:

- Copy file (COP) block.
- Aliases for global or controller variables. In the previous release, alias variables were supported only inside program organizational units (POU).
- User-defined types (UDT) for non-scalar members. In the previous release, UDT was supported only for scalar members.

For more information, see Supported Features.

PLC IDE test kit

In R2018b, you can test the compatibility of a new target IDE with Simulink PLC Coder generated code by using the PLC IDE test kit. The test kit contains a common set of example files generated for all the supported PLC target IDEs. If the new target IDE is compatible to the IEC 61131-3 standard, import, compile, and run the files in `Generic_IEC61131-3_Packages` on the new IDE. If the new IDE is similar to one of other Simulink PLC Coder supported IDEs, use the corresponding test kit files.

To access the test kit files, enter the following in the MATLAB command line:

```
cd(fullfile(matlabroot,'toolbox','plccoder','plccoderdemos','plcide_test_kit'))
```

Motion API example

The Simulating and Generating Structured Text Code for Rockwell Motion Instructions example shows you how to model Rockwell motion API instruction by using Stateflow charts for simulation and code generation. This representation can then be used to generate Structured Text code by using the `plcgeneratemotionapicode` function. For more information, see Simulation and Code Generation of Motion Instructions.

Support for Simulink.fileGenControl in PLC Coder

You can now specify the root folder for the code generation files by using the `CodeGenFolder` property of the `Simulink.fileGenControl` object. In previous releases, you used the **Code**

Output Directory option available in the **PLC Code Generation** pane in the Configuration Parameters dialog box or the `PLC_OutputDir` command-line property. The location specified in the **Code Output Directory** takes precedence over the location specified by using `Simulink.fileGenControl`.

Support for spmd in PLC Coder

You can now use the `spmd` function from the Parallel Computing Toolbox™ and generate PLC code for multiple models in parallel on workers of parallel pool. You must ensure that the codegen directory for each model is independent of one another to avoid conflicts. For example, to generate PLC code for the examples in the Simulink PLC Coder toolbox, use the following lines of code.

```
models = {'plcdemo_airport_conveyer', 'plcdemo_cruise_control', 'plcdemo_eml_tankcontrol' }
subsystems = {'Controller', 'Controller', 'TankControl'};

spmd
    mkdir(['myfolder' num2str(labindex)])
end

spmd(3)
    back=cd;
    cd(['myfolder' num2str(labindex)]);
    %do stuff...

    open_system(models{labindex});
    plcgeneratecode([models{labindex} '/' subsystems{labindex}]);
    close_system(models{labindex});
    cd(back)
end
```

R2018a

Version: 2.5

New Features

Bug Fixes

Ladder Logic Import: Convert Rockwell PLC ladder diagrams to Simulink models

In R2018a, you can import ladder diagrams created with Rockwell Automation IDEs such as RSLogix 5000 and Studio 5000 into the Simulink environment. The ladder import feature supports ladder diagram semantics such as Boolean variables, multiple rungs, basic ladder functions, and control elements like jumps. For more information, see Import Ladder Diagram into Simulink.

Use the `plcimportladder` function to import the ladder diagrams into Simulink. You can edit the imported models in Simulink and perform simulation and code generation on the imported models. For more information, see Import L5X Ladder Files into Simulink.

External Mode Logging: Collect run-time data on supported targets, visualize, and monitor logging data for Rockwell PLC targets

In R2018a, you can generate code with logging instrumentation to collect run-time data on PLC targets. You can enable this feature by using **Generate logging code** GUI option or the `PLC_GenerateLoggingCode` command-line property. The PLC target IDEs must have support for `inout` variables. See, External Mode Logging, and Generate Structured Text Code with Logging Instrumentation.

For Rockwell Automation targets, you can set up an Open Platform Communications (OPC) server and use the Simulation Data Inspector (SDI) in Simulink to visualize and monitor the logging data. For more information, see Use the Simulation Data Inspector to Visualize and Monitor the Logging Data.

Function Inlining: Control inlining of math function calls in the generated code for RSLogix IDE

In R2018a, Simulink PLC Coder attempts to inline math function calls in the generated code for target IDEs from RSLogix. You can control inlining of function calls by using **Aggressively inline Structured Text function calls** GUI option or the `PLC_EnableAggressiveInlining` command-line property. This option allows you to generate Structured Text with the least possible number of function blocks. For more information, see Aggressively Inline Structured Text Function Calls.

ssmethod Optimization: Control generation of ssmethod type for the top-level subsystem in the generated code

You can now remove the `ssmethod` type from the top-level subsystem argument interface by using the **Remove top level subsystem ssmethod type** GUI option or the `PLC_RemoveTopFBSSMethodType` command-line property. When this option is enabled, Simulink PLC Coder removes the `ssmethod` type and converts the subsystem initialization code from switch case statement to conditional `if` statement. As a result, the generated code has the same interface as the model subsystem. For more information, see Remove Top-level Subsystem ssmethod Type.

Identifier Length Customization: Specify the maximum character limit in function, type definition, and variable names

In R2018a, you can generate code with long name identifiers by using the **Override target default maximum identifier length** GUI option or the `PLC_OverrideDefaultNameLength` command-line

property. Previously, if the **Maximum identifier length** is more than the value allowed by standard versions of the target IDE, Simulink PLC Coder defaults to the maximum identifier length of the target IDE.

Now, if your custom target IDE version supports long name identifiers, you can use the **Override target default maximum identifier length** and **Maximum identifier length** options to specify the maximum number of characters in generated function, type definition, and variable names. See, PLC Coder: Symbols.

Support for using enum type as symbol names in the generated code

In R2018a, you can use enum names as the symbols names instead of enum values. To use this option, the PLC target IDE must support enum type. Use the **Override target default enum behavior** GUI option or the `PLC_GenerateEnumSymbolicName` command-line property to enable this feature. For more information, see Override Target Default enum Name Behavior.

Print diagnostic messages to console in CLI mode

In R2018a, Simulink PLC Coder displays errors, warnings, and other information in the Command Window when code generation is invoked from the command-line interface.

Emit datatypeWorksheet tags for PCWorx IDE

In R2018a, you can use the **Emit datatype worksheet tags for PCWorx** GUI option or the `PLC_EmitDatatypeWorkSheet` command-line property to control whether datatypeWorksheet tags are represented in code generated for PC WORX targets. This option allows you to have finer control and generate multiple `datatypeWorksheet` definitions. See, Emit Datatype Worksheet Tags for PCWorx.

R2017b

Version: 2.4

New Features

Bug Fixes

Code Optimization for Reusable Subsystems: Generate more efficient code for reusable subsystems

In 2017b, Simulink PLC Coder generates efficient code for reusable subsystems by performing optimization on function input, output, and parameter variables.

Function Block Instance Naming: Control naming by using instance names of reusable subsystems

Specify how Simulink PLC Coder software is to name the Function block instance it generates for the subsystem. You can use the following Configuration Parameter to select between index-based names or model subsystem names for the generated function block instances. Use the block parameters dialog box to specify separate code generation function name.

GUI Option	Command-Line Property	Description
Use subsystem instance name as function block instance name	`PLC_FBUseSubsystemInstanceName`	Specify whether to use subsystem instance names or index-based instance names.

Named Constant Inlining: Control handling of named constants in generated code

You can now control inlining of global named constants in the generated code. Use the **Inline named constants** GUI option or the `PLC_InlineNamedConstant` command-line property to enable this feature.

MATLAB Function Block Variable Reuse Control: Improve readability of the generated code

In R2017b, you can control the reuse of MATLAB Function Block variables in the generated code by using the **Reuse MATLAB Function block variable** GUI option or the `PLC_ReuseMLFcnVariable` command-line property.

Control launch behavior of the Code Generation Report

In R2017b, the **Open report automatically** GUI option or the `PLC_LaunchReport` command-line property controls the automatic opening of the Code Generation Report. By default, the report viewer is not opened automatically after PLC code generation.

Code generation folder

In R2017b, you can also use the **Code generation folder** option in Simulink preferences to specify the output folder for code generation. Previously, you could only use the **Code Output Directory** option available on the **PLC Code Generation** node in the Configuration Parameters window to set the output folder.

The location set by using PLC Coder options has higher priority over Simulink preferences.

ASCII Encoding for Structured Text XML

In R2017b, Simulink PLC Coder generates IEC61131-3 Structured Text XML with ASCII encoding. Previously, PLC coder generated Structured Text XML with UTF-8 coding causing the generated XML to become invalid when comments in the code had accented characters.

R2017a

Version: 2.3

New Features

Bug Fixes

Code Optimization for Initialization Code: Optimize generated code by removing redundant timer initialization calls

In R2017a, Simulink PLC Coder generates optimized initialization code when Stateflow charts have absolute time temporal logic.

Previously, if the Stateflow chart had absolute time temporal logic, the generated code had multiple instances of the timer initialization call. With this change, the software scans for and finds consecutive redundant timer initialization calls and removes these statements, resulting in efficient code.

rand Function Support: Generate code for rand functions on PLC IDEs that support uint32

Previously, Simulink PLC Coder did not support random number functions.

In R2017a, Simulink PLC Coder generates code for MATLAB Function blocks that use `rand` functions from the library. PLC IDEs must support the `uint32` data type. The software has conformance checks to report diagnostics for incompatible targets.

Syntax Highlighting in Code Generation Report: Read generated code more easily with syntax highlighting

In 2017a, enhancements in syntax highlighting greatly improve the readability of the code generated with Simulink PLC Coder.

Previously, the HTML code generation report used only two highlighting colors: blue for code and green for comments. With this change, PLC-specific keywords are highlighted in blue and the rest of the code is in black. Comments are still highlighted in green.

Code Optimization for Unused Stateflow Events: Generate more efficient code for unused events

In R2017a,when you have unused events in Stateflow charts, Simulink PLC Coder optimizes the generated code to remove these unused events.

FB Call ssmethod output assignment optimization

In R2017a, Simulink PLC Coder optimizes the `ssmethod` call output value assignments by generating code for step and output methods only when the output values are used.

STEP 7 and TIA Portal INOUT var check

STEP 7 and TIA Portal IDE do not support `INOUT` variables for single precision targets.

In R2017a, if the Simulink model has MATLAB Function blocks that use inplace variables, Simulink PLC Coder generates code by converting `INOUT` variables to standard input and output variables. However, if the MATLAB Function block is marked as reusable, conversion is not possible and a conformance check error is issued.

B&R Automation Studio 4 and Beckhoff TwinCAT 3: Generate code for these IDEs

In R2017a, you can generate Structured Text code dedicated to these IDEs:

- B&R Automation Studio™ 4
- Beckhoff® TwinCAT® 3

The Target IDE drop-down list in Simulink Configuration Parameters shows these IDEs as options. You can also use the command-line API to select these two targets and generate code.

R2016b

Version: 2.2

New Features

Bug Fixes

Ladder Logic Support: Generate ladder diagrams from Stateflow charts for CODESYS 3.5 IDE and Rockwell Automation AOIs

In R2016b, you can generate Ladder Diagram code in PLC Open XML format from Stateflow charts. You can import the generated code to IDEs such as CODESYS 3.5 and Rockwell Automation AOIs, and view them as ladder diagrams. You can also validate the generated code using one of the following:

- Create a validation model containing the Ladder Diagram code. You can compare the output of the model with the original Stateflow chart.
- Create testbench code. You can import the generated code and the testbench code to the CODESYS 3.5 IDE and verify your generated code against the testbench.

For more information, see Ladder Diagram Generation for PLC Controllers.

Rockwell Automation IDE Support: Generate code for RSLogix 5000 V20 and Studio 5000 Logix Designer V24 IDEs

In R2016b, you can generate Structured Text for newer versions of the following Rockwell Automation IDEs:

- RSLogix 5000 (version 20)
- Studio 5000 Logix Designer (version 24)

For more information, see Target IDE.

Multirate Support: Generate code from multirate models for Siemens IDEs and Rockwell Automation AOIs

In R2016b, you can generate code from multirate models for the following Siemens IDEs:

- Siemens SIMATIC® STEP 7
- Siemens TIA Portal
- Rockwell Automation Studio 5000 Logix Designer for AOI format
- Rockwell Automation RSLogix 5000 for AOI format

For information on support for multirate models in Simulink PLC Coder, see Generated Code Structure for Multirate Models and Multirate Model Limitations.

Global Variables for Rockwell Automation IDEs: Generate code for global variables by using INOUT variables for Rockwell Automation AOIs

In R2016b, if you generate code for Rockwell Automation IDEs using Add-On Instruction (AOI) constructs, your models can use global data, for instance, in Data Store Memory blocks. The global data appear as `INOUT` variables in generated code. The target IDEs supported for this feature are:

- Rockwell Automation Studio 5000 Logix Designer for AOI format
- Rockwell Automation RSLogix 5000 for AOI format

Previously, you could not use models with global data to generate code for Rockwell Automation IDEs using Add-On Instruction (AOI) constructs, because global variables are not supported in AOIs.

For more information on the workflow, see Generate Global Variables from Signals in Model.

Improved Code for Reusable Subsystems: Generate better reusable code for reusable subsystems

In R2016b, you can generate better reusable code for reusable subsystems.

Previously, if the same subsystem had multiple instances and some instances had constant inputs, the software generated separate function blocks for each instance. With this change, the software does not consider whether the inputs to the subsystem are constant and generates one function block for the multiple instances.

For more information, see Generate reusable code.

R2016a

Version: 2.1

New Features

Bug Fixes

INOUT Variable Support: Generate INOUT variables for MATLAB Function and Truth Table blocks that use the same name for input and output data

In R2016a, you can generate code for models that contain a MATLAB Function block or a Truth Table block when the block has input and output variables with the same name. The generated code uses `InOut` parameters for those variables.

This capability is not supported if you select:

- **Rockwell RSLogix 5000: Routine** as your target IDE, because the Rockwell Automation RSLogix 5000 routines do not support input, output or `InOut` parameters. For this target IDE, you cannot generate code if a MATLAB Function block or a Truth Table block has input and output variables with the same name.
- **KW-Software MULTIPROG 5.0** or **Phoenix Contact PC WORKX 6.0** as your target IDE, and the variables have array or structure data types.

Alias Data Type Support: Optionally preserve alias names for data types in generated code to help integration with target-specific data types

In R2016a, you can preserve alias data types from your model in generated code.

You can create an alias for a built-in Simulink data type by using the `Simulink.AliasType` class. If you assign an alias data type to signals and parameters in your model, you can preserve the alias data type in your generated code.

For instance, you can use alias names that denote safe data types in your generated code. Using these safe data types, you can conform to PLCopen safety specifications that require clear differentiation between safety-relevant signals and standard signals.

For more information, see Preserve Alias Type Names for Data Types.

Simulink Requirements Links: Embed requirements links as comments in generated code

In R2016a, if you create links to requirements documents from your model with the Simulink Verification and Validation™ software, the links appear in generated code comments. When you view the code in the Code Generation Report, you can open the links from the comments.

See View Requirements Links from Generated Code.

Simulink Design Verifier Integration: Generate code with multiple test benches from test harness models created with Simulink Design Verifier

In R2016a, if the input to your subsystem consists of multiple signal groups, you can generate code with multiple test benches.

To provide multiple signal groups as inputs, do one of the following:

- Use a Signal Builder block with multiple signal groups to provide inputs to the subsystem.
- Create a test harness model from the subsystem with Simulink Design Verifier™. In the test harness model, a Signal Builder block with one or more signal groups is created to test the model. Copy this block from the test harness model and use it to provide inputs to your subsystem.

For more information on generating multiple test benches, see Verify Generated Code with Multiple Test Benches.

64-bit Windows 7 Support for Siemens STEP 7 and RSLogix 5000 IDEs: Generate, import, and verify code for these IDEs

In R2016a, you can generate, import, and verify code in Siemens STEP 7 and RSLogix 5000 IDEs for 64-bit Windows® 7.

CODESYS 3.5 POU Block Description Support: Generate block descriptions as POU descriptions in code generated for CODESYS 3.5

In R2016a, when you generate code for the CoDeSys 3.5 IDE, Simulink PLC Coder can propagate block descriptions from the model into the `documentation` XML tag. When you import the generated code into the CoDeSys 3.5 IDE, the IDE parses the content of this tag and provides readable descriptions of the function blocks in your code.

For more information, see Propagate Block Descriptions to Code Comments.

Code generation for models containing enum to Integer Conversion

In R2016a, you can generate code from models that convert constant values from `enum` to integer data types. Use the Data Type Conversion blocks to perform this conversion.

Code Generation for subsystems with no input or output

In R2016a, you can generate code for non-empty subsystems that do not have inputs or outputs. Previously, you encountered an error when generating code from such subsystems. For code generation from a subsystem with no inputs or outputs, you must set the **Function packaging** parameter of the block to `Reusable function`.

However, you cannot generate testbench code for such subsystems because testbench code requires output from subsystems.

Support for KW-Software MULTIPROG 5.5 IDE

Simulink PLC Coder now supports version 5.5 of the KW-Software MULTIPROG IDE.

R2015b

Version: 2.0

New Features

Bug Fixes

SIEMENS TIA Portal V12 and V13 IDE Support: Generate code for these IDEs

Simulink PLC Coder supports the Siemens TIA Portal target IDE. Versions V12 and V13 are supported.

For more information on:

- How to select the IDE, see Target IDE.
- How to integrate generated code with an existing Siemens TIA Portal project, see Integrate Generated Code with Siemens TIA Portal Projects.

Streamlined Target IDE Selection: Choose target IDE more quickly

In Simulink Configuration Parameters, on the **PLC Code Generation** pane, the default **Target IDE** list shows a reduced subset of target IDEs for easier navigation. You can customize this reduced **Target IDE** list and specify more frequently used IDEs using the `plccoderpref` function. For more information, see plccoderpref.

To see all target IDEs supported by Simulink PLC Coder, select **Show full target list**. For more information, see Show full target list.

Absolute Time Temporal Logic by Using IEC 61131 Timer: Generate code for this Stateflow construct

Simulink PLC Coder supports code generation from models with Stateflow charts that use Absolute Time Temporal Logic.

For more information, see Stateflow Chart with Absolute Time Temporal Logic.

Global Variables for Siemens IDEs: Generate code for global data store memory using Simulink.Signal objects for Siemens STEP 7 and TIA Portal IDEs

Simulink PLC Coder supports code generation for global data store memory using `Simulink.Signal` objects for Siemens STEP 7 and TIA Portal IDEs.

Additional Math Function Support: Generate code for hyperbolic functions

Simulink PLC Coder supports code generation from models that contain hyperbolic functions such as `sinh`, `tanh`, etc.

Code Optimizations: Generate more efficient code for type casts

Simulink PLC Coder generates more efficient code by removing unnecessary type casts.

Linked Subsystems Code Verification: Verify that generated code results match simulation results

Simulink PLC Coder supports test bench code generation from linked subsystems.

For more information on test bench code, see Verification.

Improved code for global data store memory using Simulink.Signal objects

Simulink PLC Coder generates more efficient code for global data store memory using `Simulink.Signal` objects.

Code generation from models with atomic subcharts

Simulink PLC Coder allows code generation from models that contain Stateflow charts with atomic subcharts in them. However, before code generation, the software converts the atomic subcharts to regular subcharts.

Unnecessary variables removed from generated code for Data Store Memory blocks

If your model contains Data Store Memory blocks, the code generated by Simulink PLC Coder no longer contains unnecessary variables related to code coverage.

R2015a

Version: 1.9

New Features

Bug Fixes

Code generation for 3S-Smart Software Solutions CoDeSys V3.5 IDE

Simulink PLC Coder supports CoDeSys IDE V3.5.

Generation of code that preserves variable names in MATLAB Function blocks

The generated Structured Text from Simulink PLC Coder retains the names of variables defined in MATLAB functions.

This behavior allows you to easily map variables defined in your MATLAB code to the ones in the generated Structured Text. Earlier, certain optimizations during code generation caused reuse of variable names.

R2014b

Version: 1.8

New Features

Bug Fixes

Code generation for Rexroth IndraWorks version 13V12 IDE

Simulink PLC Coder supports Rexroth IndraWorks version 13V12 IDE.

Code generation for OMRON Sysmac Studio v1.09 IDE

Simulink PLC Coder supports OMRON® Sysmac® Studio v1.09 IDE.

Code generation support for exported functions in Stateflow

Simulink PLC Coder supports code generation for Stateflow exported functions. Functions can be defined in one Stateflow chart, and then exported and called by other charts.

Code generation support for global data store memory using Simulink.Signal object

Simulink PLC Coder supports code generation for global data store memory using `Simulink.Signal` objects. This support applies to IDEs which support global variables.

Variable names preserved for function block inputs and outputs

For PLC code generation, Simulink PLC Coder preserves function block input and output variable names in generated code. If variable names conflict with reserved names or keywords in the target, they are changed.

R2014a

Version: 1.7

New Features

Bug Fixes

Static code metrics report

Simulink PLC Coder reports key characteristics of the generated code, such as number of variables and lines of code in each function block. For more information, see Generate a Static Code Metrics Report.

Code generation for Siemens STEP 7 V5.5 IDE, B&R Automation Studio 4 IDE, and Beckhoff TwinCAT 3 IDE

Simulink PLC Coder supports Siemens STEP 7 version 5.5, B&R Automation Studio® 4, and Beckhoff TwinCAT 3.

Model block description in generated code for CoDeSys 2.3 IDE

Simulink PLC Coder generates a model block description in the code for the CoDeSys 2.3 IDE.

Simulink.Parameter description in generated code for Codesys 2.3 IDE

Simulink PLC Coder generates a `Simulink.Parameter` description in the code for the CoDeSys 2.3 IDE.

Change in Diagnostic Viewer launch behavior after code generation

After PLC code generation, the Diagnostic Viewer window showing the PLC Code Generation log no longer launches automatically. Instead, a "View diagnostics" hyperlink appears at the bottom of the Simulink model window. This link opens the Diagnostic Viewer and the PLC Code Generation log with links to the generated code.

R2013b

Version: 1.6

New Features

Bug Fixes

Masked parameters for atomic subsystems

Mask parameters for subsystems now map to function block inputs in the generated code.

To see how mask parameters map to generated code, see Generated Code Structure for Subsystem Mask Parameters.

Reusable code for intrinsic functions

For internal MATLAB functions, such as `atan2`, Simulink PLC Coder generates a single copy of the function in the generated code.

Millisecond and microsecond units with absolute-time temporal logic

You can generate code for Stateflow absolute-time temporal logic that uses the millisecond (`msec`) and microsecond (`usec`) time units.

PC WORX IDE support improvements including enhanced support for global tunable parameters

Simulink PLC Coder generates code for the PC WORX IDE with the following improvements:

- Global tunable parameters that are structures and array data types are initialized in a `PLC_INIT_PARAMETERS` function block. For more information, see Global Tunable Parameter Initialization for PC WORX.
- `floor` and `ceil` rounding is supported.
- Array type names incorporate data type description and size.

  For example, if you have a 51-element array of real data, the generated code for the array data type is:

  ```
  PLC_ARRAY_0_50_LREAL: ARRAY [0..50] OF LREAL;
  ```
- Code generation header comments and block description comments are in the body of the generated code, and are therefore visible when you import the code into the IDE.

Temporary variable minimization

The coder minimizes the number of temporary variables. This optimization improves code quality and memory usage.

Relative tolerance for test bench data comparison

When checking single and double data type values, the test bench now uses a relative error tolerance. Integer data type comparisons in the test bench still use an absolute tolerance.

To learn more about test bench data comparison, see How Test Bench Verification Works.

R2013a

Version: 1.5

New Features

Bug Fixes

Code generation for OMRON Sysmac Studio IDE

The Simulink PLC Coder software now supports OMRON Sysmac Studio Version 1.04 or later.

Code generation for multirate models in single-tasking mode

The Simulink PLC Coder software can now generate code for multirate models in single-tasking mode. For more information, see Generated Code Structure for Multirate Models and Multirate Model Restrictions.

To open an example that shows how to generate code from a multirate model, at the command line, enter:

```
plcdemo_multirate
```

R2012b

Version: 1.4

New Features

Bug Fixes

Workflow for behavioral simulation and code generation of motion instructions for RSLogix 5000 IDE

The Simulink PLC Coder software now supports a workflow for the behavioral simulation and code generation of motion instructions for the Rockwell Automation RSLogix 5000 IDE. For more information, see Simulation and Code Generation of Motion Instructions.

Code generation report with bidirectional traceability between model and code

Simulink PLC Coder now creates and displays a traceability report file. You can also opt to display the report in a model Web view. See the following Configuration Parameter options.

GUI option	Command-Line Property	Description
Generate traceability report	`PLC_GenerateReport`	Specify whether to create code generation report.
Generate model Web view	`PLC_GenerateWebview`	Include the model Web view in the code generation report to navigate between the code and model within the same window. You can share your model and generated code outside of the MATLAB environment.

For more information, see Information in Code Generation Reports.

Propagation of block descriptions to generated code comments and RSLogix 5000 AOI/routine descriptions

The Simulink PLC Coder software now propagates block comments to generated code for all target IDEs. For more information, see Propagation of Block Descriptions.

For Rockwell Automation RSLogix 5000 AOI/routine target IDEs, the coder also generates the subsystem block description text as an AOI or routine description L5X XML tag. The IDE can then import the tag as part of AOI and routine definition in the generated code.

Code generation optimizations for efficient casts and signal reuse

An Optimization pane has been added to the Configuration Parameters dialog box PLC Coder node. This pane contains the following parameters:

GUI option	Command-Line Property	Description
Signal storage reuse	`PLC_PLCEnableVarReuse`	Reuse signal memory.

GUI option	Command-Line Property	Description
Remove code from floating-point to integer conversions that wraps out-of-range values	`PLC_PLCEnableEfficientCast`	Enable code removal for efficient casts.
Loop unrolling threshold	`PLC_RollThreshold`	Specify the minimum signal or parameter width for which a for loop is generated.

For more information, see Model Architecture and Design.

Internal signals available as optional AOI outputs for debugging in RSLogix 5000 IDE

The Simulink PLC Coder software now generates code where test point outputs to the top-level subsystem are mapped to optional AOI outputs for RSLogix 5000 IDE. In the generated code, the variable tags that correspond to the test points have the property `Required=false`.

For more information, see Internal Signals for Debugging in RSLogix 5000 IDE.

Rockwell Automation RSLogix 5000 IDE Version 19

The Simulink PLC Coder software now supports Rockwell Automation RSLogix 5000 IDE Version 19.

Absolute Time Temporal Logic

The Simulink PLC Coder product now enables absolute time temporal logic for all supported IDEs. In previous releases, this capability was supported only for the Rockwell Automation RSLogix IDE. For more information, see Integrate Absolute Time Temporal Logic Code.

R2012a

Version: 1.3

New Features

Bug Fixes

Code Generation for Rockwell Automation RSLogix 5000 Routines

The Simulink PLC Coder software now generates code for routines from the Rockwell Automation RSLogix 5000 IDE.

- Load the code generated from routines without first restarting the Rockwell Automation RSLogix 5000 PLC. You can now:
- Take advantage of RSLogix user defined types (UDTs) to preserve model hierarchy in routine code and represent model.
- Observe that reusable subsystems become separate routine instances and access instance data in program UDTs.

To accommodate this capability:

- In the Configuration Parameters dialog box **PLC Code Generation** > **Target IDE** parameter, the `Rockwell RSLogix 5000 17, 18: Routine` option was added.
- In the Configuration Parameters dialog box **PLC Code Generation** > **Target IDE** parameter, the `Rockwell RSLogix 5000 17, 18` option was changed to `Rockwell RSLogix 5000 17, 18: AOI`. This renamed option continues to generate code for Add-On instruction constructs, as in previous releases.
- In the command-line `PLC_TargetIDE` parameter, the `rslogix5000_routine` option was added.

For more information, see Target IDE.

Global Tunable Parameters for Generated Code from Rockwell Automation RSLogix 5000 Add-On Instructions and Routine Formats and Phoenix Contact PC WORX

The Simulink PLC Coder software supports global tunable parameters for generated code from Rockwell Automation RSLogix 5000 Add-On instructions (AOIs) and routine formats and Phoenix Contact PC WORX. For more information on how tunable parameters are mapped, see About Tunable Parameters in the Simulink PLC Coder Environment in the Simulink PLC Coder User's Guide.

Support for Absolute Time Temporal Logic for the Rockwell Automation RSLogix 5000 IDE

The Simulink PLC Coder software now supports absolute time temporal logic in Stateflow charts for the Rockwell Automation RSLogix 5000 IDE. The coder does not support absolute time temporal logic for other target IDEs.

Note If your model uses absolute time temporal logic, you cannot create test bench code for that model.

Integration of Externally Defined Symbols in Generated Code

You can now suppress symbol definitions in the generated code. This suppression allows the generated code to refer to these symbols. You must then provide these definitions when importing the code into the PLC IDE. For more information, see Integrating Externally Defined Symbols.

Support for Configuring Tunable Parameters Using Simulink.Parameter Objects

You can now configure tunable parameters using `Simulink.Parameter` objects. In previous releases, you could only configure tunable parameters using the Configuration Parameters dialog box. For more information, see Working with Tunable Parameters in the Simulink PLC Coder Environment.

Author Creation Data, Descriptions, and Sample Times in Generated Code Header Comments

The Simulink PLC Coder generated code header now includes:

- Author names from model properties
- Creation dates from model properties
- Model descriptions from model properties
- Fundamental sample times for the model and the subsystem block for which you generate code

Support for atan2

The Simulink PLC Coder software now supports the math function `atan2`.

Convenience Dynamic Lookup Table Block

As a convenience, the DynamicLookup block has been added to the plclib/Simulink/Lookup Tables sublibrary. In previous releases, you could achieve the dynamic lookup behavior using the Prelookup block with the Interpolation Using Prelookup block. Going forward, use the plclib/Simulink/Lookup Tables/DynamicLookup block.

New Examples

The following examples are new:

- `Speed Cruise Control System Using Variable-Step Continuous Solver` — Illustrates code generation for the variable-step continuous solver. In this example, the controller subsystem has a fixed sample time, while the model has a variable-step continuous solver.
- `Mapping Tunable Parameters Defined Using Simulink.Parameter Objects to Structured Text` — Illustrates the specification of tunable parameters using `Simulink.Parameter` objects in the MATLAB base workspace.
- `Generating Structured Text for Stateflow Chart with Absolute Time Temporal Logic` — Illustrates code generation for Stateflow Chart blocks with absolute time temporal logic. This example requires the Rockwell Automation RSLogix AOI or routine format.
- `Integrating User Defined Function Blocks, Data Types, and Global Variables into Generated Structured Text` — Illustrates how to integrate user defined function blocks, data types, and global variables and constants into generated Structured Text.

R2011b

Automatic IDE Import of Subsystem Code Without Test Bench

The Simulink PLC Coder software now generates and imports subsystem code into target IDEs without the test bench. To use this feature:

1. In the Configuration Parameters dialog box, clear the **Generate testbench for subsystem** check box.
2. In the Simulink editor, right-click the subsystem and select **PLC Code Generation > Generate and Import Code for Subsystem**.

In previous releases, the coder generated and imported test bench code into the target IDE regardless of the setting of the **Generate testbench for subsystem** check box.

Subsystem Function Block Code

In generated code, the function block code of the top-level subsystem has been simplified. The coder now generates the function block code depending on whether or not the top-level subsystem has internal state. In previous releases, the coder always generated the function block code with the `ssMethodType` parameter for top-level subsystems.

Version History

This release simplifies the function block code of the top-level subsystem for generated code.

- If the top-level subsystem in the Simulink model has internal state, the generated function block for the block will have an extra first parameter `ssMethodType` of integer type. This extra parameter is in addition to the function block I/O parameters mapped from Simulink block I/O ports.

  To use the function block:

  1. Initialize the block by calling the function block with `ssMethodType` set to integer constant `SS_INITIALIZE`.
  2. If the IDE does not support symbolic constants, set `ssMethodType` to integer value 0.
  3. For each follow-up invocation, call the function block with `ssMethodType` set to constant `SS_STEP`.
  4. If the IDE does not support symbolic constants, set `ssMethodType` to integer value 1.

  These settings cause the function block to initialize or compute and return output for each time step.
- If the top-level subsystem does not have internal state, the function block code has only parameters mapped from Simulink block I/O ports. There is no `ssMethodType` parameter. To use the function block in this case, call the function block with I/O arguments.

For non-top-level subsystems, either with or without internal state, the function block code has the `ssMethodType` parameter. The generated code might have other `ssMethodType` constants to implement Simulink semantics.

New Demo

The following demo is new:

- Generating Structured Text for a Simple Simulink Subsystem without Internal State — Illustrates changes for function block prototypes in generated code.

R2011a

Version: 1.2

New Features

Bug Fixes

Support for New PLC Target IDEs

The Simulink PLC Coder software now supports code generation and automatic import of code for the Phoenix Contact PC WORX IDE.

See Supported IDE Platforms in the Simulink PLC Coder User's Guide for more information.

Generated Code File Name Can Now Be Renamed

You can now specify a custom name for the code file that you generate with Simulink PLC Coder. Use the **Function name options** parameter in the Subsystem block.

Generated Code File Header Change

The comment header in the code file that you generate with Simulink PLC Coder now includes a sample time field for the model.

Support for Lookup Table Blocks

Simulink PLC Coder models can now generate code for lookup table blocks.

Support for Fixed Point Data Types

Simulink PLC Coder models can now generate code for fixed-point data types. For more information, see Fixed-Point Data Type Limitations in the Simulink PLC Coder User's Guide.

CORDIC Trigonometric Functions

The Simulink PLC Coder product now supports code generation for CORDIC trigonometric functions. This support enables you to use trigonometric functions for PLCs that do not support these functions in built-in libraries.

To generate code for CORDIC trigonometric functions:

1. Add the Simulink Trigonometric Function block to the coder subsystem.
2. Configure the block to the desired trigonometric function.
3. From the **Approximation method** parameter, select `CORDIC`.
4. Generate code for the atomic subsystem.

64-Bit Support

The Simulink PLC Coder product supports 64-bit systems. You can still use the Simulink PLC Coder product with 32-bit IDEs.

See the MathWorks® website at Supported IDEs for a list of supported IDEs and platforms.

New Demos

The following demos are new:

- Airport Conveyer Belt Control System — Illustrates code generated for an airport conveyer belt.
- Generating Structured Text for Simulink Model with Fixed-Point Data Types — Illustrates generating fixed-point code in the Simulink PLC Coder environment.

R2010b

Version: 1.1

New Features

Support for Triggered Subsystems

You can now use the Simulink PLC Coder software to generate code from Simulink triggered subsystems. Use the Triggered Subsystem block. See How Triggered Subsystem Code Maps to Function Blocks in the Simulink PLC Coder User's Guide.

Support for New PLC Target IDEs

The Simulink PLC Coder software now supports:

- Siemens SIMATIC STEP 7 IDE
- KW-Software MULTIPROG 5.0 IDE

See Supported IDE Platforms in the Simulink PLC Coder User's Guide for more information.

Automatic Import of Generated Code

You can now automatically import Structured Text code, generated by the Simulink PLC Coder software, to your PLC IDE. In previous releases, you imported the generated code manually according to the instructions provided by the PLC IDE manufacturer.

You can take advantage of this capability for the following PLC IDEs:

- CoDeSys IDE V2.3
- Rockwell Automation RSLogix 5000 IDE
- Siemens SIMATIC STEP 7 IDE
- KW-Software MULTIPROG 5.0 IDE

See Automatically Importing Structured Text Code in the Simulink PLC Coder User's Guide for more information.

New Demo

A new Simulink PLC Coder demo, Speed Cruise Control System Using Simulink and Stateflow, illustrates code generated for a cruise control controller subsystem using a triggered subsystem.

R2010a

Version: 1.0

New Features

New Product

Simulink PLC Coder generates hardware-independent IEC 61131-3 Structured Text from Simulink models, Stateflow charts, and Embedded MATLAB® functions. The Structured Text is generated in PLCopen and other file formats supported by widely used integrated development environments (IDEs). As a result, you can compile and deploy your application to numerous programmable logic controller (PLC) and programmable automation controller (PAC) devices.

Simulink PLC Coder generates test benches that help you verify the Structured Text using PLC and PAC IDEs and simulation tools.

Key features:

- Automatic generation of IEC 61131-3 Structured Text
- Simulink support, including reusable subsystems, PID controller blocks, and lookup tables
- Stateflow support, including graphical functions, truth tables, and state machines
- Embedded MATLAB support, including if-else statements, loop constructs, and math operations
- Support for multiple data types, including Boolean, integer, enumerated, and floating-point, as well as vectors, matrices, buses, and tunable parameters
- IDE support, including B&R Automation Studio, PLCopen, Rockwell Automation RSLogix 5000, and Smart Software Solutions CoDeSys
- Test-bench creation