

## RELEASE NOTES

# NI TestStand™ ATML Toolkit

Version 2010

These release notes contain NI TestStand ATML Toolkit 2010 system requirements, installation instructions, information about new features, and other changes since the ATML Toolkit 1.0.

## Contents

---

Getting Started .....	1
New TestStand Users .....	2
New ATML Toolkit Users .....	2
Recommended System Requirements .....	2
TestStand Directory Structure .....	3
Installation Instructions.....	4
Activating ATML Toolkit Licenses .....	5
Migrating to ATML Toolkit 2010 .....	5
What's New in ATML Toolkit 2010 .....	5
Enhanced ATML Test Description Translator .....	5
Improved Incremental Update Support.....	6
Instructions for Completing the Test Program .....	7
Deleting Sequence File and Code Module Constructs .....	8
New ATML Test Results Report Style Sheet .....	8
Sequence File Updater Application .....	10
Behavior Changes .....	11

## Getting Started

---

The ATML Toolkit adds functionality to TestStand for translating an ATML Test Description (TD) file to a TestStand sequence file and LabVIEW and LabWindows™/CVI™ code modules and updates the ATML Test Results report generator.

## New TestStand Users

If you are a new TestStand user, National Instruments recommends completing the tasks listed in the *New Users* section of the *Guide to Documentation* topic of the *NI TestStand Help*, which is located by default at `<TestStand>\Doc\Help\TSHelp.chm`.

## New ATML Toolkit Users

To familiarize yourself with the ATML Toolkit, explore the *NI TestStand ATML Toolkit Help*, which is located by default at `<TestStand>\Doc\Help\TSATHelp.chm`. The *NI TestStand ATML Toolkit Help* contains information about the toolkit features and functionality. You can view the *NI TestStand ATML Toolkit Help* as a part of the *NI TestStand Help*.



**Note** The ATML Toolkit documentation refers to XML concepts and assumes that you have a working knowledge of ATML, the ATML TD standard, the ATML Test Results and Session Information standard, XML, XML Path Language (XPath), XML Schema Document (XSD), and XQuery. Refer to the World Wide Web Consortium (W3C) Web site, located at [www.w3.org](http://www.w3.org), for more information about these technologies. You can find additional resources and tutorials online to help you learn these technologies. Refer to the *ATML TD Standards* topic in the *NI TestStand ATML Toolkit Help* for more information about the ATML TD standard. Refer to the *ATML Test Results and Session Information* topic in the *NI TestStand Help* for more information about the ATML Test Results and Session Information standard.

## Recommended System Requirements

National Instruments recommends the following system requirements to run the TestStand ATML Toolkit 2010. Minimum system requirements follow in parentheses.

- TestStand 2010 (TestStand 4.2.1). Refer to the *NI TestStand Release Notes* for information about the recommended system requirements for TestStand.
- 40 MB of free hard disk space.

Use the following Microsoft operating systems with the ATML Toolkit 2010:

- Windows 7 (32-bit and 64-bit), including Starter Edition
- Windows Vista (32-bit and 64-bit) Business, Enterprise, or Ultimate Service Pack 2
- Windows XP Service Pack 3 (Service Pack 2)
- Windows Server 2008 R2 (64-bit) and Windows Server 2003 R2 (32-bit)



**Note** This version of the ATML Toolkit supports, and was tested with, the latest operating system service packs that were available at the time this version of the ATML Toolkit was released. National Instruments recommends using the ATML Toolkit with the latest operating system service pack.

Use the following Microsoft software with the ATML Toolkit 2010:

- Microsoft Internet Explorer version 7.0 or later

Use the following National Instruments application development environments (ADEs) with the ATML Toolkit 2010:

- LabVIEW 8.2.1 or later to use the ATML TD translator to translate ATML TD files to LabVIEW code modules
- LabVIEW 8.6.1 or later to use the callbacks the translator provides for building a custom code generator you can use to modify the code modules the translator generates
- LabWindows/CVI 8.1.1 or later to use the ATML TD translator to translate the ATML TD files to LabWindows/CVI code modules
- LabWindows/CVI 8.5 or later to use the callbacks the translator provides for building a custom code generator you can use to modify the code modules the translator generates

The ATML Toolkit supports version 1.01 of the ATML TD standard, which is an approved standard under IEEE 1671 Standard for Automatic Test Markup Language (ATML) for Exchanging Automatic Test Equipment and Test Information via XML. The ATML Toolkit supports version 2.02 of the ATML Test Results and Session Information schema, which is a component standard of IEEE 1636 Software Interface for Maintenance Information Collection Analysis (SIMICA). Refer to the IEEE Web site at [www.ieee.org](http://www.ieee.org) for more information about these standards.

## TestStand Directory Structure

To comply with Windows 7/Vista restrictions on writing to the Program Files directory and to improve usability for Windows XP users who do not have access to the Program Files directory, TestStand installs files in the following directories:

- **<TestStand>**—Located by default at C:\Program Files\National Instruments\TestStand on Windows 32-bit systems and at C:\Program Files (x86)\National Instruments\TestStand on Windows 64-bit systems.
- **<TestStand Public>**—Located by default at C:\Users\Public\Documents\National Instruments\TestStand on Windows 7/Vista and at C:\Documents and Settings\All Users\Documents\National Instruments\TestStand on Windows XP.

- **<TestStand Application Data>**—Hidden by default and located at C:\ProgramData\National Instruments\TestStand on Windows 7/Vista and at C:\Documents and Settings\All Users\Application Data\National Instruments\TestStand on Windows XP.
- **<TestStand Local Application Data>**—Hidden by default and located at <User Directory>\AppData\Local\National Instruments\TestStand on Windows 7/Vista and at <User Directory>\Local Settings\Application Data\National Instruments\TestStand on Windows XP.

Refer to the *TestStand Directory Structure* section of Chapter 8, *Customizing and Configuring TestStand*, of the *NI TestStand Reference Manual* for more information about these directories and the files TestStand installs.

## Installation Instructions

The ATML Toolkit installation program installs the ATML TD translator files to the <TestStand Public>\Components\Translators directory and updates the existing TestStand process models and report generator files located in the <TestStand>\Components\Models\TestStandModels directory. The toolkit also adds a Configure ATML Test Description Translator Configuration dialog box option to the Tools menu in TestStand.



**Note** Because the TestStand installer includes a dependency on the Process Model Installer component, you must uninstall TestStand 4.2.1 or TestStand 2010 to remove the process model updates the ATML Toolkit installs.

National Instruments recommends exiting all programs before running the ATML Toolkit installer. Applications that run in the background, such as virus scanning utilities, might cause the installer to take longer than average to complete.

Complete the following steps to install the ATML Toolkit.

1. Log on as an administrator or as a user with administrator privileges.
2. Insert the ATML Toolkit installation media and follow the instructions that appear on the screen.



**Note** You must exit the TestStand Sequence Editor or any TestStand User Interface before you install the toolkit.

National Instruments recommends installing the complete ATML Toolkit program for the version of TestStand you currently have installed. If you perform a custom installation and do not install all the toolkit features, you can run the installation program again later to install additional features.

After the toolkit installation completes, you can begin using the toolkit functionality in the existing TestStand installation. Refer to the *Getting Started with the ATML TD Translator* topic in the *NI TestStand ATML Toolkit Help* for more information about getting started with the toolkit features.

## Activating ATML Toolkit Licenses

When you install the ATML Toolkit, you can use the NI Activation Wizard to activate the software. If you do not activate the software, you can initiate the evaluation period for the software when you begin using it. You need the serial number and the name of the software kit to activate the ATML Toolkit. You can find both of these items on the Certificate of Ownership card included in your software kit. Refer to the *Activating Your Software* topic in the *NI TestStand Help* for more information about how to activate National Instruments software.

## Migrating to ATML Toolkit 2010

If you are migrating from the ATML Toolkit 1.0, you must run the Sequence File Updater application included with the ATML Toolkit installer to convert any sequence file you originally translated using the ATML Toolkit 1.0 to a sequence file you can incrementally update using the ATML Toolkit 2010. Refer to the [Sequence File Updater Application](#) section of this document for more information about using the Sequence File Updater application.

## What's New in ATML Toolkit 2010

---

This section describes the new features in the ATML Toolkit 2010 and other changes since the ATML Toolkit 1.0.

### Enhanced ATML Test Description Translator

The ATML TD translator now supports translating limits associated with steps translated from `<TestGroupCall>` elements and localizing error and log messages.

## Translating Limits Associated with Steps Translated from <TestGroupCall> Elements

In the case in which a TD file contains an <Action> element with a <Behavior> child element that contains a <TestGroupCall> child element, the translator in the ATML Toolkit 2010 uses the <TestLimits> element to determine the appropriate type of step to generate in the translated sequence file. Depending on the content of the <TestLimits> element, the translator generates a Pass/Fail Test, Numeric Limit Test, Multiple Numeric Limit Test, or String Value Test step. The translator in the ATML Toolkit 1.0 always generated a Pass/Fail Test step in this case.

Refer to the *Determining Test Steps to Create for <Limit> Elements* topic of the *NI TestStand ATML Toolkit Help* for more information about how the translator determines step types.

## Localization Support

The ATML Toolkit 2010 adds support for localizing user interface strings. The following toolkit features contain strings you might want to localize:

- Dialog box messages when an error occurs during translation or an incremental update
- Log file entries the translator generates for each translation or incremental update
- The string labels that display in the ATML Test Description Translator Configuration dialog box and the Update Options dialog box

The ATML Toolkit 2010 includes an `ATMLTDString.ini` string resource file, located in the `<TestStand Public>\Components\Language\English` directory. You can modify the strings in the resource file to modify the error or log messages. To support error and log messages for multiple languages, copy the resource file and place it in the required language directory in the `<TestStand Public>\Components\Language` directory.

Refer to the *Creating String Resource Files* section of Chapter 8, *Customizing and Configuring TestStand*, of the *NI TestStand Reference Manual* for more information about using string resource files in TestStand.

## Improved Incremental Update Support

The ATML Toolkit 2010 adds support for updating existing LabVIEW or LabWindows/CVI code modules the translator created for a previously translated sequence file.

- **LabVIEW Code Modules**—When incrementally updating a previously translated sequence file, the translator determines whether a VI already exists at the path you specify in the LabVIEW VI Path

control on the Code Module Settings tab of the ATML Test Description Translator Configuration dialog box. If the VI exists at the path you specified and the VI has a name that is the same as the value of the name attribute of the parent `<Action>` element, the translator updates in the VI connector pane the existing block diagram labels that correspond to the `<AutomaticGeneration>`, `<Description>`, `<Extension>`, `<IeeeStd1641>`, and `<Operations>` child elements of the `<Behavior>` element. If the VI does not exist, the translator creates a new VI.

- **LabWindows/CVI Code Modules**—When incrementally updating a previously translated sequence file, the translator determines whether a function already exists in the C source file at the path you specify in the LabWindows/CVI Source File Path control on the Code Module Settings tab of the ATML Test Description Translator Configuration dialog box. If the source file already exists at the path you specified and the file contains a function with a name that is the same as the value of the name attribute of the parent `<Action>` element, the translator updates the comments in the function definition that correspond to the `<AutomaticGeneration>`, `<Description>`, `<Extension>`, `<IeeeStd1641>`, and `<Operations>` child elements of the `<Behavior>` element. If the function does not exist, the translator creates a new function in the source file.

The translator also updates block diagram labels or function comments that correspond to the `<LocalSignal>`, `<TestLimit>`, `<Parameter>`, `<SessionDatum>`, and `<TestResult>` child elements of the parent `<Action>` element.

In addition, the ATML Toolkit 2010 adds support for incrementally updating TestStand steps, attributes, and variables created from the child elements of a `<TestGroup>` element of the `TestGroupParallel` type.

Refer to the *NI TestStand ATML Toolkit Help* for more information about how the translator updates block diagram labels or function comments.

## Instructions for Completing the Test Program

The translator cannot translate all elements in a source TD file to TestStand constructs, such as steps, sequences, or code modules. The translator stores these elements in TestStand variables, attributes, or comments, but the generated sequence file does not use the variables, attributes, or comments.

The ATML Toolkit 2010 translation log file lists the elements the translator translated but cannot implement and describes how you must implement the required functionality to complete the test program. The log file uses subsections for the sequences and steps that contain the information you

must implement. The translator generates the log file in the same directory that it generates the translated sequence file.

Refer to the *NI TestStand ATML Toolkit Help* for more information about completing the test program.

## Deleting Sequence File and Code Module Constructs

The General Settings tab of the ATML Test Description Translator Configuration dialog box now includes a Delete Previously Generated Sequence File and Code Module Constructs option, which is enabled by default.

When you enable this option, the translator deletes TestStand steps, sequences, and local variables and block diagram labels in LabVIEW code modules or function comments in LabWindows/CVI code modules created during the previous translation. The translator deletes these items when incrementally updating a sequence file if you have deleted the corresponding elements from the source TD file.

When you disable this option, the translator retains TestStand steps, sequences, and local variables and block diagram labels in LabVIEW code modules or function comments in LabWindows/CVI code modules when incrementally updating a sequence file, even if you have deleted the corresponding elements from the source TD file. The translator deletes in the retained TestStand constructs the attributes previously created for the attributes of translated elements in the source TD file. For comments in retained code modules, the translator deletes from the comments the `<ATMLTranslatorComment>` and `</ATMLTranslatorComment>` tags previously created for some elements in the source TD file.

Refer to the *NI TestStand ATML Toolkit Help* for more information about how the translator creates code module comments for some TD elements and about how the translator deletes sequence file and code module constructs depending on the change you perform in the source TD file.

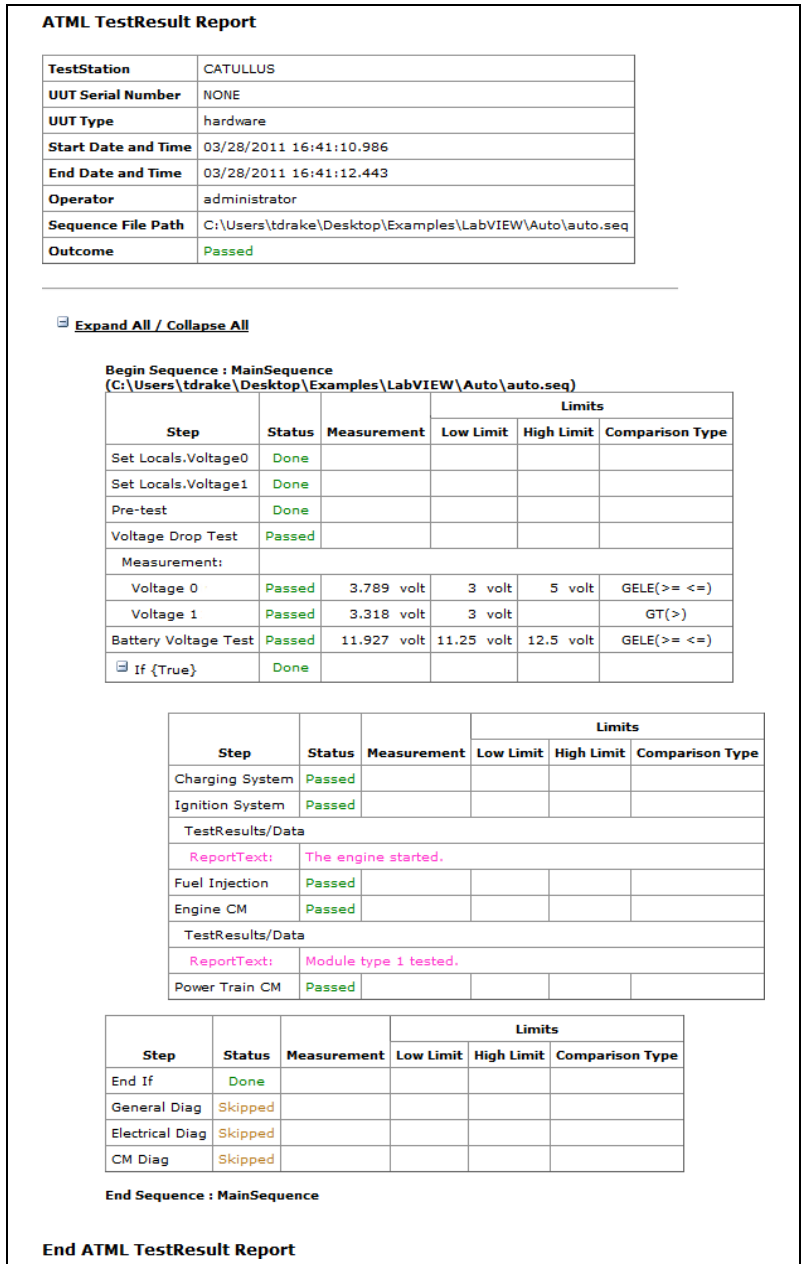
## New ATML Test Results Report Style Sheet

The ATML Toolkit 2010 includes a new ATML Test Results report style sheet, `TR_horizontal.xsl`, located in the `<TestStand>\Components\Models\TestStandModels\ATML\StyleSheets` directory. The style sheet displays an ATML Test Results report in a concise tabular format with expand and collapse sections.

Select **Configure»Report Options** to launch the Report Options dialog box. Click the **Contents** tab, select **ATML Report Document** from the Report Format ring control, and select `TR_horizontal.xsl` from the Stylesheet ring control to select the new style sheet when you generate an



ATML Test Results report. Figure 1 shows an example of a report that uses the TR\_horizontal.xml style sheet.



**Figure 1.** Report Generated with TR\_horizontal.xml Style Sheet

Refer to the *NI TestStand ATML Toolkit Help* for more information about using and customizing the style sheet and its features.

# Sequence File Updater Application

The ATML Toolkit 2010 changes the prefixes of the types contained in the `ATMLTDTypes.ini` and `ATMLTD_IEEE1641_Types.ini` type palette files and also renames the Attributes subproperty of the types as `ATMLAttributes`. As a result of these naming changes, the ATML Toolkit 2010 does not correctly incrementally update sequence files you originally translated using the ATML Toolkit 1.0. Refer to the [Behavior Changes](#) section of this document for more information about these naming changes.

You can use the Sequence File Updater application to convert sequence files you originally translated using the ATML Toolkit 1.0 to a sequence file you can use for an incremental update using the ATML Toolkit 2010.

Complete the following steps to install and launch the Sequence File Updater application.

1. Copy the `ATMLSequenceFileConverter` directory from the ATML Toolkit installer directory, located on the ATML Toolkit 2010 media, to a directory on the computer.
2. Double-click `SequenceFileConverter.exe`, located in the `ATMLSequenceFileConverter` directory, to launch the Update Sequence Files Translated with the ATML Toolkit 1.0 dialog box.
3. Use the **Path of Sequence File to be Updated** control to specify the path of the sequence file translated using the ATML Toolkit 1.0.
4. Optionally, use the **Path of Test Description File (Optional)** control to specify the path of the source TD file used to generate the sequence file, if the source TD file is available.
5. Click **OK** to update the sequence file for the ATML Toolkit 2010.

Refer to the *NI TestStand ATML Toolkit Help* for more information about how the Sequence File Updater application updates the sequence file for the ATML Toolkit 2010 and for information about using command-line syntax to specify a sequence file and a source TD file to update.

# Behavior Changes

Table 1 describes changes in behavior between the ATML Toolkit 1.0 and the ATML Toolkit 2010.

**Table 1.** Behavior Changes between the ATML Toolkit 1.0 and the ATML Toolkit 2010

Behavior	ATML Toolkit 1.0 Behavior	ATML Toolkit 2010 Behavior
<p>When translating the &lt;Datum&gt; child element of a &lt;Parameter&gt; element.</p>	<p>The translator translates the parent &lt;Action&gt; element as a TestStand step, the &lt;Parameter&gt; element as a parameter of the step, and the &lt;Datum&gt; element as a local variable of the sequence that contains the step created for the &lt;Action&gt; element. The value expression for the parameter contains the lookup string for the local variable created for the &lt;Datum&gt; element.</p>	<p>The translator translates the parent &lt;Action&gt; element as a TestStand step and the &lt;Parameter&gt; element as a parameter of the step but no longer creates a local variable when translating &lt;Datum&gt; elements of the following types:</p> <ul style="list-style-type: none"> <li>• binary</li> <li>• boolean</li> <li>• double</li> <li>• hexadecimal</li> <li>• integer</li> <li>• octal</li> <li>• string</li> <li>• unsigned integer</li> </ul> <p>Instead, the value expression for the parameter contains the value of the &lt;Parameter&gt; element.</p>
<p>When translating the &lt;ActionEntryPoints&gt; child element of the &lt;EntryPoints&gt; element.</p>	<p>The translator completes the following steps to translate the &lt;ActionEntryPoints&gt; element.</p> <ol style="list-style-type: none"> <li>1. Identifies the &lt;Action&gt; element that the actionID attribute of the first &lt;ActionEntryPoint&gt; child element of the &lt;ActionEntryPoints&gt; element references.</li> <li>2. Identifies the sequence that corresponds to the first &lt;TestGroup&gt; element that contains a reference to the &lt;Action&gt; element identified in step 1.</li> <li>3. Inserts a Goto step in the Setup step group of the sequence identified in step 2.</li> <li>4. Specifies that the destination step for the Goto step is the step that corresponds to the &lt;Action&gt; element identified in step 1.</li> <li>5. Translates the &lt;EntryPoints&gt; parent element as a Sequence Call step and inserts the step into the MainSequence.</li> </ol>	<p>The translator completes the following additional steps to add new functionality when translating the &lt;ActionEntryPoints&gt; element.</p> <ol style="list-style-type: none"> <li>1. Adds a Boolean parameter to the sequence that contains the Goto step.</li> <li>2. Specifies the value of the parameter as False.</li> <li>3. Specifies that the lookup string of the parameter is the precondition of the Goto step.</li> <li>4. Specifies that the Sequence Call step the translator created for the &lt;EntryPoints&gt; parent element passes a value of True for the Boolean parameter created in step 1.</li> </ol> <p>As a result, other Sequence Call steps that call the sequence in which the Goto step exists pass a default value of False for the Boolean parameter, and the Goto step executes only once.</p>

**Table 1.** Behavior Changes between the ATML Toolkit 1.0 and the ATML Toolkit 2010 (Continued)

Behavior	ATML Toolkit 1.0 Behavior	ATML Toolkit 2010 Behavior
<p>When translating the &lt;EntryPoints&gt; child element of a &lt;TestGroup&gt; element of the TestGroupSequence type.</p>	<p>The translator creates an EntryPoints container local variable in the translated sequence and translates individual &lt;EntryPoint&gt; elements as child containers of the EntryPoints container.</p> <p>The translator also stores all attributes of an &lt;EntryPoint&gt; element under an Attributes container local variable.</p> <p>You must specify that the first step in the translated sequence is the primary entry point because the translator does not support any other option.</p>	<p>For an &lt;EntryPoint&gt; element with an ID attribute value that matches the value of the primaryEntryPointID attribute of the parent &lt;EntryPoints&gt; element, the translator inserts a Goto step as the first step of the Main step group in the translated sequence.</p> <p>The destination for the Goto step is the step the translator creates for a &lt;Step&gt; element with an ID attribute value that is the same as the value of the stepID attribute of the &lt;EntryPoint&gt; element.</p>
<p>The ATMLTDTypes.ini type palette file, located in the &lt;TestStand Public&gt;\Components\ TypePalettes directory, contains data types from the TestDescription.xsd, Common.xsd, and HardwareCommon.xsd ATML TD schema files.</p> <p>The ATMLTD_IEEE1641_ Types.ini type palette file, located in the same directory, contains data types from the STDTSF.xsd, STDBSC.xsd, and STDTSFLib.xsd IEEE 1641 schema files.</p>	<p>The types defined in both type palette files include an ATML prefix on the type names.</p> <p>The translator translates attributes defined in the ATML and IEEE 1641 schemas as a subproperty, named Attributes, in the types defined in TestStand.</p> <p>The translator translates XML elements and attributes with the name Value as a local variable, named ATML_Value, in the translated sequence.</p>	<p>The types defined in the ATMLTDTypes.ini type palette file include an NI_ATML prefix on the type names.</p> <p>The types defined in the ATMLTD_IEEE1641_ Types.ini type palette file include an NI_ATMLSTD prefix on the type names.</p> <p>The translator translates attributes defined in the ATML and IEEE 1641 schemas as a subproperty, named ATMLAttributes, in the types defined in TestStand.</p> <p>The translator translates XML elements and attributes with the name Value as a local variable, named NI_ATMLValue, in the translated sequence.</p> <p><b>Note:</b> The ATML Toolkit 2010 does not correctly incrementally update sequence files you originally translated using the ATML Toolkit 1.0. The ATML Toolkit 2010 includes a Sequence File Updater application to convert sequence files you originally translated using the ATML Toolkit 1.0. Refer to the <a href="#">Sequence File Updater Application</a> section of this document for more information about the Sequence File Updater application.</p>

CVI, LabVIEW, National Instruments, NI, ni.com, NI TestStand, the National Instruments corporate logo, and the Eagle logo are trademarks of National Instruments Corporation. Refer to the *Trademark Information* at [ni.com/trademarks](http://ni.com/trademarks) for other National Instruments trademarks. The mark LabWindows is used under a license from Microsoft Corporation. Windows is a registered trademark of Microsoft Corporation in the United States and other countries. 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 Patent Notice* at [ni.com/patents](http://ni.com/patents). Refer to the *Export Compliance Information* at [ni.com/legal/export-compliance](http://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.