

INTRODUCTION

NI VeriStand™ Model Framework

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

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

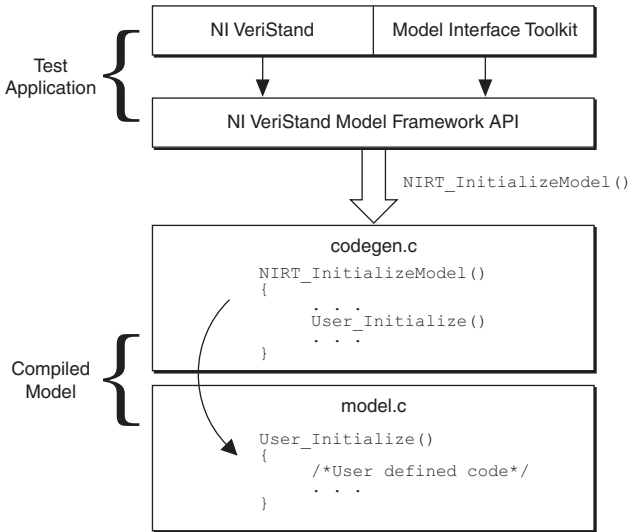
Interaction of Framework Code and Your Model Code.....	1
Installing the Model Framework	2
Components of the Model Framework.....	3
Components of Your Model Code.....	3
Overview: Model Creation and Compilation Process	4
Creating a model.h Header File.....	4
Adapting template.c to Serve as Your Model Code	5
Compiling Your Model with the NI VeriStand Model Framework	5
Where to Go Next.....	6

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 outputs
- Allow you to adjust model parameter values
- Allow you to probe model signals

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 next 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.

Components of the Model Framework

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

File	Description	Installed Location
<code>NIVERISTAND_API.h</code>	A header file that includes the following components: <ul style="list-style-type: none">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 the NI VeriStand Model Framework exports to your compiled model.	<code>RootDrive:\VeriStand\version\ModelInterface</code> where <i>RootDrive</i> is the drive where NI software installs and <i>version</i> is the version number of NI VeriStand.
<code>codegen.c</code>	Implements the common interface between your test application and your model code.	<code>RootDrive:\VeriStand\version\ModelInterface\custom</code>
<code>template.c</code>	A template for your model code. Use this file to create code that maintains interdependent structures between your model and <code>codegen.c</code> .	

Components of Your Model Code

You typically create the following two files to implement your C/C++ model:

- `model.h`—Contains the type definitions for your model parameters. This file must be named `model.h` and must include all the user-visible parameters in your model.
- `model.c`—Contains your model code. Make a copy of `template.c` and modify the copy to ensure you maintain interdependent structures, such as headers, imported and exported symbols, and functions NI software recognizes, between your model code and `codegen.c`. You can name this `.c` file however you want.

The next section of this document contains information about creating your `model.h` and `.c` files.

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 by NI software. However, these sections do not contain detailed instructions for creating and compiling a model using the Model Framework. For details about this process, refer to code comments inside the Model Framework files.

Creating a `model.h` Header File

You must create a header file named `model.h` for your model code. The `model.h` file contains the type definitions for model parameters and must include all the user-visible parameters in your model.

The following code, copied from the `model.h` file in the engine model example at `RootDrive:\VeriStand\version\ModelInterface\custom`, provides an example of the `model.h` file you must create.

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 `TO DO` comments in the `template.c` file installed by the Model Framework.

Adapting `template.c` to Serve as Your Model Code

The `template.c` file installed by the Model Framework serves as a starting point for your model code. 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 `TO DO` comments. This file also contains information about how to instantiate and access parameters.
2. Refer to the contents of `NIVERISTAND_API.h` for the type definitions you must 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 your model code to work with the NI VeriStand Model Framework, 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.



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 [exjax](#).

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

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

- `NIVERISTAND_API.h`
- `codegen.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. Refer to the `RootDrive:\VeriStand\version\ModelInterface\custom` directory, where `RootDrive` is the drive where NI software installs and `version` is the version number of NI VeriStand, for examples of model code you can explore.

Refer to the appropriate help system for the NI software you are using:

- *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.

Refer to the *NI Trademarks and Logo Guidelines* at ni.com/trademarks for more information on National Instruments trademarks. Other product and company names mentioned herein are trademarks or trade names of their respective companies. For patents covering National Instruments products/technology, refer to the appropriate location: **Help»Patents** in your software, the `patents.txt` file on your media, or the *National Instruments Patents Notice* at ni.com/patents. You can find information about end-user license agreements (EULAs) and third-party legal notices in the readme file for your NI product. Refer to the *Export Compliance Information* at ni.com/legal/export-compliance for the National Instruments global trade compliance policy and how to obtain relevant HTS codes, ECCNs, and other import/export data.

© 2009–2013 National Instruments. All rights reserved.