INTRODUCTION

NI VeriStand™ Model Framework

NI VeriStand and the NI LabVIEW Model Interface Toolkit allow you to run copies of simulation models written in C and C++. The NI VeriStand Model Framework is a group of files that provide entry points to models. To enable models to interact with NI VeriStand and the NI LabVIEW Model Interface Toolkit, you must design them to work with the NI VeriStand Model Framework.

This guide describes the components of the NI VeriStand Model Framework, the process for creating model code that is compatible with the framework, and requirements for compiling your model with the framework.

Contents

Installing the Model Framework

In the NI VeriStand installer, the NI VeriStand Model Framework feature installs the Model Framework files on the host computer. If you cannot locate the files described in the **Components of the Model Framework** section of this document, run the NI VeriStand installer again and select the **NI VeriStand Model Framework** item from the list of features to install. You do not need to reinstall other NI software to install the Model Framework.

NATIONAL INSTRUMENTS™

Components of the Model Framework

The following table describes the files in the Model Framework. These files show how you must implement your model code.

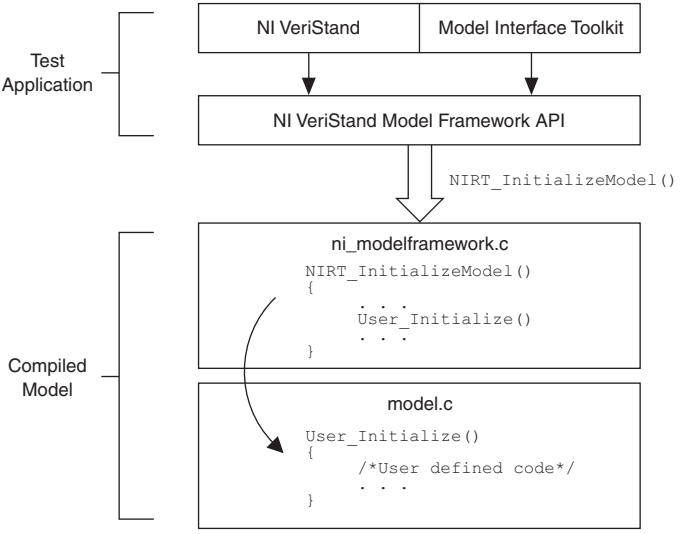
File	Description	Installed Location
`ni_modelframework.h`	A header file that includes the following components: • Type definitions that your model code can use to define properties of outward-facing components of your model, such as inports, outports, parameters, and signals. • Functions that the NI VeriStand Model Framework exports to your compiled model.	*`RootDrive`*`:\VeriStand\`*`version`*`\ModelInterface` where *`RootDrive`* is the drive where NI software installs and *`version`* is the version number of NI VeriStand.
`ni_modelframework.c`	A file that implements the common interface between your test application and your model code.	*`RootDrive`*`:\VeriStand\`*`version`*`\ModelInterface\custom\src`
`template.c`	A template for your model code. Use this file to create code that maintains interdependent structures between your model and `ni_modelframework.c`.	*`RootDrive`*`:\VeriStand\`*`version`*`\ModelInterface\custom\examples`

Interaction of Framework Code and Your Model Code

When you run your test application designed using NI VeriStand or the LabVIEW Model Interface Toolkit, the application executes functions defined in NI VeriStand Model Framework files. These functions then call functions in your model code, which convert user-defined data types, initialize your model, and take a time step.

The following illustration shows how NI software, Model Framework code, and code in your model interact. In this illustration, your test application calls a function that the Model Framework exports. That function, in turn, calls a function in your model code.

Figure 1. Interaction of NI Software, Framework Code, and Model Code



As the model executes, the test application can interact with the model in the following ways:

- Write data to model inports
- Read data from model outports
- Allow you to adjust model parameter values
- Allow you to probe model signals

Overview: Model Creation and Compilation Process

To create a model written in C/C++ that NI software can load and execute via the Model Framework, you typically perform the following tasks:

1. Create a `model.h` header file.
2. Adapt `template.c` to serve as your model code.
3. Create a makefile to compile your model code.

The following sections contain high-level guidelines about the files you need to create and customize to prepare a model for use with NI software. For details about this process, refer to code comments inside the Model Framework files.

Creating a model.h Header File

Create a header file named `model.h` that contains the type definitions for model parameters and all the user-visible parameters in your model.

Your header file should look similar to the following code copied from the `model.h` file in the engine model example at `<RootDrive>:\VeriStand\version\ModelInterface\custom\examples\engine`.

Figure 2. Example `model.h` file

```
#ifndef MODEL_h
# define MODEL_h
typedef struct {
    double a[2][2];
    double b11;
    double c12;
    double idleRPM;
    double redlineRPM;
    double temperature_timeConstant;
    double temperature_roomTemp;
    double temperature_operatingTempDelta;
    double temperature_redlineTempDelta;
} Parameters;
#endif
```

The previous example contains definitions for both scalar and vector double parameters. For information about defining parameters whose data type is something other than double, refer to comments in the `template.c` file installed by the Model Framework.

Adapting template.c to Serve as Your Model Code

Make a copy of the `template.c` file installed by the Model Framework to serve as a starting point for your model code. This allows you to maintain interdependent structures between your model code and `ni_modelframework.c`. These structures might include headers, imported and exported symbols, and functions that NI software recognizes. You can save your `.c` model file with any filename.

Use the following guidelines to modify `template.c` to serve as your own model code:

1. Refer to the contents of `template.c` for all the code you must customize, which is marked with comments. This file also contains information about how to instantiate and access parameters.
2. Refer to the contents of `ni_modelframework.h` for the type definitions you can use to define properties of outward-facing components of your model, such as inports, outports, parameters, and signals.

Compiling Your Model with the NI VeriStand Model Framework

For the NI VeriStand Model Framework to be able to provide entry points to your model to be used with NI VeriStand and the NI LabVIEW Model Interface Toolkit, you must compile the model. To compile your model, create a makefile that is appropriate for the compiler you plan to use and the operating system on which the model will run. The following list describes the type of makefile to use and the corresponding output to generate:

- Windows/NI ETS targets—Create a `.mak` makefile that generates a DLL. Refer to the National Instruments website at `ni.com/info` and enter the Info Code `exjr6s` for information about compiling a DLL for ETS targets.
- VxWorks targets—Create a `.mk` makefile that generates a `.out` file. Refer to the National Instruments website at `ni.com/info` and enter the Info Code `ex2xp2` for information about compiling a `.out` file for VxWorks targets.
- NI Linux Real-Time targets—Create a `.mk` makefile that generates an `.so` file. Refer to the National Instruments website at ni.com/info and enter the Info Code `ex6eq7` for information about compiling an `.so` file for NI Linux Real-Time targets or `exfyk9` for an example of how to use NI Linux Real-Time C/C++ tools to compile VeriStand models.

Note To determine which real-time operating system your RT target runs, refer to the National Instruments website at `ni.com/info` and enter the Info Code `exxjax`.

For examples of makefiles designed to compile models that work with the Model Framework, refer to the example `.mak` and `.mk` files installed in the *`RootDrive`*`:\VeriStand\`*`version`*`\ModelInterface\custom\examples` directory, where *`RootDrive`* is the drive where NI software installs and *`version`* is the version number of NI VeriStand.

The following files must be present when you run a makefile to compile your model code:

- `ni_modelframework.h`
- `ni_modelframework.c`
- `model.h`
- `model.c`

Where to Go Next

The Model Framework installs several example models, including their `.c` and `model.h` files and makefiles. For examples of model code you can explore, refer to the *`RootDrive`*`:\VeriStand\`*`version`*`\ModelInterface\custom\examples` directory, where *`RootDrive`* is the drive where NI software installs and *`version`* is the version number of NI VeriStand.

For information about using your model in NI software, refer to the appropriate help system:

- *NI VeriStand Help*—Available in NI VeriStand by selecting **Help»Search the NI VeriStand Help**. Browse to the **Integrating and Executing Models** section on the **Contents** tab for more information about simulating models in NI VeriStand.
- *Model Interface Toolkit Help*—Available in LabVIEW by selecting **Help»LabVIEW Help.** Browse to the **Toolkits»Model Interface Toolkit** section on the **Contents** tab for more information about simulating models in LabVIEW with the Model Interface Toolkit.