NATIONAL INSTRUMENTS
The Software is the Instrument

LABVIEW™ TEST EXECUTIVE VERSION 5.1

These installation and release notes describe system requirements, give installation instructions, and contain updated information for the *LabVIEW Test Executive Reference Manual*.

Contents

How to Proceed

Review the *Required System Configuration* section and then follow the
instructions in the *Installation Instructions* section. In addition, read the
Additions to the LabVIEW Test Executive Reference Manual and the
What's New in Test Executive 5.1 sections.

If you are a new user of the LabVIEW 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 provides an
overview of the LabVIEW Test Executive, and 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 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 Test
Executive 5.1, see 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

The installation of the LabVIEW Test Executive 5.1 does not affect a default installation of the LabVIEW Test Executive 4.0 or 5.0. If you are upgrading from the Test Executive 4.0 or 5.0, you can either uninstall it before or after installing the Test Executive 5.1. However, you must remove version 4.0 or 5.0 to avoid type-definition linkage conflicts with the Test Executive 5.1. Before uninstalling, make backups of the customized components, such as the operator interface VI, callback VIs, and test VIs.

Upgrading from Test Executive 5.0

Upgrading to the Test Executive 5.1 from version 5.0 requires only that you recompile your operator interface, and callback and test VIs to use the version 5.1 type definitions. Otherwise, your existing VIs will work without modification.

For instructions on upgrading your sequences and recompiling your test VIs, see Appendix B, *Sequence Conversion Notes*, of the *LabVIEW Test Executive Reference Manual*.

Upgrading from Test Executive 4.0

The Test Executive 5.1 incorporates many feature additions and changes that required fundamental changes to the Operator Interface VI as well as the callback VIs. If you customized these VIs, re-implement your modifications using the new Operator Interface and callback VIs included with the Test Executive 5.1.

You do not need to modify your test VIs for use with the Test Executive 5.1. For instructions on upgrading your sequences and recompiling your test VIs, see Appendix B, *Sequence Conversion Notes*, of the *LabVIEW Test Executive Reference Manual*.

Required System Configuration

Windows

For Windows, the LabVIEW Test Executive 5.1 Development System requires LabVIEW version 5.0 or later for Windows or BridgeVIEW 2.0 or later for Windows 95/NT only. Consult the *LabVIEW Release Notes* for the required system configuration for LabVIEW.

For optimal performance while running and editing test sequences in the Test Executive, Windows 3.1 users should have at least 12 MB of RAM and

Windows 95/NT users should have at least 16 MB of RAM (32 MB of RAM is recommended for Windows NT).

Macintosh and Power Macintosh

For Macintosh and Power Macintosh, the LabVIEW Test Executive 5.1 Development System requires LabVIEW version 5.0 or later for Macintosh or Power Macintosh. Consult your *LabVIEW Release Notes* for the required system configuration for LabVIEW.

For optimal performance running and editing test sequences in the Test Executive, users should have at least 16 MB of RAM.

Sun Solaris 1/2 and HP-UX

For the Sun Solaris 1/2 and HP-UX, the LabVIEW Test Executive 5.1 Development System requires LabVIEW version 5.0 for Sun or HP-UX. Consult your *LabVIEW Release Notes* for the required system configuration for LabVIEW.

For optimal performance running and editing test sequences in the Test Executive, users should have 24 MB of RAM, with 32 MB of swap space.

Installing the Test Executive

The following sections contain instructions for installing both the LabVIEW Test Executive 5.1 Development System and Run-Time System on the Windows, Sun Solaris 1/2, HP-UX, Macintosh, and Power Macintosh platforms.

The LabVIEW Test Executive 5.1 Development System and Run-Time System come on CD-ROM. Installing the Development System requires approximately 26 MB of hard disk space. Installing the Run-Time System requires approximately 18 MB of hard disk space.

☞ **Note**   *If you are using LabVIEW 4.x, you must use the Test Executive 5.0. You can find this software in the* te50 *directory on the LabVIEW Test Executive 5.1 CD-ROM.*

☞ **Note**   *Some virus detection programs may interfere with the installer program. Check the distribution disks for viruses before you begin installation. Then, turn off the automatic virus checker and run the installer. After installation, check your hard disk for viruses again and then turn on the virus checker.*

Windows 95/NT

Development System

1. Select **Run...** from the Start Menu and type
   `X:\win32\dev\disk1\setup.exe`
   where *X* is your CD drive.

2. Follow the instructions that appear on your screen.

3. When prompted by the installer to choose a destination directory, select your `LabVIEW` directory.

Run-Time System

1. Select **Run...** from the Start Menu and type
   `X:\win32\rts\disk1\setup.exe`
   where *X* is your CD drive.

2. Follow the instructions that appear on your screen.

Windows 3.1

Development System

1. Select **File»Run...** from the Program Manager and type
   `X:\win31\dev\setup.exe`
   where *X* is your CD drive.

2. Follow the instructions that appear on your screen.

3. When prompted by the installer to choose a destination directory, select your `LabVIEW` directory.

Run-Time System

1. Select **File»Run...** from the Program Manager and type
   `X:\win31\rts\setup.exe`
   where *X* is your CD drive.

2. Follow the instructions that appear on your screen.

Macintosh and Power Macintosh

Development System

1. Double click the installer icon found in the `pmac:dev` (for Power Macintosh) or `68kmac:dev` (for 680x0-based Macintosh) folder on the CD.

2. Click the **Install** button.

3. When prompted by the installer to choose a destination folder, select your `LabVIEW` folder.

Run-Time System

1. Double click the installer icon found in the `pmac:rts` (for Power Macintosh) or `68kmac:rts` (for 680x0-based Macintosh) folder on the CD.

2. Click the **Install** button.

3. When prompted by the installer to choose a destination folder, select the folder where you would like to install the Run-Time System.

Solaris 1

1. To enable superuser privileges, type `su root`. Enter the root password.

2. If the directory `/cdrom` does not exist, type the following UNIX command:
   `mkdir /cdrom`

3. Insert the Test Executive CD-ROM. Then type the following command:
   `mount -rt hsfs /dev/sr0 /cdrom`

4. To change the installation directory, type the following command:
   `cd /cdrom/sun`

5. To run the installation script, type the following command:
   `./INSTALL`

6. When prompted by the installer to choose a system, select the Development System, the Run-Time System, or both.

7. Follow the instructions that appear on your screen.

Solaris 2

1. To enable superuser privileges, type `su root`. Enter the root password.

2. If the directory `/cdrom` does not exist, type the following UNIX command:
   `mkdir /cdrom`

3. Insert the Test Executive CD-ROM. Under Solaris 2.*x*, the CD-ROM automatically mounts as soon as the CD is inserted into the drive. If this feature is disabled on your workstation, mount the CD by typing the following command:
   `mount -o ro -F hsfs /dev/dsk/c0t6d0s2 /cdrom`

4. To change the installation directory, type the following command:
   `cd /cdrom/sun`

5. To run the installation script, type the following command:
   `./INSTALL`

6. When prompted by the installer to choose a system, select the Development System, the Run-Time System, or both.

7. Follow the instructions that appear on your screen.

HP-UX

1. To enable superuser privileges, type `su root`. Enter the root password.

2. Mount the Test Executive CD-ROM on the `/cdrom` directory with the "sam" system administration utility.

3. To change to the installation directory, type the following command:
   `cd /cdrom/HP-UX`

4. To run the installation script, type the following command:
   `./INSTALL`

5. When prompted by the installer to choose a system, select the Development System, the Run-Time System, or both.

6. Follow the instructions that appear on your screen.

Installation Results

Development System

After you install the LabVIEW Test Executive 5.1 Development System, your LabVIEW directory will contain one new directory, `LVEXEC51`. The `LVEXEC51` directory contains the following Test Executive files:

- The engine (`ENGINE.LLB`)

- The operator interface VI (`OPERATOR.LLB`)

- The callback VIs (`CALLBACK.LLB`)

- The example sequences (`EXAMPLES` directory)

- Example test VIs (`TESTS` directory)

- **(Windows 95/NT only)** C tests (`TESTS` directory)

- **(Windows only)** SQL Toolkit integration examples (`DATABASE` directory)

Your LabVIEW `Project` directory contains one new directory, `LVEXEC51`, and one new file, `lvexe51.txt`.

In addition, the LabVIEW `USER.LIB` directory contains the Test Executive type definition controls in the file `LVEXEC51.LLB`.

Run-Time System

After you install the LabVIEW Test Executive 5.1 Run-Time System, the directory where you installed the software contains the following executable file, depending on the particular platform:

- **(Windows)** `LVTERT51.EXE`

- **(Solaris 1/2, HP-UX, Macintosh, and Power Macintosh)** `LVTERT51`

- Additional necessary support files, such as `DAQRV`, `SERPDRV`, and so on

What's New in Test Executive 5.1

This section describes the features added between Test Executive 5.0 and 5.1.

Additions to the LabVIEW Project Menu

The Test Executive 5.1 adds several items to the LabVIEW Project menu.

You can launch the Test Executive by selecting **Project»Test Executive»Launch Test Executive…**.

The **Project»Test Executive»Utilities** menu contains the following utilities for working with the Test Executive:

- **Save Test Executive VI for Distribution…** runs a utility that allows you to save a Test Executive VI for distribution to a test-stand computer with the Run-Time System.

- **Sequence File Converter…** launches the converter utility that you use to convert sequences created with the Test Executive 4.0 or 5.0 to the format required by the Test Executive 5.1.

- **VI Wizard…** launches a wizard utility that allows you to open template VIs for the various test and callback VIs the Test Executive 5.1 can call. These templates help ensure your VIs will meet the front panel and connector pane specifications required by the Test Executive.

Writing Test and Callback VIs

The Test Executive 5.1 uses the LabVIEW VI Server functionality to call test and callback VIs. This method results in faster calls than the method used in the Test Executive 4.0 and 5.0, but it imposes more requirements on test and callback VIs.

Specifically, the Test Executive 5.1 requires that test and callback VIs have the required controls and indicators wired to specific positions on their connector panes. Refer to the example test and callback VIs for these connector panes.

To make the transition from Test Executive 4.0 or 5.0 to 5.1 as easy as possible, the Test Executive assigns the correct connector pane to any test or callback VI that does not have the correct connector pane when the software attempts to call the VI. After assigning the connector pane, the Test Executive attempts to save the VI.

Additionally, the Test Executive 5.1 provides a wizard to aid in the creation of new test and callback VIs. To run the wizard, select **Project»Test Executive»Utility»VI Wizard…**. The wizard allows you to open template VIs for the various types of test and callback VIs. These templates have the correct front panels and connector panes for use with the Test Executive 5.1. You simply add your code to the block diagram and save the VI.

ASCII Sequence Files

The Test Executive 5.1 stores sequence files in an ASCII format similar to a Windows initialization (.ini) file. To convert your Test Executive 4.0 or 5.0 sequence files to the new format, run the Sequence File Converter by selecting **Project»Test Executive»Utility»Sequence File Converter…**.

Sequence Path Specifications

To make moving sequence files between computers easier, you can specify how the Test Executive 5.1 resolves paths to step resources, sequence VIs, and the test report file specified in a sequence file. Through the Sequence Options dialog in the Sequence Editor, you can set the Sequence Path Specification to `Absolute`, `Relative to sequence file`, or `Relative to Test Executive default`.

With a Sequence Path Specification of `Absolute`, the Test Executive expects to find step resources, sequence VIs, and the test report file in the exact location where they were when you saved the sequence file.

For a Sequence Path Specification of `Relative to sequence file`, the Test Executive expects to find test resources, sequence VIs, and the test report file in positions relative to the location of the sequence file.

When the Sequence Path Specification is set to `Relative to Test Executive default`, the Test Executive expects to find test resources, sequence VIs, and the step report file in positions relative to the path specified in the `DefaultResourceFilePath` preference in the `testexec.ini` file. For additional information, see the *Changes to testexec.ini File* section.

Localized Decimal Points

The Test Executive 5.1 supports the use of localized decimal points for display purposes. If you set your LabVIEW Development System (and the run-time executable in the case of the Run-Time System) preference to `Use localized decimal point`, the Test Executive displays and reports use your default system decimal point.

☞ **Note**   *The ASCII sequence files always denote numeric information using a period (".") as the decimal point.*

New Configuration File VIs Used in Database Components

The following database-related VIs use the LabVIEW Configuration File VIs for accessing information from the `testexec.ini` file:

- `Create DB Tables Callback.vi`
- `Per-UUT DB Logger Callback.vi`
- `Test Executive Database Browser`

Changes to testexec.ini File

The `testexec.ini` file contains changes to make it appropriate for use with the LabVIEW Configuration File VIs. The path values now are stored in a platform-independent format. Following is an example callback VI path entry from the new `testexec.ini` file:

- **(Windows)** `Login="/C/lv50/lvexec51/callback.llb/Login Callback.vi"`
- **(UNIX)** `Login="/opt/lv50/lvexec51/callback.llb/Login Callback.vi"`
- **(MacOS)** `Login="/Hard Drive/LabVIEW 5.0/callback.llb/Login Callback.vi"`

Also, the `testexec.ini` file has an additional entry in the `[Preferences]` section. The `DefaultResourceFilePath` preference specifies the path to use when the Sequence Path Specification of a sequence is set to `Relative to Test Executive default`.

Bug Fixes and Corrections in Test Executive 5.1

The following bugs are corrected in Test Executive 5.1:

- `Configure Test VI Callback.vi` calls a VI with an output named **error out** rather than **error**.
- `File Per-UUT Results Report Callback.vi` has correct internal case structure logic.
- `Test String Callback.vi` has the correct front panel controls.
- Calling C Tests works on HP-UX platform.
- Limit specifications with scientific format now compare correctly.
- Limit specifications with hexadecimal format now round correctly.
- Limit Specification dialog initializes properly with correct values.
- `Format Test Result String.vi` in the `Test Report Callback.vi` determines digits of precision correctly for the numeric information in the test report.

Additions to the LabVIEW Test Executive Reference Manual

This section contains updated information for the *LabVIEW Test Executive Reference Manual*.

Execution Model

In addition to the sequence callback VIs, each sequence might specify a Pre-Run and/or Post-Run VI. These VIs run before and after their associated sequence runs in either Test UUT or Single Pass mode. They also run before and after any group of steps you run from their associated test sequence in either Run Step(s) or Loop Step(s) mode. You specify Pre-Run and Post-Run VIs in the Sequence Option dialog of the Sequence Editor.

Creating Custom Callback VIs

When creating custom callback VIs specified by individual sequences, do not use names that are the same as the default callback VIs specified in your `testexec.ini` file.

The following type definitions referenced in Chapter 5, *Modifying the Test Executive*, of the *LabVIEW Test Executive Reference Manual* do not exist. Use the standard LabVIEW control/indicator listed in the following table.

Typedef	Standard LabVIEW Control/Indicator
TYPEDEF—String	String
TYPEDEF—Boolean	Boolean
TYPEDEF—Integer	Numeric (I32)
TYPEDEF—File Path	File Path

Calling Interface for the Test Report Callback VI

The Test Report Callback VI requires the calling interface shown in the following table.

Type	Name	Typedef	Description
Input	**Sequence Array**	Array of TYPEDEF—Sequence.ctl	Contains information about the currently loaded test sequences.
Input	**Top-Level Sequence Index**	Numeric (I32)	Specifies which sequence in the Sequence Array input is the top-level sequence in the hierarchy being executed.
Input	**Current Login Info**	TYPEDEF—Login Info.ctl	Contains login information for the current user.
Input	**Single Pass?**	Boolean	TRUE if called from Single Pass mode, FALSE if called from Test UUT mode.
Output	**Test Report**	String	Contains a report string detailing the test results for each UUT that was tested.
Output	**error out**	Standard LabVIEW error out cluster	Indicates whether an error or warning occurred in the callback VI.

Logging Results to a Database Using the LabVIEW SQL Toolkit (Windows only)

Using the LabVIEW SQL Toolkit, you can log your UUT, sequence, and step results to an SQL-compliant database. The `LVEXEC51/DATABASE/` directory contains an alternate `testexec.ini` file, as well as alternate operator interface and callback VIs that demonstrate the integration of the LabVIEW Test Executive 5.1 and the SQL Toolkit 2.0. This section of the release notes describes the use of these database-related components. All source code is provided for these VIs, so you can modify them to meet your application-specific needs. These VIs were written using SQL Toolkit 2.0. If you want to use them with a previous version of the SQL Toolkit, edit them to use the equivalent SQL VIs from the previous version.

Modifications to the System Configuration File

To use the supplied database examples, you must add several preference values to the `[Preferences]` section of the `testexec.ini` file. A sample configuration file incorporating these modifications is located in `LVEXEC51/DATABASE`. The modifications are as follows:

- `EnableDatabaseSaving=TRUE` (or `FALSE`)
- `ConnectionString=`a valid data source name for your system
- `UUTTableName=`name of table in which to store UUT results
- `SequenceTableName=`name of table in which to store sequence results
- `StepTableName=`name of table in which to store step results

The Alternate Callback VIs

The `Create DB Tables Callback.vi` and `Per-UUT DB Logger Callback.vi` are alternate callback VIs that enable database logging on your Test Executive system. These VIs are found in `LVEXEC51/DATABASE/CALLBACK.LLB`.

Create DB Tables Callback.vi

`Create DB Tables Callback.vi` is a Pre-UUT Loop callback VI. It is responsible for creating three database tables in the database specified by the `ConnectionString` preference. These three tables are referred to as the UUT Results Table, the Sequence Results Table, and the Step Results Table. The names of the tables are specified by the preferences `UUTTableName`, `SequenceTableName`, and `StepTableName`, respectively. Whether the tables are actually created is determined by the `EnableDatabaseSaving` preference.

UUT Results Table

- Serial Number

- Result

- Date

- Operator

- Top-level Sequence Name

- Test Time

Sequence Results Table

- Serial Number

- Sequence Name

- Result

- Pre-Run Execution Time

- Post-Run Execution Time

Step Results Table

- Serial Number

- Sequence Name

- Step Name

- Resource

- Result

- Execution Time

- Comment

- User Output

- Comparison Type

If the tables already exist when the `Create DB Tables Callback.vi` runs, the Test Executive will use the existing tables to record results.

Per-UUT DB Logger Callback.vi

`Per-UUT DB Logger Callback.vi` is a Post-UUT callback VI. In addition to showing the PASS, FAIL , or ABORT banner, `Per-UUT DB Logger Callback.vi` logs the result information to the tables created by `Create DB Tables Callback.vi`. Whether or not this VI actually logs the information to the database is determined by the `EnableDatabaseSaving` preference.

`Per-UUT DB Logger Callback.vi` does not necessarily log all measurement information to the Step Results Table. If the step comparison is numeric, `Per-UUT DB Logger Callback.vi` logs numeric measurement and radix information. If the comparison is string, `Per-UUT DB Logger Callback.vi` logs string measurement and limit information. If the comparison is log only, `Per-UUT DB Logger Callback.vi` logs both the numeric and the string information.

Using the Sequence Options dialog box in the Sequence Editor, you can specify `Create DB Tables Callback.vi` and `Per-UUT DB Logger Callback.vi` as the Pre-UUT Loop and Post-UUT Callback VIs, respectively, for any sequence. Alternately, you can edit the Test Executive system configuration file, `testexec.ini`, and make these callback VIs the default Pre-UUT Loop and Post-UUT callback VIs for all sequences. When you run a sequence that uses these two callback VIs (or any sequence if you modify the system configuration file), the Test Executive logs UUT Results, Sequence Results, and Step Results to the specified database on a per-UUT basis (if `EnableDatabaseSaving=TRUE`).

Operator Interface VI

Use the alternate operator interface VI found in `LVEXEC51/DATABASE/ OPERATOR.LLB` to browse the contents of the database tables specified in the system configuration file. The alternate operator interface VI has an additional button on its panel called **Database Browser...**. Click on this button to browse the database contents.

Using LabVIEW Test Shells

When you create a LabVIEW test shell, do not wire the **mode** control to the connector pane of the VI. Additionally, the load specification for the step does not have to be `Dynamic load`.

Example Sequence Using LabVIEW Test Shells

To specify an auto-configure Open Test VI callback for a sequence, select Edit Test VI Callback from the Sequence VIs ring control of the Sequence Options dialog.

Deploying the Test Executive Run-Time System to a Test-Stand Computer

For introductory information about deploying the Test Executive Run-Time System to a test-stand computer, see Chapter 6, *Deploying the Test Executive*, of the *LabVIEW Test Executive Reference Manual*.

The Test Executive Run-Time System does not include the customizable components of the Test Executive. These components include the Operator Interface VI, the system and sequence callback VIs (including Pre-Run and Post-Run VIs), and any test VIs.

When preparing these components for distribution to a test-stand computer, you must save the VIs, with their entire hierarchies, excluding those VIs already present in the Test Executive engine, to a new location (either a directory or VI library). You then must distribute the resulting directories or VI libraries along with the Run-Time System executable and `testexec.ini` file to the test stand computer.

The Test Executive 5.1 includes a utility to aid in the proper saving of your Test Executive VIs. To save a Test Executive VI for distribution with the Run-Time System, use the following steps:

1. Select **Project»Test Executive»Utilities»Save Test Executive VI for Distribution…**.

2. Using the first file dialog box, select the Test Executive VI you want to save for distribution.

3. Using the second file dialog box, choose a new location for this VI and its subVIs. You also can create new directories or a VI library at this time.

4. Choose whether to save the VIs with or without their block diagrams. Saving without diagrams is not required, but it makes the resulting files smaller.

5. LabVIEW then saves your Test Executive VI and all of its subVIs (excluding those VIs already present in the Test Executive engine) to the directory or VI library you specified in step 3.

You must follow the procedure listed in the previous paragraph for every VI required to create a complete Test Executive. These VIs include:

• The Operator Interface VI specified in your `testexec.ini` file

• All System Callback VIs and default Sequence Callback VIs specified in your `testexec.ini` file

• All Sequence Callback VIs and Pre-Run/Post-Run VIs needed by any sequence that you will execute with this Test Executive installation

• All test VIs needed by any sequence that you will execute with this Test Executive installation

To streamline the saving process, create a VI that has all of your test VIs or callback VIs on its diagram. Then save this VI using the steps listed earlier in this section. Using this streamlined method, you do not have to follow the steps for saving VIs individually for every VI used with the Test Executive. Instead, you only perform the saving procedure once.

Converting Sequence Files from Previous Versions

You access the Sequence File Converter for converting version 4.0 or 5.0 sequence files to version 5.1 by selecting **Project»Test Executive»Utilities»Sequence File Converter…**.

To convert version 3.0 sequences to version 5.1, you must first convert to version 4.0 using the 4.0 Sequence File Converter included with the Test Executive 4.0 or 5.0.

Opening a Sequence File on Launch

To have the Test Executive open a specific sequence file when launched, specify the path to the sequence file in the **Path to Seq. File** control on the Operator Interface VI. This control is located beneath the main controls on the front panel of the VI. After specifying the sequence file, pop up on the control and select **Data Operations»Make Current Value Default**. Save the changes to the VI.

When you specify a file path in the **Path to Seq. File** control, the Test Executive opens that sequence file when launched.

Calling the Test Executive as a SubVI

You can call the Test Executive Operator Interface as a subVI. The VI has the standard error input and output clusters as well as a path input to specify the initial sequence file to load, if one is desired.

Font Size Issues (UNIX only)

Depending on the configuration of your X server software, the text on the Test Executive user interface panels might be too large for the space allotted. If you observe this, change the default Application Font for LabVIEW (and the run-time executable in the case of the Run-Time System) to a smaller-size font. You can make this modification with the LabVIEW Preferences dialog box. See Chapter 7, *Customizing Your Environment*, in the *G Programming Reference Manual* for more information on changing default fonts.

Reserved Error Codes

The Test Executive reserves error codes 5000-5099. When creating LabVIEW Tests, C Tests, and callback VIs, do not use any error codes within this range.