NATIONAL INSTRUMENTS®
The Software is the Instrument®

LabVIEW® Test Executive Release Notes Version 4.0

Part Number 321022B-01

Welcome to the LabVIEW Test Executive

These release notes introduce you to the LabVIEW 4.0 Test Executive package, describe the system requirements, and contain updated information for the documentation.

Contents

How to Proceed 1
  Special Note to Users Upgrading from a Previous Version of the Test Executive....................2
Required System Configuration 2
  Windows 3.1 and Windows NT 2
  Macintosh and Power Macintosh 3
  Sun SPARCstation 3
Additions to the LabVIEW Test Executive Reference Manual....................3
  Set Limit Specification 3
  Sequence Callback VIs 4
    Callback VI Management....................5
  Removing a Sequence Callback VI from a Test Sequence 5
  Dynamic Loading of the Sequence Editor and Test VIs 6
    Replacing a Test VI with a VI that has the Same Name 6
  Changes to the TYPEDEF Controls 7

How to Proceed

Scan the *Required System Configuration* section and then follow the installation instructions in Chapter 1, *Introduction,* in the *LabVIEW Test Executive Reference Manual*. In addition, be sure to read the *Additions to the LabVIEW Test Executive Reference Manual* section.

If you are a new user of the LabVIEW 4.0 Test Executive, or even if you are an experienced user, work through Chapter 2, *Getting Started,* in the *LabVIEW Test Executive Reference Manual*. The chapter gives an overview of how the LabVIEW 4.0 Test Executive works. It shows how to start the Test Executive, load and run test sequences, and create new or edit existing test sequences. It also describes the example test sequences that are delivered with the Test Executive.

If you have used a previous version of the LabVIEW Test Executive and want to convert your existing test sequences to run in the new LabVIEW 4.0 Test Executive, you should read Appendix B, *Sequence Conversion Notes,* in the *LabVIEW Test Executive Reference Manual*.

Special Note to Users Upgrading from a Previous Version of the Test Executive

If you install the LabVIEW Test Executive over a previous version, the program overwrites the following file.

```
engine.llb
```

The LabVIEW 4.0 Test Executive moves the following files to an `OLDFILES` subdirectory.

```
test_vis.llb
operator.llb
callback.llb
testexec.lib
```

If the Test Executive is installed in the default location (inside the `LabVIEW` directory), the `controls.llb` file will be added to `user.lib`; otherwise it will be in the `LVEXEC4` directory.

Required System Configuration

Windows 3.1 and Windows NT

For Windows 3.1 or Windows NT, the LabVIEW 4.0 Test Executive requires LabVIEW version 4.0 for Windows 3.1/NT. Consult your *LabVIEW Release Notes* for the required system configuration for LabVIEW.

For optimal performance while running and editing test sequences in the Test Executive, Windows users should have at least 12 MB of RAM and Windows NT users should have 16 MB of RAM. You can use the LabVIEW Test Executive with less system memory if you use Windows virtual memory, but performance may suffer.

The LabVIEW Test Executive requires approximately 9.5 MB of disk space.

Macintosh and Power Macintosh

For Macintosh and Power Macintosh, the LabVIEW 4.0 Test Executive requires LabVIEW version 4.0 for Macintosh. Consult your *LabVIEW Release Notes* for the required system configuration for LabVIEW.

For optimal performance while running and editing test sequences in the Test Executive, 12 MB or more of RAM is recommended. You may need more memory, depending on the size of your application and the amount of data that your application manipulates.

The LabVIEW Test Executive requires approximately 9.5 MB of disk space.

Sun SPARCstation

For the Sun SPARCstation, the LabVIEW 4.0 Test Executive requires LabVIEW version 4.0 for Sun. Consult your *LabVIEW Release Notes* for the required system configuration for LabVIEW.

The SPARCstation should have 24 MB of RAM, with 32 MB of swap space storage. LabVIEW for Sun can run on less than 24 MB of RAM, but performance will suffer.

The LabVIEW Test Executive requires approximately 18 MB of disk space during installation. After installation, the Test Executive uses approximately 12 MB of disk space.

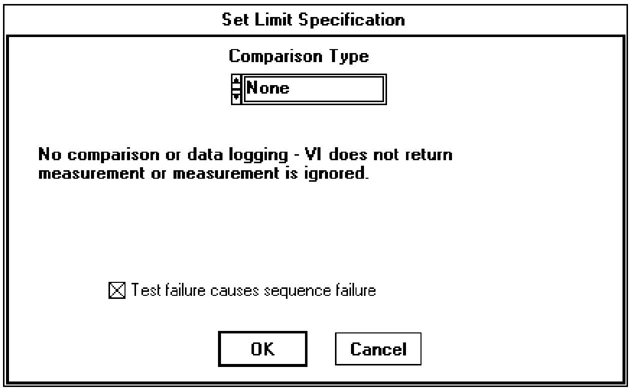
Additions to the LabVIEW Test Executive Reference Manual

This section contains additional information to the *LabVIEW Test Executive Reference Manual*.

Set Limit Specification

The information in this section belongs in the *Test Editor Controls* section of Chapter 4, *Creating Test VIs and Test Sequences*.

The Set Limit Specification dialog box, which is used in the Sequence Editor, has an additional control, called *Test failure causes sequence failure*. When this option is enabled (the default setting), the Test Executive uses the test result when determining whether a UUT passes or fails. This is the standard behavior for the Test Executive. If this checkbox is not selected, The Test Executive *does not use* the result of the test for which you are setting the limit specification to determine whether the UUT passes or fails. This means that a test within a test sequence can fail, but the UUT can still pass the test sequence. The Set Limit Specification dialog box operates exactly as before, except for the addition of the new control, shown in the following illustration.



No comparison or data logging - VI does not return measurement or measurement is ignored.

☒ Test failure causes sequence failure

You can use the Test failure causes sequence failure control in conjunction with the new GOTO statement in your test sequences to repeat portions of a test sequence or to branch within a test sequence without causing a UUT to fail.

Sequence Callback VIs

The information in this section belongs in the *Sequence VIs* subsection of Chapter 4, *Creating Test VIs and Test Sequences*.

If you use a PreRun or PostRun VI in your test sequence, it must have both Test Data and error indicators on its front panel—like other test VIs. Recall that the contents of the Test Data indicator are not used for PreRun and PostRun VIs.

If you decide to build other sequence callback VIs for your test sequence, you should use the default callback VIs as a template. This ensures that your callback VI has the proper controls and indicators on its front panel. You can find these template VIs in `CALLBACK.LLB`.

Consult the following table to determine which template VI to use to build a specific callback VI.

Callback Name in Edit Sequence VI Dialog Box	**Default Callback VI Name in CALLBACK.LLB**
Pre-UUT loop callback	`Pre-UUT Loop Callback.vi`
Pre-UUT callback	`Pre-UUT Callback.vi`
Post-UUT loop callback	`Post-UUT Loop Callback.vi`
Post-UUT callback	`Post-UUT Callback.vi`
Test Report callback	`Test Report Callback.vi`
Edit Test VI callback	`Open Test VI Callback.vi`
Post Run-Loop Test callback	`Post Run-Loop Test Callback.vi`
Test Failure callback	`Test Failure Callback.vi`

Callback VI Management

Keep in mind that the Test Executive calls the default callback VIs in `CALLBACK.LLB` unless you specify otherwise. It is strongly recommended that you always build new callback VIs using a copy of the default callback VIs. That way, you will always have a clean and complete template VI to use later. Also, future versions of the LabVIEW Test Executive may overwrite `CALLBACK.LLB`, so it is recommended that you store your callback VIs in a different library.

Removing a Sequence Callback VI from a Test Sequence

To remove a sequence callback VI from a test sequence, open the Edit Sequence VIs dialog box from the Sequence Editor and choose the callback VI you want to remove through the Select VI control. Be sure to delete the entire file path shown in the VI Path control in the Edit Sequence VIs dialog box. Then click the OK button to return to the Sequence Editor. Close the Sequence Editor to return to the Test Executive. Whenever the Test Executive runs the test sequence and encounters an empty path in the list of sequence callback VIs, it calls the default callback VI stored in `CALLBACK.LLB`. Because the Test Executive always executes the default callback VI when you have not assigned a specific callback VI, it is very important that you do not modify the default callback VIs in `CALLBACK.LLB`.

Note: ***Any sequence callback VIs you remove from a sequence remain in memory until you quit the Test Executive.***

Dynamic Loading of the Sequence Editor and Test VIs

The information in this section belongs in the *Creating or Editing a Test Sequence* section of Chapter 4, *Creating Test VIs and Test Sequences*.

The LabVIEW 4.0 Test Executive dynamically loads test VIs for a sequence when the test sequence is loaded into the Test Executive. As described in the *LabVIEW Test Executive Reference Manual*, you can control when a test VI will be dynamically loaded. Keep the following information in mind when working in the Sequence Editor.

Replacing a Test VI with a VI That Has the Same Name

LabVIEW allows only one instance of a VI with a particular name to be loaded into memory at a time. Consequently, the Test Executive does not allow two separate tests to use test VIs that have the same name, even if the VIs reside in different locations on the computer. For example, if you use a test VI called `Check Power Supply.vi` that is stored in the VI library `mytests.llb` for Test A, you cannot use another test VI called `Check Power Supply.vi` from a library called `newtests.llb` for Test B. You can only select one version of `Check Power Supply.vi` for use in the test sequence.

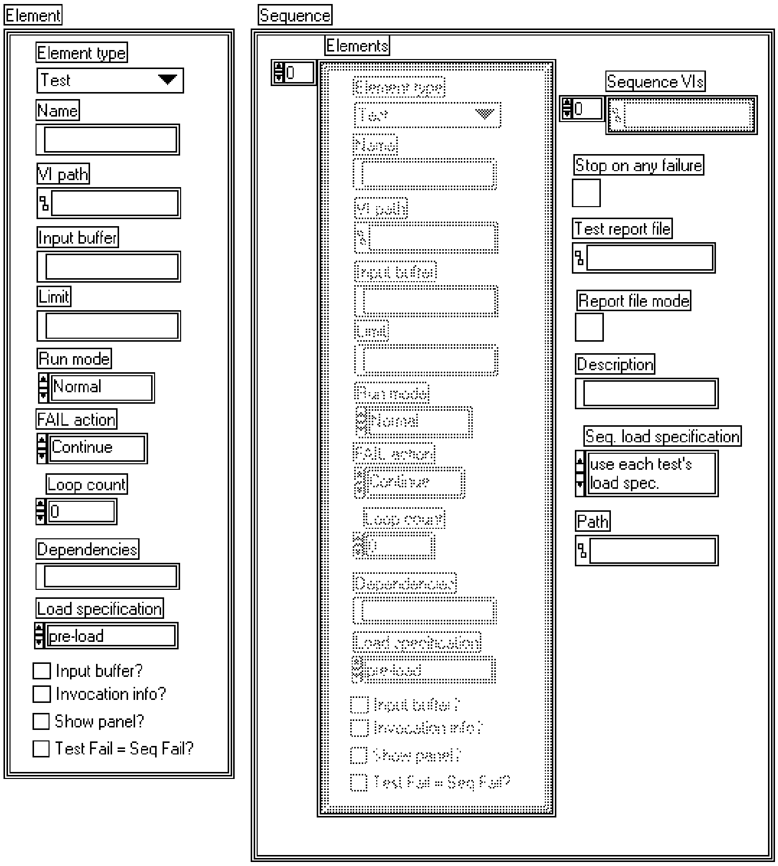
If you try to assign a test VI that has the same name, but different file path, as another test VI, the Test Executive alerts you and prompts you to choose whether all tests should use the new VI. To make the Test Executive unload the old test VI and load the new one, you must exit the Sequence Editor. Be aware that the old test VI remains in memory and is used in the execution of the test sequence if any of the following conditions apply.

- The front panel of the test VI is open.
- Another VI in memory uses the test VI as a subVI.
- The test VI has been recompiled, but you did not save changes when the Test Executive attempted to unload the VI from memory.

Changes to the Typedef Controls

The information in this section belongs in *The Test Executive Typedef Controls* section of Chapter 5, *Modifying the Test Executive*.

The `TYPEDEF - Sequence Element.ctl` and `TYPEDEF - Sequence.ctl` controls have changed to accommodate the new Test failure causes sequence failure control. The new controls appear in the following illustration.



*321022B·

321022B-01

April 1996