

Custom Error Handling In LabVIEW

 [Print this Page](#)

The LabVIEW environment includes an internal error file that is used to provide meaningful error messages to the user. However, a LabVIEW developer may want the ability to use custom error codes along with custom error messages. The following document describes why custom error handling is useful along with different ways to create and use custom error codes and messages in the LabVIEW environment, executables, and shared libraries.

Table of Contents:

- [Why Custom Error Handling?](#)
- [Creating Custom Error Files](#)
- [Creating a Custom Error Handler](#)

Why Custom Error Handling?

Custom error handling reduces debugging time for large applications and provides meaningful information for end-user applications.

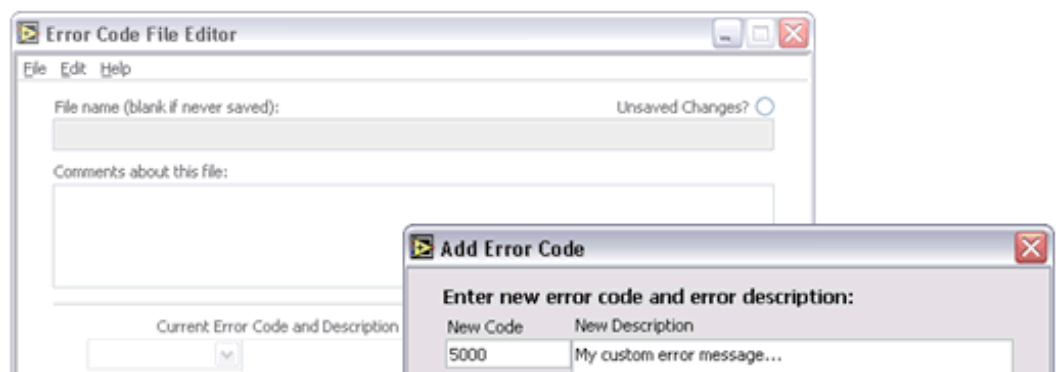
When developing a large application, error handling is of utmost importance. The ability to use custom error codes and messages can reduce debugging time by providing the developer with application-specific information on where and why the error occurred.

When developing an application to be distributed, custom error handling can provide meaningful information for the end user. For example, a "Generic file I/O error" provides no useful information to the end user as to why the error occurred or how to correct it. On the other hand, a custom error message such as "*The data file you specified cannot be found in the User Data folder. Please provide a 2D I32 binary data file in this directory.*" provides an application-specific error message along with instruction to resolve the error. This is especially useful when dealing with built executables and shared libraries, where the user does not have access to the block diagram and cannot debug the application.

Creating Custom Error Files

While editing LabVIEW's internal error file would be a bad idea (see note below), a user can create an additional error file with custom error codes and corresponding error messages. By doing so, the user can leverage the existing Simple Error Handler and General Error Handler without making changes to the code. The Simple Error Handler and General Error Handler will first reference LabVIEW's internal error file and then the user error files. If an error code is found in multiple files, the error handler VIs will report both error messages. To create a custom error file, complete the following steps:

Select Tools->Advanced->Edit Error Codes and choose to create a new error code file. From the Error Code File Editor, choose to Add a new error.





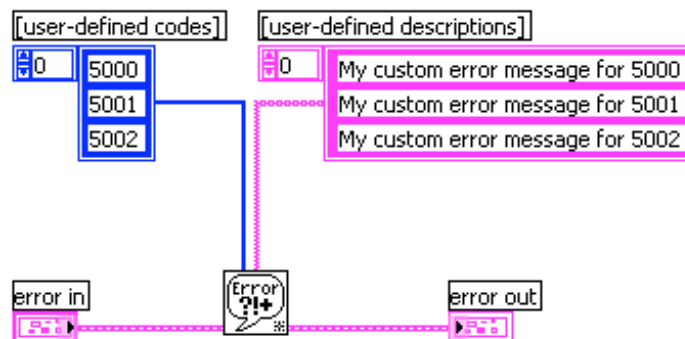
In the Add Error Code window, type in the error code that you want to create and its corresponding description. It is suggested to create custom error codes in the range of 5000 to 9999 to prevent duplicate error codes with LabVIEW's internal error file. Add as many custom error codes as necessary. A user can even add comments about the custom error code file in the Error Code File Editor. Upon completion, select File->Save as and save the file to the LabVIEW <version>\user.lib\errors directory. The file must be named xxx-errors.txt, where xxx is the supplied name. For example, when saving My Custom Error file, save the file as MyCustomErrors-errors.txt. Keep in mind that changes to custom error code files **take effect only after restarting LabVIEW**.

When building an executable (EXE) or shared library (DLL) with the Application Builder, make sure to add the custom error file as a support file (under the Source Files tab). For an executable, the custom error file will be saved in the <executable destination directory>\user.lib\errors directory. For a shared library, the custom error file will be saved in the National Instruments\Shared\LabVIEW Run-Time\<version>\Errors directory. When distributing a shared library, make sure to install the custom error file in this directory.

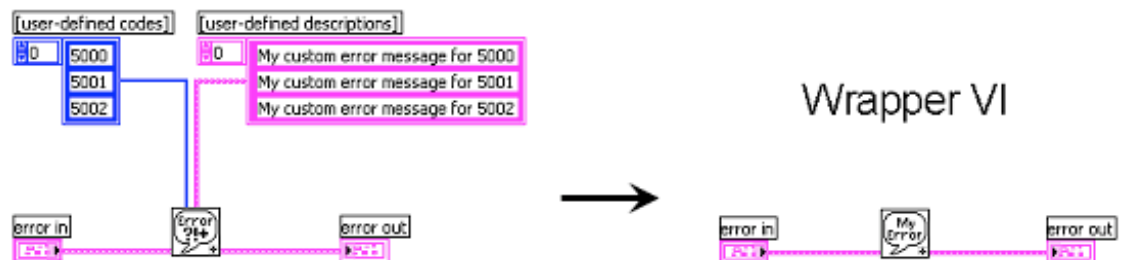
Note: Editing LabVIEW's internal error file could potentially cause the loss of useful predefined error codes and descriptions. The only way to restore this information would be to uninstall and reinstall LabVIEW. Also, running an application on a different machine or building the VI into an executable or shared library could cause inconsistent error messages.

Creating a Custom Error Handler

A user can programmatically leverage custom error handling by providing custom error codes and messages to the General Error Handler. To do so, pass an array of error codes to the [user-defined codes] input terminal and an array of error descriptions to the [user-defined descriptions] input terminal, as shown below.



This allows the user to keep the custom error codes local to the application as well as to programmatically select custom error messages for certain error codes. Another option for the user is to create a custom error handler by utilizing a wrapper VI, which includes the user-defined codes and descriptions, as shown below.



Creating a custom error handler makes distribution of an executable simple. As opposed to the custom error file technique, no support files are needed when creating a executable or shared library since the custom error codes and descriptions are a part of the source code.

[Print this Page](#)

Your Feedback!

◀ Poor | Excellent ▶

Yes No

Document Quality?

1 2 3 4 5

Answered Your Question?

Yes No

Document needs work? Please tell us why.

Please [Contact NI](#) for all product and support inquiries.

Submit ▶