NI-TClk Synchronization Help

For National Instruments SMC-Based Digitizers, Signal Generators, and Digital Waveform Generators/Analyzers

February 2007, 370956F-01

This help file contains information about how to synchronize National Instruments digitizers, signal generators, and digital waveform generator/analyzers based on the <u>Synchronization and Memory Core</u> (<u>SMC</u>) technology. This help file also contains programming flows and programming reference information for NI-TClk VIs and properties, and NI-TClk C functions and attributes.

To navigate this help file, use the **Contents**, **Index**, and **Search** tabs to the left of this window.

To comment on National Instruments documentation, refer to the <u>National</u> <u>Instruments Web site</u>.

© 2003–2007 National Instruments Corporation. All rights reserved.

NI-TClk Overview

You can use TClk synchronization to have multiple instruments simultaneously respond to triggers, to align sample clocks on instruments, and/or to simultaneously start multiple instruments. TClk synchronization is supported on National Instruments digitizers, signal generators, and digital waveform generator/analyzers that are based on the <u>SMC</u> technology.

The NI-TClk <u>library</u> allows you to synchronize NI modular instruments in the following situations:

- In a single or in multiple PXI chassis
- In a single PC
- When triggers are used for any purpose, the same or different, by an individual instrument. For example, when a start trigger from one modular instrument is used as the start trigger on another modular instrument, or when a start trigger from one modular instrument is used as the reference trigger on another modular instrument.
- When sample clock rates are equal, and also when the sample clock rates are different.



Note Refer to <u>NI-TClk External Sample Clock and External Sample Clock Timebase</u> <u>Considerations</u> and <u>NI-TClk Sample Clock Delay</u> for more information about using the sample clocks.

NI-TClk does not impose any limitations on the number of modules or chassis that you can synchronize.

The TClk Signal

The TClk synchronization method is based on a periodic signal used for trigger transmission. This signal is known as the Trigger Clock (TClk). The TClk signal is not shared by the devices. Instead, the TClk signal is internally generated by each device from the sample clock or sample clock timebase for that device. <u>Sync Pulse and Sync Pulse Clock</u> are additional signals used during TClk synchronization.

To generate TClk signals, devices must have a common reference clock (for internal sample clocks) or a common sample clock or sample clock timebase (for <u>external sample clocks or sample clock timebases</u>).



Note Common sample clock timebases apply only to devices with decimation, such as digitizers.

Example

The following figure shows an example of synchronizing two devices that have internal sample clocks and a common 10 MHz reference clock.



Device 1 is the Start Trigger master, with a sample clock rate of 50 MHz. Device 2 is the Start Trigger slave, with a sample clock rate of 20 MHz. The TClk frequency is 5 MHz.

The Effect of TClk on Trigger Response Time

Because devices react to TClk-synchronized triggers with delays and uncertainties that are comparable to the TClk period (200 ns or more), the trigger reaction time for synchronized devices is higher than that for individual devices. However, this condition does not impact time stamping of the reference triggers for acquisitions, so the reference trigger positions are correctly reported for TClk-synchronized devices.

Instrument Drivers and NI-TClk

Instrument drivers in the context of this document refer to libraries of VIs or functions for controlling individual devices. For example, NI-SCOPE, NI-FGEN, and NI-HSDIO are instrument drivers. NI-TClk is a *library* designed for use with instrument drivers.

High-level NI-TClk VIs or functions are different from typical instrument driver VIs or functions in that they accept arrays of sessions (as opposed to individual sessions). Session is a software concept for communicating and interacting with a particular physical instrument or a subset of a physical instrument. For example, you can use an NI-SCOPE session to communicate with a digitizer, and an NI-HSDIO acquisition session to communicate with the acquisition subset of a digital waveform generator/analyzer.

In LabVIEW, these sessions are obtained from your instrument driver by using the *<driver name>* Get Session Reference VI—for example, the niScope Get Session Reference VI. In C, NI-TClk accepts regular instrument driver sessions.

SMC Technology

For more information about the SMC technology, refer to the National Instruments Developer Zone document, <u>National Instruments</u> <u>Synchronization and Memory Core—A Modern Architecture for Mixed-Signal Test</u>.

NI-TClk Homogeneous and Heterogeneous Triggers

NI-TClk provides a simplified interface for synchronizing systems with <u>homogeneous</u> triggers in a single PXI chassis or PC. NI-TClk also provides a flexible interface for other supported systems.

Triggers Supported by NI-TClk

The following table summarizes the triggers supported by NI-TClk.

Trigger	Acquisition	Generation
Start Trigger	Yes	Yes
Reference Trigger	Yes	N/A
Script Trigger 0 – 3	N/A	Yes
Pause Trigger	No	Yes

Individual instruments may support some or all of the triggers listed in the previous table. Instruments may also support triggers other than the ones listed in the previous table. These other triggers, as well as pause triggers for acquisition sessions, are not supported by NI-TClk and are not relevant to the definition of homogeneous triggers. For information about which triggers your device supports, refer to the device user documentation.

Homogeneous Triggers

For purposes of TClk synchronization, synchronized sessions are defined to have homogeneous triggers if the triggers are synchronized. To get synchronous triggers, set *only* the master session (the first session by default) to use a software trigger. If not explicitly set, slave sessions use the same trigger source as the master session.



Note When a supported trigger is not configured, it is referred to as *None*. Refer to <u>Comparison of Terminology between NI-TClk, NI-HSDIO, NI-SCOPE, and NI-FGEN</u> for more information about trigger terminology.

Also, for each of the other <u>triggers supported by NI-TClk</u>, one of the following statements must be true:

- Trigger is not supported by any of the sessions.
- Trigger is configured as None by all sessions that support it.
- One of the sessions, called the trigger master session, is configured to receive the trigger from the external world (in case of external triggers) or to generate that trigger (in case of None or software triggers), and the corresponding trigger on all other sessions is configured so that it comes from the trigger on the trigger master session.

If there is a trigger master session, non-master sessions must use the triggers that they receive from the trigger master session for the same purpose as the trigger master session. For example, the reference trigger from the reference trigger master session must be used as the reference trigger for the non-master sessions.

The type of trigger (for example digital edge, analog hysteresis, or software) for the trigger master session is not relevant for this definition. The type of trigger for non-master sessions must be digital edge or digital level because the trigger coming from the trigger master is a digital signal.

With TClk synchronization, all sessions react to a synchronized trigger at the same time.

Heterogeneous Triggers

Sessions are defined to have heterogeneous triggers if they do not have homogeneous triggers. If any of the conditions from the definition of homogeneous triggers are not true, the sessions are considered to have heterogeneous triggers.

Examples

Example 1: Homogeneous Triggers

Sessions A and B have triggers as indicated in the following table.

Trigger	Session A (Generation)	Session B (Generation)
Start Trigger	Configured as software trigger	Configured to use start trigger from session A
Reference Trigger	Not supported	Not supported
Script Trigger 0 – 3	Not supported	Not supported
Pause Trigger	Not supported	Not supported

Sessions A and B have homogeneous triggers.

Example 2: Homogeneous Triggers

Sessions A, B, C, and D have triggers as indicated in the following table.

Trigger	Session A (Acquisition)	Session B (Acquisition)	Session C (Generation)	Session D (Generation)
Start Trigger	Configured as None	Configured to use start trigger from session A	Configured to use start trigger from session A	Configured to use start trigger from session A
Reference Trigger	Configured to use reference trigger from session B	Configured as analog edge trigger	Not supported	Not supported
Script Trigger 0 – 3	Not supported	Not supported	Configured as None	Configured as None
Pause Trigger	Not supported	Not supported	Configured as None	Configured as None

Sessions A, B, C, and D have homogeneous triggers. Notice that session A is the start trigger master session, and session B is the reference trigger master session.

Example 3: Heterogeneous Triggers

Sessions A, B, and C have triggers as indicated in the following table.

Trigger	Session A (Generation)	Session B (Generation)	Session C (Generation)
Start Trigger	Configured as software trigger	Configured to use start trigger from session A	Configured to use start trigger from session A
Reference Trigger	Not supported	Not supported	Not supported
Script Trigger 0	Configured as digital edge	Configured as None	Configured as None
Script Trigger 1 – 3	Configured as None	Configured as None	Configured as None
Pause Trigger	Configured as None	Configured as digital level	Configured to use pause trigger from session B

Sessions A, B, and C have heterogeneous triggers because neither the script trigger nor the pause trigger comply with any of the three required conditions, of which at least one condition must be met, for the reference, script, and pause triggers.

Related Topics

niTClk Configure For Homogeneous Triggers VI

<u>niTClk_ConfigureForHomogeneousTriggers</u> function

NI-TClk Sync Pulse and Sync Pulse Clock

Sync Pulse

The Sync Pulse signal is used to synchronize <u>TClk signals</u> on different devices. When you run the <u>niTClk Synchronize</u> VI or the <u>niTClk_Synchronize</u> function, NI-TClk causes one of the synchronized devices to produce a series of Sync Pulses, which *must* be received by all the synchronized devices. The Sync Pulse is used to measure the misalignment between TClks on different devices, as shown in the following figure.



If all the devices are in one PXI chassis or one PC, the niTClk Synchronize VI or the niTClk_Synchronize function routes the Sync Pulse automatically, assuming that a <u>PXI trigger line or a RTSI line</u> is available.

If the devices are in <u>multiple PXI chassis</u>, or if the devices are in a PXI chassis with <u>multiple bus segments (18-slot chassis</u>), the Sync Pulse signal is not automatically routed, so you must configure the signal routing for the system.

Sync Pulse Clock

Sync Pulse Clock is another signal used during the synchronization of TClk signals on different devices. For PXI devices, this signal must come from PXI_CLK10. For PCI devices, it must come from RTSI 7. Typically, the Reference Clock is used as the Sync Pulse Clock, as shown in the previous figure.

For both PXI and PCI devices, the signal must be 10 MHz. The niTClk Synchronize VI or the niTClk_Synchronize function routes this signal automatically unless you use the <u>Sync Pulse Clock Source</u> property or the <u>NITCLK_ATTR_SYNC_PULSE_CLOCK_SOURCE</u> attribute.

NI-TClk PXI and RTSI Trigger Lines

NI-TClk uses the PXI trigger lines and the RTSI trigger lines to share signals between devices.

Using the PXI Trigger Lines



Caution Because PXI_CLK10 and the PXI trigger lines are essential for synchronization, you must use a PXI chassis. You cannot use a CompactPCI chassis that is not also a PXI chassis.

To use the routing provided by the high-level NI-TClk VIs or functions, and to avoid getting errors, configure your PXI system in MAX.

Configuring your PXI system requires two steps: identifying the PXI system controller and identifying the PXI chassis.



Note If you want to directly drive certain PXI trigger lines with a device that is *not* listed under **Devices and Interfaces**»**NI-DAQmx Devices** in the MAX Configuration tree, you also need to statically reserve those PXI trigger lines in MAX.

For more information about configuring your PXI system, refer to the *Measurement & Automation Explorer Help for PXI*. To access this help file, launch MAX, and select **Help»Help Topics»PXI**.

Using the RTSI Trigger Lines

To use the routing provided by the high-level NI-TClk VIs or functions and to avoid getting errors, connect the devices with a RTSI cable and create a new device entry for the RTSI cable in MAX. For information about connecting your devices with a RTSI cable, refer to the device documentation.



Note If you want to directly drive certain RTSI lines with a device that is *not* listed under **Devices and Interfaces»NI-DAQmx Devices** in the MAX Configuration tree, you need to reserve those RTSI lines in MAX.

For more information about using the RTSI trigger lines, refer to the *Measurement & Automation Explorer Help for NI-DAQmx*. To access this help file, launch MAX, and select **Help»Help Topics»NI-DAQmx**.

NI-TClk External Sample Clock and External Sample Clock Timebase Considerations

NI-TClk works with internal and external sample clocks, and with internal and external sample clock timebases for the NI modular instruments that support them.

If all of the sessions that you are synchronizing have external sample clocks or external sample clock timebases, then the sample clock or sample clock timebase rates of all the sessions must be equal.

If some of the sessions that you are synchronizing have internal sample clocks and other sessions have external sample clocks or external sample clock timebases, then the rates of all the external sample clocks or the external sample clock timebases must be equal to one of the internal sample clocks.

NI-TClk Comparison of Terminology between NI-TClk, NI-HSDIO, NI-SCOPE, and NI-FGEN

Some of the terminology used in NI-TClk and NI-HSDIO differs from the terminology used in NI-SCOPE and NI-FGEN. The following table lists the equivalent names used by NI-TClk and these drivers.

NI-TClk and NI-HSDIO Signal Names	NI-SCOPE Signal Names	NI-FGEN Signal Names
Reference Clock	Input Clock Output Clock Reference Clock	Reference Clock Board Clock
Sample Clock	Sample Clock	Update Clock Sample Clock
Start Trigger	Acquisition Arm Trigger Start Trigger	Trigger Start Trigger
Reference Trigger	Trigger Stop Trigger	N/A
None (no trigger configured)	Immediate Trigger	Immediate Trigger
RTSI 7	RTSI_CLK	RTSI_7

NI-TClk Synchronization in a Single PXI Chassis or a Single PC with Homogeneous Triggers

If you are synchronizing several instrument sessions with <u>homogeneous</u> <u>triggers</u> in a single PXI chassis or a single PC, then use the VIs or functions listed in the following table.

LabVIEW VI	C Function
niTClk Configure For Homogeneous Triggers VI	niTClk ConfigureForHomogeneousTriggers
niTClk Synchronize VI	niTClk Synchronize
niTClk Initiate VI	niTClk Initiate
(optional) <u>niTClk Is Done</u> VI <i>or</i> <u>niTClk Wait Until Done</u> VI	(optional) niTClk_IsDone or niTClk_WaitUntilDone
N/A	niTClk GetExtendedErrorInfo



Note Before using these NI-TClk VIs or functions, verify that your system is configured as specified in <u>PXI Trigger Lines and RTSI Lines</u>.

Programming Examples

During the instrument driver installation, NI-TClk programming examples were installed along with programming examples for the instrument driver. To access these examples, navigate to **Start»Programs»National Instruments»<***driver name***>*Examples**.

NI-TClk VI Programming Flow

Homogeneous Trigger, Single Chassis or PC



NI-TClk C Functions Programming Flow

Homogeneous Trigger, Single Chassis or PC



NI-TClk Synchronization in a Single PXI Chassis or a Single PC with Heterogeneous Triggers

If you are synchronizing several NI modular instruments with <u>heterogeneous triggers</u> in a single PXI chassis or a single PC, then use the instrument driver functions to route the triggers so that the triggers can be shared, and use the NI-TClk VIs or functions and properties or attributes listed in the following table.

LabVIEW VI	C Function
<u>niTClk Synchronize</u> VI	niTClk Synchronize
<u>niTClk Initiate</u> VI	<u>niTClk Initiate</u>
(optional) <u>niTClk Is Done</u> VI <i>or</i> <u>niTClk Wait Until Done</u> VI	(optional) niTClk_IsDone or niTClk_WaitUntilDone
N/A	niTClk GetExtendedErrorInfo

M

Also, use the NI-TClk properties or attributes listed in the following table to specify how the triggers are shared between devices.

LabVIEW Property	C Attribute
Start Trigger Master Session	NITCLK ATTR START TRIGGER MASTER SESSION
Reference Trigger Master Session	NITCLK ATTR REF TRIGGER MASTER SESSION
Script Trigger Master Session	NITCLK ATTR SCRIPT TRIGGER MASTER SESSION
Pause Trigger Master Session	NITCLK ATTR PAUSE TRIGGER MASTER SESSION

Note Before using these NI-TClk VIs or functions, verify that your system is configured as specified in <u>PXI Trigger Lines and RTSI Lines</u>.

When the sample clock rates, sample clock timebase rates, and/or the sample counts are different in acquisition sessions sharing the reference trigger, you should also set the holdoff attributes for the reference trigger master using the instrument driver.

Programming Examples

During the instrument driver installation, NI-TClk programming examples were installed along with programming examples for the instrument driver. To access these examples, navigate to **Start»Programs»National Instruments»<***driver name***>*Examples**.

NI-TClk VI Programming Flow

Heterogeneous Trigger, Single Chassis or PC



NI-TClk C Functions Programming Flow

Heterogeneous Trigger, Single Chassis or PC



NI-TClk Synchronization in Multiple PXI Chassis

If devices are in multiple PXI chassis, the <u>Sync Pulse</u> signal is *not* automatically routed, and you must configure the system so that *all* of the PXI_CLK10 signals in *all* of the chassis are driven from the same clock source.



Note To minimize phase differences between the chassis, use matched-length cables to drive $\mathsf{PXI_CLK10}$ to the different chassis.

Routing the Triggers

When you synchronize several NI modular instruments in multiple PXI chassis with <u>homogeneous or heterogeneous triggers</u>, use the instrument driver VIs or functions to route the triggers so that the triggers can be shared. Also, use the following NI-TClk VIs or functions:

LabVIEW VI	C Function
<u>niTClk Synchronize</u> VI	<u>niTClk Synchronize</u>
<u>niTClk Initiate</u> VI	<u>niTClk Initiate</u>
(optional) <u>niTClk Is Done</u> VI <i>or</i> <u>niTClk Wait Until Done</u> VI	(optional) niTClk_IsDone or niTClk_WaitUntilDone
N/A	niTClk GetExtendedErrorInfo

Also, use the NI-TClk properties or attributes listed in the following table to specify how the triggers are shared between devices.

LabVIEW Property	C Attribute
Start Trigger Master Session	NITCLK ATTR START TRIGGER MASTER SESSION
Reference Trigger Master Session	NITCLK ATTR REF TRIGGER MASTER SESSION
Script Trigger Master Session	NITCLK ATTR SCRIPT TRIGGER MASTER SESSION
Pause Trigger Master Session	NITCLK ATTR PAUSE TRIGGER MASTER SESSION

When the sample clock rates, sample clock timebase rates, and/or the sample counts are different in acquisition sessions sharing the reference trigger, you should also set the holdoff attributes for the reference trigger master using the instrument driver.



Note To get synchronous triggers, set *only* the master session (the first session by default) to use a software trigger. Sending a trigger to each device individually may not keep the devices in sync.

Routing the Sync Pulse Signal

To route the <u>Sync Pulse</u> signal in a multiple chassis system, use PXI synchronization modules, such as the NI PXI-6653, in each chassis. If you install these modules in Slot 2, you can conveniently use them to supply a common 10 MHz clock signal to all the chassis.

Use the following NI-TClk properties or attributes to specify the routing of the <u>Sync Pulse</u> signal:

LabVIEW Property	C Attribute
Sync Pulse Source	NITCLK ATTR SYNC PULSE SOURCE
Export Sync Pulse Output Terminal	NITCLK ATTR EXPORTED SYNC PULSE OUTPUT TERMINAL

Example

For example, if chassis A contains the device that is exporting the Sync Pulse signal, and chassis A is connected through the PXI synchronization modules and cables to chassis B and chassis C, then the following signal routing is recommended:

- 1. Route the Sync Pulse signal from the device that is exporting this signal to the PXI synchronization module using a <u>PXI trigger line</u> (line X).
- 2. Resynchronize to PXI_CLK10 in the PXI synchronization module and route the Sync Pulse signal to the following locations:
 - a. Another PXI trigger line (line Y) in the same chassis (where line Y is different from line X)
 - b. The PXI synchronization modules in chassis B and chassis C
- 3. Program the devices in chassis A to receive the signal from the PXI trigger line Y.
- 4. Route the Sync Pulse signal—*without resynchronizing*—from the PXI synchronization module on chassis B and chassis C to the PXI trigger line Y on chassis B and chassis C.
- 5. Program the devices in chassis B and chassis C to receive the Sync Pulse signal from PXI trigger line Y.

If PXI trigger line Y is not available in chassis B and chassis C, you can use any other available line. Using the same line in all chassis simplifies programming.

When using this configuration, keep the total signal propagation delay of the cables and the PXI synchronization modules to less than 100 ns.

For more information about synchronizing multiple PXI chassis, contact National Instruments technical support. For information on 18-slot PXI chassis, refer to <u>Configuring PXI Chassis with Multiple Bus Segments</u> (18-Slot Chassis).

NI-TClk VI Programming Flow

Multiple Chassis



NI-TCIk C Functions Programming Flow Multiple Chassis



NI-TClk Configuring PXI Chassis with Multiple Bus Segments (18-Slot Chassis)

Synchronizing modules in a PXI chassis that has multiple bus segments (for example, the NI PXI-1045) requires configuring the crosspoint switches through Measurement & Automation Explorer (MAX). The crosspoint switches are the routing modules used to drive PXI triggers across segments. Using the <u>niTClk Configure For Homogeneous</u> Triggers VI or the <u>niTClk ConfigureForHomogeneousTriggers</u> function to route triggers automatically configures the crosspoint switches. However, if you want to manually route the triggers and the master and slave devices are in different segments (as is necessary in a multi-chassis system), you must manually configure the crosspoint switches.

To manually configure the crosspoint switches in the chassis, complete the following steps:

- 1. Launch MAX.
- 2. Identify the chassis by following the procedures described in the Measurement & Automation Explorer Help for PXI.
- 3. Select the identified chassis name in the configuration tree and click the Triggers tab in the right pane. The Triggers tab contains the PXI trigger line reservations and routing options for the crosspoint switches.
- 4. Click the Trigger Routing tab and verify that all PXI trigger lines that carry manually configured triggers driven by the master device (for example, Start, Stop, and Exported Sync Pulse) have routing configured as away from the segment containing the master device. Also, verify that these PXI trigger lines are reserved so that no other devices can use them. You can change the configurations by using the list boxes next to each line to select Towards Slot 1, Outwards from Middle, or Away from Slot 1 for each manually configured trigger.
- 5. If you are using a Slot 2 controller to resynchronize the Exported Sync Pulse from the master device and send it back to all the

devices as the Sync Pulse, you must configure the routing for the PXI trigger line with the Sync Pulse as away from the first segment. The first segment should contain the Slot 2 controller.



Note Do not reserve the PXI trigger line with the Sync Pulse—the Slot 2 controller automatically reserves that line.
NI-TCIk Sample Clock Delay

If you are using the internal sample clock, you can delay the sample clock on some sessions relative to other sessions by setting the <u>Sample Clock Delay</u> property or the <u>NITCLK_ATTR_SAMPLE_CLOCK_DELAY</u> attribute.

NI-TCIk Performance Optimization

To ensure correct synchronization, you must call the <u>niTClk Synchronize</u> VI or the <u>niTClk_Synchronize</u> function every time you change a clocking or sample rate attribute for a TClk-synchronized device. Each call to the niTClk Synchronize VI or the niTClk_Synchronize function adds time to the application execution.

To optimize the performance of your application, set the clocking and sample rate attributes *only once*, then call the niTClk Synchronize VI or the niTClk_Synchronize function.

You do not need to call the niTClk Synchronize VI or the niTClk_Synchronize function every time you initiate or start an acquisition *unless* you are changing attributes that affect clocking or sample rate. These attributes are listed in the table below. For the NI 5922 digitizer, this list of attributes also includes channel-based attributes.

Caution NI-TClk does *not* detect any changes to the relevant attributes. Therefore, no errors are reported if the devices become unsynchronized because of an attribute change. You must remember to resynchronize if any of these attributes change.



Note You should call the <u>niTClk Configure for Homogenous Trigers</u> VI the <u>niTClk ConfigureForHomogeneousTriggers</u> function any time any driver attribute changes value.

Driver	LabVIEW Property	C Attribute	Comment
NI- FGEN	Sample Rate	NIFGEN_ATTR_ARB_SAMPLE_RATE	
NI- FGEN	Update Clock Source	NIFGEN_ATTR_UPDATE_CLOCK_SOURCE	
NI- FGEN	Clock Mode	NIFGEN_ATTR_CLOCK_MODE	
NI- FGEN	Reference Clock Source	NIFGEN_ATTR_REF_CLOCK_SOURCE	
NI- FGEN	Reference Clock Frequency	NIFGEN_ATTR_REF_CLOCK_FREQUENCY	
NI-	Frequency	NIFGEN_ATTR_FUNC_FREQUENCY	Only applies

FGEN			to some devices
NI- FGEN	Waveform	NIFGEN_ATTR_FUNC_WAVEFORM	Only applies to some devices
NI- FGEN	OSP Enabled	NIFGEN_ATTR_OSP_ENABLED	
NI- FGEN	IQ Rate	NIFGEN_ATTR_OSP_IQ_RATE	
NI- FGEN	CIC Interpolation Factor	NIFGEN_ATTR_OSP_CIC_FILTER_INTERPOLATION	
NI- FGEN	FIR Interpolation Factor	NIFGEN_ATTR_OSP_FIR_FILTER_INTERPOLATION	
NI- HSDIO	RefClk.Source	NIHSDIO_ATTR_REF_CLOCK_SOURCE	
NI- HSDIO	RefClk.Rate	NIHSDIO_ATTR_REF_CLOCK_RATE	
NI- HSDIO	RefClk.Impedance	NIHSDIO_ATTR_REF_CLOCK_IMPEDANCE	Only with external reference clock
NI- HSDIO	SampClk.Source	NIHSDIO_ATTR_SAMPLE_CLOCK_SOURCE	
NI- HSDIO	SampClk.Rate	NIHSDIO_ATTR_SAMPLE_CLOCK_RATE	
NI- HSDIO	SampClk.Impedance	NIHSDIO_ATTR_SAMPLE_CLOCK_IMPEDANCE	Only with external sample clock
NI- SCOPE	Min Sample Rate	NISCOPE_ATTR_MIN_SAMPLE_RATE	
NI- SCOPE	Input Clock Source	NISCOPE_ATTR_CLOCK_SOURCE	
NI- SCOPE	Reference Clock Rate	NISCOPE_ATTR_REF_CLK_RATE	
NI- SCOPE	Sample Clock Timebase Source	NISCOPE_ATTR_SAMP_CLK_TIMEBASE_SRC	
NI- SCOPE	Sample Clock Timebase Rate	NISCOPE_ATTR_SAMP_CLK_TIMEBASE_RATE	
NI- SCOPE	Sample Clock Timebase Divsor	NISCOPE_ATTR_SAMP_CLK_TIMEBASE_DIV	
NI- SCOPE	Max Input Frequency	NISCOPE_ATTR_MAX_INPUT_FREQUENCY	Only applies to some devices

NI-TClk Synchronization Repeatability Optimization

Every call to the <u>niTClk Synchronize</u> VI or <u>niTClk_Synchronize</u> function causes NI-TClk to measure the time between the Sync Pulse Clock and TClk. The imprecision associated with this measurement is usually very small relative to the <u>TClk timebase</u>. NI-TClk adjusts TClks and TClk timebases on the devices based on the measurement. Some devices can adjust their TClk timebases with very fine resolution, usually using an oscillator phase DAC and a Phase Locked Loop (PLL). These devices will appear to have synchronization jitter from repeated calls to the niTClk Synchronize VI or niTClk_Synchronize function.

You can eliminate the jitter associated with TClk measurements for most devices by setting the oscillator phase DAC value directly through the individual product drivers as follows:

- 1. Configure the devices for acquisition or generation synchronized with NI-TClk.
- 2. After the acquisition or generation is completed, but before you call Close on the product driver, read the oscillator phase DAC attribute using the individual product drivers for each synchronized device.
- 3. Store these values.
- 4. Before running the program again, change the program to set the phase DAC attributes to the stored values using the individual product drivers for each synchronized device before calling the niTClk Synchronize VI or the niTClk_Synchronize function.

When you follow this procedure, NI-TClk adjusts TClks but not TClk timebases on the synchronized devices, and the synchronization jitter is minimized.

NI-TClk VI Reference Help

The following table summarizes the VIs and properties suitable for different chassis and PC configurations.

	PXI C	hassis/PC Configurat	ion
	<u>Single PXI</u> <u>Chassis/PC,</u> <u>Homogeneous</u> <u>Triggers</u>	<u>Single PXI</u> <u>Chassis/PC,</u> <u>Heterogeneous</u> <u>Triggers</u>	<u>Multiple PXI</u> <u>Chassis</u>
VIs			
niTClk Configure For Homogeneous Triggers	Yes	_	
niTClk Synchronize	Yes	Yes	Yes
niTClk Initiate	Yes	Yes	Yes
niTClk Is Done	Yes	Yes	Yes
niTClk Wait Until Done	Yes	Yes	Yes
Property Node			
niTClk Property Node	—	Yes	Yes
Properties			
Active Channels	_	S	S
Start Trigger Master Session	_	Т	Т
Reference Trigger Master Session	_	Т	Т
Script Trigger Master Session	_	Т	Т
Pause Trigger Master Session	_	Т	Т
Sync Pulse Source	—	—	Yes
Export Sync Pulse Output Terminal	—	—	Yes
Sync Pulse Clock Source	R	R	—
Sample Clock Delay	Optional	Optional	Optional
R: Devices with RTSI trigger bus only; NI-TCIk	used when you want to	o control RTSI 7 directly	/ instead of through

T: Use attributes that pertain to the triggers that you are synchronizing S: Use with <u>Script Trigger Master Session</u> only

niTClk Configure For Homogeneous Triggers

Configures the properties commonly required for the TClk synchronization of device sessions with <u>homogeneous triggers</u> in a <u>single PXI chassis or a single PC</u>.

Use the niTClk Configure For Homogeneous Triggers VI to configure the properties for the <u>reference clocks</u>, <u>start triggers</u>, <u>reference triggers</u>, <u>script triggers</u>, and <u>pause triggers</u>.

If the niTClk Configure For Homogeneous Triggers VI cannot perform all the steps appropriate for the given sessions, it returns an error. If an error is returned, use the instrument driver VIs and properties for signal routing, along with the following NI-TClk properties:

- Start Trigger Master Session
- <u>Reference Trigger Master Session</u>
- Script Trigger Master Session
- Pause Trigger Master Session

<u>Details</u>



sessions is an array of session references that are being synchronized.

You obtain session references from the instrument driver by using *<driver name>* Get Session Reference VI, where *<driver name>* is the name of the instrument driver—for example, the niScope Get Session Reference VI.

- error in describes error conditions that occur before this VI or function runs.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE. If **status** is TRUE, this VI propagates the error and does not do anything else.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The **source** string includes the name of the VI that produced the error and often offers additional information about the error.

- **Sessions out** passes the array of session references to the next VI.
- **error out** contains error information. If **error in** indicates that an error occurred before this VI or function ran, **error out** contains the same error information. Otherwise, it describes the error status that this VI or function produces.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where and why an error occurred. The **source** string includes the name of the VI that produced the error and often offers additional information about the error.

The LabVIEW Simple Error Handler VI and the LabVIEW General Error Handler VI use the **source** string to deliver messages that may help you to troubleshoot your application.

niTClk Configure For Homogeneous Triggers Details

Reference Clocks

The niTClk Configure For Homogeneous Triggers VI configures the reference clocks if they are needed. Specifically, if the internal sample clocks or internal sample clock timebases are used, and the reference clock source is not configured—or is set to None (no trigger configured) —the niTClk Configure For Homogeneous Triggers VI configures the following:

- **PXI**—The reference clock source on all devices is set to be the 10 MHz PXI backplane clock (PXI_CLK10).
- **PCI**—One of the devices exports its 10 MHz onboard reference clock to RTSI 7. The reference clock source on all devices is set to be RTSI 7.

If the reference clock source is set to a value other than None, the niTClk Configure For Homogeneous Triggers VI does not configure the reference clock source.

Start Triggers

If the start trigger is set to None (no trigger configured) for all sessions, the sessions are configured to share the start trigger. The start trigger is shared by:

- Implicitly exporting the start trigger from one session
- Configuring the other sessions for digital edge start triggers with sources corresponding to the exported start trigger
- Setting the <u>Start Trigger Master Session</u> property to the session that is exporting the trigger for all sessions

If the start triggers are None for all except one session, the niTClk Configure For Homogeneous Triggers VI configures the sessions to share the start trigger from the one excepted session. The start trigger is shared by:

- Implicitly exporting start trigger from the session with the start trigger that is not None
- Configuring the other sessions for digital edge start triggers with sources corresponding to the exported start trigger

• Setting the <u>Start Trigger Master Session</u> property to the session that is exporting the trigger for all sessions

If start triggers are configured for all sessions, The niTClk Configure For Homogeneous Triggers VI does not affect the start triggers. Start triggers are considered to be configured for all sessions if either of the following conditions is true:

- No session has a start trigger that is None
- One session has a start trigger that is None, and all other sessions have start triggers other than None. The one session with the None trigger must have the Start Trigger Master Session property set to itself, indicating that the session itself is the start trigger master

Reference Triggers

The niTClk Configure For Homogeneous Triggers VI configures sessions that support reference triggers to share the reference triggers if the reference triggers are None (no trigger configured) for all except one session. The reference triggers are shared by:

- Implicitly exporting the reference trigger from the session whose reference trigger is not None
- Configuring the other sessions that support the reference trigger for digital-edge reference triggers with sources corresponding to the exported reference trigger
- Setting the <u>Reference Trigger Master Session</u> property to the session that is exporting the trigger for all sessions that support reference trigger

If the reference triggers are configured for all sessions that support reference triggers, The niTClk Configure For Homogeneous Triggers VI does not affect the reference triggers. Reference triggers are considered to be configured for all sessions if either one or the other of the following conditions is true:

- No session has a reference trigger that is None
- One session has a reference trigger that is None, and all other sessions have reference triggers other than None. The one session with the None trigger must have the <u>Reference Trigger</u>

<u>Master Session</u> property set to itself, indicating that the session itself is the reference trigger master

Reference Trigger Holdoffs

For acquisition sessions, the niTClk Configure For Homogeneous Triggers VI configures the holdoff attributes (which are instrument driver specific) on the reference trigger master session so that the session does not recognize the reference trigger before the other sessions are ready. This condition is only relevant when the sample clock rates, sample clock timebase rates, sample counts, holdoffs, and/or any delays for the acquisitions are different.

Script Triggers 0, 1, 2, and 3

Note This section applies to each of Script Triggers 0, 1, 2, and 3.

The niTClk Configure For Homogeneous Triggers VI configures sessions that support script triggers to share them, if the script triggers are None (no trigger configured) for all except one session. The script triggers are shared in the following ways:

- Implicitly exporting the script trigger from the session whose script trigger is not None
- Configuring the other sessions that support the script trigger for digital-edge script triggers with sources corresponding to the exported script trigger
- Setting the <u>Script Trigger Master Session</u> property to the session that is exporting the trigger for all sessions that support script triggers

If the script triggers are configured for all sessions that support script triggers, The niTClk Configure For Homogeneous Triggers VI does not affect script triggers. Script triggers are considered to be configured for all sessions if either one or the other of the following conditions are true:

- No session has a script trigger that is None
- One session has a script trigger that is None, and all other sessions have script triggers other than None. The one session with the None trigger must have the <u>Script Trigger Master Session</u> property set to itself, indicating that the session itself is the script trigger master

Pause Triggers

The niTClk Configure For Homogeneous Triggers VI configures generation sessions that support pause triggers to share them, if the pause triggers are None (no trigger configured) for all except one session. The pause triggers are shared by:

- Implicitly exporting the pause trigger from the session whose script trigger is not None
- Configuring the other sessions that support the pause trigger for digital-edge pause triggers with sources corresponding to the exported pause trigger
- Setting the <u>Pause Trigger Master Session</u> property to the session that is exporting the trigger for all sessions that support script triggers

If the pause triggers are configured for all generation sessions that support pause triggers, The niTClk Configure For Homogeneous Triggers VI does not affect pause triggers. Pause triggers are considered to be configured for all sessions if either one or the other of the following conditions is true:

- No session has a pause trigger that is None
- One session has a pause trigger that is None, and all other sessions have pause triggers other than None. The one session with the None trigger must have the <u>Pause Trigger Master Session</u> property set to itself, indicating that the session itself is the pause trigger master



Note TClk synchronization is not supported for pause triggers on acquisition sessions.

niTClk Initiate

Initiates the acquisition and/or generation sessions specified, taking into consideration any special requirements needed for synchronization.

For example, the session exporting the TClk-synchronized start trigger is initiated after all the sessions that import the TClk-synchronized start trigger.



error and often offers additional information about the error.

The LabVIEW Simple Error Handler VI and the LabVIEW General Error Handler VI use the **source** string to deliver messages that may help you to troubleshoot your application.

niTClk Is Done

Monitors the progress of the acquisitions and/or generations corresponding to sessions.



sessions is an array of session references that are being synchronized.

You obtain session references from the instrument driver by using *<driver name>* Get Session Reference VI, where *<driver name>* is the name of the instrument driver—for example, the niScope Get Session Reference VI.

- error in describes error conditions that occur before this VI or function runs.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE. If **status** is TRUE, this VI propagates the error and does not do anything else.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The **source** string includes the name of the VI that produced the error and often offers additional information about the error.



- **biline done** indicates that the operation is done. The operation is done when each session has completed without any errors or when any one of the sessions reports an error.
- **error out** contains error information. If **error in** indicates that an error occurred before this VI or function ran, **error out** contains the same error information. Otherwise, it describes the error status that this VI or function produces.
 - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
 - **source** identifies where and why an error occurred. The **source** string includes the name of the VI that produced the error and often offers additional information about the error.

The LabVIEW Simple Error Handler VI and the LabVIEW General Error Handler VI use the **source** string to deliver messages that may help you to troubleshoot your application.

niTClk Synchronize

Synchronizes the TClk signals on the given sessions. After the niTClk Synchronize VI executes, <u>TClk signals</u> from all sessions are synchronized.



Note Before using this NI-TClk VI, verify that your system is configured as specified in <u>PXI</u> <u>Trigger Lines and RTSI Lines</u>.

sessions	NI-TCIK sessions out
minimum T clock period – error in (no error) –	error out



sessions is an array of session references that are being synchronized.

You obtain session references from the instrument driver by using *<driver name>* Get Session Reference VI, where *<driver name>* is the name of the instrument driver—for example, the niScope Get Session Reference VI.

minimum T clock period is the minimal period of TClk, expressed in seconds. Supported values are between 0.0 and 0.050 s (50 ms). Minimal period for a single chassis/PC is 200 ns. If the specified value is less than 200 ns, NI-TClk automatically coerces **minimum T clock period** to 200 ns. For multichassis synchronization, adjust this value to account for propagation delays through the various devices and cables.

error in describes error conditions that occur before this VI or function runs.

- **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE. If **status** is TRUE, this VI propagates the error and does not do anything else.
- **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.

source identifies where an error occurred. The **source** string includes the name of the VI that produced the error and often offers additional information about the error.

sessions out passes the array of session references to the next VI.

error out contains error information. If **error in** indicates that an error occurred before this VI or function ran, **error out** contains the same error information. Otherwise, it describes the error status that this VI or function produces.

- **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
- **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a

warning code.

abc

source identifies where and why an error occurred. The source string includes the name of the VI that produced the error and often offers additional information about the error.

The LabVIEW Simple Error Handler VI and the LabVIEW General Error Handler VI use the **source** string to deliver messages that may help you to troubleshoot your application.

niTClk Wait Until Done

Call this VI to pause execution of your program until the acquisitions and/or generations corresponding to sessions are done or until the function returns a timeout error.

niTClk Wait Until Done is a blocking VI that periodically checks the operation status. It returns control to the calling program if the operation completes successfully or an error occurs (including a timeout error).

This VI is most useful for finite data operations that you expect to complete within a certain time.



sessions is an array of session references that are being synchronized.

You obtain session references from the instrument driver by using *<driver name>* Get Session Reference VI, where *<driver name>* is the name of the instrument driver—for example, the niScope Get Session Reference VI.

- **timeout** is the amount of time in seconds that the niTClk Wait Until Done VI waits for the sessions to complete. If **timeout** is exceeded, the niTClk Wait Until Done VI returns an error.
- error in describes error conditions that occur before this VI or function runs.
 - **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE. If **status** is TRUE, this VI propagates the error and does not do anything else.
 - **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** identifies where an error occurred. The **source** string includes the name of the VI that produced the error and often offers additional information about the error.



2.5

error out contains error information. If **error in** indicates that an error occurred before this VI or function ran, **error out** contains the same error information. Otherwise, it describes the error status that this VI or function produces.

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.

source identifies where and why an error occurred. The **source** string includes the name of the VI that produced the error and often offers additional information about the error.

The LabVIEW Simple Error Handler VI and the LabVIEW General Error Handler VI use the **source** string to deliver messages that may help you to troubleshoot your application.

niTClk Property Node

A LabVIEW Property Node with the NI-TClk class preselected. Gets (reads) and/or sets (writes) properties of a session reference.



session reference refers to the instrument session.

session reference is obtained from instrument driver by using the *<driver name>* Get Session Reference VI, where *<driver name>* is the instrument driver name—for example, the niScope Get Session Reference VI.

error in describes error conditions that occur before this property node runs.

The default is no error. If an error occurred before this property node runs, the property node passes the **error in** value to **error out**. If an error occurs while this property node runs, it runs normally and sets its own error status in **error out**. Use the LabVIEW Simple Error Handler or General Error Handler VIs to display the description of the error code. Use **error in** and **error out** to check errors and to specify execution order by wiring **error out** from one node to **error in** of the next node.

- **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE. If **status** is TRUE, this VI propagates the error and does not do anything else.
- **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a negative error code. If **status** is FALSE, **code** is 0 or a warning code.
- **source** identifies where an error occurred. The **source** string includes the name of the VI that produced the error and often offers additional information about the error.

session reference returns session reference input unchanged.

error out contains error information. If **error in** indicates that an error occurred before this VI or function ran, **error out** contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the **error out** indicator on the front panel and select **Explain Error** from the shortcut menu for more information about the error.

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

- **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.
- **source** identifies where and why an error occurred. The

source string includes the name of the VI that produced the error and often offers additional information about the error.

The LabVIEW Simple Error Handler VI and the LabVIEW General Error Handler VI use the **source** string to deliver messages that may help you to troubleshoot your application.

NI-TClk Active Channels Property

Short Name: ActiveChans

Use this property in conjunction with the <u>Script Trigger Master Session</u> property to specify which script trigger (scriptTrigger0, scriptTrigger1, scriptTrigger2 scriptTrigger3) the Script Trigger Master Session applies to.

To specify Script Trigger Master Session for more than one script trigger, specify the Active Channel and the Script Trigger Master Session properties multiple times. The following example shows how to specify Script Trigger Master Session for scriptTrigger0 and scriptTrigger1.



The property is named Active Channels for consistency with similar properties in NI-SCOPE, NI-HSDIO, and NI-FGEN.

The following table lists the characteristics of this property.

Data TypeViStringPermissionsWrite Only

NI-TClk Export Sync Pulse Output Terminal Property

Short Name: ExportSyncPulseOutputTerm

Specifies the destination of the <u>Sync Pulse</u>. This property is most often used when synchronizing a <u>multichassis system</u>.

Values

Empty string. Empty string is a valid value, indicating that the signal is not exported.

PXI Devices

"PXI_Trig0" through "PXI_Trig7", PXI_STAR (Slots 3 through 15), and device-specific settings

PCI Devices

"RTSI_0" through "RTSI_7" and device-specific settings

Examples of Device-Specific Settings

- NI PXI-5122 supports "PFI0" and "PFI1"
- NI PXI-5421 supports "PFI0", "PFI1", "PFI4", and "PFI5"
- NI PXI-6551/6552 supports "PFI0", "PFI1", "PFI2", and "PFI3"

Default Value

Empty string

The following table lists the characteristics of this property.

Data Type	ViString
Permissions	R/W
Corresponding Active Channels	N/A

NI-TClk Pause Trigger Master Session Property

Short Name: PauseTrigMasterSession

Specifies the pause trigger master session.

For external triggers, the session that originally receives the trigger. For None (no trigger configured) or software triggers, the session that originally generates the trigger.

The following table lists the characteristics of this property.

Permissions R/W	Data Type	session reference
Openne and the station Observation NI/A	Permissions	R/W
Corresponding <u>Active Channels</u> N/A	Corresponding Active Channels	N/A

NI-TClk Reference Trigger Master Session Property

Short Name: RefTrigMasterSession

Specifies the reference trigger master session.

For external triggers, the session that originally receives the trigger. For None (no trigger configured) or software triggers, the session that originally generates the trigger.

The following table lists the characteristics of this property.

Permissions R/W	Data Type	session reference
Openne and the station Observation NI/A	Permissions	R/W
Corresponding <u>Active Channels</u> N/A	Corresponding Active Channels	N/A

NI-TCIk Sample Clock Delay Property

Short Name: SampleClkDelay

Specifies the delay, in seconds, to apply to the session sample clock relative to the other synchronized sessions. During synchronization, NI-TClk aligns the sample clocks on the synchronized devices. If you want to delay the sample clocks, set this property before calling the <u>niTClk</u> <u>Synchronize</u> VI.

The values for the Sample Clock Delay property range between minus one and plus one period of the sample clock.

One sample clock period is equal to (1/sample clock rate). For example, for a session with sample rate of 100 MS/s, you can specify sample clock delays between -10.0 ns and +10.0 ns.

The default value is 0.

The following table lists the characteristics of this property.

Data Type	ViReal64
Permissions	R/W
Corresponding Active Channels	N/A

NI-TClk Script Trigger Master Session Property

Short Name: ScriptTrigMasterSession

Specifies the script trigger master session.

For external triggers, the session that originally receives the trigger. For None (no trigger configured) or software triggers, the session that originally generates the trigger.

The following table lists the characteristics of this property.

Data Type	session reference
Permissions	R/W
Corresponding Active Channels	scriptTrigger0, scriptTrigger1, scriptTrigger2, and scriptTrigger3

If the Active Channels property is not specified or is set to an empty string, scriptTrigger0 is assumed for backward compatibility.

NI-TClk Start Trigger Master Session Property

Short Name: StartTrigMasterSession

Specifies the start trigger master session.

For external triggers, the session that originally receives the trigger. For None (no trigger configured) or software triggers, the session that originally generates the trigger.

The following table lists the characteristics of this property.

Permissions R/W	Data Type	session reference
Openne and the station Observation NI/A	Permissions	R/W
Corresponding <u>Active Channels</u> N/A	Corresponding Active Channels	N/A
NI-TClk Sync Pulse Clock Source Property

Short Name: SyncPulseClkSrc

Specifies the <u>Sync Pulse Clock</u> source. This property is typically used to synchronize PCI devices when you want to control RTSI 7 yourself. Make sure that a 10 MHz clock is driven onto RTSI 7.

Values

PCI Devices

"RTSI_7" "None"

PXI Devices

"PXI_CLK10" "None"

Default Value

"None" directs the <u>niTClk Synchronize</u> to create the necessary routes. For PCI, one of the synchronized devices drives a 10 MHz clock on RTSI 7 unless that line is already being driven.

Remarks

The following table lists the characteristics of this property.

Data Type	ViString
Permissions	R/W
Corresponding Active Channels	N/A

NI-TCIk Sync Pulse Source Property

Short Name: SyncPulseClk

Specifies the <u>Sync Pulse</u> source. This property is most often used when synchronizing a <u>multichassis system</u>.

Values

Empty string.

PXI Devices

 $"PXI_Trig0"$ through "PXI_Trig7" and device-specific settings

PCI Devices

"RTSI_0" through "RTSI_7" and device-specific settings

Examples of Device-Specific Settings

- NI PXI-5122 supports "PFI0" and "PFI1"
- NI PXI-5421 supports "PFI0", "PFI1", "PFI4", and "PFI5"
- NI PXI-6551/6552 supports "PFI0", "PFI1", "PFI2", and "PFI3"

Default Value

The default value (empty string) directs the <u>niTClk Synchronize</u> VI to set this property when all the synchronized devices are in one PXI chassis or in one PC. To synchronize a <u>multichassis system</u>, you must set this property before calling the niTClk Synchronize VI.

Remarks

The following table lists the characteristics of this property.

Data Type	ViString
Permissions	R/W
Corresponding Active Channels	N/A

NI-TClk C Function Reference Help

The following table summarizes the functions and attributes suitable for different chassis and PC configurations.

	PXI Chassis/PC Configuration		
	Single PXI Chassis/PC, Homogeneous Triggers	Single PXI Chassis/PC, Heterogeneous Triggers	Mı Ch
Functions	<u>.</u>		
niTClk ConfigureForHomogeneousTriggers	Yes	—	
niTClk Synchronize	Yes	Yes	
niTClk_Initiate	Yes	Yes	
niTClk IsDone	Yes	Yes	
niTClk WaitUntilDone	Yes	Yes	
niTClk GetExtendedErrorInfo	Yes	Yes	
Attributes	•	•	-
NITCLK ATTR START TRIGGER MASTER SESSION	—	Т	
NITCLK ATTR REF TRIGGER MASTER SESSION		Т	
NITCLK ATTR SCRIPT TRIGGER MASTER SESSION		Т	
NITCLK ATTR PAUSE TRIGGER MASTER SESSION		Т	
NITCLK ATTR SYNC PULSE SOURCE		—	
NITCLK ATTR EXPORTED SYNC PULSE OUTPUT TERMINAL		—	
NITCLK ATTR SYNC PULSE CLOCK SOURCE	R	R	
NITCLK ATTR SAMPLE CLOCK DELAY	Optional	Optional	Ot
R: Devices with RTSI trigger bus only; used when you want to cont	rol RTSI 7 direct	ly instead of throu	gh I

T: Use attributes that pertain to the triggers that you are synchronizing

NI-TClk Functions

Class/Panel Name	Function Name
Configure For Homogeneous Triggers	niTClk ConfigureForHomogeneousTriggers
Synchronize	niTClk Synchronize
Initiate	niTClk Initiate
Wait Until Done	niTClk WaitUntilDone
Get Extended Error Info	niTClk GetExtendedErrorInfo
Advanced	
Is Done	niTClk IsDone
Set Attribute	
Set Attribute ViReal64	niTClk SetAttributeViReal64
Set Attribute ViSession	niTClk SetAttributeViSession
Set Attribute ViString	niTClk SetAttributeViString
Get Attribute	
Get Attribute ViReal64	niTClk GetAttributeViReal64
Get Attribute ViSession	niTClk GetAttributeViSession
Get Attribute ViString	niTClk GetAttributeViString

NI-TClk High-Level Functions

Configure For Homogeneous Triggers <u>niTClk ConfigureForHomogeneousTriggers</u>

Synchronize

Initiate

Wait Until Done

Get Extended Error Info

niTClk ConfigureForHomoger niTClk Synchronize niTClk Initiate niTClk WaitUntilDone niTClk GetExtendedErrorInfo

niTClk_ConfigureForHomogeneousTriggers

C Function Prototype

ViStatus niTClk_ConfigureForHomogeneousTriggers (ViUInt32 sessionCount, ViSession sessions []);

Purpose

Configures the attributes commonly required for the TClk synchronization of device sessions with <u>homogeneous triggers</u> in a <u>single PXI chassis or</u> <u>a single PC</u>.

Use niTClk_ConfigureForHomogeneousTriggers to configure the attributes for the reference clocks, start triggers, reference triggers, script triggers, and pause triggers.

If niTClk_ConfigureForHomogeneousTriggers cannot perform all the steps appropriate for the given sessions, it returns an error. If an error is returned, use the instrument driver functions and attributes for signal routing, along with the following NI-TClk attributes:

- <u>NITCLK_ATTR_START_TRIGGER_MASTER_SESSION</u>
- <u>NITCLK_ATTR_REF_TRIGGER_MASTER_SESSION</u>
- <u>NITCLK_ATTR_SCRIPT_TRIGGER_MASTER_SESSION</u>
- <u>NITCLK_ATTR_PAUSE_TRIGGER_MASTER_SESSION</u>

niTClk_ConfigureForHomogeneousTriggers affects the following clocks and triggers:

- Reference Clocks
- Start Triggers
- Reference Triggers
- Script Triggers
- Pause Triggers

Parameters

NameTypeDescriptionsessionCountViUInt32Number of elements in the sessions array.sessionsViSession []sessions is an array of sessions that are being synchronized.

Return Value

niTClk_Synchronize

C Function Prototype

ViStatus niTClk_Synchronize (ViUInt32 sessionCount, ViSession sessions [], ViReal64 minTime);

Purpose

Synchronizes the <u>TClk signals</u> on the given sessions. After niTClk_Synchronize executes, TClk signals from all sessions are synchronized.



Note Before using this NI-TClk function, verify that your system is configured as specified in <u>PXI Trigger Lines and RTSI Lines</u>.

Parameters

Name	Туре	Description
sessionCount	ViUInt32	Number of elements in the sessions array.
sessions	ViSession []	sessions is an array of sessions that are being synchronized.
minTime	ViReal64	Minimal period of TClk, expressed in seconds. Supported values are between 0.0 s and 0.050 s (50 ms). Minimal period for a single chassis/PC is 200 ns. If the specified value is less than 200 ns, NI-TClk automatically coerces minTime to 200 ns. For multichassis synchronization, adjust this value to account for propagation delays through the various devices and cables.

Return Value

niTClk_Initiate

Specific Function

C Function Prototype

ViStatus niTClk_Initiate (ViUInt32 sessionCount, ViSession sessions []);

Purpose

Initiates the acquisition or generation sessions specified, taking into consideration any special requirements needed for synchronization.

For example, the session exporting the TClk-synchronized start trigger is not initiated until after niTClk_Initiate initiates all the sessions that import the TClk-synchronized start trigger.

Parameters

NameTypeDescriptionsessionCountViUInt32Number of elements in the sessions array.sessionsViSession []sessions is an array of sessions that are being synchronized.

Return Value

niTClk_WaitUntilDone

C Function Prototype

ViStatus niTClk_WaitUntilDone (ViUInt32 sessionCount, ViSession sessions [], ViReal64 timeout);

Purpose

Call this function to pause execution of your program until the acquisitions and/or generations corresponding to sessions are done or until the function returns a timeout error.

niTClk_WaitUntilDone is a blocking function that periodically checks the operation status. It returns control to the calling program if the operation completes successfully or an error occurs (including a timeout error).

This function is most useful for finite data operations that you expect to complete within a certain time.

Parameters

Name	Туре	Description
sessionCount	ViUInt32	Number of elements in the sessions array.
sessions	ViSession []	sessions is an array of sessions that are being synchronized.
timeout	ViReal64	The amount of time in seconds that $niTClk_WaitUntilDone$ waits for the sessions to complete. If timeout is exceeded, $niTClk_WaitUntilDone$ returns an error.

Return Value

niTClk_GetExtendedErrorInfo

Specific Function

C Function Prototype

ViStatus niTClk_GetExtendedErrorInfo (ViChar errorString [], ViUInt32 errorStringSize);

Purpose

Reports extended error information for the most recent NI-TClk function that returned an error.

To establish the function that returned an error, use the return values of the individual functions because once niTClk_GetExtendedErrorInfo reports an **errorString**, it does not report an empty string again.

Parameters

Name	Туре	Description
errorString	ViChar []	Extended error description. If errorString is NULL, then it is not large enough to hold the entire error description. In this case, the return value of niTClk_GetExtendedErrorInfo is the size that you should use for niTClk_GetExtendedErrorInfo to return the full error string.
errorStringSize	ViUInt32	Size of the errorString . If errorStringSize is 0, then it is not large enough to hold the entire error description. In this case, the return value of niTClk_GetExtendedErrorInfo is the size that you should use for niTClk_GetExtendedErrorInfo to return the full error string.

Return Value

NI-TClk Advanced Function

Is Done <u>niTClk_IsDone</u>

niTClk_IsDone

Specific Function

C Function Prototype

ViStatus niTClk_IsDone (ViUInt32 sessionCount, ViSession sessions [], ViBoolean *done);

Purpose

Monitors the progress of the acquisitions and/or generations corresponding to sessions.

Parameters

Name	Туре	Description
sessionCount	ViUInt32	Number of elements in the sessions array.
sessions	ViSession []	sessions is an array of sessions that are being synchronized.
done	ViBoolean	Indicates that the operation is done. The operation is done when each session has completed without any errors or when any one of the sessions reports an error.

Return Value

NI-TCIk Get Attribute Functions

Get Attribute ViReal64 <u>niTClk_GetAttributeViReal64</u> Get Attribute ViSession niTClk GetAttributeViSession Get Attribute ViString <u>niTClk_GetAttributeViString</u>

niTClk_GetAttributeViReal64

C Function Prototype

ViStatus niTClk_GetAttributeViReal64 (ViSession session, ViConstString channelName, ViAttr attributeId, ViReal64 *value);
Purpose

Gets the value of an NI-TClk ViReal64 attribute.

Parameters

Name	Туре	Description
session	ViSession	session references the sessions being synchronized.
channelName	ViConstString	Pass VI_NULL or an empty string.
attributeId	ViAttr	The ID of the attribute that you want to get.
		Supported Attribute
		NITCLK ATTR SAMPLE CLOCK DELAY
value	ViReal64	The value that you are getting.

Return Value

niTClk_GetAttributeViSession

C Function Prototype

ViStatus niTClk_GetAttributeViSession (ViSession session, ViConstString channelName, ViAttr attributeId, ViSession *value);

Purpose

Gets the value of an NI-TClk ViSession attribute.

Parameters

Name	Туре	Description
session	ViSession	session references the sessions being synchronized.
channelName	ViConstString	Pass VI_NULL or an empty string, except for <u>NITCLK ATTR SCRIPT TRIGGER MASTER SESSION</u> , for which you should specify scriptTrigger0, scriptTrigger1, scriptTrigger2, or scriptTrigger3. VI_NULL and the empty string are treated as scriptTrigger0 for NITCLK_ATTR_SCRIPT_TRIGGER_MASTER_SESSION.
attributeld	ViAttr	The ID of the attribute that you want to get.
		Supported Attributes
		NITCLK ATTR START TRIGGER MASTER SESSION NITCLK ATTR REF TRIGGER MASTER SESSION NITCLK ATTR SCRIPT TRIGGER MASTER SESSION NITCLK ATTR PAUSE TRIGGER MASTER SESSION
value	ViSession	The value that you are getting.

Return Value

niTClk_GetAttributeViString

C Function Prototype

ViStatus niTClk_GetAttributeViString (ViSession session, ViConstString channelName, ViAttr attributeId, ViInt32 bufSize, ViChar value []);

Purpose

This function queries the value of an NI-TClk ViString attribute.

You must provide a ViChar array to serve as a buffer for the value. You pass the number of bytes in the buffer as **bufSize**. If the current value of the attribute, including the terminating NULL byte, is larger than the size you indicate in **bufSize**, the function copies **bufSize** minus 1 bytes into the buffer, places an ASCII NULL byte at the end of the buffer, and returns the array size that you must pass to get the entire value. For example, if the value is "123456" and **bufSize** is 4, the function places "123" into the buffer and returns 7.

If you want to call niTClk_GetAttributeViString just to get the required array size, pass 0 for **bufSize** and VI_NULL for the **value**.

Parameters

Name	Туре	Description
session	ViSession	session references the sessions being synchronized.
channelName	ViConstString	Pass VI_NULL or an empty string.
attributeld	ViAttr	The ID of the attribute that you want to get.
		Supported Attributes
		NITCLK ATTR SYNC PULSE SOURCE NITCLK ATTR SYNC PULSE CLOCK SOURCE NITCLK ATTR EXPORTED SYNC PULSE OUTPUT TERMINAL
bufSize	Vilnt32	The number of bytes in the ViChar array that you specify for the value parameter.
value	ViChar []	The value that you are getting.

Return Value

The status code returned by the function.

- A value of 0 indicates success.
- A negative value indicates an error.
- A value greater than 0 indicates a warning.

Use <u>niTClk_GetExtendedErrorInfo</u> to get detailed information about individual errors and warnings.

If you pass NULL for the value or 0 for the buffer size, or if the size of value prevents the function from reporting the value in its entirety, this function returns the number of characters needed to report the value.

NI-TCIk Set Attribute Functions

Set Attribute ViString

Set Attribute ViReal64 <u>niTClk_SetAttributeViReal64</u> Set Attribute ViSession <u>niTClk_SetAttributeViSession</u> <u>niTClk_SetAttributeViString</u>

niTClk_SetAttributeViReal64

C Function Prototype

ViStatus niTClk_SetAttributeViReal64 (ViSession session, ViConstString channelName, ViAttr attributeId, ViReal64 value);

Purpose

Sets the value of an NI-TClk VIReal64 attribute.

niTClk_SetAttributeViReal64 is a low-level function that you can use to set the values of NI-TClk attributes.

NI-TClk contains high-level functions that set most of the attributes. It is best to use the high-level functions as much as possible.

Parameters

Name	Туре	Description
session	ViSession	session references the sessions being synchronized.
channelName	ViConstString	Pass VI_NULL or an empty string.
attributeId	ViAttr	The ID of the attribute that you want to set.
		Supported Attribute
		NITCLK ATTR SAMPLE CLOCK DELAY
value	ViReal64	The value for the attribute.

Return Value

niTClk_SetAttributeViSession

C Function Prototype

ViStatus niTClk_SetAttributeViSession (ViSession session, ViConstString channelName, ViAttr attributeId, ViSession value);

Purpose

Sets the value of an NI-TClk ViSession attribute.

niTClk_SetAttributeViSession is a low-level function that you can use to set the values of NI-TClk attributes.

NI-TClk contains high-level functions that set most of the attributes. It is best to use the high-level functions as much as possible.

Parameters

Name	Туре	Description
session	ViSession	session references the sessions being synchronized.
channelName	ViConstString	Pass VI_NULL or an empty string, except for <u>NITCLK ATTR SCRIPT TRIGGER MASTER SESSION</u> , for which you should specify scriptTrigger0, scriptTrigger1, scriptTrigger2, or scriptTrigger3. VI_NULL and the empty string are treated as scriptTrigger0 for NITCLK_ATTR_SCRIPT_TRIGGER_MASTER_SESSION.
attributeId	ViAttr	The ID of the attribute that you want to set.
		Supported Attributes
		NITCLK ATTR START TRIGGER MASTER SESSION NITCLK ATTR REF TRIGGER MASTER SESSION NITCLK ATTR SCRIPT TRIGGER MASTER SESSION NITCLK ATTR PAUSE TRIGGER MASTER SESSION
value	ViSession	The value for the attribute.

Return Value

niTClk_SetAttributeViString

C Function Prototype

ViStatus niTClk_SetAttributeViString (ViSession session, ViConstString channelName, ViAttr attributeId, ViConstString value);

Purpose

Sets the value of an NI-TClk VIString attribute.

niTClk_SetAttributeViString is a low-level function that you can use to set the values of NI-TClk attributes.

NI-TClk contains high-level functions that set most of the attributes. It is best to use the high-level functions as much as possible.

Parameters

Name	Туре	Description
session	ViSession	session references the sessions being synchronized.
channelName	ViConstString	Pass VI_NULL or an empty string.
attributeld	ViAttr	Pass the ID of the attribute that you want to set.
		Supported Attributes
		NITCLK ATTR SYNC PULSE SOURCE NITCLK ATTR SYNC PULSE CLOCK SOURCE NITCLK ATTR EXPORTED SYNC PULSE OUTPUT TERMINAL
value	ViConstString	Pass the value for the attribute.

Return Value

NI-TClk Functions Listed Alphabetically

niTClk_ConfigureForHomogeneousTriggers
niTClk_Get Attribute ViReal64
niTClk_Get Attribute ViSession
niTClk_Get Attribute ViString
niTClk_Get Extended Error Info
niTClk_IsDone
niTClk_IsDone
niTClk_SetAttributeViReal64
niTClk_SetAttributeViSession
niTClk_SetAttributeViString
niTClk_Synchronize
niTClk_WaitUntilDone

NI-TClk Attributes Listed Alphabetically

Exported Sync Pulse Output Terminal NITCLK ATTR PAUSE TRIGGER MASTER SESSION Pause Trigger Master Session Reference Trigger Master Session NITCLK ATTR REF TRIGGER MASTER SESSION NITCLK ATTR SAMPLE CLOCK DELAY Sample Clock Delay Script Trigger Master Session NITCLK ATTR SCRIPT TRIGGER MASTER SESSION Start Trigger Master Session NITCLK ATTR START_TRIGGER_MASTER_SESSION Sync Pulse Clock Source NITCLK ATTR SYNC PULSE CLOCK SOURCE NITCLK ATTR SYNC PULSE SOURCE Sync Pulse Source

NITCLK ATTR EXPORTED SYNC PULSE OUTPUT TERMINAL

NITCLK_ATTR_EXPORTED_SYNC_PULSE_OUTF

Data Type Access Coercion High-Level Functions

ViString R/W None

niTClk Synchronize

Description

Specifies the destination of the <u>Sync Pulse</u>. This attribute is most often used when synchronizing a <u>multichassis system</u>.

Values

Empty string. Empty string is a valid value, indicating that the signal is not exported.

PXI Devices

"PXI_Trig0" through "PXI_Trig7" and device-specific settings

PCI Devices

"RTSI_0" through "RTSI_7" and device-specific settings

Examples of Device-Specific Settings

- NI PXI-5122 supports "PFI0" and "PFI1"
- NI PXI-5421 supports "PFI0", "PFI1", "PFI4", and "PFI5"
- NI PXI-6551/6552 supports "PFI0", "PFI1", "PFI2", and "PFI3"

Default Value

Empty string

NITCLK_ATTR_PAUSE_TRIGGER_MASTER_SES

Data Type Access Coercion High-Level Functions

ViSession R/W None <u>niTClk ConfigureForHomogeneousTriggers</u>

Description

Specifies the pause trigger master session.

For external triggers, the session that originally receives the trigger. For None (no trigger configured) or software triggers, the session that originally generates the trigger.

NITCLK_ATTR_REF_TRIGGER_MASTER_SESSIC

Data Type Access Coercion High-Level Functions

ViSession R/W None <u>niTClk ConfigureForHomogeneousTriggers</u>

Description

Specifies the reference trigger master session.

For external triggers, the session that originally receives the trigger. For None (no trigger configured) or software triggers, the session that originally generates the trigger.

NITCLK_ATTR_SAMPLE_CLOCK_DELAY

Data Type Access Coercion High-Level Functions

ViReal64 R/W None None

Description

Specifies the delay, in seconds, to apply to the session sample clock relative to the other synchronized sessions. During synchronization, NI-TClk aligns the sample clocks on the synchronized devices. If you want to delay the sample clocks, set this attribute before calling <u>niTClk_Synchronize</u>.

Values

Between minus one and plus one period of the sample clock.

One sample clock period is equal to (1/sample clock rate). For example, for a session with sample rate of 100 MS/s, you can specify sample clock delays between -10.0 ns and +10.0 ns.

Default Value

0
NITCLK_ATTR_SCRIPT_TRIGGER_MASTER_SES

Data Type Access Coercion High-Level Functions

ViSession R/W None <u>niTClk ConfigureForHomogeneousTriggers</u>

Specifies the script trigger master session.

For external triggers, the session that originally receives the trigger. For None (no trigger configured) or software triggers, the session that originally generates the trigger.

NITCLK_ATTR_START_TRIGGER_MASTER_SES

Data Type Access Coercion High-Level Functions

ViSession R/W None <u>niTClk ConfigureForHomogeneousTriggers</u>

Specifies the start trigger master session.

For external triggers, the session that originally receives the trigger. For None (no trigger configured) or software triggers, the session that originally generates the trigger.

NITCLK_ATTR_SYNC_PULSE_CLOCK_SOURCE

Data Type Access Coercion High-Level Functions

ViString R/W None <u>niTClk ConfigureForHomogeneousTriggers</u>

Specifies the <u>Sync Pulse Clock</u> source. This attribute is typically used to synchronize PCI devices when you want to control RTSI 7 yourself. Make sure that a 10 MHz clock is driven onto RTSI 7.

Values

PCI Devices

"RTSI_7" "None"

PXI Devices

"PXI_CLK10" "None"

Default Value

"None" directs <u>niTClk_Synchronize</u> to create the necessary routes. For PCI, one of the synchronized devices drives a 10 MHz clock on RTSI 7 unless that line is already being driven.

NITCLK_ATTR_SYNC_PULSE_SOURCE

Data Type Access Coercion High-Level Functions

ViString R/W None

niTClk Synchronize

Specifies the <u>Sync Pulse</u> source. This attribute is most often used when synchronizing a <u>multichassis system</u>.

Values

Empty string.

PXI Devices

 $"PXI_Trig0"$ through "PXI_Trig7" and device-specific settings

PCI Devices

"RTSI_0" through "RTSI_7" and device-specific settings

Examples of Device-Specific Settings

- NI PXI-5122 supports "PFI0" and "PFI1"
- NI PXI-5421 supports "PFI0", "PFI1", "PFI2", and "PFI3"
- NI PXI-6551/6552 supports "PFI0", "PFI1", "PFI2", and "PFI3"

Default Value

Empty string. This default value directs <u>niTClk_Synchronize</u> to set this attribute when all the synchronized devices are in one PXI chassis. To synchronize a <u>multichassis system</u>, you must set this attribute before calling <u>niTClk_Synchronize</u>.

NI-TClk Return Value

The status code returned by the function.

- A value of 0 indicates success.
- A negative value indicates an error.
- A value greater than 0 indicates a warning.

Use <u>niTClk_GetExtendedErrorInfo</u> to get detailed information about individual errors and warnings.