
SystemLink API Reference

2025-07-22



Contents

API Reference	11
SystemLink Nodes and Functions Reference for NXG and G Web Development	
Software	13
Alarm Functions	14
Acknowledge Alarm	15
Multiple	16
Single	18
Add Alarm Notes	19
Multiple	20
Single	21
Build Alarm ID	23
Build Alarm Set Transition	23
Multiple Notifications	24
Single Notification	25
Clear Alarm	27
Delete Alarm	29
Create Alarm Query Filter	30
Get Alarm Information	35
Get Notification Strategies	36
Query Alarms	37
No Transitions	38
With Transitions	40
Set Alarm	43
Asset Utilization Functions	46
End Utilization	47
Start Utilization	48
All	48
Multiple	50
Single	52
Configuration Functions	53
Close Configuration	54
Open Configuration	55

User	56
User (Desktop).....	58
API Key	61
API Key (Desktop).....	63
Auto (Desktop)	65
Auto	66
Use Workspace	67
Name.....	68
Instance	69
List Workspaces	71
File Transfer Functions	72
Create Filter	73
Extension	74
Name.....	75
Property Value.....	77
Size in Bytes.....	79
Created Time Stamp	80
Delete	82
Single	82
Multiple.....	83
Download.....	84
Multiple.....	85
Single	86
Get File Metadata	88
Get Property	88
Merge Metadata Properties	90
Open File	91
Query Files.....	93
Send	94
Buffer.....	95
Path	97
Message Functions.....	99
Close	100
Flush	101
Get Max Queue Size Bytes	102
Get Queue Size Bytes.....	103

Open	104
Publish	105
Read	107
Subscribe	109
Unsubscribe	110
Tag Functions	111
Advanced Functions	113
Generate Tag Cluster	113
Boolean.....	114
DBL	116
I32	118
String.....	120
Timestamp.....	122
U64.....	123
Multi Read	125
Multi Write	126
Build Path.....	127
Close Tag	128
Multiple.....	129
Single	130
Delete Tag	131
Multiple.....	131
Single	132
Historian Functions	133
Configure Tag History Window	133
Count.....	134
Decimated	135
Configure Retention.....	137
Count.....	137
Duration	139
None	140
Permanent.....	141
Multi Read History	142
Read Tag History.....	144
Boolean.....	145
DBL	147

I32	148
String.....	150
Timestamp.....	152
U64.....	154
Open Tag	156
Query Tags.....	158
Read Tag.....	160
Boolean	161
DBL	162
I32	164
String.....	166
Timestamp.....	167
U64.....	169
Read Tag Settings.....	171
Reset Aggregates	172
Multiple.....	173
Single	174
Write Tag.....	175
Boolean	176
DBL	177
I32	178
String.....	180
Timestamp.....	181
U64.....	183
Write Tag Settings.....	184
TDM Reader Functions	186
Close File	187
Multiple.....	187
Single	188
Create Filter.....	189
Name.....	190
Property	192
Size in Bytes.....	193
Created Time Stamp	195
Get TDM File Metadata	197
List File Contents	197

GroupsAndChannels	198
Channels.....	199
Open File	201
Query TDMS Files	202
Read Data.....	204
1Chan Boolean	206
NChan Boolean.....	209
1Chan Double	212
NChan Double.....	214
1Chan I16	218
NChan I16.....	220
1Chan I32	224
NChan I32.....	226
1Chan I8	230
NChan I8.....	232
1Chan Single	236
NChan Single.....	239
1Chan String	242
NChan String.....	245
1Chan Timestamp	248
NChan Timestamp	251
1Chan U16	254
NChan U16.....	256
1Chan U32	260
NChan U32.....	262
1Chan U8	266
NChan U8.....	268
1Chan Waveform	272
NChan Waveform	274
Utility Functions.....	278
Get Minion ID.....	278
Get System Name.....	279
SystemLink VIs and Functions Reference for LabVIEW	279
Configuration Functions.....	280
Close Configuration	281
Open Configuration	282

Open Configuration (HTTP Auto)	282
Open Configuration (HTTP)	283
Open Configuration (API Key)	285
Use Workspace	287
Use Workspace (Name)	287
Use Workspace (Instance)	288
List Workspaces	289
Tag Functions	290
Build Path	291
Open Tag	292
Query Tags	293
Close Tag	295
Read Tag	296
Read Tag (DBL)	297
Read Tag (I32)	298
Read Tag (U64)	299
Read Tag (String)	300
Read Tag (Boolean)	302
Read Tag (Timestamp)	303
Write Tag	304
Reset Aggregates	305
Reset Aggregates (Multiple)	306
Reset Aggregates (Single)	307
Delete Tag	307
Delete Tag (Multiple)	308
Delete Tag (Single)	308
Tag Properties	309
Advanced Tag Functions	310
Generate Tag Cluster	310
Multi Write	312
Multi Read	312
Tag Historian Functions	313
Configure Window	314
Configure Window (Undecimated)	314
Configure Window (Decimated)	315
Read Tag History	316

Multi Read History	318
Configure Retention.....	319
Configure Retention (Duration).....	320
Configure Retention (Count)	321
Configure Retention (None).....	321
Configure Retention (Permanent)	322
Messages Functions.....	323
Open	324
Open (HTTP)	324
Close	325
Publish	326
Read	327
Subscribe	329
Unsubscribe	330
Flush	331
Message Properties	331
File Transfer Functions	332
Send	334
File (HTTP)	334
Buffer (HTTP).....	335
Create Filter.....	337
Create Filter (Extension)	337
Create Filter (Name).....	338
Create Filter (Property Value)	339
Create Filter (Size)	340
Create Filter (Time)	341
Query Files.....	342
Download.....	344
Download (Single)	344
Download (Multiple)	345
Close	346
Get Property	347
Set Property.....	348
Update Metadata Properties	349
Merge Metadata Properties	350
Property Node.....	351

Delete	352
Advanced Functions	352
Create	353
Send Packet.....	355
Send Last Packet	356
Close	357
Register for Available File Events	358
Open	359
Open (HTTP)	359
Unregister for Available File Events	360
Register for Packet Events	360
Download (Async)	361
Download (Async) (Single)	362
Download (Async) (Multiple)	363
Unregister for Packet Events	364
Test Monitor Functions	365
Add Test Step Measurement	366
Create Test Result.....	367
Create Test Step	368
Delete Test Result.....	370
Delete Test Step	370
Generate Named Value	371
Initialize Child Step from Step.....	372
Initialize Test Result.....	374
Initialize Test Step	375
Initialize Test Step from Result	376
Read Named Value Data	378
Update Test Result.....	379
Update Test Step	381
Utilities Functions	382
Get Minion ID.....	382
Get System Name.....	383
Alarm Functions	383
Acknowledge Alarm.....	385
Acknowledge Alarm (Multiple)	385
Acknowledge Alarm (Single)	386

Add Alarm Notes.....	387
Add Alarm Notes (Multiple)	387
Add Alarm Notes (Single)	388
Property Node.....	389
Build Alarm ID	390
Build Alarm Set Transition	391
Build Alarm Set Transition (Multiple).....	391
Build Alarm Set Transition (Single).....	393
Clear Alarm	395
Create Alarm Query Filter.....	396
Delete Alarm	397
Get Notification Strategies	397
Query Alarms.....	398
Set Alarm	401
Asset Functions.....	403
Asset Utilization Functions.....	403
Start Utilization.....	404
Start Utilization (All).....	404
Start Utilization (Multiple)	406
Start Utilization (Single)	407
End Utilization.....	408
List All Assets.....	409
Build Asset Tag Path	409
HTTP API Reference.....	410
Python API Reference	412
.NET API Reference	412

API Reference

G, HTTP, Python, and .NET APIs allow you to programmatically access SystemLink data services outside of NI SystemLink Web Application.

Use the following table to choose which API meets your needs depending on your programming environment and the data services you want to access.

API	How to access	Data Services
G (SystemLink nodes, or functions)	LabVIEW 2017 or later	• Message • Tag • Tag Historian • Server Configuration • File Transfer • Test Monitor • Alarm • Asset Utilization • DataFinder
	LabVIEW NXG 5.0 or later	• Message • Tag • Tag Historian • Server Configuration • File Transfer • Alarm • Asset Utilization • TDM Reader • Utilities
	• G Web Development Software 2021 • LabVIEW NXG Web Module 5.0 or later	• Message • Tag • Tag Historian • Server Configuration • Alarms • TDM Reader

API	How to access	Data Services
Python	• JupyterHub • NI SystemLink API Clients for Python (Tag only)	• File Ingestion • TDM Reader • Tag • Test Monitor • Asset Management
HTTP	Any language-specific HTTP client	• Auth Service • User Service • Alarm • Asset Management • File • Message • OPC UA • Tag • Tag Historian • TDM Reader • Test Monitor • Systems State Manager • Salt Service • Analysis Automation Configuration • Analysis Automation Package Creation • Data Analysis • Data Cart • DataPlugin Service • Data Navigation Utils • Data Navigation Data Service • Data Indexing Configuration • Data Preparation Configuration • Federation Configuration
.NET	Any language-specific .NET client	• Core • Message

API	How to access	Data Services
		• Tag • File Ingestion • Test Monitor • Asset Management
NI (Logos/Corba)	DIAdem	DataFinder
COM (DIAdem; Programming languages)	Object-oriented script interface	• Analysis Automation Package Creation • Data Preparation Package Creation • File Service • File Transfer • Tags



Note

The Test Monitor API is only available in LabVIEW 2018 or later.

Related concepts:

- [HTTP API Reference](#)
- [Python API Reference](#)

SystemLink Nodes and Functions Reference for NXG and G Web Development Software

Use SystemLink nodes in NXG or G Web Development Software to communicate data between your test and measurement systems and your SystemLink Server account.



Note G Web Development Software does not support the Asset Utilization, File Transfer, or Utilities APIs.

[API Reference](#)

G, HTTP, Python, and .NET APIs allow you to programmatically access SystemLink data services outside of NI SystemLink Web Application.

[Alarm Functions](#)

Create alarms to implement rules and manage conditions affecting your systems, tests, or assets.

[Asset Utilization Functions](#)

Track how long and how often you use your assets in test systems.

[Configuration Functions](#)

Manage the HTTP or HTTPS connection with your server.

[File Transfer Functions](#)

Upload files to your SystemLink Server, where you can also update, delete, or download them.

[Message Functions](#)

Communicate as a network queue using the publish-subscribe model. Use messages to send commands, status updates, and data between your systems and server.

[Tag Functions](#)

Use tags to read and write data values between systems and SystemLink Server.

[TDM Reader Functions](#)

Visualize and interact with waveform data in TDMS files.

[Utility Functions](#)

Get information about systems connected to your server.

[API Reference](#)

G, HTTP, Python, and .NET APIs allow you to programmatically access SystemLink data services outside of NI SystemLink Web Application.

Alarm Functions

Create alarms to implement rules and manage conditions affecting your systems, tests, or assets.

[SystemLink Nodes and Functions Reference for NXG and G Web Development Software](#)

Use SystemLink nodes in NXG or G Web Development Software to communicate data between your test and measurement systems and your SystemLink Server account.

[Acknowledge Alarm](#)

Acknowledges one or more active alarm instances on the server.

[Add Alarm Notes](#)

Adds one or more notes, or comments, to an alarm instance.

[Build Alarm ID](#)

Creates an alarm ID using the names of the system and application where the alarm originated.

[Build Alarm Set Transition](#)

Creates a `set transition` to apply when an alarm condition is met.

[Clear Alarm](#)

Clears a set alarm instance on the server if the node does not detect a condition to trigger the alarm.

[Delete Alarm](#)

Removes the alarm instance identified by `alarm` from the server.

[Create Alarm Query Filter](#)

Returns a group of values you specify to query the server for matching alarm instances.

[Get Alarm Information](#)

Returns information, such as notes, severity levels, transition details, and more, about an alarm instance.

[Get Notification Strategies](#)

Queries the server for all alarm notification strategies.

[Query Alarms](#)

Searches the server for alarm instances that match the `filter`.

[Set Alarm](#)

Triggers an alarm instance to notify subscribers of an issue or a state requiring attention when an application detects a condition.

[API Reference](#)

G, HTTP, Python, and .NET APIs allow you to programmatically access SystemLink data services outside of NI SystemLink Web Application.

[SystemLink Nodes and Functions Reference for NXG and G Web Development Software](#)

Use SystemLink nodes in NXG or G Web Development Software to communicate data between your test and measurement systems and your SystemLink Server account.

Acknowledge Alarm

Acknowledges one or more active alarm instances on the server.

[Alarm Functions](#)

Create alarms to implement rules and manage conditions affecting your systems, tests, or assets.

[Multiple](#)

Acknowledges a group of active alarm instances on the server with a single call.

Single

Acknowledges an active alarm instance on the server.

API Reference

G, HTTP, Python, and .NET APIs allow you to programmatically access SystemLink data services outside of NI SystemLink Web Application.

SystemLink Nodes and Functions Reference for NXG and G Web Development Software

Use SystemLink nodes in NXG or G Web Development Software to communicate data between your test and measurement systems and your SystemLink Server account.

Alarm Functions

Create alarms to implement rules and manage conditions affecting your systems, tests, or assets.

Multiple

Acknowledges a group of active alarm instances on the server with a single call.

An alarm instance remains active until you acknowledge and clear it. Typically, the application that sets the alarm clears it when the condition(s) that triggered the alarm no longer exist. An operator must acknowledge the alarms in the user interface or call this node.

Use Query Alarms to find alarm instances you want to acknowledge.

Inputs/Outputs

`datatype_icon`

alarms in

Group of active alarm instances you need to acknowledge.

`datatype_icon`

note

Information you want to include about the alarm acknowledgement.

`datatype_icon`

clear alarms

A Boolean that determines whether to clear alarms without waiting for the application that set the alarms to do it.

Default value: False

datatype_icon

error in

Error conditions that occur before this node runs.

The node responds to this input according to standard error behavior.

Standard Error Behavior

Default value: No error

datatype_icon

alarms out

Group of active alarm instances you acknowledged.

datatype_icon

acknowledged

An array of Booleans that determine whether your team acknowledged the alarm instances.

datatype_icon

error out

Error information.

The node produces this output according to standard error behavior.

Standard Error Behavior

Single

Acknowledges an active alarm instance on the server.

An alarm instance remains active until you acknowledge and clear it. Typically, the application that sets the alarm clears it when the condition(s) that triggered the alarm no longer exist. An operator must acknowledge the alarm in the user interface or call this node.

Use Query Alarms to find alarm instances you want to acknowledge.

Inputs/Outputs

datatype_icon

alarm in

Active alarm instance you need to acknowledge.

datatype_icon

note

Information you want to include about the alarm acknowledgement.

datatype_icon

clear alarm

A Boolean that determines whether to clear an alarm without waiting for the application that set the alarm to do it.

Default value: False

datatype_icon

error in

Error conditions that occur before this node runs.

The node responds to this input according to standard error behavior.

Standard Error Behavior

Default value: No error

datatype_icon

alarm out

Active alarm instance you acknowledged.

datatype_icon

acknowledged?

An array of Booleans that determine whether your team acknowledged the alarm instances.

datatype_icon

error out

Error information.

The node produces this output according to standard error behavior.

Standard Error Behavior

Add Alarm Notes

Adds one or more notes, or comments, to an alarm instance.

Alarm Functions

Create alarms to implement rules and manage conditions affecting your systems, tests, or assets.

Multiple

Add a group of notes to an alarm instance to record information about it.

Single

Add a note to an alarm instance to record information about it.

API Reference

G, HTTP, Python, and .NET APIs allow you to programmatically access SystemLink data services outside of NI SystemLink Web Application.

SystemLink Nodes and Functions Reference for NXG and G Web Development Software

Use SystemLink nodes in NXG or G Web Development Software to communicate data between your test and measurement systems and your SystemLink Server account.

Alarm Functions

Create alarms to implement rules and manage conditions affecting your systems, tests, or assets.

Multiple

Add a group of notes to an alarm instance to record information about it.

Inputs/Outputs

`datatype_icon`

alarm in

Alarm instance to which you want to add a note about on the server.

`datatype_icon`

notes

Information you want to include about the alarm acknowledgement.

`datatype_icon`

time

Timestamp of when you created a note. Leave this input unwired to use the current time on the server.

`datatype_icon`

error in

Error conditions that occur before this node runs.

The node responds to this input according to standard error behavior.

Standard Error Behavior

Default value: No error

datatype_icon

alarm out

Alarm instance to which you added a note about on the server.

datatype_icon

error out

Error information.

The node produces this output according to standard error behavior.

Standard Error Behavior

Single

Add a note to an alarm instance to record information about it.

Inputs/Outputs

datatype_icon

alarm in

Alarm instance to which you want to add a note about on the server.

datatype_icon

note

Information you want to include about the alarm acknowledgement.

datatype_icon

time

Timestamp of when you created a note. Leave this input unwired to use the current time on the server.

datatype_icon

error in

Error conditions that occur before this node runs.

The node responds to this input according to standard error behavior.

[Standard Error Behavior](#)

Default value: No error

datatype_icon

alarm out

Alarm instance to which you added a note about on the server.

datatype_icon

error out

Error information.

The node produces this output according to standard error behavior.

[Standard Error Behavior](#)

Build Alarm ID

Creates an alarm ID using the names of the system and application where the alarm originated.

Inputs/Outputs

`datatype_icon`

alarm name

Name of the alarm triggered.

`datatype_icon`

namespace

Name to overwrite the default namespace.

The namespace of the alarm ID defaults to the name of the top-level VI where the alarm triggered. If you want to override the default namespace, wire a name to the namespace input.

`datatype_icon`

alarm id

Unique identifier for a process or condition tracked by an alarm.

Build Alarm Set Transition

Creates a `set transition` to apply when an alarm condition is met.

Alarm Functions

Create alarms to implement rules and manage conditions affecting your systems, tests, or assets.

Multiple Notifications

Creates a `set transition` to apply to multiple notification strategies when an alarm condition is met and the alarm changes in severity.

Single Notification

Creates a `set transition` to apply to a notification strategy when an alarm condition is met and the alarm changes in severity.

API Reference

G, HTTP, Python, and .NET APIs allow you to programmatically access SystemLink data services outside of NI SystemLink Web Application.

SystemLink Nodes and Functions Reference for NXG and G Web Development Software

Use SystemLink nodes in NXG or G Web Development Software to communicate data between your test and measurement systems and your SystemLink Server account.

Alarm Functions

Create alarms to implement rules and manage conditions affecting your systems, tests, or assets.

Multiple Notifications

Creates a `set transition` to apply to multiple notification strategies when an alarm condition is met and the alarm changes in severity.

Wire `set transition` to Set Alarm to create or update an alarm instance on the server.

Inputs/Outputs

`datatype_icon`

occurred at

Timestamp of when a system or application triggered an alarm instance.

`datatype_icon`

value

Data value that caused the application to set the alarm.

`datatype_icon`

condition

Description of what caused the transition.

For example, `Greater than 90% CPU` or `Low disk space`.

`datatype_icon`

severity level

Number that indicates how severe an alarm instance is. In the user interface, severity levels range from 1 to 4, from lowest to highest.

`datatype_icon`

notification strategy ids

IDs of the notification strategies that will trigger if the request creates an alarm instance or transitions to a new highest severity level.

`datatype_icon`

keywords

Words or phrases associated with the transition.

`datatype_icon`

properties

Key-value pairs associated with the transition.

`datatype_icon`

set transition

State change and severity level that meet a specific condition to create or update an alarm instance.

Single Notification

Creates a `set transition` to apply to a notification strategy when an alarm condition is met and the alarm changes in severity.

Wire `set transition` to `Set Alarm` to create or update an alarm instance on the server.

Inputs/Outputs

`datatype_icon`

occurred at

Timestamp of when a system or application triggered an alarm instance.

`datatype_icon`

value

Data value that caused the application to set the alarm.

`datatype_icon`

condition

Description of what caused the transition.

For example, `Greater than 90% CPU` or `Low disk space`.

`datatype_icon`

severity level

Number that indicates how severe an alarm instance is. In the user interface, severity levels range from 1 to 4, from lowest to highest.

`datatype_icon`

notification strategy id

ID of the notification strategy that will trigger if the request creates an alarm instance or transitions to a new highest severity level.

`datatype_icon`

keywords

Words or phrases associated with the transition.

datatype_icon

properties

Key-value pairs associated with the transition.

datatype_icon

set transition

State change and severity level that meet a specific condition to create or update an alarm instance.

Clear Alarm

Clears a set alarm instance on the server if the node does not detect a condition to trigger the alarm.

An alarm instance remains active until you acknowledge and clear it. Typically, the application that sets the alarm clears it when the condition(s) that triggered the alarm no longer exist.

Inputs/Outputs

datatype_icon

configuration

Connection to the server.

datatype_icon

alarm id

Unique identifier for a process or condition tracked by an alarm. The Build Alarm ID node concatenates the names of the system and top-level VI where the rule triggered

with the `alarm_name` into a path.

For example, an `alarm_id` may look like `<system>.<namespace>.<alarm_name>`.

`datatype_icon`

condition

Description of what caused the transition.

For example, `Greater than 90% CPU or Low disk space`.

`datatype_icon`

error in

Error conditions that occur before this node runs.

The node responds to this input according to standard error behavior.

Standard Error Behavior

Default value: No error

`datatype_icon`

alarm instance

Cleared alarm instance, if an instance exists. If no instance exists, this output returns an empty class instance.

`datatype_icon`

cleared?

Boolean that determines whether a set alarm instance exists. If no alarm instance exists, it returns `False` and `alarm_instance` returns default data.

`datatype_icon`

error out

Error information.

The node produces this output according to standard error behavior.

Standard Error Behavior

Delete Alarm

Removes the alarm instance identified by `alarm` from the server.

Only delete an active alarm as a last resort. Instead, you should clear and acknowledge it or force clear and acknowledge it.

Inputs/Outputs

`datatype_icon`

alarm

Alarm instance you want to delete on the server.

`datatype_icon`

error in

Error conditions that occur before this node runs.

The node responds to this input according to standard error behavior.

Standard Error Behavior

Default value: No error

`datatype_icon`

error out

Error information.

The node produces this output according to standard error behavior.

Standard Error Behavior

Create Alarm Query Filter

Returns a group of values you specify to query the server for matching alarm instances.

Wire this cluster to Query Filter to search the server.

Use the properties in the cluster to specify the values you want to match on the server. The G Type contains multiple properties. Refer to the following table to learn about the properties you can use to filter alarm instances from the server.

Property	Description	Example
alarm id	Unique identifier for a process or condition tracked by an alarm.	<code>System.Health.DiskSpaceLow.Bdd9u!4aMd!\$pYrf*CnaIZ2tbCt%?</code>
acknowledged after	Timestamp representing the earliest time to return alarm instances for. The server returns alarm instances your team acknowledged between acknowledged after and acknowledged before.	<code>2019-01-14T23:15:53.7270539Z</code>
acknowledged before	Timestamp representing the latest time to return alarm instances for. The server returns alarm instances	<code>2019-01-15T23:15:53.7270539Z</code>

Property	Description	Example
	your team acknowledged between acknowledged after and acknowledged before.	
channel	Resource associated with the alarm. For example, if an application sets an alarm because a wireless network went down, the channel could be the name of the network and the resource type classifies the alarm as the network.	<code>CRIO1.System.Health.DiskSpaceUsePercentage</code>
created by	Identifier for who or what created a given alarm instance. This property normally identifies the rule engine that requested an alarm instance be created.	<code>TagRuleEngine</code>
current severity level max	Highest current severity level you want to return alarm instances for. The server returns alarm	3

Property	Description	Example
	instances with current severity levels between current severity level min and current severity level max.	
current severity level min	Lowest current severity level you want to return alarm instances for. The server returns alarm instances with current severity levels between current severity level min and current severity level max.	2
description	Notes you gave an alarm instance.	Disk space on tester-1 is high. Disk usage was 95% w this alarm was created.
display name	Title you gave the alarm instance.	Low disk space on tester-1
highest severity level max	Highest severity level an instance ever reached. The server returns alarm instances with severity levels between highest severity level min and highest severity level max.	4
highest severity level min	Highest severity level an alarm instance ever	2

Property	Description	Example
	reached. The server returns alarm instances with severity levels between highest severity level min and highest severity level max.	
is acknowledged	Enum that determines whether to return acknowledged, unacknowledged, or all alarm instances.	Any
is active	Enum that determines whether to return active, inactive, or all alarm instances.	False
is clear	Enum that determines whether to return cleared, active, or all alarm instances.	True
occurred after	Timestamp specifying when an alarm instance occurred. The server returns alarm instances with timestamps between occurred after and occurred before.	2019-01-14T23:15:53.7270539Z

Property	Description	Example
occurred before	Timestamp specifying when an alarm instance occurred. The server returns alarm instances with timestamps between occurred after and occurred before.	2019-01-15T23:15:53.7270539Z
keywords	Words or phrases associated with an alarm instance.	keyword1
properties	Key-value pairs associated with an alarm instance.	Slot Number = -1
resource type	Type of resource associated with the alarm. For example, if an application sets an alarm because a wireless network went down, the channel may be the name of the network where the alarm triggered and the resource type may classify the alarm as the network.	Network

Get Alarm Information

Returns information, such as notes, severity levels, transition details, and more, about an alarm instance.

Inputs/Outputs

datatype_icon

alarm in

Active alarm instance you need to acknowledge.

datatype_icon

retrieve transitions?

A Boolean that determines whether to obtain information about the transitions the alarm instance went through.

datatype_icon

error in

Error conditions that occur before this node runs.

The node responds to this input according to standard error behavior.

Standard Error Behavior

Default value: No error

datatype_icon

alarm out

Alarm instance you obtained information about.

datatype_icon

information

Details about the alarm instance you specified, such as when and where it occurred, severity levels, and more.

datatype_icon

transitions

Transition history of the alarm instance.

datatype_icon

error out

Error information.

The node produces this output according to standard error behavior.

[Standard Error Behavior](#)

Get Notification Strategies

Queries the server for all alarm notification strategies.

The notifications strategies are then sent to Set Alarm when a condition is triggered.

Inputs/Outputs

datatype_icon

configuration

Connection to the server.

datatype_icon

error in

Error conditions that occur before this node runs.

The node responds to this input according to standard error behavior.

Standard Error Behavior

Default value: No error

`datatype_icon`

notification strategies

Plans established to notify the specified users with the right information whenever an alarm triggers on a system or application.

`datatype_icon`

error out

Error information.

The node produces this output according to standard error behavior.

Standard Error Behavior

Query Alarms

Searches the server for alarm instances that match the `filter`.

Alarm Functions

Create alarms to implement rules and manage conditions affecting your systems, tests, or assets.

No Transitions

Searches the server for alarm instances that match the `filter` without obtaining the transition history of an alarm.

With Transitions

Searches the server for alarm instances that match the `filter` while obtaining the transition history of an alarm.

API Reference

G, HTTP, Python, and .NET APIs allow you to programmatically access SystemLink data services outside of NI SystemLink Web Application.

SystemLink Nodes and Functions Reference for NXG and G Web Development Software

Use SystemLink nodes in NXG or G Web Development Software to communicate data between your test and measurement systems and your SystemLink Server account.

Alarm Functions

Create alarms to implement rules and manage conditions affecting your systems, tests, or assets.

No Transitions

Searches the server for alarm instances that match the `filter` without obtaining the transition history of an alarm.

Inputs/Outputs

`datatype_icon`

limit to configured workspace?

Boolean that determines whether this node only queries for alarms in the workspace you specify in `configuration`.

By default, the node queries for alarms from all accessible workspaces.

`datatype_icon`

configuration

Connection to the server.

`datatype_icon`

filter

Properties and values used to find and return matching alarm instances.

Wire Filter G Type to this input.

`datatype_icon`

only most recent?

Boolean that determines whether the server should return only the most recent instance of each `alarm id`.

datatype_icon

order of results

Order in which the results are sorted based on the **updated at** field of an alarm instance.

Ascending	Sorts the alarm instances returned in least to greatest order.
Descending	Sorts the alarm instances returned in greatest to least order.

Default value: Ascending

datatype_icon

error in

Error conditions that occur before this node runs.

The node responds to this input according to standard error behavior.

Standard Error Behavior

Default value: No error

datatype_icon

skip

Number of alarm instances to skip on the server.

Default value: 0

`datatype_icon`

take

Number of alarm instances requested from the server.

If you leave `skip` and `take` empty, the query returns the first 1,000 results by default.

`datatype_icon`

alarms

Alarm instances matching the `filter` returned from the server.

`datatype_icon`

alarm information

Collection of notes about an alarm instance or group of alarm instances.

`datatype_icon`

total count

Total number of alarm instances matching the `filter` before taking `skip` and `take` into account on the server.

`datatype_icon`

error out

Error information.

The node produces this output according to standard error behavior.

Standard Error Behavior

With Transitions

Searches the server for alarm instances that match the `filter` while obtaining the

transition history of an alarm.

Inputs/Outputs

datatype_icon

limit to configured workspace?

Boolean that determines whether this node only queries for alarms in the workspace you specify in `configuration`.

By default, the node queries for alarms from all accessible workspaces.

datatype_icon

configuration

Connection to the server.

datatype_icon

filter

Properties and values used to find and return matching alarm instances.

Wire Filter G Type to this input.

datatype_icon

only most recent?

Boolean that determines whether the server should return only the most recent instance of each `alarm id`.

datatype_icon

order of results

Order in which the results are sorted based on the **updated at** field of an alarm instance.

Ascending	Sorts the alarm instances returned in least to greatest order.
Descending	Sorts the alarm instances returned in greatest to least order.

Default value: Ascending

`datatype_icon`

error in

Error conditions that occur before this node runs.

The node responds to this input according to standard error behavior. For more information, refer to ***Standard Error Behavior***.

Default value: No error

`datatype_icon`

skip

Number of alarm instances to skip on the server.

Default value: 0

`datatype_icon`

take

Number of alarm instances requested from the server.

If you leave `skip` and `take` empty, the query returns the first 1,000 results by default.

`datatype_icon`

alarms

Alarm instances matching the `filter` returned from the server.

datatype_icon

alarm information

Collection of notes about an alarm instance or group of alarm instances.

datatype_icon

total count

Total number of alarm instances matching the `filter` before taking `skip` and `take` into account on the server.

datatype_icon

error out

Error information.

The node produces this output according to standard error behavior. For more information, refer to ***Standard Error Behavior***.

Related information:

- [Standard Error Behavior](#)

Set Alarm

Triggers an alarm instance to notify subscribers of an issue or a state requiring attention when an application detects a condition.

Wire Clear Alarm after this node to clear the alarms associated with the alarm instance.

Inputs/Outputs

datatype_icon

configuration

Connection to the server.

datatype_icon

additional information

Variety of data about the new alarm instance, such as the application or user who created it and keywords and properties associated with it.

This node ignores this input when updating an existing alarm instance

datatype_icon

alarm id

Unique identifier for a process or condition tracked by an alarm. The Build Alarm ID node concatenates the names of the system and top-level VI where the rule triggered with the `alarm name` into a path.

For example, an `alarm id` may look like `<system>.<namespace>.<alarm name>`.

datatype_icon

set transition

State change and severity level that meet a specific condition to create or update an alarm instance.

Wire Build Alarm Set Transition to this input.

datatype_icon

display name

Title you want to give the alarm instance.

This node ignores this input when updating an existing alarm instance.

datatype_icon

channel

Resource associated with the alarm. For example, if an application sets an alarm because a wireless network went down, the `channel` could be the name of the network and the `resource type` classifies the alarm as the network.

This node ignores this input when it updates an existing alarm instance.

`datatype_icon`

error in

Error conditions that occur before this node runs.

The node responds to this input according to standard error behavior.

Standard Error Behavior

Default value: No error

`datatype_icon`

resource type

Type of resource associated with the alarm. For example, if an application sets an alarm because a wireless network went down, the `channel` may be the name of the network where the alarm triggered and the `resource type` may classify the alarm as the network.

This node ignores this input when it updates an existing alarm instance.

`datatype_icon`

alarm

Alarm instance created or updated.

`datatype_icon`

alarm instance created or updated?

Boolean that determines whether an alarm instance with the provided `set transition` already exists. If an alarm instance already exists or the current severity level already occurred, it returns `False` and `alarm` returns default data.

`datatype_icon`

error out

Error information.

The node produces this output according to standard error behavior.

Standard Error Behavior

Asset Utilization Functions

Track how long and how often you use your assets in test systems.



Note G Web Development Software does not support this API.

SystemLink Nodes and Functions Reference for NXG and G Web Development Software

Use SystemLink nodes in NXG or G Web Development Software to communicate data between your test and measurement systems and your SystemLink Server account.

End Utilization

Stops collecting usage data about an asset or group of assets when a test or measurement task finishes.

Start Utilization

Collects usage data about an asset, group of assets, or all assets when a test or measurement task starts. G Web Development Software does not support this function.

API Reference

G, HTTP, Python, and .NET APIs allow you to programmatically access SystemLink data services outside of NI SystemLink Web Application.

SystemLink Nodes and Functions Reference for NXG and G Web Development Software

Use SystemLink nodes in NXG or G Web Development Software to communicate data

between your test and measurement systems and your SystemLink Server account.

End Utilization

Stops collecting usage data about an asset or group of assets when a test or measurement task finishes.



Note You can only run this node on a SystemLink system, or minion, or on the server itself. If you run this node on an unmanaged system, the data obtained will not transmit to the server. Instead, the data transfer will fail silently.



Note G Web Development Software does not support this function.

Inputs/Outputs

`datatype_icon`

asset utilization

Asset utilization task reference on the server.

`datatype_icon`

error in

Error conditions that occur before this node runs.

The node responds to this input according to standard error behavior.

Standard Error Behavior

Default value: No error

`datatype_icon`

error out

Error information.

The node produces this output according to standard error behavior.

Standard Error Behavior

Start Utilization

Collects usage data about an asset, group of assets, or all assets when a test or measurement task starts. G Web Development Software does not support this function.

Asset Utilization Functions

Track how long and how often you use your assets in test systems.

All

Collects usage data about every asset when a test or measurement task starts.

Multiple

Collects usage data about a group of assets when a test or measurement task starts.

Single

Collects usage data about an asset when a test or measurement task starts.

API Reference

G, HTTP, Python, and .NET APIs allow you to programmatically access SystemLink data services outside of NI SystemLink Web Application.

SystemLink Nodes and Functions Reference for NXG and G Web Development Software

Use SystemLink nodes in NXG or G Web Development Software to communicate data between your test and measurement systems and your SystemLink Server account.

Asset Utilization Functions

Track how long and how often you use your assets in test systems.

All

Collects usage data about every asset when a test or measurement task starts.



Note You can only run this node on a SystemLink system, or minion, or on the server itself. If you run this node on an unmanaged system, the data obtained will not transmit to the server. Instead, the data transfer will fail silently.



Note G Web Development Software does not support this function.

Inputs/Outputs

datatype_icon

task name

Name of the test or measurement task you want the asset to perform.

datatype_icon

utilization category

Reason the `user` is using the asset.

Default value: General

datatype_icon

user

Name of the person using the asset.

datatype_icon

error in

Error conditions that occur before this node runs.

The node responds to this input according to standard error behavior.

Standard Error Behavior

Default value: No error

datatype_icon

asset utilization

Asset utilization task reference on the server.

datatype_icon

error out

Error information.

The node produces this output according to standard error behavior.

Standard Error Behavior**Multiple**

Collects usage data about a group of assets when a test or measurement task starts.



Note You can only run this node on a SystemLink system, or minion, or on the server itself. If you run this node on an unmanaged system, the data obtained will not transmit to the server. Instead, the data transfer will fail silently.



Note G Web Development Software does not support this function.

Inputs/Outputs

datatype_icon

task name

Name of the test or measurement task you want the asset to perform.

datatype_icon

asset names

Names of the devices.

datatype_icon

utilization category

Reason the `user` is using the asset.

Default value: General

datatype_icon

user

Name of the person using the asset.

datatype_icon

error in

Error conditions that occur before this node runs.

The node responds to this input according to standard error behavior.

Standard Error Behavior

Default value: No error

datatype_icon

asset utilization

Asset utilization task reference on the server.

datatype_icon

error out

Error information.

The node produces this output according to standard error behavior.

Standard Error Behavior

Single

Collects usage data about an asset when a test or measurement task starts.



Note You can only run this node on a SystemLink system, or minion, or on the server itself. If you run this node on an unmanaged system, the data obtained will not transmit to the server. Instead, the data transfer will fail silently.



Note G Web Development Software does not support this function.

Inputs/Outputs

`datatype_icon`

task name

Name of the test or measurement task you want the asset to perform.

`datatype_icon`

asset name

Name of the device.

`datatype_icon`

utilization category

Reason the `user` is using the asset.

Default value: General

`datatype_icon`

user

Name of the person using the asset.

datatype_icon

error in

Error conditions that occur before this node runs.

The node responds to this input according to standard error behavior.

Standard Error Behavior

Default value: No error

datatype_icon

asset utilization

Asset utilization task reference on the server.

datatype_icon

error out

Error information.

The node produces this output according to standard error behavior.

Standard Error Behavior

Configuration Functions

Manage the HTTP or HTTPS connection with your server.

SystemLink Nodes and Functions Reference for NXG and G Web Development Software

Use SystemLink nodes in NXG or G Web Development Software to communicate data between your test and measurement systems and your SystemLink Server account.

Close Configuration

Terminates connections associated with the configuration and invalidates open references to SystemLink objects on SystemLink Server.

Open Configuration

Manage the HTTP or HTTPS connection with your server.

[Use Workspace](#)

Specifies the active workspace nodes that reference this configuration use to perform operations. The active workspace determines what resources the node can access on the SystemLink server and where new resources are created.

[List Workspaces](#)

Lists workspaces available on the SystemLink server.

[API Reference](#)

G, HTTP, Python, and .NET APIs allow you to programmatically access SystemLink data services outside of NI SystemLink Web Application.

[SystemLink Nodes and Functions Reference for NXG and G Web Development Software](#)

Use SystemLink nodes in NXG or G Web Development Software to communicate data between your test and measurement systems and your SystemLink Server account.

Close Configuration

Terminates connections associated with the configuration and invalidates open references to SystemLink objects on SystemLink Server.

Inputs/Outputs

datatype_icon

configuration

Connection to the server.

If you leave this input unwired, this node uses the default connection. It automatically retrieves your SystemLink Server credentials when you execute the application on a managed system or server.

datatype_icon

error in

Error conditions that occur before this node runs.

Unlike most nodes, this node runs normally even if an error occurs before this node runs.

Standard Error Behavior

Default value: no error

`datatype_icon`

error out

Error information.

The node produces this output according to standard error behavior.

Standard Error Behavior

Programming Patterns

- Transferring Data Using Tags
- Sending Messages Between Systems

Open Configuration

Manage the HTTP or HTTPS connection with your server.

Configuration Functions

Manage the HTTP or HTTPS connection with your server.

User

Initiates a SystemLink Server connection using HTTP or HTTPS. The function authenticates the connection with a valid `username` and `password`.

User (Desktop)

Initializes an HTTP connection for SystemLink data services to communicate with your SystemLink Server using your SystemLink Server login credentials and a CA certificate for authentication.

API Key

Initializes an HTTPS connection for SystemLink data services to communicate with SystemLink using an API key for authentication.

API Key (Desktop)

Initializes an HTTPS connection for SystemLink data services to communicate with SystemLink using an API key and a CA certificate for authentication.

Auto (Desktop)

Initializes an HTTP connection for SystemLink data services to communicate with your SystemLink Server by automatically retrieving credentials when executed on a managed system.

Auto

Initializes an HTTP connection for SystemLink data services to communicate with your SystemLink Server by automatically retrieving credentials when executed on a managed system.

API Reference

G, HTTP, Python, and .NET APIs allow you to programmatically access SystemLink data services outside of NI SystemLink Web Application.

SystemLink Nodes and Functions Reference for NXG and G Web Development Software

Use SystemLink nodes in NXG or G Web Development Software to communicate data between your test and measurement systems and your SystemLink Server account.

Configuration Functions

Manage the HTTP or HTTPS connection with your server.

User

Initiates a SystemLink Server connection using HTTP or HTTPS. The function authenticates the connection with a valid `username` and `password`.

Inputs/Outputs

`datatype_icon`

server url

URL of the server where this node sends the web request.

Default value: `http://localhost:80`

URL Syntax

Part	Definition	Example
Scheme	Protocol the web service uses.	<code>http://</code>
Host	Domain name or IP address of the service.	<code>localhost</code>
Port	HTTP port of the web service. If you don't specify a port, the web service	<code>80</code>

Part	Definition	Example
	uses the default port.	

Use the following syntax to prompt a connection to the SystemLink Server:

`https://host:<port>.`

To host a web application on a SystemLink Server, leave the `server url` input empty.

`datatype_icon`

username

Login username for the server.

`datatype_icon`

password

Login password for the server.

`datatype_icon`

error in

Error conditions that occur before this node runs.

The node responds to this input according to standard error behavior.

Standard Error Behavior

Default value: No error

`datatype_icon`

configuration

Connection to the server.

`datatype_icon`

error out

Error information.

The node produces this output according to standard error behavior.

Standard Error Behavior

Programming Patterns

- Transferring Data Using Tags
- Sending Messages Between Systems

Node Configuration for Hosting Your Web Application on SystemLink Server

When you host your web application on SystemLink Server, leave the `server_url`, `username`, and `password` inputs of the Open Configuration (User) node empty. To minimize security vulnerabilities, SystemLink Server will use your LDAP or Windows user account credentials for authentication.

User (Desktop)

Initializes an HTTP connection for SystemLink data services to communicate with your SystemLink Server using your SystemLink Server login credentials and a CA certificate for authentication.



Note G Web Development Software does not support this function.

Inputs/Outputs

`datatype_icon`

verify server?

A Boolean that determines whether this VI will verify the connection is secured by a

valid certificate.

The default value is True.

`datatype_icon`

server url

URL of the server where this node sends the web request.

Default value: `http://localhost:80`

URL Syntax

Part	Definition	Example
Scheme	Protocol the web service uses.	<code>http://</code>
Host	Domain name or IP address of the service.	<code>localhost</code>
Port	HTTP port of the web service. If you don't specify a port, the web service uses the default port.	<code>80</code>

Use the following syntax to prompt a connection to the SystemLink Server:

`https://host:<port>.`

To host a web application on a SystemLink Server, leave the `server url` input empty.

`datatype_icon`

username

Login username for the server.

`datatype_icon`

password

Login password for the server.

`datatype_icon`

ca certificate file

File path to where a custom or private CA certificate (`.cer`) is located on the system.

The CA certificate authenticates a secure sockets layer to communicate data between your application and the server. If you leave this input unwired, the VI uses the system defaults, which includes public CA certificates.

datatype_icon

error in

Error conditions that occur before this node runs.

The node responds to this input according to standard error behavior.

Standard Error Behavior

Default value: No error

datatype_icon

configuration

Connection to the server.

datatype_icon

error out

Error information.

The node produces this output according to standard error behavior.

Standard Error Behavior**Programming Patterns**

- Transferring Data Using Tags
- Sending Messages Between Systems

Node Configuration for Hosting Your Web Application on SystemLink Server

When you host your web application on SystemLink Server, leave the `server url`, `username`, and `password` inputs of the Open Configuration (User) node empty. To minimize security vulnerabilities, SystemLink Server will use your LDAP or Windows user account credentials for authentication.

API Key

Initializes an HTTPS connection for SystemLink data services to communicate with SystemLink using an API key for authentication.

Inputs/Outputs

`datatype_icon`

`server url`

URL of the server where this node sends the web request.

Default value: `http://localhost:80`

URL Syntax

Part	Definition	Example
Scheme	Protocol the web service uses.	<code>http://</code>
Host	Domain name or IP address of the service.	<code>localhost</code>
Port	HTTP port of the web service. If you don't specify a port, the web service uses the default port.	<code>80</code>

Use the following syntax to prompt a connection to the SystemLink Server:

`https://host:<port>`.

To host a web application on a SystemLink Server, leave the `server url` input empty.

datatype_icon

api key

Unique passcode the node uses to authenticate communication between the application and SystemLink server.

datatype_icon

error in

Error conditions that occur before this node runs.

The node responds to this input according to standard error behavior.

Standard Error Behavior

Default value: No error

datatype_icon

configuration

Connection to the server.

datatype_icon

error out

Error information.

The node produces this output according to standard error behavior.

Standard Error Behavior

Programming Patterns

- Transferring Data Using Tags
- Sending Messages Between Systems

API Key (Desktop)

Initializes an HTTPS connection for SystemLink data services to communicate with SystemLink using an API key and a CA certificate for authentication.



Note G Web Development Software does not support this function.

Inputs/Outputs

`datatype_icon`

verify server?

A Boolean that determines whether this VI will verify the connection is secured by a valid certificate.

The default value is True.

`datatype_icon`

server url

URL of the server where this node sends the web request.

Default value: `http://localhost:80`

URL Syntax

Part	Definition	Example
Scheme	Protocol the web service uses.	<code>http://</code>
Host	Domain name or IP address of the service.	<code>localhost</code>
Port	HTTP port of the web service. If you don't specify a port, the web service uses the default port.	<code>80</code>

Use the following syntax to prompt a connection to the SystemLink Server:

`https://host:<port>`.

To host a web application on a SystemLink Server, leave the `server url` input

empty.

datatype_icon

ca certificate file

File path to where a custom or private CA certificate (`.cer`) is located on the system.

The CA certificate authenticates a secure sockets layer to communicate data between your application and the server. If you leave this input unwired, the VI uses the system defaults, which includes public CA certificates.

datatype_icon

api key

Unique passcode the node uses to authenticate communication between the application and SystemLink.

datatype_icon

error in

Error conditions that occur before this node runs.

The node responds to this input according to standard error behavior.

Standard Error Behavior

Default value: No error

datatype_icon

configuration

Connection to the server.

datatype_icon

error out

Error information.

The node produces this output according to standard error behavior.

Standard Error Behavior

Programming Patterns

- Transferring Data Using Tags
- Sending Messages Between Systems

Auto (Desktop)

Initializes an HTTP connection for SystemLink data services to communicate with your SystemLink Server by automatically retrieving credentials when executed on a managed system.

This function runs on your SystemLink server without additional configuration information.



Note G Web Development Software does not support this function.

Inputs/Outputs

datatype_icon

require secure?

A Boolean that determines whether to use only an HTTPS connection. If set to True, your data services only communicate with the SystemLink Server over an HTTPS connection. The default value is True.

This input also allows local connections over HTTP in addition to HTTPS connections.



Note If the server is configured to use HTTP instead of HTTPS, set the value

to False or this function returns an error. Furthermore, using HTTP remotely insecurely exposes credentials and data to the network

datatype_icon

error in

Error conditions that occur before this node runs.

The node responds to this input according to standard error behavior.

Standard Error Behavior

Default value: No error

datatype_icon

configuration

Connection to the server.

datatype_icon

error out

Error information.

The node produces this output according to standard error behavior.

Standard Error Behavior

Auto

Initializes an HTTP connection for SystemLink data services to communicate with your SystemLink Server by automatically retrieving credentials when executed on a managed system.

This function runs on your SystemLink server without additional configuration information.

Inputs/Outputs

datatype_icon

error in

Error conditions that occur before this node runs.

The node responds to this input according to standard error behavior.

Standard Error Behavior

Default value: No error

datatype_icon

configuration

Connection to the server.

datatype_icon

error out

Error information.

The node produces this output according to standard error behavior.

Standard Error Behavior

Use Workspace

Specifies the active workspace nodes that reference this configuration use to perform operations. The active workspace determines what resources the node can access on the SystemLink server and where new resources are created.

Configuration Functions

Manage the HTTP or HTTPS connection with your server.

Name

Specifies, by workspace name, the active workspace a node that references this

configuration uses to perform operations.

Instance

Specifies, by workspace instance, the workspace that any node referencing this configuration uses to perform operations.

API Reference

G, HTTP, Python, and .NET APIs allow you to programmatically access SystemLink data services outside of NI SystemLink Web Application.

SystemLink Nodes and Functions Reference for NXG and G Web Development Software

Use SystemLink nodes in NXG or G Web Development Software to communicate data between your test and measurement systems and your SystemLink Server account.

Configuration Functions

Manage the HTTP or HTTPS connection with your server.

Name

Specifies, by workspace name, the active workspace a node that references this configuration uses to perform operations.

The active workspace determines what resources the node can access on the SystemLink server and where new resources are created.

Inputs/Outputs

datatype_icon

HTTP configuration in

Configuration to communicate with the server over HTTP.

datatype_icon

workspace name

Name of the workspace you want to use. The workspace you specify determines what resources the node can access on the SystemLink server and where new resources are created.

datatype_icon

error in

Error conditions that occur before this node runs.

The node responds to this input according to standard error behavior.

Standard Error Behavior

Default value: No error

datatype_icon

HTTP configuration out

Configuration to communicate with the server over HTTP.

datatype_icon

error out

Error information.

The node produces this output according to standard error behavior.

Standard Error Behavior**Instance**

Specifies, by workspace instance, the workspace that any node referencing this configuration uses to perform operations.

The active workspace determines what resources the node can access on the SystemLink server and where new resources are created.

Inputs/Outputs

datatype_icon

HTTP configuration in

Configuration to communicate with the server over HTTP.

datatype_icon

workspace

Cluster specifying the workspace you want to use. The workspace you specify determines what resources the node can access on the SystemLink server and where new resources are created.

datatype_icon

error in

Error conditions that occur before this node runs.

The node responds to this input according to standard error behavior.

Standard Error Behavior

Default value: No error

datatype_icon

HTTP configuration out

Configuration to communicate with the server over HTTP.

datatype_icon

error out

Error information.

The node produces this output according to standard error behavior.

Standard Error Behavior

List Workspaces

Lists workspaces available on the SystemLink server. SystemLink uses workspaces to organize and control access to data on the server.

This node returns only workspaces that the current Configuration can access. By default, this node does not return archived workspaces.

Inputs/Outputs

datatype_icon

configuration

Connection to the server.

If you leave this input unwired, this node uses the default connection. It automatically retrieves your SystemLink Server credentials when you execute the application on a managed system or server.

datatype_icon

include archived?

Boolean that specifies whether or not to include archived workspaces in the list.

datatype_icon

error in

Error conditions that occur before this node runs.

The node responds to this input according to standard error behavior.

Standard Error Behavior

Default value: No error

datatype_icon

configuration out

Connection to the server.

datatype_icon

workspaces

Workspaces this node can access using the current configuration. By default, this node does not return archived workspaces.

datatype_icon

error out

Error information.

The node produces this output according to standard error behavior.

Standard Error Behavior

File Transfer Functions

Upload files to your SystemLink Server, where you can also update, delete, or download them.



Note G Web Development Software does not support this API.

SystemLink Nodes and Functions Reference for NXG and G Web Development Software

Use SystemLink nodes in NXG or G Web Development Software to communicate data between your test and measurement systems and your SystemLink Server account.

Create Filter

Creates or adds to an existing filter to query files on the server.

Delete

Removes a file or a group of files from the server.

Download

Receives a file or an array of files from the server.

Get File Metadata

Returns the metadata of a file on the server.

Get Property

Returns the property `value` of the file that corresponds to the `key`.

Merge Metadata Properties

Updates the metadata properties of a file.

Open File

Opens a reference to an existing file on the SystemLink file service.

Query Files

Searches the SystemLink file service for files matching `filter` in using an HTTP configuration.

Send

Sends a local file or data string to the server using the configuration you specify.

API Reference

G, HTTP, Python, and .NET APIs allow you to programmatically access SystemLink data services outside of NI SystemLink Web Application.

SystemLink Nodes and Functions Reference for NXG and G Web Development Software

Use SystemLink nodes in NXG or G Web Development Software to communicate data between your test and measurement systems and your SystemLink Server account.

Create Filter

Creates or adds to an existing filter to query files on the server. G Web Development Software does not support this function.

File Transfer Functions

Upload files to your SystemLink Server, where you can also update, delete, or download them.

Extension

Creates or adds a file extension query to an existing file query filter.

Name

Creates or adds a file name query to an existing file query filter.

Property Value

Creates or adds a property query to an existing file query filter.

Size in Bytes

Creates or adds a size query to an existing file query filter.

Created Time Stamp

Creates or adds a timestamp query to an existing file query filter.

API Reference

G, HTTP, Python, and .NET APIs allow you to programmatically access SystemLink data services outside of NI SystemLink Web Application.

[SystemLink Nodes and Functions Reference for NXG and G Web Development Software](#)

Use SystemLink nodes in NXG or G Web Development Software to communicate data between your test and measurement systems and your SystemLink Server account.

[File Transfer Functions](#)

Upload files to your SystemLink Server, where you can also update, delete, or download them.

Extension

Creates or adds a file extension query to an existing file query filter.

When you wire this function to Query Files, the results only contain files that have a name property that matches the file extension query. If you provide multiple extension queries, the last extension query overrides all preceding ones.



Note G Web Development Software does not support this function.

Inputs/Outputs

datatype_icon

operator

Criteria for how to filter files on the server.

Default value: equal

datatype_icon

filter in

Query or queries you want to use find matching files on the server.

Default value: new filter

datatype_icon

extension

File extension to add.

datatype_icon

error in

Error conditions that occur before this node runs.

The node responds to this input according to standard error behavior.

Standard Error Behavior

Default value: No error

datatype_icon

filter out

Updated query or queries to find matching files on the server.

datatype_icon

error out

Error information.

The node produces this output according to standard error behavior.

Standard Error Behavior

Name

Creates or adds a file name query to an existing file query filter.



Note G Web Development Software does not support this function.

Inputs/Outputs

datatype_icon

operator

Criteria for how to filter files on the server.

Default value: contains

datatype_icon

filter in

Query or queries you want to use find matching files on the server.

Default value: new filter

datatype_icon

name

File name you want to add to the query filter.

datatype_icon

error in

Error conditions that occur before this node runs.

The node responds to this input according to standard error behavior.

Standard Error Behavior

Default value: No error

datatype_icon

filter out

Updated query or queries to find matching files on the server.

datatype_icon

error out

Error information.

The node produces this output according to standard error behavior.

Standard Error Behavior

Property Value

Creates or adds a property query to an existing file query filter.

When you wire this function to Query Files, the results contain only files that have property attributes that satisfy the query entry. If you specify multiple property queries for the same property, only files that match all queries open.



Note G Web Development Software does not support this function.

Inputs/Outputs

datatype_icon

operator

Criteria for how to filter files on the server.

Default value: contains

datatype_icon

filter in

Query or queries you want to use find matching files on the server.

Default value: new filter

datatype_icon

key

Unique identifier for the property you want to locate.

datatype_icon

value

Metadata property to add.

datatype_icon

error in

Error conditions that occur before this node runs.

The node responds to this input according to standard error behavior.

Standard Error Behavior

Default value: No error

datatype_icon

filter out

Updated query or queries to find matching files on the server.

datatype_icon

error out

Error information.

The node produces this output according to standard error behavior.

Standard Error Behavior

Size in Bytes

Creates or adds a size query to an existing file query filter.

When you wire this function to Query Files, the results only contain files that have size attributes that satisfy the query entry. If you provide multiple size queries with the same operator, the last size query overrides preceding ones.



Note G Web Development Software does not support this function.

Inputs/Outputs

datatype_icon

operator

Criteria for how to filter files on the server.

Default value: greater than or equal

datatype_icon

filter in

Query or queries you want to use find matching files on the server.

Default value: new filter

datatype_icon

size

Size, in bytes, of the file or files for which you want to query.

datatype_icon

error in

Error conditions that occur before this node runs.

The node responds to this input according to standard error behavior.

Standard Error Behavior

Default value: No error

datatype_icon

filter out

Updated query or queries to find matching files on the server.

datatype_icon

error out

Error information.

The node produces this output according to standard error behavior.

Standard Error Behavior

Created Time Stamp

Creates or adds a timestamp query to an existing file query filter.

When you wire this function to Query Files, the results only contain files that have timestamp attributes that satisfy the query entry. If you provide multiple timestamp queries with the same operator, the last timestamp query provided overrides preceding ones.



Note G Web Development Software does not support this function.

Inputs/Outputs

datatype_icon

operator

Criteria for how to filter files on the server.

Default value: greater than or equal

datatype_icon

filter in

Query or queries you want to use find matching files on the server.

Default value: new filter

datatype_icon

time stamp

Date and time in which the file was created.

datatype_icon

error in

Error conditions that occur before this node runs.

The node responds to this input according to standard error behavior.

Standard Error Behavior

Default value: No error

datatype_icon

filter out

Updated query or queries to find matching files on the server.

datatype_icon

error out

Error information.

The node produces this output according to standard error behavior.

Standard Error Behavior

Delete

Removes a file or a group of files from the server. G Web Development Software does not support this function.

File Transfer Functions

Upload files to your SystemLink Server, where you can also update, delete, or download them.

Single

Removes a file from the server.

Multiple

Removes a group of files from the server.

API Reference

G, HTTP, Python, and .NET APIs allow you to programmatically access SystemLink data services outside of NI SystemLink Web Application.

SystemLink Nodes and Functions Reference for NXG and G Web Development Software

Use SystemLink nodes in NXG or G Web Development Software to communicate data between your test and measurement systems and your SystemLink Server account.

File Transfer Functions

Upload files to your SystemLink Server, where you can also update, delete, or download them.

Single

Removes a file from the server.



Note G Web Development Software does not support this function.

Inputs/Outputs

datatype_icon

file

File on the server.

datatype_icon

error in

Error conditions that occur before this node runs.

The node responds to this input according to standard error behavior.

Standard Error Behavior

Default value: No error

datatype_icon

error out

Error information.

The node produces this output according to standard error behavior.

Standard Error Behavior

Multiple

Removes a group of files from the server.



Note G Web Development Software does not support this function.

Inputs/Outputs

datatype_icon

files

Files on the server.

datatype_icon

error in

Error conditions that occur before this node runs.

The node responds to this input according to standard error behavior.

Standard Error Behavior

Default value: No error

datatype_icon

error out

Error information.

The node produces this output according to standard error behavior.

Standard Error Behavior

Download

Receives a file or an array of files from the server. G Web Development Software does not support this function.

File Transfer Functions

Upload files to your SystemLink Server, where you can also update, delete, or download them.

Multiple

Receives an array of files from the server.

Single

Receives a file from the server.

API Reference

G, HTTP, Python, and .NET APIs allow you to programmatically access SystemLink data services outside of NI SystemLink Web Application.

[SystemLink Nodes and Functions Reference for NXG and G Web Development Software](#)

Use SystemLink nodes in NXG or G Web Development Software to communicate data between your test and measurement systems and your SystemLink Server account.

[File Transfer Functions](#)

Upload files to your SystemLink Server, where you can also update, delete, or download them.

Multiple

Receives an array of files from the server.

When you call this function, it blocks additional calls until the files are downloaded or it returns an error.



Note G Web Development Software does not support this function.

Inputs/Outputs

`datatype_icon`

files in

Files on the server.

`datatype_icon`

paths

Location where you want to save the files.

`datatype_icon`

error in

Error conditions that occur before this node runs.

The node responds to this input according to standard error behavior.

Standard Error Behavior

Default value: No error

datatype_icon

files out

Files on the server.

datatype_icon

paths out

Location where you saved the files.

datatype_icon

error out

Error information.

The node produces this output according to standard error behavior.

Standard Error Behavior

Single

Receives a file from the server.



Note G Web Development Software does not support this function.

When you call this function, it blocks additional calls until the file is downloaded or it returns an error.

Inputs/Outputs

datatype_icon

file in

File on the server.

datatype_icon

selected path

Location where you want to save the file.

datatype_icon

error in

Error conditions that occur before this node runs.

The node responds to this input according to standard error behavior.

Standard Error Behavior

Default value: No error

datatype_icon

file out

File downloaded from the server.

datatype_icon

path out

Location where you saved the file.

datatype_icon

error out

Error information.

The node produces this output according to standard error behavior.

Standard Error Behavior

Get File Metadata

Returns the metadata of a file on the server.



Note G Web Development Software does not support this function.

Inputs/Outputs

`datatype_icon`

file in

File on the server.

`datatype_icon`

file out

File on the server.

`datatype_icon`

id

Unique file identifier the SystemLink file service generates to identify the file on the server.

`datatype_icon`

metadata

Metadata properties of the file you specify.

Get Property

Returns the property `value` of the file that corresponds to the `key`.



Note G Web Development Software does not support this function.

Inputs/Outputs

datatype_icon

file in

File on the server.

datatype_icon

key

Unique identifier for the property you want to locate.

datatype_icon

error in

Error conditions that occur before this node runs.

The node responds to this input according to standard error behavior. For more information, refer to ***Standard Error Behavior***.

Default value: No error

datatype_icon

file out

File on the server.

datatype_icon

value

Property metadata returned.

datatype_icon

found?

Boolean that determines whether the function returns property.

datatype_icon

error out

Error information.

The node produces this output according to standard error behavior. For more information, refer to ***Standard Error Behavior***.

Merge Metadata Properties

Updates the metadata properties of a file.

Merge Metadata Properties verifies and merges properties according to the latest values available on the server. Use this function to safely modify the metadata properties of a file.



Note G Web Development Software does not support this function.

Inputs/Outputs

datatype_icon

file in

File on the server.

datatype_icon

add

Metadata properties you want to add to or overwrite on the file.

datatype_icon

remove

Metadata properties you want to remove from the file.

datatype_icon

error in

Error conditions that occur before this node runs.

The node responds to this input according to standard error behavior.

Standard Error Behavior

Default value: No error

datatype_icon

file out

File on the server.

datatype_icon

error out

Error information.

The node produces this output according to standard error behavior.

Standard Error Behavior

Open File

Opens a reference to an existing file on the SystemLink file service.



Note G Web Development Software does not support this function.

Inputs/Outputs

datatype_icon

configuration

Connection to the server.

If you leave this input unwired, this node uses the default connection. It automatically retrieves your SystemLink Server credentials when you execute the application on a managed system or server.

datatype_icon

file id

Unique identifier the server generates to identify the file on the server.

datatype_icon

error in

Error conditions that occur before this node runs.

The node responds to this input according to standard error behavior.

Standard Error Behavior

Default value: No error

datatype_icon

file

File reference returned from the server.

datatype_icon

error out

Error information.

The node produces this output according to standard error behavior.

Standard Error Behavior

Query Files

Searches the SystemLink file service for files matching `filter` in using an HTTP configuration.

If you do not provide a filter, this function opens references to every file on the service. The SystemLink file service limits the maximum number of results returned in a single query to 1,000 files.



Note G Web Development Software does not support this function.

Inputs/Outputs

datatype_icon

configuration

Connection to the server.

If you leave this input unwired, this node uses the default connection. It automatically retrieves your SystemLink Server credentials when you execute the application on a managed system or server.

datatype_icon

filter in

Query or queries you want to use find matching files on the server.

Default value: new filter

datatype_icon

limit to configured workspace?

Boolean that determines whether this node only queries for tags in the workspace you specify in `configuration`.

By default, the node queries for tags from all accessible workspaces.

`datatype_icon`

error in

Error conditions that occur before this node runs.

The node responds to this input according to standard error behavior.

Standard Error Behavior

Default value: No error

`datatype_icon`

files

Files returned from the server.

`datatype_icon`

error out

Error information.

The node produces this output according to standard error behavior.

Standard Error Behavior**Send**

Sends a local file or data string to the server using the configuration you specify. G Web Development Software does not support this function.

File Transfer Functions

Upload files to your SystemLink Server, where you can also update, delete, or download them.

Buffer

Uploads a buffer of data to create a new file on the SystemLink file service using the HTTP configuration.

Path

Sends a local file to the SystemLink file service using the HTTP configuration.

API Reference

G, HTTP, Python, and .NET APIs allow you to programmatically access SystemLink data services outside of NI SystemLink Web Application.

SystemLink Nodes and Functions Reference for NXG and G Web Development Software

Use SystemLink nodes in NXG or G Web Development Software to communicate data between your test and measurement systems and your SystemLink Server account.

File Transfer Functions

Upload files to your SystemLink Server, where you can also update, delete, or download them.

Buffer

Uploads a buffer of data to create a new file on the SystemLink file service using the HTTP configuration.



Note G Web Development Software does not support this function.

Inputs/Outputs

datatype_icon

configuration

Connection to the server.

If you leave this input unwired, this node uses the default connection. It automatically retrieves your SystemLink Server credentials when you execute the application on a managed system or server.

datatype_icon

data

A string containing the contents of the file. When the SystemLink file service completes uploading the contents to the server, it creates the file.

datatype_icon

file name

Name to give the file after the SystemLink file service uploads it to the server.

datatype_icon

properties

Metadata about the file to send to the server.



Note The SystemLink file service uses the current file name to name the file on the server.

datatype_icon

error in

Error conditions that occur before this node runs.

The node responds to this input according to standard error behavior.

Standard Error Behavior

Default value: No error

datatype_icon

timeout

Time, in milliseconds, that this VI waits for a file before timing out.

Default value: 60,000 milliseconds

datatype_icon

id

Unique file identifier the SystemLink file service generates to identify the file on the server.

datatype_icon

error out

Error information.

The node produces this output according to standard error behavior.

Standard Error Behavior

Path

Sends a local file to the SystemLink file service using the HTTP configuration.



Note G Web Development Software does not support this function.

Inputs/Outputs

datatype_icon

configuration

Connection to the server.

If you leave this input unwired, this node uses the default connection. It automatically retrieves your SystemLink Server credentials when you execute the application on a managed system or server.

datatype_icon

path

Path to the file to send.

datatype_icon

properties

Metadata about the file to send to the server.



Note The SystemLink file service uses the current file name to name the file on the server.

datatype_icon

error in

Error conditions that occur before this node runs.

The node responds to this input according to standard error behavior.

Standard Error Behavior

Default value: No error

datatype_icon

timeout

Time, in milliseconds, that this VI waits for a file before timing out.

Default value: 60,000 milliseconds

datatype_icon

id

Unique file identifier the SystemLink file service generates to identify the file on the server.

datatype_icon

error out

Error information.

The node produces this output according to standard error behavior.

Standard Error Behavior

Message Functions

Communicate as a network queue using the publish-subscribe model. Use messages to send commands, status updates, and data between your systems and server.

SystemLink Nodes and Functions Reference for NXG and G Web Development Software

Use SystemLink nodes in NXG or G Web Development Software to communicate data between your test and measurement systems and your SystemLink Server account.

Close

Terminates the message session with the server.

Flush

Clears all queued messages associated with the session from the queue. .

Get Max Queue Size Bytes

Obtains the maximum number of bytes that the queue can hold.

Get Queue Size Bytes

Obtains the current number of bytes for the message queue associated with the session.

Open

Creates a message session with the server.

Publish

Writes a message to a topic, which immediately sends the message to subscribers of the topic.

Read

Returns a message from the queue.

Subscribe

Registers a session to receive messages published to a specific **topic**.

Unsubscribe

Unregisters a session from a specific topic to stop receiving messages.

API Reference

G, HTTP, Python, and .NET APIs allow you to programmatically access SystemLink data services outside of NI SystemLink Web Application.

[SystemLink Nodes and Functions Reference for NXG and G Web Development Software](#)

Use SystemLink nodes in NXG or G Web Development Software to communicate data between your test and measurement systems and your SystemLink Server account.

Close

Terminates the message session with the server.

Inputs/Outputs

datatype_icon

message session

Synchronous message session established with the server.

datatype_icon

error in

Error conditions that occur before this node runs.

Unlike most nodes, this node runs normally even if an error occurs before this node runs. For more information, refer to ***Standard Error Behavior***.

Default value: no error

datatype_icon

configuration

Connection to the server.

datatype_icon

error out

Error information.

The node produces this output according to standard error behavior. For more information, refer to ***Standard Error Behavior***.

Related information:

- [Standard Error Behavior](#)

Flush

Clears all queued messages associated with the session from the queue.

.

Inputs/Outputs

datatype_icon

message session in

Synchronous message session established with the server.

datatype_icon

error in

Error conditions that occur before this node runs.

The node responds to this input according to standard error behavior.

Standard Error Behavior

Default value: No error

datatype_icon

message session out

Synchronous message session established with the server.

datatype_icon

error out

Error information.

The node produces this output according to standard error behavior.

Standard Error Behavior**Get Max Queue Size Bytes**

Obtains the maximum number of bytes that the queue can hold.

When the current queue size plus the size of a message being queued exceeds the maximum queue size, the queue rejects new messages until queue space becomes available.

Inputs/Outputs

datatype_icon

message session in

Synchronous message session established with the server.

datatype_icon

error in

Error conditions that occur before this node runs.

The node responds to this input according to standard error behavior.

Standard Error Behavior

Default value: No error

datatype_icon

message session out

Synchronous message session established with the server.

datatype_icon

max queue size

Total number of bytes the queue can contain before preventing a message from being queued.

datatype_icon

error out

Error information.

The node produces this output according to standard error behavior.

Standard Error Behavior

Get Queue Size Bytes

Obtains the current number of bytes for the message queue associated with the session.

Inputs/Outputs

datatype_icon

message session in

Synchronous message session established with the server.

datatype_icon

error in

Error conditions that occur before this node runs.

The node responds to this input according to standard error behavior.

Standard Error Behavior

Default value: No error

datatype_icon

message session out

Synchronous message session established with the server.

datatype_icon

queue size

Size of the queue in bytes.

datatype_icon

error out

Error information.

The node produces this output according to standard error behavior.

Standard Error Behavior

Open

Creates a message session with the server.

Inputs/Outputs

datatype_icon

configuration

Connection to the server.

If you leave this input unwired, this node uses the default connection. It automatically retrieves your SystemLink Server credentials when you execute the application on a managed system or server.

datatype_icon

error in

Error conditions that occur before this node runs.

The node responds to this input according to standard error behavior.

Standard Error Behavior

Default value: No error

datatype_icon

message session

Synchronous message session established with the server.

datatype_icon

error out

Error information.

The node produces this output according to standard error behavior.

Standard Error Behavior

Programming Patterns

- Sending Messages Between Systems

Publish

Writes a message to a topic, which immediately sends the message to subscribers of the topic.

Subscribers of a topic receive every message published to the topic.

Inputs/Outputs

datatype_icon

message session in

Synchronous message session established with the server.

datatype_icon

topic

Channel to which endpoints publish or subscribe messages.

datatype_icon

message

Text published for subscribers of a topic to read, if a message exists.

datatype_icon

error in

Error conditions that occur before this node runs.

The node responds to this input according to standard error behavior.

Standard Error Behavior

Default value: No error

datatype_icon

message session out

Synchronous message session established with the server.

datatype_icon

error out

Error information.

The node produces this output according to standard error behavior.

Standard Error Behavior**Programming Patterns**

- Sending Messages Between Systems

Read

Returns a message from the queue.

To read a message, you must subscribe to the topic the message is published to. If you do not subscribe to the topic, you cannot receive the message. Therefore, you cannot read it.

If there are no queued messages, the call synchronously blocks up to the timeout specified or to the maximum timeout the web service defines, whichever is smaller. The SystemLink Server times out after 10 seconds.

Inputs/Outputs

datatype_icon

message session in

Synchronous message session established with the server.

datatype_icon

timeout

Time, in milliseconds, that this node waits for an enqueued message before timing out.

Default value: 100

datatype_icon

error in

Error conditions that occur before this node runs.

The node responds to this input according to standard error behavior.

Standard Error Behavior

Default value: No error

datatype_icon

message session out

Synchronous message session established with the server.

datatype_icon

topic

Channel to which endpoints publish or subscribe messages.

Ensure topic names are unique to avoid name collisions.

datatype_icon

message

Text published for subscribers of a topic to read, if a message exists.

datatype_icon

error out

Error information.

The node produces this output according to standard error behavior.

Standard Error Behavior

datatype_icon

dequeue status

Information about the current state of the node.

If True, the node received an incoming message. If False, the node timed out.

Programming Patterns

- Sending Messages Between Systems

Subscribe

Registers a session to receive messages published to a specific **topic**.

Subscribers of a topic receive every message published to it. Wire several instances of this VI in a series to subscribe the multiple topics.

Inputs/Outputs

datatype_icon

message session in

Synchronous message session established with the server.

datatype_icon

topic

Channel to which endpoints publish or subscribe messages.

datatype_icon

error in

Error conditions that occur before this node runs.

The node responds to this input according to standard error behavior.

Standard Error Behavior

Default value: No error

datatype_icon

message session out

Synchronous message session established with the server.

datatype_icon

error out

Error information.

The node produces this output according to standard error behavior.

Standard Error Behavior

Programming Patterns

- Sending Messages Between Systems

Unsubscribe

Unregisters a session from a specific topic to stop receiving messages.

Endpoints continue to receive messages in the queue after you call Unsubscribe, if the application published them prior to you calling Unsubscribe.

Inputs/Outputs

datatype_icon

message session in

Synchronous message session established with the server.

datatype_icon

topic

Channel to which endpoints publish or subscribe messages.

datatype_icon

error in

Error conditions that occur before this node runs.

The node responds to this input according to standard error behavior.

Standard Error Behavior

Default value: No error

datatype_icon

message session out

Synchronous message session established with the server.

datatype_icon

error out

Error information.

The node produces this output according to standard error behavior.

Standard Error Behavior

Tag Functions

Use tags to read and write data values between systems and SystemLink Server.

SystemLink Nodes and Functions Reference for NXG and G Web Development Software

Use SystemLink nodes in NXG or G Web Development Software to communicate data

between your test and measurement systems and your SystemLink Server account.

[Advanced Functions](#)

Read or write multiple times in a single call to improve performance.

[Build Path](#)

Identifies the system and top-level application that generates the tag as a tag path.

[Close Tag](#)

Closes a tag reference or an array of tag references.

[Delete Tag](#)

Removes a tag or group of tags on the server.

[Historian Functions](#)

Retain historical tag values to review, troubleshoot, or analyze data over time.

[Open Tag](#)

Opens a reference to a tag on the server. If no reference of the tag exists on the server, the node creates one.

[Query Tags](#)

Searches the server for tags matching the criteria you specify.

[Read Tag](#)

Returns the value of the tag. The data type of the tag must match the configuration of the function.

[Read Tag Settings](#)

Returns the settings of a tag.

[Reset Aggregates](#)

Clears the aggregate information for a tag reference.

[Write Tag](#)

Specifies the value of the tag. The data type of the tag must match the configuration of the function.

[Write Tag Settings](#)

Specifies the settings of a tag. The node replaces existing tag settings on the server with the new settings.

[API Reference](#)

G, HTTP, Python, and .NET APIs allow you to programmatically access SystemLink data services outside of NI SystemLink Web Application.

[SystemLink Nodes and Functions Reference for NXG and G Web Development Software](#)

Use SystemLink nodes in NXG or G Web Development Software to communicate data between your test and measurement systems and your SystemLink Server account.

Advanced Functions

Read or write multiple times in a single call to improve performance.

Tag Functions

Use tags to read and write data values between systems and SystemLink Server.

Generate Tag Cluster

Creates or adds to an array of tag clusters, which specifies the tag data values for Multi Write to communicate.

Multi Read

Returns the value of an array of tags in a single operation.

Multi Write

Specifies values for multiple tags in a single operation.

API Reference

G, HTTP, Python, and .NET APIs allow you to programmatically access SystemLink data services outside of NI SystemLink Web Application.

SystemLink Nodes and Functions Reference for NXG and G Web Development Software

Use SystemLink nodes in NXG or G Web Development Software to communicate data between your test and measurement systems and your SystemLink Server account.

Tag Functions

Use tags to read and write data values between systems and SystemLink Server.

Generate Tag Cluster

Creates or adds to an array of tag clusters, which specifies the tag data values for Multi Write to communicate.

Advanced Functions

Read or write multiple times in a single call to improve performance.

Boolean

Creates or adds to an array of tag clusters, which specifies the tag data values for Multi Write to communicate.

DBL

Creates or adds to an array of tag clusters, which specifies the tag data values for Multi Write to communicate.

I32

Creates or adds to an array of tag clusters, which specifies the tag data values for Multi Write to communicate.

String

Creates or adds to an array of tag clusters, which specifies the tag data values for Multi Write to communicate.

Timestamp

Creates or adds to an array of tag clusters, which specifies the tag data values for Multi Write to communicate.

U64

Creates or adds to an array of tag clusters, which specifies the tag data values for Multi Write to communicate.

API Reference

G, HTTP, Python, and .NET APIs allow you to programmatically access SystemLink data services outside of NI SystemLink Web Application.

SystemLink Nodes and Functions Reference for NXG and G Web Development Software

Use SystemLink nodes in NXG or G Web Development Software to communicate data between your test and measurement systems and your SystemLink Server account.

Tag Functions

Use tags to read and write data values between systems and SystemLink Server.

Advanced Functions

Read or write multiple times in a single call to improve performance.

Boolean

Creates or adds to an array of tag clusters, which specifies the tag data values for Multi Write to communicate.

The data type of the tag cluster must be a Boolean.

Inputs/Outputs

`datatype_icon`

time stamp

Date and time the server obtained the most recent data value.

Leave the `time stamp` input empty to have the server manage the timestamp.

`datatype_icon`

tag clusters in

Clusters of tags the web service references to communicate data.

datatype_icon

tag in

Tag the server references to communicate measurement data.

datatype_icon

value

Data value of the tag or tags.

datatype_icon

error in

Error conditions that occur before this node runs.

The node responds to this input according to standard error behavior.

[Standard Error Behavior](#)

Default value: No error

datatype_icon

tag clusters out

Clusters of tags the web service references to communicate data.

datatype_icon

tag out

Tag the server references to communicate measurement data.

datatype_icon

error out

Error information.

The node produces this output according to standard error behavior.

Standard Error Behavior**Programming Patterns**

Wire this node to Multi Write to record values for multiple tags in a single operation. This improves performance compared to writing each tag individually.

DBL

Creates or adds to an array of tag clusters, which specifies the tag data values for Multi Write to communicate.

The data type of the tag cluster must be a double-precision, floating-point number.

Inputs/Outputs

`datatype_icon`

time stamp

Date and time the server obtained the most recent data value.

Leave the `time stamp` input empty to have the server manage the timestamp.

`datatype_icon`

tag clusters in

Clusters of tags the web service references to communicate data.

`datatype_icon`

tag in

Tag the server references to communicate measurement data.

datatype_icon

value

Data value of the tag or tags.

datatype_icon

error in

Error conditions that occur before this node runs.

The node responds to this input according to standard error behavior.

Standard Error Behavior

Default value: No error

datatype_icon

tag clusters out

Clusters of tags the web service references to communicate data.

datatype_icon

tag out

Tag the server references to communicate measurement data.

datatype_icon

error out

Error information.

The node produces this output according to standard error behavior.

Standard Error Behavior

Programming Patterns

Wire this node to Multi Write to record values for multiple tags in a single operation. This improves performance compared to writing each tag individually.

I32

Creates or adds to an array of tag clusters, which specifies the tag data values for Multi Write to communicate.

The data type of the tag cluster must be a 32-bit integer.

Inputs/Outputs

`datatype_icon`

time stamp

Date and time the server obtained the most recent data value.

Leave the `time stamp` input empty to have the server manage the timestamp.

`datatype_icon`

tag clusters in

Clusters of tags the web service references to communicate data.

`datatype_icon`

tag in

Tag the server references to communicate measurement data.

`datatype_icon`

value

Data value of the tag or tags.

datatype_icon

error in

Error conditions that occur before this node runs.

The node responds to this input according to standard error behavior.

Standard Error Behavior

Default value: No error

datatype_icon

tag clusters out

Clusters of tags the web service references to communicate data.

datatype_icon

tag out

Tag the server references to communicate measurement data.

datatype_icon

error out

Error information.

The node produces this output according to standard error behavior.

Standard Error Behavior

Programming Patterns

Wire this node to Multi Write to record values for multiple tags in a single operation. This improves performance compared to writing each tag individually.

String

Creates or adds to an array of tag clusters, which specifies the tag data values for Multi Write to communicate.

The data type of the tag cluster must be a string.

Inputs/Outputs

`datatype_icon`

time stamp

Date and time the server obtained the most recent data value.

Leave the `time stamp` input empty to have the server manage the timestamp.

`datatype_icon`

tag clusters in

Clusters of tags the web service references to communicate data.

`datatype_icon`

tag in

Tag the server references to communicate measurement data.

`datatype_icon`

value

Data value of the tag or tags.

datatype_icon

error in

Error conditions that occur before this node runs.

The node responds to this input according to standard error behavior.

Standard Error Behavior

Default value: No error

datatype_icon

tag clusters out

Clusters of tags the web service references to communicate data.

datatype_icon

tag out

Tag the server references to communicate measurement data.

datatype_icon

error out

Error information.

The node produces this output according to standard error behavior.

Standard Error Behavior

Programming Patterns

Wire this node to Multi Write to record values for multiple tags in a single operation. This improves performance compared to writing each tag individually.

Timestamp

Creates or adds to an array of tag clusters, which specifies the tag data values for Multi Write to communicate.

The data type of the tag cluster must be a timestamp.

Inputs/Outputs

`datatype_icon`

time stamp

Date and time the server obtained the most recent data value.

Leave the `time stamp` input empty to have the server manage the timestamp.

`datatype_icon`

tag clusters in

Clusters of tags the web service references to communicate data.

`datatype_icon`

tag in

Tag the server references to communicate measurement data.

`datatype_icon`

value

Data value of the tag or tags.

`datatype_icon`

error in

Error conditions that occur before this node runs.

The node responds to this input according to standard error behavior.

Standard Error Behavior

Default value: No error

datatype_icon

tag clusters out

Clusters of tags the web service references to communicate data.

datatype_icon

tag out

Tag the server references to communicate measurement data.

datatype_icon

error out

Error information.

The node produces this output according to standard error behavior.

Standard Error Behavior

Programming Patterns

Wire this node to Multi Write to record values for multiple tags in a single operation. This improves performance compared to writing each tag individually.

U64

Creates or adds to an array of tag clusters, which specifies the tag data values for Multi Write to communicate.

The data type of the tag cluster must be an unsigned 64-bit integer.

Inputs/Outputs

`datatype_icon`

time stamp

Date and time the server obtained the most recent data value.

Leave the `time stamp` input empty to have the server manage the timestamp.

`datatype_icon`

tag clusters in

Clusters of tags the web service references to communicate data.

`datatype_icon`

tag in

Tag the server references to communicate measurement data.

`datatype_icon`

value

Data value of the tag or tags.

`datatype_icon`

error in

Error conditions that occur before this node runs.

The node responds to this input according to standard error behavior.

Standard Error Behavior

Default value: No error

`datatype_icon`

tag clusters out

Clusters of tags the web service references to communicate data.

datatype_icon

tag out

Tag the server references to communicate measurement data.

datatype_icon

error out

Error information.

The node produces this output according to standard error behavior.

Standard Error Behavior**Programming Patterns**

Wire this node to Multi Write to record values for multiple tags in a single operation. This improves performance compared to writing each tag individually.

Multi Read

Returns the value of an array of tags in a single operation.

The tags may be different data types.

Inputs/Outputs

datatype_icon

tags in

Tags the server references to communicate measurement data.

datatype_icon

error in

Error conditions that occur before this node runs.

The node responds to this input according to standard error behavior.

Standard Error Behavior

Default value: No error

datatype_icon

tags out

Tags the server references to communicate measurement data.

datatype_icon

tag clusters

Clusters of tags the web service references to communicate data.

datatype_icon

error out

Error information.

The node produces this output according to standard error behavior.

Standard Error Behavior**Multi Write**

Specifies values for multiple tags in a single operation.

The tags may be of different data types.

Inputs/Outputs

`datatype_icon`

tag clusters

Clusters of tags the web service references to communicate data.

`datatype_icon`

error in

Error conditions that occur before this node runs.

The node responds to this input according to standard error behavior.

Standard Error Behavior

Default value: No error

`datatype_icon`

error out

Error information.

The node produces this output according to standard error behavior.

Standard Error Behavior

Programming Patterns

Wire this node to Generate Tag Cluster to build the `tag_clusters` input.

Build Path

Identifies the system and top-level application that generates the tag as a tag path.

If you leave `namespace` unwired, it defaults to the name of the top-level VI.

Inputs/Outputs

datatype_icon

tag name

Name of the tag whose path you want to build.

datatype_icon

namespace

Namespace containing the tag.

If you leave this input unwired, Build Path uses the name of the top-level VI.

datatype_icon

path

Path to the tag reference.

Close Tag

Closes a tag reference or an array of tag references.

Tag Functions

Use tags to read and write data values between systems and SystemLink Server.

Multiple

Closes an array of tag references.

Single

Closes a tag reference.

API Reference

G, HTTP, Python, and .NET APIs allow you to programmatically access SystemLink data services outside of NI SystemLink Web Application.

SystemLink Nodes and Functions Reference for NXG and G Web Development Software

Use SystemLink nodes in NXG or G Web Development Software to communicate data between your test and measurement systems and your SystemLink Server account.

Tag Functions

Use tags to read and write data values between systems and SystemLink Server.

Multiple

Closes an array of tag references.

Inputs/Outputs

datatype_icon

tags

Tags the server references to communicate measurement data.

datatype_icon

error in

Error conditions that occur before this node runs.

Unlike most nodes, this node runs normally even if an error occurs before this node runs. For more information, refer to ***Standard Error Behavior***.

Default value: no error

datatype_icon

error out

Error information.

The node produces this output according to standard error behavior. For more information, refer to ***Standard Error Behavior***.

Programming Patterns

- Transferring Data Using Tags

Related information:

- [Standard Error Behavior](#)

Single

Closes a tag reference.

Inputs/Outputs

datatype_icon

tag

Tag the server references to communicate measurement data.

datatype_icon

error in

Error conditions that occur before this node runs.

Unlike most nodes, this node runs normally even if an error occurs before this node runs. For more information, refer to ***Standard Error Behavior***.

Default value: no error

datatype_icon

error out

Error information.

The node produces this output according to standard error behavior. For more information, refer to ***Standard Error Behavior***.

Programming Patterns

- Transferring Data Using Tags

Related information:

- [Standard Error Behavior](#)

Delete Tag

Removes a tag or group of tags on the server.

[Tag Functions](#)

Use tags to read and write data values between systems and SystemLink Server.

[Multiple](#)

Removes a group of tags on the server.

[Single](#)

Removes a tag on the server.

[API Reference](#)

G, HTTP, Python, and .NET APIs allow you to programmatically access SystemLink data services outside of NI SystemLink Web Application.

[SystemLink Nodes and Functions Reference for NXG and G Web Development Software](#)

Use SystemLink nodes in NXG or G Web Development Software to communicate data between your test and measurement systems and your SystemLink Server account.

[Tag Functions](#)

Use tags to read and write data values between systems and SystemLink Server.

Multiple

Removes a group of tags on the server.

Inputs/Outputs

`datatype_icon`

tags

Tags the server references to communicate measurement data.

`datatype_icon`

error in

Error conditions that occur before this node runs.

The node responds to this input according to standard error behavior.

Standard Error Behavior

Default value: No error

datatype_icon

error out

Error information.

The node produces this output according to standard error behavior.

Standard Error Behavior

Single

Removes a tag on the server.

Inputs/Outputs

datatype_icon

tag

Tag the server references to communicate measurement data.

datatype_icon

error in

Error conditions that occur before this node runs.

The node responds to this input according to standard error behavior.

Standard Error Behavior

Default value: No error

`datatype_icon`

error out

Error information.

The node produces this output according to standard error behavior.

[Standard Error Behavior](#)

Historian Functions

Retain historical tag values to review, troubleshoot, or analyze data over time.

[Tag Functions](#)

Use tags to read and write data values between systems and SystemLink Server.

[Configure Tag History Window](#)

Defines the range of historical tag values to read with Read Tag History or Multi Read Tag History.

[Configure Retention](#)

Defines how long a tag retains historical values.

[Multi Read History](#)

Returns the historical values from multiple tags in a single operation.

[Read Tag History](#)

Returns the historical values from a single tag. The data type of the tag must match the polymorphic instance.

[API Reference](#)

G, HTTP, Python, and .NET APIs allow you to programmatically access SystemLink data services outside of NI SystemLink Web Application.

[SystemLink Nodes and Functions Reference for NXG and G Web Development Software](#)

Use SystemLink nodes in NXG or G Web Development Software to communicate data between your test and measurement systems and your SystemLink Server account.

[Tag Functions](#)

Use tags to read and write data values between systems and SystemLink Server.

Configure Tag History Window

Defines the range of historical tag values to read with Read Tag History or Multi Read

Tag History.

Historian Functions

Retain historical tag values to review, troubleshoot, or analyze data over time.

Count

Defines the range of historical tag values to read with Read Tag History or Multi Read Tag History.

Decimated

Defines the range of historical tag values to read with Read Tag History or Multi Read Tag History and decimates at set time intervals.

API Reference

G, HTTP, Python, and .NET APIs allow you to programmatically access SystemLink data services outside of NI SystemLink Web Application.

SystemLink Nodes and Functions Reference for NXG and G Web Development Software

Use SystemLink nodes in NXG or G Web Development Software to communicate data between your test and measurement systems and your SystemLink Server account.

Tag Functions

Use tags to read and write data values between systems and SystemLink Server.

Historian Functions

Retain historical tag values to review, troubleshoot, or analyze data over time.

Count

Defines the range of historical tag values to read with Read Tag History or Multi Read Tag History.

The default is to read the oldest 1,000 historical values.

Inputs/Outputs

`datatype_icon`

take

Number of historical values requested from the server.

If this input is empty, Read Tag History returns the first 1,000 results by default.

datatype_icon

start

The time that limits the returned historical values to those written at or after the given time.

datatype_icon

end

The time that limits the returned historical values to those written at or before the given time.

datatype_icon

newest first?

A Boolean that determines whether to return the newest or oldest historical tag values first.

The default is False, which returns the oldest historical values first.

datatype_icon

history window

The configured time range to read historical tag values.

Wire `history window` to Read Tag History or Multi Read History to define the historical values to return.

Decimated

Defines the range of historical tag values to read with Read Tag History or Multi Read Tag History and decimates at set time intervals.

Inputs/Outputs

`datatype_icon`

decimated

The approximate number of intervals to divide the historical tag values in the set time range.

`datatype_icon`

start

The time that limits the returned historical values to those written at or after the given time.

`datatype_icon`

end

The time that limits the returned historical values to those written at or before the given time.

`datatype_icon`

newest first?

A Boolean that determines whether to return the newest or oldest historical tag values first.

The default is False, which returns the oldest historical values first.

`datatype_icon`

history window

The configured time range to read historical tag values.

Wire `history window` to Read Tag History or Multi Read History to define the historical values to return.

Configure Retention

Defines how long a tag retains historical values.

Historian Functions

Retain historical tag values to review, troubleshoot, or analyze data over time.

Count

Creates a retention policy to specify the maximum number of tags stored in tag history.

Duration

Creates a retention policy for historical tag values to expire after a set number of days they were written.

None

Creates a retention policy to never retain historical tag values.

Permanent

Creates a retention policy to retain all historical tag values.

API Reference

G, HTTP, Python, and .NET APIs allow you to programmatically access SystemLink data services outside of NI SystemLink Web Application.

SystemLink Nodes and Functions Reference for NXG and G Web Development Software

Use SystemLink nodes in NXG or G Web Development Software to communicate data between your test and measurement systems and your SystemLink Server account.

Tag Functions

Use tags to read and write data values between systems and SystemLink Server.

Historian Functions

Retain historical tag values to review, troubleshoot, or analyze data over time.

Count

Creates a retention policy to specify the maximum number of tags stored in tag history.

Inputs/Outputs

`datatype_icon`

tag in

Reference to a tag on the server.

Use Open Tag or Query Tags to obtain this reference.

datatype_icon

maximum count

The number of historical tag values to retain.

The default is 10,000.

datatype_icon

error in

Error conditions that occur before this node runs.

The node responds to this input according to standard error behavior.

Standard Error Behavior

Default value: No error

datatype_icon

tag out

Reference to a specific tag on SystemLink Server.

datatype_icon

error out

Error information.

The node produces this output according to standard error behavior.

Standard Error Behavior

Duration

Creates a retention policy for historical tag values to expire after a set number of days they were written.

Inputs/Outputs

datatype_icon

tag in

Reference to a tag on the server.

Use Open Tag or Query Tags to obtain this reference.

datatype_icon

days

The number of days tag values stay in the tag history before they expire.

The default is 30 days.

datatype_icon

error in

Error conditions that occur before this node runs.

The node responds to this input according to standard error behavior.

Standard Error Behavior

Default value: No error

datatype_icon

tag out

Reference to a specific tag on SystemLink Server.

datatype_icon

error out

Error information.

The node produces this output according to standard error behavior.

Standard Error Behavior

None

Creates a retention policy to never retain historical tag values.

Inputs/Outputs

datatype_icon

tag in

Reference to a tag on the server.

Use Open Tag or Query Tags to obtain this reference.

datatype_icon

error in

Error conditions that occur before this node runs.

The node responds to this input according to standard error behavior.

Standard Error Behavior

Default value: No error

datatype_icon

tag out

Reference to a specific tag on SystemLink Server.

datatype_icon

error out

Error information.

The node produces this output according to standard error behavior.

Standard Error Behavior

Permanent

Creates a retention policy to retain all historical tag values.



Note Any tag that uses this retention policy may use a large amount of storage space on SystemLink Server.

Inputs/Outputs

datatype_icon

tag in

Reference to a tag on the server.

Use Open Tag or Query Tags to obtain this reference.

datatype_icon

error in

Error conditions that occur before this node runs.

The node responds to this input according to standard error behavior.

Standard Error Behavior

Default value: No error

datatype_icon

tag out

Reference to a specific tag on SystemLink Server.

datatype_icon

error in

Error conditions that occur before this node runs.

The node responds to this input according to standard error behavior.

Standard Error Behavior

Default value: No error

Multi Read History

Returns the historical values from multiple tags in a single operation.

Inputs/Outputs

datatype_icon

tags in

Reference to a specific tag or a group of tags on SystemLink Server.

Use Open Tag or Query Tags to obtain this reference.

datatype_icon

window

Time range you specify to read historical tag values.

The default is set to read all values decimated to 1,000.

datatype_icon

error in

Error conditions that occur before this node runs.

The node responds to this input according to standard error behavior.

Standard Error Behavior

Default value: No error

datatype_icon

next window

Next page of historical tag data, if available.

datatype_icon

tags out

Reference to a specific tag or a group of tags on SystemLink Server.

Use Open Tag or Query Tags to obtain this reference.

datatype_icon

values

Data values of the tag history.

datatype_icon

end of data?

Boolean that indicates whether the node read the last of the data for the tag.

datatype_icon

error out

Error information.

The node produces this output according to standard error behavior.

Standard Error Behavior

Read Tag History

Returns the historical values from a single tag. The data type of the tag must match the polymorphic instance.

Historian Functions

Retain historical tag values to review, troubleshoot, or analyze data over time.

Boolean

Returns the historical tags of a single tag. The data type of the tag must be a Boolean.

DBL

Returns the historical values of a single tag. The data type of the tag must be a double-precision, floating-point number.

I32

Returns the historical values from a single tag. The data type of the tag must be a 32-bit integer.

String

Returns the historical values from a single tag. The data type of the tag must be a string.

Timestamp

Returns the historical values of a single tag. The data type of the tag must be a timestamp.

U64

Returns the historical values from a single tag. The data type of the tag must be an unsigned 64-bit integer.

API Reference

G, HTTP, Python, and .NET APIs allow you to programmatically access SystemLink data services outside of NI SystemLink Web Application.

SystemLink Nodes and Functions Reference for NXG and G Web Development Software

Use SystemLink nodes in NXG or G Web Development Software to communicate data between your test and measurement systems and your SystemLink Server account.

Tag Functions

Use tags to read and write data values between systems and SystemLink Server.

Historian Functions

Retain historical tag values to review, troubleshoot, or analyze data over time.

Boolean

Returns the historical tags of a single tag. The data type of the tag must be a Boolean.

Inputs/Outputs

`datatype_icon`

tag in

Reference to a tag on the server.

Use Open Tag or Query Tags to obtain this reference.

`datatype_icon`

window

Time range you specify to read historical tag values.

The default is set to read all values decimated to 1,000.

`datatype_icon`

error in

Error conditions that occur before this node runs.

The node responds to this input according to standard error behavior.

Standard Error Behavior

Default value: No error

datatype_icon

next window

Next page of historical tag data, if available.

datatype_icon

tag out

Reference to a specific tag on SystemLink Server.

datatype_icon

values

Data value of the tag or tags.

datatype_icon

time stamps

Dates and times the server obtained the most recent historical data values.

datatype_icon

error out

Error information.

The node produces this output according to standard error behavior.

Standard Error Behavior

datatype_icon

end of data?

Boolean that indicates whether the node read the last of the data for the tag.

DBL

Returns the historical values of a single tag. The data type of the tag must be a double-precision, floating-point number.

Inputs/Outputs

`datatype_icon`

tag in

Reference to a tag on the server.

Use Open Tag or Query Tags to obtain this reference.

`datatype_icon`

window

Time range you specify to read historical tag values.

The default is set to read all values decimated to 1,000.

`datatype_icon`

error in

Error conditions that occur before this node runs.

The node responds to this input according to standard error behavior.

Standard Error Behavior

Default value: No error

`datatype_icon`

next window

Next page of historical tag data, if available.

datatype_icon

tag out

Reference to a specific tag on SystemLink Server.

datatype_icon

values

Data value of the tag or tags.

datatype_icon

time stamps

Dates and times the server obtained the most recent historical data values.

datatype_icon

error out

Error information.

The node produces this output according to standard error behavior.

Standard Error Behavior

datatype_icon

end of data?

Boolean that indicates whether the node read the last of the data for the tag.

I32

Returns the historical values from a single tag. The data type of the tag must be a 32-bit integer.

Inputs/Outputs

datatype_icon

tag in

Reference to a tag on the server.

Use Open Tag or Query Tags to obtain this reference.

datatype_icon

window

Time range you specify to read historical tag values.

The default is set to read all values decimated to 1,000.

datatype_icon

error in

Error conditions that occur before this node runs.

The node responds to this input according to standard error behavior.

Standard Error Behavior

Default value: No error

datatype_icon

next window

Next page of historical tag data, if available.

datatype_icon

tag out

Reference to a specific tag on SystemLink Server.

datatype_icon

values

Data value of the tag or tags.

datatype_icon

time stamps

Dates and times the server obtained the most recent historical data values.

datatype_icon

error out

Error information.

The node produces this output according to standard error behavior.

Standard Error Behavior

datatype_icon

end of data?

Boolean that indicates whether the node read the last of the data for the tag.

String

Returns the historical values from a single tag. The data type of the tag must be a string.

Inputs/Outputs

datatype_icon

tag in

Reference to a tag on the server.

Use Open Tag or Query Tags to obtain this reference.

datatype_icon

window

Time range you specify to read historical tag values.

The default is set to read all values decimated to 1,000.

datatype_icon

error in

Error conditions that occur before this node runs.

The node responds to this input according to standard error behavior.

Standard Error Behavior

Default value: No error

datatype_icon

next window

Next page of historical tag data, if available.

datatype_icon

tag out

Reference to a specific tag on SystemLink Server.

datatype_icon

values

Data value of the tag or tags.

datatype_icon

time stamps

Dates and times the server obtained the most recent historical data values.

datatype_icon

error out

Error information.

The node produces this output according to standard error behavior.

Standard Error Behavior

datatype_icon

end of data?

Boolean that indicates whether the node read the last of the data for the tag.

Timestamp

Returns the historical values of a single tag. The data type of the tag must be a timestamp.

Inputs/Outputs

datatype_icon

tag in

Reference to a tag on the server.

Use Open Tag or Query Tags to obtain this reference.

datatype_icon

window

Time range you specify to read historical tag values.

The default is set to read all values decimated to 1,000.

datatype_icon

error in

Error conditions that occur before this node runs.

The node responds to this input according to standard error behavior.

Standard Error Behavior

Default value: No error

datatype_icon

next window

Next page of historical tag data, if available.

datatype_icon

tag out

Reference to a specific tag on SystemLink Server.

datatype_icon

values

Data value of the tag or tags.

datatype_icon

time stamps

Dates and times the server obtained the most recent historical data values.

datatype_icon

error out

Error information.

The node produces this output according to standard error behavior.

Standard Error Behavior

datatype_icon

end of data?

Boolean that indicates whether the node read the last of the data for the tag.

U64

Returns the historical values from a single tag. The data type of the tag must be an unsigned 64-bit integer.

Inputs/Outputs

datatype_icon

tag in

Reference to a tag on the server.

Use Open Tag or Query Tags to obtain this reference.

datatype_icon

window

Time range you specify to read historical tag values.

The default is set to read all values decimated to 1,000.

datatype_icon

error in

Error conditions that occur before this node runs.

The node responds to this input according to standard error behavior.

Standard Error Behavior

Default value: No error

datatype_icon

next window

Next page of historical tag data, if available.

datatype_icon

tag out

Reference to a specific tag on SystemLink Server.

datatype_icon

values

Data value of the tag or tags.

datatype_icon

time stamps

Dates and times the server obtained the most recent historical data values.

datatype_icon

error out

Error information.

The node produces this output according to standard error behavior.

Standard Error Behavior

datatype_icon

end of data?

Boolean that indicates whether the node read the last of the data for the tag.

Open Tag

Opens a reference to a tag on the server. If no reference of the tag exists on the server, the node creates one.

Inputs/Outputs

datatype_icon

configuration

Connection to the server.

If you leave this input unwired, this node uses the default connection. It automatically retrieves your SystemLink Server credentials when you execute the application on a managed system or server.

datatype_icon

path

Path to a tag reference.



Note Tag paths do not support spaces. If an illegal character is present, the node returns an error.

datatype_icon

data type

Data type of the tag reference.

If a tag with the same path but different data type already exists, Open Tag returns an error. Use Query Tags to open existing tag references of unknown data types.

datatype_icon

error in

Error conditions that occur before this node runs.

The node responds to this input according to standard error behavior.

Standard Error Behavior

Default value: No error

datatype_icon

tag

Tag the server references to communicate measurement data.

datatype_icon

error out

Error information.

The node produces this output according to standard error behavior.

Standard Error Behavior

Programming Patterns

- Transferring Data Using Tags

Using an Asterisk in a Tag Path

To set tag permissions for a group of tags or query a collection of tags, use an asterisk to match any sequence of zero or more characters in a tag path. For example, to return every tag that begins with `example.path` on the web server, enter

`example.path*` in to the `path` input.

Query Tags

Searches the server for tags matching the criteria you specify.

If an input is empty, this node opens references to every tag on the server.

Inputs/Outputs

`datatype_icon`

configuration

Connection to the server.

If you leave this input unwired, this node uses the default connection. It automatically retrieves your SystemLink Server credentials when you execute the application on a managed system or server.

`datatype_icon`

paths

Paths to tag references.



Note Tag paths do not support spaces. If an illegal character is present, the node returns an error.

`datatype_icon`

keywords

Specific words provided to find tags.

`datatype_icon`

properties

Key-value pairs that exist as metadata on a tag.

datatype_icon

limit to configured workspace?

Boolean that determines whether this node only queries for tags in the workspace you specify in `configuration`.

By default, the node queries for tags from all accessible workspaces.

datatype_icon

error in

Error conditions that occur before this node runs.

The node responds to this input according to standard error behavior.

Standard Error Behavior

Default value: No error

datatype_icon

skip

Number of tags skipped to limit the amount returned from the server.

If the input is left empty, the query returns the first 1000 results by default.

datatype_icon

take

Number of tags requested from the server.

If the input is left empty, the query returns the first 1000 results by default.

datatype_icon

tags

Tags the server references to communicate measurement data.

datatype_icon

total count

Total number of tags on the server.

datatype_icon

error out

Error information.

The node produces this output according to standard error behavior.

Standard Error Behavior

Read Tag

Returns the value of the tag. The data type of the tag must match the configuration of the function.

Tag Functions

Use tags to read and write data values between systems and SystemLink Server.

Boolean

Returns the value of the tag.

DBL

Returns the value of the tag.

I32

Returns the value of the tag.

String

Returns the value of the tag.

Timestamp

Returns the value of the tag.

U64

Returns the value of the tag.

API Reference

G, HTTP, Python, and .NET APIs allow you to programmatically access SystemLink data services outside of NI SystemLink Web Application.

SystemLink Nodes and Functions Reference for NXG and G Web Development Software

Use SystemLink nodes in NXG or G Web Development Software to communicate data between your test and measurement systems and your SystemLink Server account.

Tag Functions

Use tags to read and write data values between systems and SystemLink Server.

Boolean

Returns the value of the tag.

The data type of the tag must be a Boolean.

Inputs/Outputs

datatype_icon

tag in

Tag the server references to communicate measurement data.

datatype_icon

error in

Error conditions that occur before this node runs.

The node responds to this input according to standard error behavior.

Standard Error Behavior

Default value: No error

datatype_icon

count

Total number of times a system or group of systems writes to a tag.

datatype_icon

tag out

Tag the server references to communicate measurement data.

datatype_icon

value

Data value of the tag or tags.

datatype_icon

time stamp

Date and time the server obtained the most recent data value.

datatype_icon

error out

Error information.

The node produces this output according to standard error behavior.

[Standard Error Behavior](#)

Programming Patterns

- Transferring Data Using Tags

DBL

Returns the value of the tag.

The data type of the tag must be a double-precision, floating-point number.

Inputs/Outputs

datatype_icon

tag in

Tag the server references to communicate measurement data.

datatype_icon

error in

Error conditions that occur before this node runs.

The node responds to this input according to standard error behavior.

Standard Error Behavior

Default value: No error

datatype_icon

aggregates

Values the tag collects since calling Reset Aggregates.

If a tag is set to collect aggregates, every write to the tag updates the aggregate of the tag.

datatype_icon

tag out

Tag the server references to communicate measurement data.

datatype_icon

value

Data value of the tag or tags.

datatype_icon

time stamp

Date and time the server obtained the most recent data value.

datatype_icon

error out

Error information.

The node produces this output according to standard error behavior.

Standard Error Behavior**Programming Patterns**

- Transferring Data Using Tags

I32

Returns the value of the tag.

The data type of the tag must be a 32-bit integer.

Inputs/Outputs

datatype_icon

tag in

Tag the server references to communicate measurement data.

datatype_icon

error in

Error conditions that occur before this node runs.

The node responds to this input according to standard error behavior.

Standard Error Behavior

Default value: No error

datatype_icon

aggregates

Values the tag collects since calling Reset Aggregates.

If a tag is set to collect aggregates, every write to the tag updates the aggregate of the tag.

datatype_icon

tag out

Tag the server references to communicate measurement data.

datatype_icon

value

Data value of the tag or tags.

datatype_icon

time stamp

Date and time the server obtained the most recent data value.

datatype_icon

error out

Error information.

The node produces this output according to standard error behavior.

Standard Error Behavior

Programming Patterns

- Transferring Data Using Tags

String

Returns the value of the tag.

The data type of the tag must be a string.

Inputs/Outputs

datatype_icon

tag in

Tag the server references to communicate measurement data.

datatype_icon

error in

Error conditions that occur before this node runs.

The node responds to this input according to standard error behavior.

Standard Error Behavior

Default value: No error

datatype_icon

count

Total number of times a system or group of systems writes to a tag.

datatype_icon

tag out

Tag the server references to communicate measurement data.

datatype_icon

value

Data value of the tag or tags.

datatype_icon

time stamp

Date and time the server obtained the most recent data value.

datatype_icon

error out

Error information.

The node produces this output according to standard error behavior.

[Standard Error Behavior](#)**Programming Patterns**

- Transferring Data Using Tags

Timestamp

Returns the value of the tag.

The data type of the tag must be a timestamp.

Inputs/Outputs

datatype_icon

tag in

Tag the server references to communicate measurement data.

datatype_icon

error in

Error conditions that occur before this node runs.

The node responds to this input according to standard error behavior.

Standard Error Behavior

Default value: No error

datatype_icon

count

Total number of times a system or group of systems writes to a tag.

datatype_icon

tag out

Tag the server references to communicate measurement data.

datatype_icon

value

Data value of the tag or tags.

datatype_icon

time stamp

Date and time the server obtained the most recent data value.

datatype_icon

error out

Error information.

The node produces this output according to standard error behavior.

Standard Error Behavior**Programming Patterns**

- Transferring Data Using Tags

U64

Returns the value of the tag.

The data type of the tag must be an unsigned 64-bit integer.

Inputs/Outputs

datatype_icon

tag in

Tag the server references to communicate measurement data.

datatype_icon

error in

Error conditions that occur before this node runs.

The node responds to this input according to standard error behavior.

Standard Error Behavior

Default value: No error

datatype_icon

aggregates

Values the tag collects since calling Reset Aggregates.

If a tag is set to collect aggregates, every write to the tag updates the aggregate of the tag.

datatype_icon

tag out

Tag the server references to communicate measurement data.

datatype_icon

value

Data value of the tag or tags.

datatype_icon

time stamp

Date and time the server obtained the most recent data value.

datatype_icon

error out

Error information.

The node produces this output according to standard error behavior.

Standard Error Behavior**Programming Patterns**

- Transferring Data Using Tags

Read Tag Settings

Returns the settings of a tag.

A tag setting is a type of permission for the tag, such as read, write, modify, delete, and create. Based on the settings, a tag can complete any of the permissions an administrator specifies.

Inputs/Outputs

datatype_icon

tag in

Tag the server references to communicate measurement data.

datatype_icon

error in

Error conditions that occur before this node runs.

The node responds to this input according to standard error behavior.

Standard Error Behavior

Default value: No error

datatype_icon

retention type

Aging policy which the tag server uses.

datatype_icon

tag out

Tag the server references to communicate measurement data.

datatype_icon

properties

Key-value pairs that exist as metadata on a tag.

datatype_icon

keywords

Specific words provided to find tags.

datatype_icon

error out

Error information.

The node produces this output according to standard error behavior.

Standard Error Behavior

datatype_icon

collect aggregates?

Boolean that determines if the node collects aggregate tag data.

Reset Aggregates

Clears the aggregate information for a tag reference.

Tag Functions

Use tags to read and write data values between systems and SystemLink Server.

Multiple

Clears the aggregate information for an array of tag references.

Single

Clears the aggregate information for a tag reference.

API Reference

G, HTTP, Python, and .NET APIs allow you to programmatically access SystemLink data

services outside of NI SystemLink Web Application.

[SystemLink Nodes and Functions Reference for NXG and G Web Development Software](#)

Use SystemLink nodes in NXG or G Web Development Software to communicate data between your test and measurement systems and your SystemLink Server account.

[Tag Functions](#)

Use tags to read and write data values between systems and SystemLink Server.

Multiple

Clears the aggregate information for an array of tag references.

Inputs/Outputs

datatype_icon

tags in

Tags the server references to communicate measurement data.

datatype_icon

error in

Error conditions that occur before this node runs.

The node responds to this input according to standard error behavior.

[Standard Error Behavior](#)

Default value: No error

datatype_icon

tags out

Tags the server references to communicate measurement data.

datatype_icon

error out

Error information.

The node produces this output according to standard error behavior.

Standard Error Behavior**Single**

Clears the aggregate information for a tag reference.

Inputs/Outputs

datatype_icon

tag in

Tag the server references to communicate measurement data.

datatype_icon

error in

Error conditions that occur before this node runs.

The node responds to this input according to standard error behavior.

Standard Error Behavior

Default value: No error

datatype_icon

tag out

Tag the server references to communicate measurement data.

datatype_icon

error out

Error information.

The node produces this output according to standard error behavior.

Standard Error Behavior

Write Tag

Specifies the value of the tag. The data type of the tag must match the configuration of the function.

Tag Functions

Use tags to read and write data values between systems and SystemLink Server.

Boolean

Specifies the value of the tag.

DBL

Specifies the value of the tag.

I32

Specifies the value of the tag.

String

Specifies the value of the tag.

Timestamp

Specifies the value of the tag.

U64

Specifies the value of the tag.

API Reference

G, HTTP, Python, and .NET APIs allow you to programmatically access SystemLink data services outside of NI SystemLink Web Application.

SystemLink Nodes and Functions Reference for NXG and G Web Development Software

Use SystemLink nodes in NXG or G Web Development Software to communicate data between your test and measurement systems and your SystemLink Server account.

Tag Functions

Use tags to read and write data values between systems and SystemLink Server.

Boolean

Specifies the value of the tag.

The data type of the tag must be a Boolean.

Inputs/Outputs

`datatype_icon`

tag in

Tag the server references to communicate measurement data.

`datatype_icon`

value

Data value of the tag or tags.

`datatype_icon`

time stamp

Date and time the server obtained the most recent data value.

Leave the `time stamp` input empty to have the server manage the timestamp.

`datatype_icon`

error in

Error conditions that occur before this node runs.

The node responds to this input according to standard error behavior.

Standard Error Behavior

Default value: No error

`datatype_icon`

tag out

Tag the server references to communicate measurement data.

datatype_icon

error out

Error information.

The node produces this output according to standard error behavior.

Standard Error Behavior**Programming Patterns**

Transferring Data Using Tags

DBL

Specifies the value of the tag.

The data type of the tag must be a double-precision, floating-point number.

Inputs/Outputs

datatype_icon

tag in

Tag the server references to communicate measurement data.

datatype_icon

value

Data value of the tag or tags.

datatype_icon

time stamp

Date and time the server obtained the most recent data value.

Leave the `time stamp` input empty to have the server manage the timestamp.

`datatype_icon`

error in

Error conditions that occur before this node runs.

The node responds to this input according to standard error behavior.

Standard Error Behavior

Default value: No error

`datatype_icon`

tag out

Tag the server references to communicate measurement data.

`datatype_icon`

error out

Error information.

The node produces this output according to standard error behavior.

Standard Error Behavior**Programming Patterns**

Transferring Data Using Tags

132

Specifies the value of the tag.

The data type of the tag must be a 32-bit integer.

Inputs/Outputs

`datatype_icon`

tag in

Tag the server references to communicate measurement data.

`datatype_icon`

value

Data value of the tag or tags.

`datatype_icon`

time stamp

Date and time the server obtained the most recent data value.

Leave the `time stamp` input empty to have the server manage the timestamp.

`datatype_icon`

error in

Error conditions that occur before this node runs.

The node responds to this input according to standard error behavior.

Standard Error Behavior

Default value: No error

`datatype_icon`

tag out

Tag the server references to communicate measurement data.

datatype_icon

error out

Error information.

The node produces this output according to standard error behavior.

Standard Error Behavior

Programming Patterns

Transferring Data Using Tags

String

Specifies the value of the tag.

The data type of the tag must be a string.

Inputs/Outputs

datatype_icon

tag in

Tag the server references to communicate measurement data.

datatype_icon

value

Data value of the tag or tags.

datatype_icon

time stamp

Date and time the server obtained the most recent data value.

Leave the `time stamp` input empty to have the server manage the timestamp.
`datatype_icon`

error in

Error conditions that occur before this node runs.

The node responds to this input according to standard error behavior.

Standard Error Behavior

Default value: No error

`datatype_icon`

tag out

Tag the server references to communicate measurement data.

`datatype_icon`

error out

Error information.

The node produces this output according to standard error behavior.

Standard Error Behavior

Programming Patterns

Transferring Data Using Tags

Timestamp

Specifies the value of the tag.

The data type of the tag must be a timestamp.

Inputs/Outputs

datatype_icon

tag in

Tag the server references to communicate measurement data.

datatype_icon

value

Data value of the tag or tags.

datatype_icon

time stamp

Date and time the server obtained the most recent data value.

Leave the `time stamp` input empty to have the server manage the timestamp.

datatype_icon

error in

Error conditions that occur before this node runs.

The node responds to this input according to standard error behavior.

Standard Error Behavior

Default value: No error

datatype_icon

tag out

Tag the server references to communicate measurement data.

datatype_icon

error out

Error information.

The node produces this output according to standard error behavior.

Standard Error Behavior**Programming Patterns**

Transferring Data Using Tags

U64

Specifies the value of the tag. The data type of the tag must be an unsigned 64-bit integer.

Inputs/Outputs

`datatype_icon`

tag in

Tag the server references to communicate measurement data.

`datatype_icon`

value

Data value of the tag or tags.

`datatype_icon`

time stamp

Date and time the server obtained the most recent data value.

Leave the `time stamp` input empty to have the server manage the timestamp.

`datatype_icon`

error in

Error conditions that occur before this node runs.

The node responds to this input according to standard error behavior. For more information, refer to ***Standard Error Behavior***.

Default value: No error

datatype_icon

error out

Error information.

The node produces this output according to standard error behavior. For more information, refer to ***Standard Error Behavior***.

datatype_icon

tag out

Tag the server references to communicate measurement data.

Programming Patterns

Transferring Data Using Tags

Related information:

- [Standard Error Behavior](#)

Write Tag Settings

Specifies the settings of a tag. The node replaces existing tag settings on the server with the new settings.

A tag setting is a type of permission for the tag, such as read, write, modify, delete, and create. Based on the settings, a tag can complete any of the permissions an

administrator specifies.

Inputs/Outputs

datatype_icon

retention type

Aging policy which the tag server uses.

datatype_icon

tag in

Tag the server references to communicate measurement data.

datatype_icon

properties

Key-value pairs that exist as metadata on a tag.

datatype_icon

keywords

Specific words provided to find tags.

datatype_icon

error in

Error conditions that occur before this node runs.

The node responds to this input according to standard error behavior.

Standard Error Behavior

Default value: No error

datatype_icon

collect aggregates?

Boolean that determines if the node collects aggregate tag data.

datatype_icon

tag out

Tag the server references to communicate measurement data.

datatype_icon

error out

Error information.

The node produces this output according to standard error behavior.

Standard Error Behavior

TDM Reader Functions

Visualize and interact with waveform data in TDMS files.

SystemLink Nodes and Functions Reference for NXG and G Web Development Software

Use SystemLink nodes in NXG or G Web Development Software to communicate data between your test and measurement systems and your SystemLink Server account.

Close File

Closes a reference to a TDMS file or group of TDMS files on the server.

Create Filter

Creates or adds to an existing filter to query TDMS files on the server.

Get TDM File Metadata

Returns metadata for a TDM file you specify.

List File Contents

Returns channel group and channel names from a TDMS file.

Open File

Opens a reference to a TDMS file on the server.

Query TDMS Files

Searches the server for files matching `filter` in.

Read Data

Returns data from a TDMS file.

API Reference

G, HTTP, Python, and .NET APIs allow you to programmatically access SystemLink data services outside of NI SystemLink Web Application.

SystemLink Nodes and Functions Reference for NXG and G Web Development Software

Use SystemLink nodes in NXG or G Web Development Software to communicate data between your test and measurement systems and your SystemLink Server account.

Close File

Closes a reference to a TDMS file or group of TDMS files on the server.

TDM Reader Functions

Visualize and interact with waveform data in TDMS files.

Multiple

Closes a reference to a group of TDMS files on the server.

Single

Closes a reference to a TDMS file on the server.

API Reference

G, HTTP, Python, and .NET APIs allow you to programmatically access SystemLink data services outside of NI SystemLink Web Application.

SystemLink Nodes and Functions Reference for NXG and G Web Development Software

Use SystemLink nodes in NXG or G Web Development Software to communicate data between your test and measurement systems and your SystemLink Server account.

TDM Reader Functions

Visualize and interact with waveform data in TDMS files.

Multiple

Closes a reference to a group of TDMS files on the server.

Even though it is not required, NI recommends you close TDMS file reference(s) to free resources your program may use in the future.

Inputs/Outputs

datatype_icon

tdm files

TDMS files reference on the server.

Use Open File to obtain the reference.

datatype_icon

error in

Error conditions that occur before this node runs.

The node responds to this input according to standard error behavior.

Standard Error Behavior

Default value: No error

datatype_icon

error out

Error information.

The node produces this output according to standard error behavior.

Standard Error Behavior**Single**

Closes a reference to a TDMS file on the server.

Even though it is not required, NI recommends you close TDMS file reference(s) to free any resources your program may use in the future.

Inputs/Outputs

datatype_icon

tdm file

TDMS file reference on the server.

Use Open File to obtain the reference.

datatype_icon

error in

Error conditions that occur before this node runs.

The node responds to this input according to standard error behavior.

Standard Error Behavior

Default value: No error

datatype_icon

error out

Error information.

The node produces this output according to standard error behavior.

Standard Error Behavior

Create Filter

Creates or adds to an existing filter to query TDMS files on the server.

TDM Reader Functions

Visualize and interact with waveform data in TDMS files.

Name

Creates or adds a file name query to an existing TDMS file query filter.

Property

Creates or adds a property query to an existing TDMS file query filter.

Size in Bytes

Creates or adds a size query to an existing TDMS file query filter.

[Created Time Stamp](#)

Creates or adds a timestamp query to an existing TDMS file query filter.

[API Reference](#)

G, HTTP, Python, and .NET APIs allow you to programmatically access SystemLink data services outside of NI SystemLink Web Application.

[SystemLink Nodes and Functions Reference for NXG and G Web Development Software](#)

Use SystemLink nodes in NXG or G Web Development Software to communicate data between your test and measurement systems and your SystemLink Server account.

[TDM Reader Functions](#)

Visualize and interact with waveform data in TDMS files.

Name

Creates or adds a file name query to an existing TDMS file query filter.

If you specify multiple name queries, Query TDMS Files opens only the TDMS files that match all queries.

Inputs/Outputs

datatype_icon

operator

Criteria for how to filter TDMS files on the server.

contains	The alarm instance(s) returned must contain the value you specify.
not contains	The alarm instance(s) returned must not contain the value you specify.
equal	The alarm instance(s) returned must equal the value you specify.
not equal	The alarm instance(s) returned must not equal the value you specify.

datatype_icon

filter in

Query or queries you want to use to find matching TDMS files on the server.

datatype_icon

name

File name you want to add to the query filter.

datatype_icon

error in

Error conditions that occur before this node runs.

The node responds to this input according to standard error behavior.

Standard Error Behavior

Default value: No error

datatype_icon

filter out

Updated query or queries to find matching TDMS files on the server.

datatype_icon

error out

Error information.

The node produces this output according to standard error behavior.

Standard Error Behavior

Property

Creates or adds a property query to an existing TDMS file query filter.

If you specify multiple property queries, Query TDMS Files opens only the TDMS files that match all queries.

Inputs/Outputs

datatype_icon

operator

Criteria for how to filter TDMS files on the server.

contains	The alarm instance(s) returned must contain the value you specify.
not contains	The alarm instance(s) returned must not contain the value you specify.
equal	The alarm instance(s) returned must equal the value you specify.
not equal	The alarm instance(s) returned must not equal the value you specify.

datatype_icon

filter in

Query or queries you want to use to find matching TDMS files on the server.

datatype_icon

key

Unique identifier for the property you want to locate.

datatype_icon

value

Metadata property you want to add to the TDMS file.

datatype_icon

error in

Error conditions that occur before this node runs.

The node responds to this input according to standard error behavior.

Standard Error Behavior

Default value: No error

datatype_icon

filter out

Updated query or queries to find matching TDMS files on the server.

datatype_icon

error out

Error information.

The node produces this output according to standard error behavior.

Standard Error BehaviorSize in Bytes

Creates or adds a size query to an existing TDMS file query filter.

When you wire this node to Query TDMS Files, the results only contain TDMS files that match size attributes in the range specified by one or more query entries.

Inputs/Outputs

datatype_icon

operator

Criteria for how to filter TDMS files on the server.

contains	The alarm instance(s) returned must contain the value you specify.
not contains	The alarm instance(s) returned must not contain the value you specify.
equal	The alarm instance(s) returned must equal the value you specify.
not equal	The alarm instance(s) returned must not equal the value you specify.

datatype_icon

filter in

Query or queries you want to use to find matching TDMS files on the server.

datatype_icon

size

Size, in bytes, of the TDMS file(s) for which you want to query.

datatype_icon

error in

Error conditions that occur before this node runs.

The node responds to this input according to standard error behavior.

Standard Error Behavior

Default value: No error

datatype_icon

filter out

Updated query or queries to find matching TDMS files on the server.

datatype_icon

error out

Error information.

The node produces this output according to standard error behavior.

Standard Error Behavior

Created Time Stamp

Creates or adds a timestamp query to an existing TDMS file query filter.

If you specify multiple timestamp queries, Query TDMS Files opens only the TDMS files that match the last timestamp query provided.

Inputs/Outputs

datatype_icon

operator

Criteria for how to filter TDMS files on the server.

contains	The alarm instance(s) returned must contain the value you specify.
not contains	The alarm instance(s) returned must not contain the value you specify.
equal	The alarm instance(s) returned must equal the value you specify.

not equal	The alarm instance(s) returned must not equal the value you specify.
-----------	--

datatype_icon

filter in

Query or queries you want to use to find matching TDMS files on the server.

datatype_icon

time stamp

Creation date and time of the TDMS file(s) you want to query.

datatype_icon

error in

Error conditions that occur before this node runs.

The node responds to this input according to standard error behavior.

Standard Error Behavior

Default value: No error

datatype_icon

filter out

Updated query or queries to find matching TDMS files on the server.

datatype_icon

error out

Error information.

The node produces this output according to standard error behavior.

Standard Error Behavior

Get TDM File Metadata

Returns metadata for a TDM file you specify.

Inputs/Outputs

`datatype_icon`

file in

File on the server.

`datatype_icon`

file out

File on the server.

`datatype_icon`

id

Unique file identifier the SystemLink file service generates to identify the file on the server.

`datatype_icon`

metadata

Metadata properties of the file you specify.

List File Contents

Returns channel group and channel names from a TDMS file.

TDM Reader Functions

Visualize and interact with waveform data in TDMS files.

GroupsAndChannels

Returns channel group and channel names from a TDMS file.

[Channels](#)

Returns channel names from channel group in a TDMS file.

[API Reference](#)

G, HTTP, Python, and .NET APIs allow you to programmatically access SystemLink data services outside of NI SystemLink Web Application.

[SystemLink Nodes and Functions Reference for NXG and G Web Development Software](#)

Use SystemLink nodes in NXG or G Web Development Software to communicate data between your test and measurement systems and your SystemLink Server account.

[TDM Reader Functions](#)

Visualize and interact with waveform data in TDMS files.

GroupsAndChannels

Returns channel group and channel names from a TDMS file.

Inputs/Outputs

datatype_icon

tdm file in

TDMS file reference on the server.

Use Query TDMS Files to obtain the reference.

datatype_icon

error in

Error conditions that occur before this node runs.

The node responds to this input according to standard error behavior.

[Standard Error Behavior](#)

Default value: No error

datatype_icon

tdm file out

TDMS file reference on the server.

datatype_icon

group names

Names of the channel groups in the TDMS file.

datatype_icon

channel names per group

Names of the channels in the TDMS file.

datatype_icon

error out

Error information.

The node produces this output according to standard error behavior.

[Standard Error Behavior](#)

Channels

Returns channel names from channel group in a TDMS file.

Inputs/Outputs

datatype_icon

tdm file in

TDMS file reference on the server.

Use Query TDMS Files to obtain the reference.

datatype_icon

group name in

Channel group to analyze.

datatype_icon

error in

Error conditions that occur before this node runs.

The node responds to this input according to standard error behavior.

Standard Error Behavior

Default value: No error

datatype_icon

tdm file out

TDMS file reference on the server.

datatype_icon

group name out

Channel group on which you performed the operation.

datatype_icon

channel names

Names of the channels associated with the group you specified.

datatype_icon

error out

Error information.

The node produces this output according to standard error behavior.

Standard Error Behavior

Open File

Opens a reference to a TDMS file on the server.

Inputs/Outputs

datatype_icon

configuration

Connection to the server.

If you leave this input unwired, this node uses the default connection. It automatically retrieves your SystemLink Server credentials when you execute the application on a managed system or server.

datatype_icon

file id

Unique file identifier the server generates to identify the TDMS file on the server.

datatype_icon

error in

Error conditions that occur before this node runs.

The node responds to this input according to standard error behavior.

Standard Error Behavior

Default value: No error

datatype_icon

tdm file

TDMS file reference returned from the server.

datatype_icon

error out

Error information.

The node produces this output according to standard error behavior.

Standard Error Behavior

Query TDMS Files

Searches the server for files matching `filter in`.

If you do not provide a filter, this VI opens references to every TDMS file on the server. The server limits the maximum number of results returned in a single query to 1,000 TDMS files.

Inputs/Outputs

datatype_icon

configuration

Connection to the server.

datatype_icon

filter in

Query or queries you want to use find matching TDMS files on the server.

Wire Create Filter to this input to construct the filter you want to use.

datatype_icon

limit to configured workspace?

Boolean that determines whether this node only queries for tags in the workspace you specify in `configuration`.

By default, the node queries for tags from all accessible workspaces.

`datatype_icon`

error in

Error conditions that occur before this node runs.

The node responds to this input according to standard error behavior.

Standard Error Behavior

Default value: No error

`datatype_icon`

skip

Number of files skipped to limit the amount returned from the server.

If the input is left empty, the query returns the first 1000 results by default.

`datatype_icon`

take

Number of files requested from the server.

If the input is left empty, the query returns the first 1000 results by default.

`datatype_icon`

tdm files

TDMS files reference returned from the server.

`datatype_icon`

filter out

Updated query or queries to find matching TDMS files on the server.

`datatype_icon`

error out

Error information.

The node produces this output according to standard error behavior.

Standard Error Behavior

Read Data

Returns data from a TDMS file.

TDM Reader Functions

Visualize and interact with waveform data in TDMS files.

1Chan Boolean

Returns Boolean data from the channel specified in `channel name in`.

NChan Boolean

Returns Boolean data from the channels specified in `channel names in`.

1Chan Double

Returns double-precision, floating-point number data from the channel specified in `channel name in`.

NChan Double

Returns double-precision, floating-point number data from the channels specified in `channel names in`.

1Chan I16

Returns 16-bit integer data from the channel specified in `channel name in`.

NChan I16

Returns 16-bit integer data from the channels specified in `channel names in`.

1Chan I32

Returns 32-bit integer data from the channel specified in `channel name in`.

NChan I32

Returns 32-bit integer data from the channels specified in `channel names in`.

1Chan I8

Reads 8-bit integer data from the channel specified in `channel name in`.

NChan I8

Reads 8-bit integer data from the channels specified in `channel names in`.

1Chan Single

Returns single-precision, floating-point number data from the channel specified in `channel name in`.

NChan Single

Returns single-precision, floating-point number data from the channels specified in `channel names in`.

1Chan String

Returns string data from the channel specified in `channel name in`.

NChan String

Returns string data from the channels specified in `channel names in`.

1Chan Timestamp

Returns timestamp data from the channel specified in `channel name in`.

NChan Timestamp

Returns timestamp data from the channels specified in `channel names in`.

1Chan U16

Returns unsigned 16-bit integer data from the channel specified in `channel name in`.

NChan U16

Returns unsigned 16-bit integer data from the channels specified in `channel names in`.

1Chan U32

Returns unsigned 32-bit integer data from the channel specified in `channel name in`.

NChan U32

Returns unsigned 32-bit integer data from the channels specified in `channel names in`.

1Chan U8

Returns unsigned 8-bit integer data from the channel specified in `channel name in`.

NChan U8

Returns unsigned 8-bit integer data from the channels specified in `channel names in`.

1Chan Waveform

Returns waveform data from the channel specified in `channel name in`.

[NChan Waveform](#)

Returns waveform data from the channels specified in `channels name in`.

[API Reference](#)

G, HTTP, Python, and .NET APIs allow you to programmatically access SystemLink data services outside of NI SystemLink Web Application.

[SystemLink Nodes and Functions Reference for NXG and G Web Development Software](#)

Use SystemLink nodes in NXG or G Web Development Software to communicate data between your test and measurement systems and your SystemLink Server account.

[TDM Reader Functions](#)

Visualize and interact with waveform data in TDMS files.

1Chan Boolean

Returns Boolean data from the channel specified in `channel name in`.

Inputs/Outputs

`datatype_icon`

count

Maximum number of data elements to read for each channel.

If you use the default value, this node reads all available elements.

Default value: -1

`datatype_icon`

offset

Number of data elements into the TDMS file at which this node begins to read.

Default value: 0

`datatype_icon`

tdm file in

TDMS file reference on the server.

Use Query TDMS Files to obtain the reference.

datatype_icon

group name in

Channel group to analyze.

If you do not wire `group name in`, this node reads the channel name from the next group in the TDMS file each time it executes.

datatype_icon

channel name in

Channel to read data from.

If you do not wire `channel name in`, this node reads the first channel in the group.

datatype_icon

error in

Error conditions that occur before this node runs.

The node responds to this input according to standard error behavior.

Standard Error Behavior

Default value: No error

datatype_icon

tdm file out

TDMS file reference on the server.

datatype_icon

group name out

Channel group returned from the TDMS file.

datatype_icon

channel name out

Name of the channel associated with the group you specified.

datatype_icon

data

1D array of Boolean data read from the TDMS file.

datatype_icon

error out

Error information.

The node produces this output according to standard error behavior.

Standard Error Behavior

datatype_icon

end of channel data?

Boolean that indicates whether the node read the last of the data for the channel in `channel name out`.

datatype_icon

end of groups?

Boolean that indicates whether the node read data from the last group in the TDMS file.

NChan Boolean

Returns Boolean data from the channels specified in `channel names in`.

Inputs/Outputs

`datatype_icon`

count

Maximum number of data elements to read for each channel.

If you use the default value, this node reads all available elements.

Default value: -1

`datatype_icon`

offset

Number of data elements into the TDMS file at which this node begins to read.

Default value: 0

`datatype_icon`

tdm file in

TDMS file reference on the server.

Use Query TDMS Files to obtain the reference.

`datatype_icon`

group name in

Channel group to analyze.

If you do not wire `group name in`, this node reads the channel names from the next group in the TDMS file each time it executes.

datatype_icon

channel names in

Channels to read data from.

If you do not wire `channel name in`, this node reads all channels in the group.

datatype_icon

return channels in file order?

Boolean specifying whether the node returns data channels in the same order as they exist in the TDMS file.

True	Return data channels in the same order as they exist in the TDMS file.
False	Return data channels in the same order as you specify in <code>channel names in</code> .

Default value: False

datatype_icon

error in

Error conditions that occur before this node runs.

The node responds to this input according to standard error behavior.

Standard Error Behavior

Default value: No error

datatype_icon

tdm file out

TDMS file reference on the server.

datatype_icon

group name out

Channel group returned from the TDMS file.

datatype_icon

channel names out

Names of the channels associated with the group you specified.

The order of the names returned correspond to the order of the `data` array when `return channels in file order?` is `True`.

datatype_icon

data

2D array of Boolean data read from the TDMS file.

datatype_icon

error out

Error information.

The node produces this output according to standard error behavior.

Standard Error Behavior

datatype_icon

end of channel data?

A Boolean that indicates whether the node read the last of the data for any channel in `channel names out`.

datatype_icon

end of groups?

Boolean that indicates whether the node read data from the last group in the TDMS file.

1Chan Double

Returns double-precision, floating-point number data from the channel specified in `channel name in`.

Inputs/Outputs

`datatype_icon`

count

Maximum number of data elements to read for each channel.

If you use the default value, this node reads all available elements.

Default value: -1

`datatype_icon`

offset

Number of data elements into the TDMS file at which this node begins to read.

Default value: 0

`datatype_icon`

tdm file in

TDMS file reference on the server.

Use Query TDMS Files to obtain the reference.

`datatype_icon`

group name in

Channel group to analyze.

If you do not wire `group name in`, this node reads the channel name from the next group in the TDMS file each time it executes.

datatype_icon

channel name in

Channel to read data from.

If you do not wire `channel name in`, this node reads the first channel in the group.

datatype_icon

error in

Error conditions that occur before this node runs.

The node responds to this input according to standard error behavior. For more information, refer to ***Standard Error Behavior***.

Default value: No error

datatype_icon

tdm file out

TDMS file reference on the server.

datatype_icon

group name out

Channel group returned from the TDMS file.

datatype_icon

channel name out

Name of the channel associated with the group you specified.

datatype_icon

data

1D array of double-precision, floating-point number data read from the TDMS file.

datatype_icon

error out

Error information.

The node produces this output according to standard error behavior. For more information, refer to ***Standard Error Behavior***.

datatype_icon

end of channel data?

Boolean that indicates whether the node read the last of the data for the channel in `channel name out`.

datatype_icon

end of groups?

Boolean that indicates whether the node read data from the last group in the TDMS file.

Related information:

- [Standard Error Behavior](#)

NChan Double

Returns double-precision, floating-point number data from the channels specified in

channel names in.

Inputs/Outputs

datatype_icon

count

Maximum number of data elements to read for each channel.

If you use the default value, this node reads all available elements.

Default value: -1

datatype_icon

offset

Number of data elements into the TDMS file at which this node begins to read.

Default value: 0

datatype_icon

tdm file in

TDMS file reference on the server.

Use Query TDMS Files to obtain the reference.

datatype_icon

group name in

Channel group to analyze.

If you do not wire `group name in`, this node reads the channel name from the next group in the TDMS file each time it executes.

datatype_icon

channel names in

Channels to read data from.

If you do not wire `channel name in`, this node reads all channels in the group.

`datatype_icon`

return channels in file order?

Boolean specifying whether the node returns data channels in the same order as they exist in the TDMS file.

True	Return data channels in the same order as they exist in the TDMS file.
False	Return data channels in the same order as you specify in <code>channel names in</code> .

Default value: False

`datatype_icon`

error in

Error conditions that occur before this node runs.

The node responds to this input according to standard error behavior. For more information, refer to ***Standard Error Behavior***.

Default value: No error

`datatype_icon`

tdm file out

TDMS file reference on the server.

`datatype_icon`

group name out

Channel group returned from the TDMS file.

`datatype_icon`

channel names out

Names of the channels associated with the group you specified.

The order of the names returned correspond to the order of the `data` array when `return channels in file order?` is `True`.

`datatype_icon`

data

2D array of double-precision, floating-point number data read from the TDMS file.

`datatype_icon`

error out

Error information.

The node produces this output according to standard error behavior. For more information, refer to ***Standard Error Behavior***.

`datatype_icon`

end of channel data?

A Boolean that indicates whether the node read the last of the data for any channel in `channel names out`.

`datatype_icon`

end of groups?

Boolean that indicates whether the node read data from the last group in the TDMS

file.

Related information:

- [Standard Error Behavior](#)

1Chan I16

Returns 16-bit integer data from the channel specified in `channel name in`.

Inputs/Outputs

`datatype_icon`

count

Maximum number of data elements to read for each channel.

If you use the default value, this node reads all available elements.

Default value: -1

`datatype_icon`

offset

Number of data elements into the TDMS file at which this node begins to read.

Default value: 0

`datatype_icon`

tdm file in

TDMS file reference on the server.

Use Query TDMS Files to obtain the reference.

`datatype_icon`

group name in

Channel group to analyze.

If you do not wire `group name in`, this node reads the channel name from the next group in the TDMS file each time it executes.

datatype_icon

channel name in

Channel to read data from.

If you do not wire `channel name in`, this node reads the first channel in the group.

datatype_icon

error in

Error conditions that occur before this node runs.

The node responds to this input according to standard error behavior. For more information, refer to ***Standard Error Behavior***.

Default value: No error

datatype_icon

tdm file out

TDMS file reference on the server.

datatype_icon

group name out

Channel group returned from the TDMS file.

datatype_icon

channel name out

Name of the channel associated with the group you specified.

datatype_icon

data

1D array of 16-bit integer data read from the TDMS file.

datatype_icon

error out

Error information.

The node produces this output according to standard error behavior. For more information, refer to ***Standard Error Behavior***.

datatype_icon

end of channel data?

Boolean that indicates whether the node read the last of the data for the channel in `channel name out`.

datatype_icon

end of groups?

Boolean that indicates whether the node read data from the last group in the TDMS file.

Related information:

- [Standard Error Behavior](#)

NChan I16

Returns 16-bit integer data from the channels specified in `channel names in`.

Inputs/Outputs

datatype_icon

count

Maximum number of data elements to read for each channel.

If you use the default value, this node reads all available elements.

Default value: -1

datatype_icon

offset

Number of data elements into the TDMS file at which this node begins to read.

Default value: 0

datatype_icon

tdm file in

TDMS file reference on the server.

Use Query TDMS Files to obtain the reference.

datatype_icon

group name in

Channel group to analyze.

If you do not wire `group name in`, this node reads the channel name from the next group in the TDMS file each time it executes.

datatype_icon

channel names in

Channels to read data from.

If you do not wire `channel name in`, this node reads all channels in the group.

`datatype_icon`

return channels in file order?

Boolean specifying whether the node returns data channels in the same order as they exist in the TDMS file.

True	Return data channels in the same order as they exist in the TDMS file.
False	Return data channels in the same order as you specify in <code>channel names in</code> .

Default value: False

`datatype_icon`

error in

Error conditions that occur before this node runs.

The node responds to this input according to standard error behavior.

Standard Error Behavior

Default value: No error

`datatype_icon`

tdm file out

TDMS file reference on the server.

`datatype_icon`

group name out

Channel group returned from the TDMS file.

datatype_icon

channel names out

Names of the channels associated with the group you specified.

The order of the names returned correspond to the order of the `data` array when `return channels in file order?` is `True`.

datatype_icon

data

2D array of 16-bit integer data read from the TDMS file.

datatype_icon

error out

Error information.

The node produces this output according to standard error behavior.

Standard Error Behavior

datatype_icon

end of channel data?

A Boolean that indicates whether the node read the last of the data for any channel in `channel names out`.

datatype_icon

end of groups?

Boolean that indicates whether the node read data from the last group in the TDMS file.

1Chan I32

Returns 32-bit integer data from the channel specified in `channel name in`.

Inputs/Outputs

`datatype_icon`

count

Maximum number of data elements to read for each channel.

If you use the default value, this node reads all available elements.

Default value: -1

`datatype_icon`

offset

Number of data elements into the TDMS file at which this node begins to read.

Default value: 0

`datatype_icon`

tdm file in

TDMS file reference on the server.

Use Query TDMS Files to obtain the reference.

`datatype_icon`

group name in

Channel group to analyze.

If you do not wire `group name in`, this node reads the channel name from the next group in the TDMS file each time it executes.

datatype_icon

channel name in

Channel to read data from.

If you do not wire `channel name in`, this node reads the first channel in the group.

datatype_icon

error in

Error conditions that occur before this node runs.

The node responds to this input according to standard error behavior.

Standard Error Behavior

Default value: No error

datatype_icon

tdm file out

TDMS file reference on the server.

datatype_icon

group name out

Channel group returned from the TDMS file.

datatype_icon

channel name out

Name of the channel associated with the group you specified.

datatype_icon

data

1D array of 32-bit integer data read from the TDMS file.

datatype_icon

error out

Error information.

The node produces this output according to standard error behavior.

Standard Error Behavior

datatype_icon

end of channel data?

Boolean that indicates whether the node read the last of the data for the channel in `channel name out`.

datatype_icon

end of groups?

Boolean that indicates whether the node read data from the last group in the TDMS file.

NChan I32

Returns 32-bit integer data from the channels specified in `channel names in`.

Inputs/Outputs

datatype_icon

count

Maximum number of data elements to read for each channel.

If you use the default value, this node reads all available elements.

Default value: -1

datatype_icon

offset

Number of data elements into the TDMS file at which this node begins to read.

Default value: 0

datatype_icon

tdm file in

TDMS file reference on the server.

Use Query TDMS Files to obtain the reference.

datatype_icon

group name in

Channel group to analyze.

If you do not wire `group name in`, this node reads the channel name from the next group in the TDMS file each time it executes.

datatype_icon

channel names in

Channels to read data from.

If you do not wire `channel name in`, this node reads all channels in the group.

`datatype_icon`

return channels in file order?

Boolean specifying whether the node returns data channels in the same order as they exist in the TDMS file.

True	Return data channels in the same order as they exist in the TDMS file.
False	Return data channels in the same order as you specify in <code>channel names in</code> .

Default value: False

`datatype_icon`

error in

Error conditions that occur before this node runs.

The node responds to this input according to standard error behavior.

Standard Error Behavior

Default value: No error

`datatype_icon`

tdm file out

TDMS file reference on the server.

`datatype_icon`

group name out

Channel group returned from the TDMS file.

datatype_icon

channel names out

Names of the channels associated with the group you specified.

The order of the names returned correspond to the order of the `data` array when `return channels in file order?` is `True`.

datatype_icon

data

2D array of 32-bit integer data read from the TDMS file.

datatype_icon

error out

Error information.

The node produces this output according to standard error behavior.

Standard Error Behavior

datatype_icon

end of channel data?

A Boolean that indicates whether the node read the last of the data for any channel in `channel names out`.

datatype_icon

end of groups?

Boolean that indicates whether the node read data from the last group in the TDMS file.

1Chan I8

Reads 8-bit integer data from the channel specified in `channel name in`.

Inputs/Outputs

`datatype_icon`

count

Maximum number of data elements to read for each channel.

If you use the default value, this node reads all available elements.

Default value: -1

`datatype_icon`

offset

Number of data elements into the TDMS file at which this node begins to read.

Default value: 0

`datatype_icon`

tdm file in

TDMS file reference on the server.

Use Query TDMS Files to obtain the reference.

`datatype_icon`

group name in

Channel group to analyze.

If you do not wire `group name in`, this node reads the channel name from the next group in the TDMS file each time it executes.

datatype_icon

channel name in

Channel to read data from.

If you do not wire `channel name in`, this node reads the first channel in the group.

datatype_icon

error in

Error conditions that occur before this node runs.

The node responds to this input according to standard error behavior. For more information, refer to ***Standard Error Behavior***.

Default value: No error

datatype_icon

tdm file out

TDMS file reference on the server.

datatype_icon

group name out

Channel group returned from the TDMS file.

datatype_icon

channel name out

Name of the channel associated with the group you specified.

datatype_icon

data

1D array of 8-bit integer data read from the TDMS file.

datatype_icon

error out

Error information.

The node produces this output according to standard error behavior. For more information, refer to ***Standard Error Behavior***.

datatype_icon

end of channel data?

Boolean that indicates whether the node read the last of the data for the channel in `channel name out`.

datatype_icon

end of groups?

Boolean that indicates whether the node read data from the last group in the TDMS file.

Related information:

- [Standard Error Behavior](#)

NChan I8

Reads 8-bit integer data from the channels specified in `channel names in`.

Inputs/Outputs

datatype_icon

count

Maximum number of data elements to read for each channel.

If you use the default value, this node reads all available elements.

Default value: -1

datatype_icon

offset

Number of data elements into the TDMS file at which this node begins to read.

Default value: 0

datatype_icon

tdm file in

TDMS file reference on the server.

Use Query TDMS Files to obtain the reference.

datatype_icon

group name in

Channel group to analyze.

If you do not wire `group name in`, this node reads the channel name from the next group in the TDMS file each time it executes.

datatype_icon

channel names in

Channels to read data from.

If you do not wire `channel name in`, this node reads all channels in the group.

`datatype_icon`

return channels in file order?

Boolean specifying whether the node returns data channels in the same order as they exist in the TDMS file.

True	Return data channels in the same order as they exist in the TDMS file.
False	Return data channels in the same order as you specify in <code>channel names in</code> .

Default value: False

`datatype_icon`

error in

Error conditions that occur before this node runs.

The node responds to this input according to standard error behavior.

Standard Error Behavior

Default value: No error

`datatype_icon`

tdm file out

TDMS file reference on the server.

`datatype_icon`

group name out

Channel group returned from the TDMS file.

datatype_icon

channel names out

Names of the channels associated with the group you specified.

The order of the names returned correspond to the order of the `data` array when `return channels in file order?` is `True`.

datatype_icon

data

2D array of 8-bit integer data read from the TDMS file.

datatype_icon

error out

Error information.

The node produces this output according to standard error behavior.

Standard Error Behavior

datatype_icon

end of channel data?

A Boolean that indicates whether the node read the last of the data for any channel in `channel names out`.

datatype_icon

end of groups?

Boolean that indicates whether the node read data from the last group in the TDMS file.

1Chan Single

Returns single-precision, floating-point number data from the channel specified in `channel name in`.

Inputs/Outputs

`datatype_icon`

count

Maximum number of data elements to read for each channel.

If you use the default value, this node reads all available elements.

Default value: -1

`datatype_icon`

offset

Number of data elements into the TDMS file at which this node begins to read.

Default value: 0

`datatype_icon`

tdm file in

TDMS file reference on the server.

Use Query TDMS Files to obtain the reference.

`datatype_icon`

group name in

Channel group to analyze.

If you do not wire `group name in`, this node reads the channel name from the next group in the TDMS file each time it executes.

`datatype_icon`

channel name in

Channel to read data from.

If you do not wire `channel name in`, this node reads the first channel in the group.

`datatype_icon`

data

1D array of single-precision, floating-point number data read from the TDMS file.

`datatype_icon`

error in

Error conditions that occur before this node runs.

The node responds to this input according to standard error behavior.

Standard Error Behavior

Default value: No error

`datatype_icon`

tdm file out

TDMS file reference on the server.

`datatype_icon`

group name out

Channel group returned from the TDMS file.

datatype_icon

channel name out

Name of the channel associated with the group you specified.

datatype_icon

data

1D array of single-precision, floating-point number data read from the TDMS file.

datatype_icon

error out

Error information.

The node produces this output according to standard error behavior.

Standard Error Behavior

datatype_icon

end of channel data?

Boolean that indicates whether the node read the last of the data for the channel in `channel name out`.

datatype_icon

end of groups?

Boolean that indicates whether the node read data from the last group in the TDMS file.

NChan Single

Returns single-precision, floating-point number data from the channels specified in `channel names in`.

Inputs/Outputs

`datatype_icon`

count

Maximum number of data elements to read for each channel.

If you use the default value, this node reads all available elements.

Default value: -1

`datatype_icon`

offset

Number of data elements into the TDMS file at which this node begins to read.

Default value: 0

`datatype_icon`

tdm file in

TDMS file reference on the server.

Use Query TDMS Files to obtain the reference.

`datatype_icon`

group name in

Channel group to analyze.

If you do not wire `group name in`, this node reads the channel name from the next

group in the TMDS file each time it executes.

datatype_icon

channel names in

Channels to read data from.

If you do not wire `channel name in`, this node reads all channels in the group.

datatype_icon

return channels in file order?

Boolean specifying whether the node returns data channels in the same order as they exist in the TDMS file.

True	Return data channels in the same order as they exist in the TDMS file.
False	Return data channels in the same order as you specify in <code>channel names in</code> .

Default value: False

datatype_icon

error in

Error conditions that occur before this node runs.

The node responds to this input according to standard error behavior.

Standard Error Behavior

Default value: No error

datatype_icon

tdm file out

TDMS file reference on the server.

datatype_icon

group name out

Channel group returned from the TDMS file.

datatype_icon

channel names out

Names of the channels associated with the group you specified.

The order of the names returned correspond to the order of the `data` array when `return channels in file order?` is `True`.

datatype_icon

data

2D array of single-precision, floating-point number data read from the TDMS file.

datatype_icon

error out

Error information.

The node produces this output according to standard error behavior.

Standard Error Behavior

datatype_icon

end of channel data?

A Boolean that indicates whether the node read the last of the data for any channel in

`channel` names out.

`datatype_icon`

end of groups?

Boolean that indicates whether the node read data from the last group in the TDMS file.

1Chan String

Returns string data from the channel specified in `channel` name in.

Inputs/Outputs

`datatype_icon`

count

Maximum number of data elements to read for each channel.

If you use the default value, this node reads all available elements.

Default value: -1

`datatype_icon`

offset

Number of data elements into the TDMS file at which this node begins to read.

Default value: 0

`datatype_icon`

tdm file in

TDMS file reference on the server.

Use Query TDMS Files to obtain the reference.

datatype_icon

group name in

Channel group to analyze.

If you do not wire `group name in`, this node reads the channel name from the next group in the TDMS file each time it executes.

datatype_icon

channel name in

Channel to read data from.

If you do not wire `channel name in`, this node reads the first channel in the group.

datatype_icon

error in

Error conditions that occur before this node runs.

The node responds to this input according to standard error behavior. For more information, refer to ***Standard Error Behavior***.

Default value: No error

datatype_icon

tdm file out

TDMS file reference on the server.

datatype_icon

group name out

Channel group returned from the TDMS file.

datatype_icon

channel name out

Name of the channel associated with the group you specified.

datatype_icon

data

1D array of string data read from the TDMS file.

datatype_icon

error out

Error information.

The node produces this output according to standard error behavior. For more information, refer to ***Standard Error Behavior***.

datatype_icon

end of channel data?

Boolean that indicates whether the node read the last of the data for the channel in `channel name out`.

datatype_icon

end of groups?

Boolean that indicates whether the node read data from the last group in the TDMS file.

Related information:

- [Standard Error Behavior](#)

NChan String

Returns string data from the channels specified in `channel names in`.

Inputs/Outputs

`datatype_icon`

count

Maximum number of data elements to read for each channel.

If you use the default value, this node reads all available elements.

Default value: -1

`datatype_icon`

offset

Number of data elements into the TDMS file at which this node begins to read.

Default value: 0

`datatype_icon`

tdm file in

TDMS file reference on the server.

Use Query TDMS Files to obtain the reference.

`datatype_icon`

group name in

Channel group to analyze.

If you do not wire `group name in`, this node reads the channel name from the next group in the TDMS file each time it executes.

datatype_icon

channel names in

Channels to read data from.

If you do not wire `channel name in`, this node reads all channels in the group.

datatype_icon

return channels in file order?

Boolean specifying whether the node returns data channels in the same order as they exist in the TDMS file.

True	Return data channels in the same order as they exist in the TDMS file.
False	Return data channels in the same order as you specify in <code>channel names in</code> .

Default value: False

datatype_icon

error in

Error conditions that occur before this node runs.

The node responds to this input according to standard error behavior. For more information, refer to ***Standard Error Behavior***.

Default value: No error

datatype_icon

tdm file out

TDMS file reference on the server.

datatype_icon

group name out

Channel group returned from the TDMS file.

datatype_icon

channel names out

Names of the channels associated with the group you specified.

The order of the names returned correspond to the order of the `data` array when `return channels in file order?` is `True`.

datatype_icon

data

2D array of string data read from the TDMS file.

datatype_icon

error out

Error information.

The node produces this output according to standard error behavior. For more information, refer to ***Standard Error Behavior***.

datatype_icon

end of channel data?

A Boolean that indicates whether the node read the last of the data for any channel in `channel names out`.

datatype_icon

end of groups?

Boolean that indicates whether the node read data from the last group in the TDMS file.

Related information:

- [Standard Error Behavior](#)

1Chan Timestamp

Returns timestamp data from the channel specified in `channel name in`.

Inputs/Outputs

`datatype_icon`

count

Maximum number of data elements to read for each channel.

If you use the default value, this node reads all available elements.

Default value: -1

`datatype_icon`

offset

Number of data elements into the TDMS file at which this node begins to read.

Default value: 0

`datatype_icon`

tdm file in

TDMS file reference on the server.

Use Query TDMS Files to obtain the reference.

datatype_icon

group name in

Channel group to analyze.

If you do not wire `group name in`, this node reads the channel name from the next group in the TDMS file each time it executes.

datatype_icon

channel name in

Channel to read data from.

If you do not wire `channel name in`, this node reads the first channel in the group.

datatype_icon

error in

Error conditions that occur before this node runs.

The node responds to this input according to standard error behavior. For more information, refer to ***Standard Error Behavior***.

Default value: No error

datatype_icon

tdm file out

TDMS file reference on the server.

datatype_icon

group name out

Channel group returned from the TDMS file.

datatype_icon

channel name out

Name of the channel associated with the group you specified.

datatype_icon

data

1D array of timestamp data read from the TDMS file.

datatype_icon

error out

Error information.

The node produces this output according to standard error behavior. For more information, refer to ***Standard Error Behavior***.

datatype_icon

end of channel data?

Boolean that indicates whether the node read the last of the data for the channel in `channel name out`.

datatype_icon

end of groups?

Boolean that indicates whether the node read data from the last group in the TDMS file.

Related information:

- [Standard Error Behavior](#)

NChan Timestamp

Returns timestamp data from the channels specified in `channel names in`.

Inputs/Outputs

`datatype_icon`

count

Maximum number of data elements to read for each channel.

If you use the default value, this node reads all available elements.

Default value: -1

`datatype_icon`

offset

Number of data elements into the TDMS file at which this node begins to read.

Default value: 0

`datatype_icon`

tdm file in

TDMS file reference on the server.

Use Query TDMS Files to obtain the reference.

`datatype_icon`

group name in

Channel group to analyze.

If you do not wire `group name in`, this node reads the channel name from the next group in the TDMS file each time it executes.

datatype_icon

channel names in

Channels to read data from.

If you do not wire `channel name in`, this node reads all channels in the group.

datatype_icon

return channels in file order?

Boolean specifying whether the node returns data channels in the same order as they exist in the TDMS file.

True	Return data channels in the same order as they exist in the TDMS file.
False	Return data channels in the same order as you specify in <code>channel names in</code> .

Default value: False

datatype_icon

error in

Error conditions that occur before this node runs.

The node responds to this input according to standard error behavior.

Standard Error Behavior

Default value: No error

datatype_icon

tdm file out

TDMS file reference on the server.

datatype_icon

group name out

Channel group returned from the TDMS file.

datatype_icon

channel names out

Names of the channels associated with the group you specified.

The order of the names returned correspond to the order of the `data` array when `return channels in file order?` is `True`.

datatype_icon

data

2D array of timestamp data read from the TDMS file.

datatype_icon

error out

Error information.

The node produces this output according to standard error behavior.

Standard Error Behavior

datatype_icon

end of channel data?

A Boolean that indicates whether the node read the last of the data for any channel in `channel names out`.

datatype_icon

end of groups?

Boolean that indicates whether the node read data from the last group in the TDMS file.

1Chan U16

Returns unsigned 16-bit integer data from the channel specified in `channel name in`.

Inputs/Outputs

`datatype_icon`

count

Maximum number of data elements to read for each channel.

If you use the default value, this node reads all available elements.

Default value: -1

`datatype_icon`

offset

Number of data elements into the TDMS file at which this node begins to read.

Default value: 0

`datatype_icon`

tdm file in

TDMS file reference on the server.

Use Query TDMS Files to obtain the reference.

`datatype_icon`

group name in

Channel group to analyze.

If you do not wire `group name in`, this node reads the channel name from the next group in the TDMS file each time it executes.

datatype_icon

channel name in

Channel to read data from.

If you do not wire `channel name in`, this node reads the first channel in the group.

datatype_icon

error in

Error conditions that occur before this node runs.

The node responds to this input according to standard error behavior.

Standard Error Behavior

Default value: No error

datatype_icon

tdm file out

TDMS file reference on the server.

datatype_icon

group name out

Channel group returned from the TDMS file.

datatype_icon

channel name out

Name of the channel associated with the group you specified.

datatype_icon

data

1D array of unsigned 16-bit integer data read from the TDMS file.

datatype_icon

error out

Error information.

The node produces this output according to standard error behavior.

Standard Error Behavior

datatype_icon

end of channel data?

Boolean that indicates whether the node read the last of the data for the channel in `channel name out`.

datatype_icon

end of groups?

Boolean that indicates whether the node read data from the last group in the TDMS file.

NChan U16

Returns unsigned 16-bit integer data from the channels specified in `channel names in`.

Inputs/Outputs

datatype_icon

count

Maximum number of data elements to read for each channel.

If you use the default value, this node reads all available elements.

Default value: -1

datatype_icon

offset

Number of data elements into the TDMS file at which this node begins to read.

Default value: 0

datatype_icon

tdm file in

TDMS file reference on the server.

Use Query TDMS Files to obtain the reference.

datatype_icon

group name in

Channel group to analyze.

If you do not wire `group name in`, this node reads the channel name from the next group in the TDMS file each time it executes.

datatype_icon

channel names in

Channels to read data from.

If you do not wire `channel name in`, this node reads all channels in the group.

`datatype_icon`

return channels in file order?

Boolean specifying whether the node returns data channels in the same order as they exist in the TDMS file.

True	Return data channels in the same order as they exist in the TDMS file.
False	Return data channels in the same order as you specify in <code>channel names in</code> .

Default value: False

`datatype_icon`

error in

Error conditions that occur before this node runs.

The node responds to this input according to standard error behavior.

Standard Error Behavior

Default value: No error

`datatype_icon`

tdm file out

TDMS file reference on the server.

`datatype_icon`

group name out

Channel group returned from the TDMS file.

datatype_icon

channel names out

Names of the channels associated with the group you specified.

The order of the names returned correspond to the order of the `data` array when `return channels in file order?` is `True`.

datatype_icon

data

2D array of unsigned 16-bit integer data read from the TDMS file.

datatype_icon

error out

Error information.

The node produces this output according to standard error behavior.

Standard Error Behavior

datatype_icon

end of channel data?

A Boolean that indicates whether the node read the last of the data for any channel in `channel names out`.

datatype_icon

end of groups?

Boolean that indicates whether the node read data from the last group in the TDMS file.

1Chan U32

Returns unsigned 32-bit integer data from the channel specified in `channel name in`.

Inputs/Outputs

`datatype_icon`

count

Maximum number of data elements to read for each channel.

If you use the default value, this node reads all available elements.

Default value: -1

`datatype_icon`

offset

Number of data elements into the TDMS file at which this node begins to read.

Default value: 0

`datatype_icon`

tdm file in

TDMS file reference on the server.

Use Query TDMS Files to obtain the reference.

`datatype_icon`

group name in

Channel group to analyze.

If you do not wire `group name in`, this node reads the channel name from the next group in the TDMS file each time it executes.

datatype_icon

channel name in

Channel to read data from.

If you do not wire `channel name in`, this node reads the first channel in the group.

datatype_icon

error in

Error conditions that occur before this node runs.

The node responds to this input according to standard error behavior. For more information, refer to ***Standard Error Behavior***.

Default value: No error

datatype_icon

tdm file out

TDMS file reference on the server.

datatype_icon

group name out

Channel group returned from the TDMS file.

datatype_icon

channel name out

Name of the channel associated with the group you specified.

datatype_icon

data

1D array of unsigned 32-bit integer data read from the TDMS file.

datatype_icon

error out

Error information.

The node produces this output according to standard error behavior. For more information, refer to ***Standard Error Behavior***.

datatype_icon

end of channel data?

Boolean that indicates whether the node read the last of the data for the channel in `channel name out`.

datatype_icon

end of groups?

Boolean that indicates whether the node read data from the last group in the TDMS file.

Related information:

- [Standard Error Behavior](#)

NChan U32

Returns unsigned 32-bit integer data from the channels specified in `channel names`

`in.`

Inputs/Outputs

`datatype_icon`

count

Maximum number of data elements to read for each channel.

If you use the default value, this node reads all available elements.

Default value: -1

`datatype_icon`

offset

Number of data elements into the TDMS file at which this node begins to read.

Default value: 0

`datatype_icon`

tdm file in

TDMS file reference on the server.

Use Query TDMS Files to obtain the reference.

`datatype_icon`

group name in

Channel group to analyze.

If you do not wire `group name in`, this node reads the channel name from the next group in the TDMS file each time it executes.

`datatype_icon`

channel names in

Channels to read data from.

If you do not wire `channel name in`, this node reads all channels in the group.

`datatype_icon`

return channels in file order?

Boolean specifying whether the node returns data channels in the same order as they exist in the TDMS file.

True	Return data channels in the same order as they exist in the TDMS file.
False	Return data channels in the same order as you specify in <code>channel names in</code> .

Default value: False

`datatype_icon`

error in

Error conditions that occur before this node runs.

The node responds to this input according to standard error behavior.

Standard Error Behavior

Default value: No error

`datatype_icon`

tdm file out

TDMS file reference on the server.

`datatype_icon`

group name out

Channel group returned from the TDMS file.

`datatype_icon`

channel names out

Names of the channels associated with the group you specified.

The order of the names returned correspond to the order of the `data` array when `return channels in file order?` is `True`.

`datatype_icon`

data

2D array of unsigned 32-bit integer data read from the TDMS file.

`datatype_icon`

error out

Error information.

The node produces this output according to standard error behavior.

Standard Error Behavior

`datatype_icon`

end of channel data?

A Boolean that indicates whether the node read the last of the data for any channel in `channel names out`.

`datatype_icon`

end of groups?

Boolean that indicates whether the node read data from the last group in the TDMS file.

1Chan U8

Returns unsigned 8-bit integer data from the channel specified in `channel_name` in.

Inputs/Outputs

`datatype_icon`

count

Maximum number of data elements to read for each channel.

If you use the default value, this node reads all available elements.

Default value: -1

`datatype_icon`

offset

Number of data elements into the TDMS file at which this node begins to read.

Default value: 0

`datatype_icon`

tdm file in

TDMS file reference on the server.

Use Query TDMS Files to obtain the reference.

`datatype_icon`

group name in

Channel group to analyze.

If you do not wire `group name in`, this node reads the channel name from the next group in the TDMS file each time it executes.

datatype_icon

channel name in

Channel to read data from.

If you do not wire `channel name in`, this node reads the first channel in the group.

datatype_icon

error in

Error conditions that occur before this node runs.

The node responds to this input according to standard error behavior. For more information, refer to ***Standard Error Behavior***.

Default value: No error

datatype_icon

tdm file out

TDMS file reference on the server.

datatype_icon

group name out

Channel group returned from the TDMS file.

datatype_icon

channel name out

Name of the channel associated with the group you specified.

datatype_icon

data

1D array of unsigned 8-bit integer data read from the TDMS file.

datatype_icon

error out

Error information.

The node produces this output according to standard error behavior. For more information, refer to ***Standard Error Behavior***.

datatype_icon

end of channel data?

Boolean that indicates whether the node read the last of the data for the channel in `channel name out`.

datatype_icon

end of groups?

Boolean that indicates whether the node read data from the last group in the TDMS file.

Related information:

- [Standard Error Behavior](#)

NChan U8

Returns unsigned 8-bit integer data from the channels specified in `channel names`

`in.`

Inputs/Outputs

`datatype_icon`

count

Maximum number of data elements to read for each channel.

If you use the default value, this node reads all available elements.

Default value: -1

`datatype_icon`

offset

Number of data elements into the TDMS file at which this node begins to read.

Default value: 0

`datatype_icon`

tdm file in

TDMS file reference on the server.

Use Query TDMS Files to obtain the reference.

`datatype_icon`

group name in

Channel group to analyze.

If you do not wire `group name in`, this node reads the channel name from the next group in the TDMS file each time it executes.

`datatype_icon`

channel names in

Channels to read data from.

If you do not wire `channel name in`, this node reads all channels in the group.

datatype_icon

return channels in file order?

Boolean specifying whether the node returns data channels in the same order as they exist in the TDMS file.

True	Return data channels in the same order as they exist in the TDMS file.
False	Return data channels in the same order as you specify in <code>channel names in</code> .

Default value: False

datatype_icon

error in

Error conditions that occur before this node runs.

The node responds to this input according to standard error behavior. For more information, refer to ***Standard Error Behavior***.

Default value: No error

datatype_icon

tdm file out

TDMS file reference on the server.

datatype_icon

group name out

Channel group returned from the TDMS file.

`datatype_icon`

channel names out

Names of the channels associated with the group you specified.

The order of the names returned correspond to the order of the `data` array when `return channels in file order?` is `True`.

`datatype_icon`

data

2D array of unsigned 8-bit integer data read from the TDMS file.

`datatype_icon`

error out

Error information.

The node produces this output according to standard error behavior. For more information, refer to ***Standard Error Behavior***.

`datatype_icon`

end of channel data?

A Boolean that indicates whether the node read the last of the data for any channel in `channel names out`.

`datatype_icon`

end of groups?

Boolean that indicates whether the node read data from the last group in the TDMS

file.

Related information:

- [Standard Error Behavior](#)

1Chan Waveform

Returns waveform data from the channel specified in `channel name in`.

Inputs/Outputs

`datatype_icon`

count

Maximum number of data elements to read for each channel.

If you use the default value, this node reads all available elements.

Default value: -1

`datatype_icon`

offset

Number of data elements into the TDMS file at which this node begins to read.

Default value: 0

`datatype_icon`

tdm file in

TDMS file reference on the server.

Use Query TDMS Files to obtain the reference.

`datatype_icon`

group name in

Channel group to analyze.

If you do not wire `group name in`, this node reads the channel name from the next group in the TDMS file each time it executes.

datatype_icon

channel name in

Channel to read data from.

If you do not wire `channel name in`, this node reads the first channel in the group.

datatype_icon

error in

Error conditions that occur before this node runs.

The node responds to this input according to standard error behavior. For more information, refer to ***Standard Error Behavior***.

Default value: No error

datatype_icon

tdm file out

TDMS file reference on the server.

datatype_icon

group name out

Channel group returned from the TDMS file.

datatype_icon

channel name out

Name of the channel associated with the group you specified.

datatype_icon

data

Waveform data read from the TDMS file.

datatype_icon

error out

Error information.

The node produces this output according to standard error behavior. For more information, refer to ***Standard Error Behavior***.

datatype_icon

end of channel data?

Boolean that indicates whether the node read the last of the data for the channel in `channel name out`.

datatype_icon

end of groups?

Boolean that indicates whether the node read data from the last group in the TDMS file.

Related information:

- [Standard Error Behavior](#)

NChan Waveform

Returns waveform data from the channels specified in `channels name in`.

Inputs/Outputs

datatype_icon

count

Maximum number of data elements to read for each channel.

If you use the default value, this node reads all available elements.

Default value: -1

datatype_icon

offset

Number of data elements into the TDMS file at which this node begins to read.

Default value: 0

datatype_icon

tdm file in

TDMS file reference on the server.

Use Query TDMS Files to obtain the reference.

datatype_icon

group name in

Channel group to analyze.

If you do not wire `group name in`, this node reads the channel name from the next group in the TDMS file each time it executes.

datatype_icon

channel names in

Channels to read data from.

If you do not wire `channel name in`, this node reads all channels in the group.

datatype_icon

return channels in file order?

Boolean specifying whether the node returns data channels in the same order as they exist in the TDMS file.

True	Return data channels in the same order as they exist in the TDMS file.
False	Return data channels in the same order as you specify in <code>channel names in</code> .

Default value: False

datatype_icon

error in

Error conditions that occur before this node runs.

The node responds to this input according to standard error behavior. For more information, refer to ***Standard Error Behavior***.

Default value: No error

datatype_icon

tdm file out

TDMS file reference on the server.

datatype_icon

group name out

Channel group returned from the TDMS file.

datatype_icon

channel names out

Names of the channels associated with the group you specified.

The order of the names returned correspond to the order of the `data` array when `return channels in file order?` is `True`.

datatype_icon

data

1D array of waveform data read from the TDMS file.

datatype_icon

error out

Error information.

The node produces this output according to standard error behavior. For more information, refer to ***Standard Error Behavior***.

datatype_icon

end of channel data?

A Boolean that indicates whether the node read the last of the data for any channel in `channel names out`.

datatype_icon

end of groups?

Boolean that indicates whether the node read data from the last group in the TDMS

file.

Related information:

- [Standard Error Behavior](#)

Utility Functions

Get information about systems connected to your server.



Note G Web Development Software does not support this API.

[SystemLink Nodes and Functions Reference for NXG and G Web Development Software](#)

Use SystemLink nodes in NXG or G Web Development Software to communicate data between your test and measurement systems and your SystemLink Server account.

[Get Minion ID](#)

Obtains the minion ID of the Salt system, if one is available.

[Get System Name](#)

Obtains the name of the system, if one is available.

[API Reference](#)

G, HTTP, Python, and .NET APIs allow you to programmatically access SystemLink data services outside of NI SystemLink Web Application.

[SystemLink Nodes and Functions Reference for NXG and G Web Development Software](#)

Use SystemLink nodes in NXG or G Web Development Software to communicate data between your test and measurement systems and your SystemLink Server account.

Get Minion ID

Obtains the minion ID of the Salt system, if one is available.

If a minion ID is not available, this VI returns an empty string.



Note G Web Development Software does not support this function.

Inputs/Outputs

datatype_icon

minion id

Minion ID of the Salt system.

Get System Name

Obtains the name of the system, if one is available.

If a system name is not available, this VI returns an empty string.



Note G Web Development Software does not support this function.

Inputs/Outputs

datatype_icon

system name

Name of the system.

SystemLink VIs and Functions Reference for LabVIEW

Use SystemLink functions in LabVIEW to communicate data between your test and measurement systems and your SystemLink Server account.

SystemLink data services allow you to efficiently manage and monitor your tests, assets, and systems data from anywhere in the world. Refer to the following list to select which SystemLink API you want to learn more about.

Configuration Functions

Manage the HTTP or AMQP connection with your server.

File Transfer Functions

Upload files to your SystemLink Server, where you can also update, delete, or download them.

Tag Functions

Send pieces of data between distributed systems. Use tags to send and receive data from one system to other systems, or SystemLink Server.

Messages Functions

Communicate as a network queue using the publish-subscribe model. Use messages to send commands, status updates, and data between systems and applications.

Test Monitor Functions

Manage and communicate your test results and test steps between your test systems and SystemLink Server.

Utility Functions

Obtain system names and minion IDs to assign tag paths and properties to your data.

Alarm Functions

Create alarms to implement rules and manage conditions affecting your systems, tests, or assets.

Asset Utilization Functions

Track how long and how often you use your assets in test systems.

Configuration Functions

Manage the connection between SystemLink data services and SystemLink.

Open Configuration

Initializes an HTTP or AMQP connection for SystemLink data services to communicate with SystemLink Server.

Close Configuration

Terminates connections associated with the configuration and invalidates open references to SystemLink objects on SystemLink Server.

Use Workspace

Specifies the active workspace a VI that references this configuration uses to perform operations.

List Workspaces

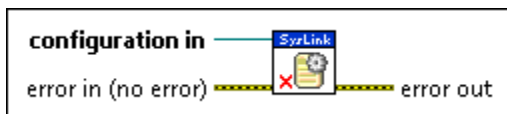
Lists workspaces available on the SystemLink server.

Close Configuration

Terminates connections associated with the configuration and invalidates open references to SystemLink objects on SystemLink Server.

Programming Patterns

- Transferring Data Using Tags
- Sending Messages Between Systems



configuration in

The connection to the server.

error in

Error conditions that occur before this VI runs. Unlike most VIs, this VI runs normally even if an error occurred before this VI runs.

error out

Error information. This VI produces this output according to [standard error behavior](#).

Open Configuration

Initializes an HTTP connection for SystemLink data services to communicate with SystemLink Server.

Open Configuration (HTTP Auto)

Initializes an HTTP connection for SystemLink data services to communicate with your SystemLink Server by automatically retrieving credentials when executed on a managed system.

Open Configuration (HTTP)

Initializes an HTTP connection for SystemLink data services to communicate with your SystemLink Server using your SystemLink Server login credentials.

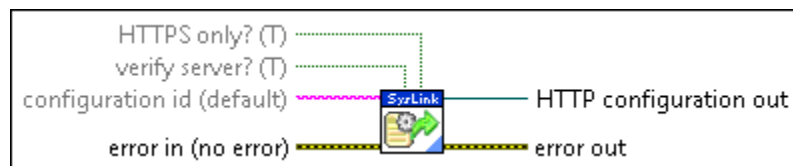
Open Configuration (HTTP Auto)

Initializes an HTTP connection for SystemLink data services to communicate with your SystemLink Server by automatically retrieving credentials when executed on a managed system.

This VI runs on SystemLink Server without additional configuration information.

Programming Patterns

- Transferring Data Using Tags
- Sending Messages Between Systems
- Acquiring Test Results
- Transferring Files from Disk to the Server
- Transferring Files from Memory to the Server



configuration id

Name of the configuration you want to connect to the SystemLink master or SystemLink Server. If you leave the `configuration id` unwired, Open Configuration uses the default configuration.

error in

Error conditions that occur before this VI runs. The VI responds to this input according to [standard error behavior](#).

verify server?

A Boolean that determines whether this VI will verify the connection is secured by a valid certificate. The default value is True.

This input also allows local connections over HTTP in addition to HTTPS connections.

HTTPS only?

A Boolean that determines whether to use only an HTTPS connection. If set to True, your data services only communicate with the SystemLink Server over an HTTPS connection. The default value is True.



Note If the server is configured to use HTTP instead of HTTPS, you have to set the value to False or this VI returns an error. Furthermore, using HTTP remotely insecurely exposes credentials and data to the network.

HTTP configuration out

HTTP connection to the server.

error out

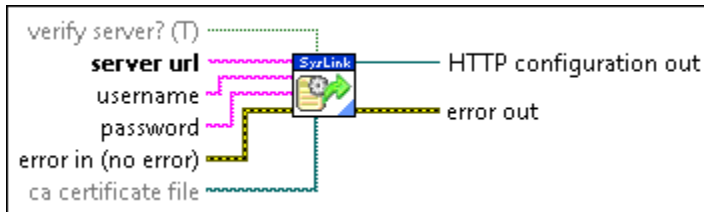
Error information. This VI produces this output according to [standard error behavior](#).

Open Configuration (HTTP)

Initializes an HTTP connection for SystemLink data services to communicate with your SystemLink Server using your SystemLink Server login credentials.

Programming Patterns

- Transferring Data Using Tags
- Sending Messages Between Systems
- Acquiring Test Results
- Transferring Files from Disk to the Server
- Transferring Files from Memory to the Server



verify server?

A Boolean that determines whether this VI will verify the connection is secured by a valid certificate. The default value is True.

This input also allows local connections over HTTP in addition to HTTPS connections.

server url

URL of the server where this VI sends the web request. If you leave `server url` unwired, this VI uses `http://localhost:80` as the default URL.

The following table describes how to use the proper URL syntax for the `server url` input.

Part	Definition	Example
Scheme	Protocol the web service uses.	<code>http://</code>
Host	Domain name or IP address of the service.	<code>localhost</code>
Port	HTTP port of the web service. If you don't specify a port, the web service uses the default port.	<code>80</code>

Use the following syntax to prompt a connection to the SystemLink Server:

`http://host:<port>`.

username

Login username for the server.

password

Login password for the server.

error in

Error conditions that occur before this VI runs. The VI responds to this input according to [standard error behavior](#).

ca certificate file

File path to where a custom or private CA certificate (`.cer`) is located on the system. The CA certificate authenticates a secure sockets layer to communicate data between your application and the server. If you leave this input unwired, the VI uses the system defaults, which includes public CA certificates.

HTTP configuration out

HTTP connection to the server.

error out

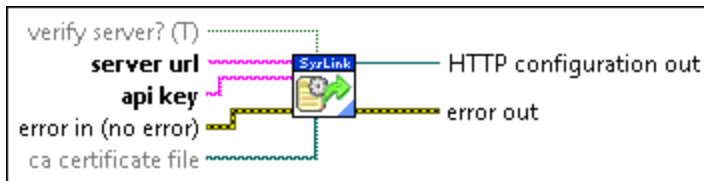
Error information. This VI produces this output according to [standard error behavior](#).

Open Configuration (API Key)

Initializes an HTTP connection for SystemLink data services to communicate with SystemLink using an API key for authentication. The VI authenticates the connection with an API key obtained from SystemLink.

Programming Patterns

- Transferring Data Using Tags
- Sending Messages Between Systems
- Acquiring Test Results
- Transferring Files from Disk to the Server
- Transferring Files from Memory to the Server



verify server?

A Boolean that determines whether this VI will verify the connection is secured by a valid certificate. The default value is True.

This input also allows local connections over HTTP in addition to HTTPS connections.

server url

URL of the server where this VI sends the web request. If you leave `server url` unwired, this VI uses `http://localhost:80` as the default URL.

The following table describes how to use the proper URL syntax for the `server url` input.

Part	Definition	Example
Scheme	Protocol the web service uses.	<code>http://</code>
Host	Domain name or IP address of the service.	<code>localhost</code>
Port	HTTP port of the web service. If you don't specify a port, the web service uses the default port.	<code>80</code>

api key

Unique passcode the VI uses to authenticate communication between the application and SystemLink.

error in

Error conditions that occur before this VI runs. The VI responds to this input according to [standard error behavior](#).

ca certificate file

File path to where a custom or private CA certificate (`.cer`) is located on the system.

The CA certificate authenticates a secure sockets layer to communicate data between your application and the server. If you leave this input unwired, the VI uses the system defaults, which includes public CA certificates.

HTTP configuration out

HTTP connection to the server.

error out

Error information. This VI produces this output according to [standard error behavior](#).

Use Workspace

Specifies the active workspace a VI that references this configuration uses to perform operations. The active workspace determines what resources the node can access on the SystemLink server and where new resources are created.

Use Workspace (Name)

Specifies, by name, the workspace that any VI referencing this configuration uses to perform operations..

Use Workspace (Instance)

Specifies, by instance, the workspace that any VI referencing this configuration uses to perform operations.

Use Workspace (Name)

Specifies, by name, the workspace that any VI referencing this configuration uses to perform operations. The active workspace determines what resources the VI can access on the SystemLink server and where new resources are created.



HTTP configuration in

Configuration to communicate with the server over HTTP.

workspace name

Name of the workspace you want to use. The workspace you specify determines what resources the VI can access on the SystemLink server and where new resources are created.

error in

Error conditions that occur before this VI runs. The VI responds to this input according to [standard error behavior](#).

HTTP configuration out

Configuration to communicate with the server over HTTP.

error out

Error information. This VI produces this output according to [standard error behavior](#).

Use Workspace (Instance)

Specifies, by workspace instance, the workspace that any VI referencing this configuration uses to perform operations. The active workspace determines what resources the VI can access on the SystemLink server and where new resources are created.



HTTP configuration in

Configuration to communicate with the server over HTTP.

workspace

Instance of the workspace you want to use. The workspace you specify determines what resources the VI can access on the SystemLink server and where new resources

are created.

error in

Error conditions that occur before this VI runs. The VI responds to this input according to [standard error behavior](#).

HTTP configuration out

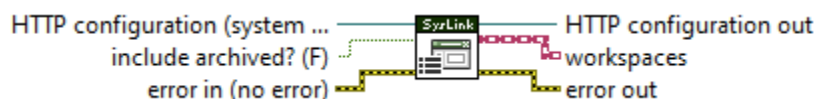
Configuration to communicate with the server over HTTP.

error out

Error information. This VI produces this output according to [standard error behavior](#).

List Workspaces

Lists workspaces available on the SystemLink server. SystemLink uses workspaces to organize and control access to data on the server.



configuration

Name of the configuration you want to connect to the SystemLink master or SystemLink Server. If you leave the `configuration id` unwired, Open Configuration uses the default configuration.

include archived?

Boolean that specifies whether or not to include archived workspaces in the list. The default value is False.

error in

Error conditions that occur before this VI runs. The VI responds to this input according to [standard error behavior](#).

HTTP configuration out

HTTP connection to the server.

workspaces

Workspaces this VI can access using the current configuration. By default, this VI does not return archived workspaces.

error out

Error conditions that occur after this VI runs. The VI responds to this input according to [standard error behavior](#).

Tag Functions

Use tags to read and write data values between systems and SystemLink Server.

[Build Path](#)

Identifies the system and top-level application that generates the tag as a tag path.

[Open Tag](#)

Opens a reference to a tag on the server. If the tag does not exist on the server, this VI creates it.

[Query Tags](#)

Searches the server for tags matching the criteria you specify.

[Close Tag](#)

Closes a tag reference or an array of tag references.

[Read Tag](#)

Returns the value of the tag. The data type of the tag must match the polymorphic instance.

[Write Tag](#)

Specifies the value of the tag.

Reset Aggregates

Clears the aggregate information for a tag reference or group of tag references.

Delete Tag

Removes a tag or group of tags on the server.

Tag Properties

Returns information about an open tag reference so you can modify its configuration.

Tag History Functions

Retain historical tag values to review, troubleshoot, or analyze data over time.

Advanced Tag Functions

Read or write multiple times in a single call to improve performance.

Build Path

Identifies the system and top-level application that generates the tag as a tag path.

If you leave `namespace` unwired, it defaults to the name of the top-level VI.



tag name

Name you give the tag.

namespace

Name of the resource or application that generates the tag.

path

Path to the tag reference.

A tag path uses dot separators to separate each component of the path. For example, a tag path looks like `<system>.<namespace>.<tag>`.

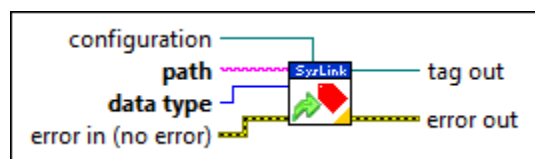
Open Tag

Opens a reference to a tag on the server. If the tag does not exist on the server, this VI creates it.

Use Query Tags to open existing tags of unknown data types.

Programming Patterns

- Transferring Data Using Tags



configuration

Connection to the server.

If you leave this input unwired, this VI uses the default connection. It automatically retrieves your SystemLink Server credentials when you execute the application on a managed system or on the server.

data type

Data type of the tag reference.

If a tag with the same path but different data type already exists, Open Tag returns an error. Use Query Tags to open existing tag references of unknown data types.

path

Path to the tag to open.

error in

Error conditions that occur before this VI runs. The VI responds to this input according to [standard error behavior](#).

tag out

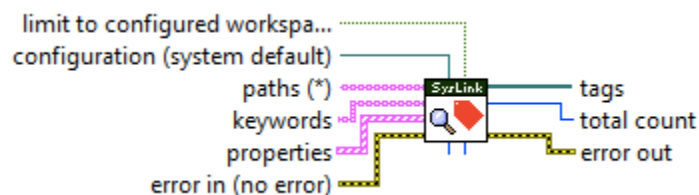
Reference to a specific tag on SystemLink Server.

error out

Error information. This VI produces this output according to [standard error behavior](#).

Query Tags

Searches the server for tags matching the criteria you specify.



configuration

Connection to the server.

If you leave this input unwired, this VI uses the default connection. It automatically retrieves your SystemLink Server credentials when you execute the application on a managed system or on the server.

paths

Tag search paths to include in the query. Each search path may contain wildcard (*) characters to match multiple tags that share a prefix or suffix.

For example, enter `<System>.*` to obtain all tags for a system or `*.<Application>.*` to get all tags for a single application across all systems.

keywords

Words associated with a tag. The server returns only tags that have all of the provided keywords.

properties

Key-value pairs associated with a tag. The server returns only tags with all of the provided properties.

error in

Error conditions that occur before this VI runs. The VI responds to this input according to [standard error behavior](#).

skip

Number of tags to skip.

If `skip` and `take` are left empty, the query returns the first 1,000 results by default.

take

Number of tags requested from the server.

If `skip` and `take` are left empty, the query returns the first 1,000 results by default.

limit to configured workspace?

Boolean that determines whether this VI queries only for tags in the workspace you specify in `configuration`. By default, the VI queries for tags from all accessible workspaces.

tags

References to the tags matched by the query on SystemLink Server.

totalCount

Total number of tags matched by the query before taking `skip` and `take` into account on SystemLink Server.

error out

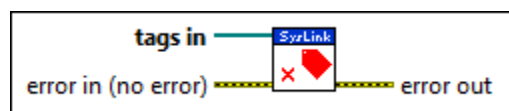
Error information. This VI produces this output according to [standard error behavior](#).

Close Tag

Closes a tag reference or an array of tag references.

Programming Patterns

- Transferring Data Using Tags



tags in

Reference to a specific tag or a group of tags on SystemLink Server.

Use Open Tag or Query Tags to obtain this reference.

error in

Error conditions that occur before this VI runs. Unlike most VIs, this VI runs normally even if an error occurred before this VI runs.

error out

Error information. This VI produces this output according to [standard error behavior](#).

Read Tag

Returns the value of the tag. The data type of the tag must match the polymorphic instance.

[Read Tag \(DBL\)](#)

Returns the value of the tag. The data type of the tag must be a double-precision, floating-point number.

[Read Tag \(I32\)](#)

Returns the value of the tag. The data type of the tag must be a 32-bit integer.

[Read Tag \(U64\)](#)

Returns the value of the tag. The data type of the tag must be an unsigned 64-bit integer.

[Read Tag \(String\)](#)

Returns the value of the tag. The data type of the tag must be a string.

[Read Tag \(Boolean\)](#)

Returns the value of the tag. The data type of the tag must be a Boolean.

[Read Tag \(Timestamp\)](#)

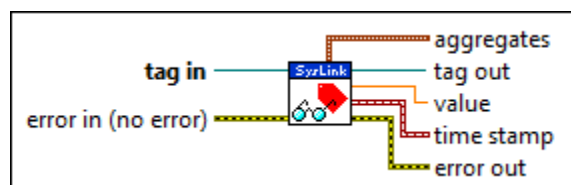
Returns the value of the tag. The data type of the tag must be a timestamp.

Read Tag (DBL)

Returns the value of the tag. The data type of the tag must be a double-precision, floating-point number.

Programming Patterns

- Transferring Data Using Tags



tag in

Reference to a tag on the server.

Use Open Tag or Query Tags to obtain this reference.

error in

Error conditions that occur before this VI runs. The VI responds to this input according to [standard error behavior](#).

aggregates

Values the tag collects since calling Reset Aggregates. If a tag is set to collect aggregates, the tag updates the aggregate every time you write to it.

tag out

Reference to a specific tag on SystemLink Server.

Use Open Tag or Query Tags to obtain this reference.

value

Data value of the tag.

time stamp

Date and time associated with the most recent data value.

error out

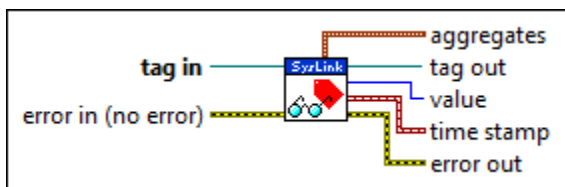
Error information. This VI produces this output according to [standard error behavior](#).

Read Tag (I32)

Returns the value of the tag. The data type of the tag must be a 32-bit integer.

Programming Patterns

- Transferring Data Using Tags



tag in

Reference to a tag on the server.

Use Open Tag or Query Tags to obtain this reference.

error in

Error conditions that occur before this VI runs. The VI responds to this input according to [standard error behavior](#).

aggregates

Values the tag collects since calling Reset Aggregates. If a tag is set to collect aggregates, the tag updates the aggregate every time you write to it.

tag out

Reference to a specific tag on SystemLink Server.

Use Open Tag or Query Tags to obtain this reference.

value

Data value of the tag or tags.

time stamp

Date and time associated with the most recent data value.

error out

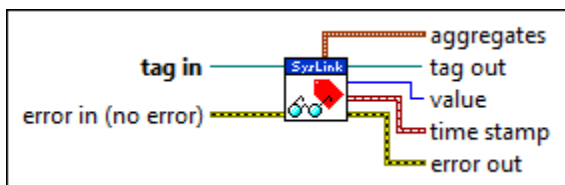
Error information. This VI produces this output according to [standard error behavior](#).

Read Tag (U64)

Returns the value of the tag. The data type of the tag must be an unsigned 64-bit integer.

Programming Patterns

- Transferring Data Using Tags



tag in

Reference to a tag on the server.

Use Open Tag or Query Tags to obtain this reference.

error in

Error conditions that occur before this VI runs. The VI responds to this input according to [standard error behavior](#).

aggregates

Values the tag collects since calling Reset Aggregates. If a tag is set to collect aggregates, the tag updates the aggregate every time you write to it.

tag out

Reference to a specific tag on SystemLink Server.

Use Open Tag or Query Tags to obtain this reference.

value

Data value of the tag or tags.

time stamp

Date and time associated with the most recent data value.

error out

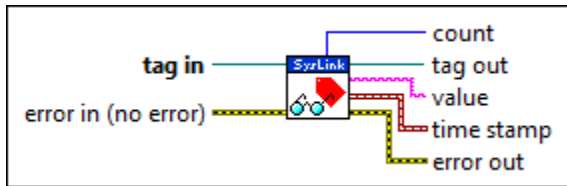
Error information. This VI produces this output according to [standard error behavior](#).

Read Tag (String)

Returns the value of the tag. The data type of the tag must be a string.

Programming Patterns

- Transferring Data Using Tags



tag in

Reference to a tag on the server.

Use Open Tag or Query Tags to obtain this reference.

error in

Error conditions that occur before this VI runs. The VI responds to this input according to [standard error behavior](#).

count

Total number of times a system or group of systems writes to a tag after calling Reset Aggregates the last time.



Note The tag only tracks this number if Collect Aggregates is set to true.

tag out

Reference to a specific tag on SystemLink Server.

Use Open Tag or Query Tags to obtain this reference.

value

Data value of the tag or tags.

time stamp

Date and time associated with the most recent data value.

error out

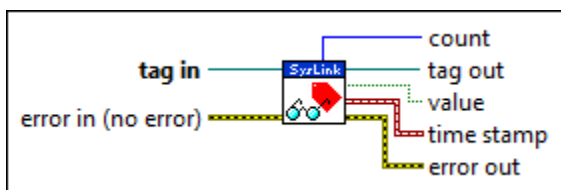
Error information. This VI produces this output according to [standard error behavior](#).

Read Tag (Boolean)

Returns the value of the tag. The data type of the tag must be a Boolean.

Programming Patterns

- Transferring Data Using Tags



tag in

Reference to a tag on the server.

Use Open Tag or Query Tags to obtain this reference.

error in

Error conditions that occur before this VI runs. The VI responds to this input according to [standard error behavior](#).

count

Total number of times a system or group of systems writes to a tag after calling Reset Aggregates the last time.



Note The tag only tracks this number if Collect Aggregates is set to true.

tag out

Reference to a specific tag on SystemLink Server.

Use Open Tag or Query Tags to obtain this reference.

value

Data value of the tag or tags.

time stamp

Date and time associated with the most recent data value.

error out

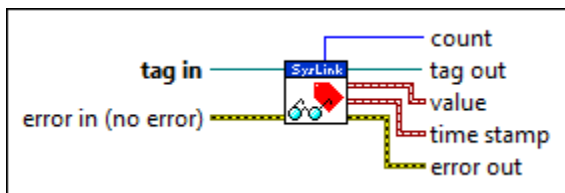
Error information. This VI produces this output according to [standard error behavior](#).

Read Tag (Timestamp)

Returns the value of the tag. The data type of the tag must be a timestamp.

Programming Patterns

- Transferring Data Using Tags



tag in

Reference to a tag on the server.

Use Open Tag or Query Tags to obtain this reference.

error in

Error conditions that occur before this VI runs. The VI responds to this input according to [standard error behavior](#).

count

Total number of times a system or group of systems writes to a tag after calling Reset Aggregates the last time.



Note The tag only tracks this number if Collect Aggregates is set to true.

tag out

Reference to a specific tag on SystemLink Server.

Use Open Tag or Query Tags to obtain this reference.

value

Data value of the tag or tags.

time stamp

Date and time associated with the most recent data value.

error out

Error information. This VI produces this output according to [standard error behavior](#).

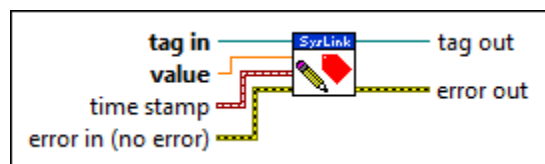
Write Tag

Specifies the value of the tag.

The data type of the tag must match the polymorphic instance of the VI.

Programming Patterns

- Transferring Data Using Tags



tag in

Reference to a tag on the server.

Use Open Tag or Query Tags to obtain this reference.

value

Data value of the tag. Wire a constant or control to this input of the same data type as the polymorphic instance.

time stamp

Date and time to associate with the data value.

If you leave `time stamp` unwired, the server assigns a date and time when it receives the value. NI recommends you do not use this input in most cases. Only use `time stamp` when the value is associated with a measurement or something that occurred at a specific time.

error in

Error conditions that occur before this VI runs. The VI responds to this input according to [standard error behavior](#).

tag out

Reference to a specific tag on SystemLink Server.

Use Open Tag or Query Tags to obtain this reference.

error out

Error information. This VI produces this output according to [standard error behavior](#).

Reset Aggregates

Clears the aggregate information for a tag reference or group of tag references.

Reset Aggregates (Multiple)

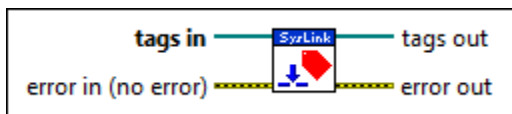
Clears the aggregate information for a group of tag references.

Reset Aggregates (Single)

Clears the aggregate information for a tag reference.

Reset Aggregates (Multiple)

Clears the aggregate information for a group of tag references.



tags in

Reference to a specific tag or a group of tags on SystemLink Server.

Use Open Tag or Query Tags to obtain this reference.

error in

Error conditions that occur before this VI runs. The VI responds to this input according to [standard error behavior](#).

tags out

Reference to a specific tag or group of tags on SystemLink Server.

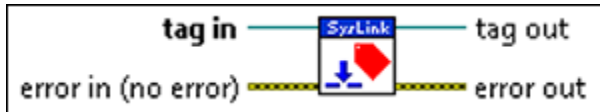
Use Open Tag or Query Tags to obtain this reference.

error out

Error information. This VI produces this output according to [standard error behavior](#).

Reset Aggregates (Single)

Clears the aggregate information for a tag reference.



tag in

Reference to a tag on the server.

Use Open Tag or Query Tags to obtain this reference.

error in

Error conditions that occur before this VI runs. The VI responds to this input according to [standard error behavior](#).

tag out

Reference to a specific tag on SystemLink Server.

Use Open Tag or Query Tags to obtain this reference.

error out

Error information. This VI produces this output according to [standard error behavior](#).

Delete Tag

Removes a tag or group of tags on the server.

Delete Tag (Multiple)

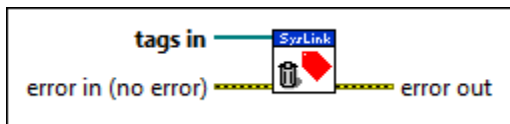
Removes a group of tags on the server.

Delete Tag (Single)

Removes a tag on the server.

Delete Tag (Multiple)

Removes a group of tags on the server.



tags in

Reference to a specific tag or a group of tags on SystemLink Server.

Use Open Tag or Query Tags to obtain this reference.

error in

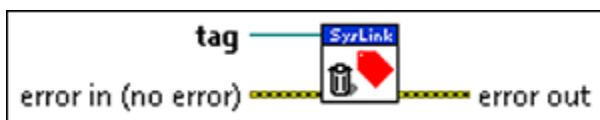
Error conditions that occur before this VI runs. Unlike most VIs, this VI closes tag references on the server even if an error occurred before this VI runs. However, this VI does not delete tag references from the server if an error occurred before this VI runs.

error out

Error information. This VI produces this output according to [standard error behavior](#).

Delete Tag (Single)

Removes a tag on the server.



tag

Reference to a specific tag on SystemLink Server.

Use Open Tag or Query Tags to obtain this reference.

error in

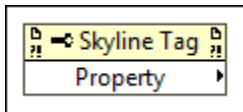
Error conditions that occur before this VI runs. Unlike most VIs, this VI closes tag references on the server even if an error occurred before this VI runs. However, this VI does not delete tag references from the server if an error occurred before this VI runs.

error out

Error information. This VI produces this output according to [standard error behavior](#).

Tag Properties

Returns information about an open tag reference so you can modify its configuration.



reference

Control reference associated with the object whose properties you want to configure.

error in

Error conditions that occur before this VI runs. The VI responds to this input according to [standard error behavior](#).

reference out

Control reference specified by `reference`. If you wrote values to this VI, the value of `reference out` may differ from `reference in`.

error out

Error information. This VI produces this output according to [standard error behavior](#).

Advanced Tag Functions

Read or write multiple times in a single call to improve performance.

[Generate Tag Cluster](#)

Creates or adds to an array of tag clusters, which specifies the tag data values for Multi Write to communicate.

[Multi Write](#)

Specifies values for multiple tags in a single operation.

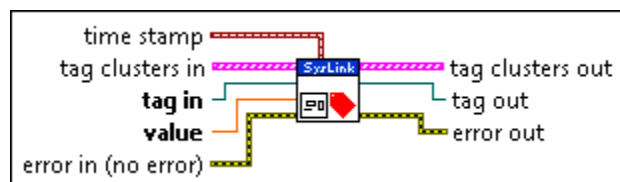
[Multi Read](#)

Returns the value of an array of tags in a single operation.

Generate Tag Cluster

Creates or adds to an array of tag clusters, which specifies the tag data values for Multi Write to communicate.

The data type of the tag must match the polymorphic instance of the VI.



time stamp

Date and time to associate with the data value.

If you leave `time stamp` unwired, the server assigns a date and time when it receives

the value. NI recommends you do not use this input in most cases. Only use `timestamp` when the value is associated with a measurement or something that occurred at a specific time.

tag clusters in

Clusters of tag write information from Generate Tag Cluster.

tag in

Reference to a tag on the server.

Use Open Tag or Query Tags to obtain this reference.

value

Data value of the tag. Wire a constant or control to this input of the same data type as the polymorphic instance.

error in

Error conditions that occur before this VI runs. The VI responds to this input according to [standard error behavior](#).

tag clusters out

Clusters of information to write to tags. Wire this output to Generate Tag Cluster to add additional values before wiring it to Multi Write, which sends all the values to the server at the same time.

tag out

Reference to a specific tag on SystemLink Server.

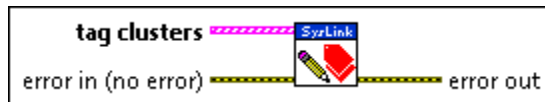
Use Open Tag or Query Tags to obtain this reference.

error out

Error information. This VI produces this output according to [standard error behavior](#).

Multi Write

Specifies values for multiple tags in a single operation.



tag clusters

Clusters of tag write information output from Generate Tag Cluster.

error in

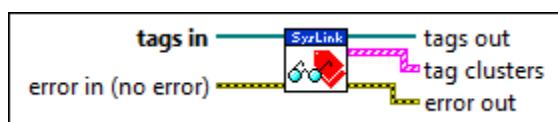
Error conditions that occur before this VI runs. The VI responds to this input according to [standard error behavior](#).

error out

Error information. This VI produces this output according to [standard error behavior](#).

Multi Read

Returns the value of an array of tags in a single operation.



tags in

Reference to a specific tag or a group of tags on SystemLink Server.

Use Open Tag or Query Tags to obtain this reference.

error in

Error conditions that occur before this VI runs. The VI responds to this input according

to [standard error behavior](#).

tags out

Reference to a specific tag or group of tags on SystemLink Server.

Use Open Tag or Query Tags to obtain this reference.

tag clusters

Clusters of tag value information.

error out

Error information. This VI produces this output according to [standard error behavior](#).

Tag Historian Functions

Retain historical tag values to review, troubleshoot, or analyze data over time.



Note The 19.5 Tag Historian API does not work with 19.6 SystemLink Server or later. However, the 19.6 Tag Historian API works with all versions of SystemLink as well as the 19.5 Tag Historian API.

[Configure Tag History Window](#)

Defines the range of historical tag values to read with Read Tag History or Multi Read Tag History.

[Read Tag History](#)

Returns the historical values from a single tag. The data type of the tag must match the polymorphic instance.

[Multi Read History](#)

Returns the historical values from multiple tags in a single operation.

[Configure Retention](#)

Defines how long a tag retains historical values.

Configure Window

Define the range of historical tag values to read with Read Tag History or Multi Read Tag History.

Configure Window (Undecimated)

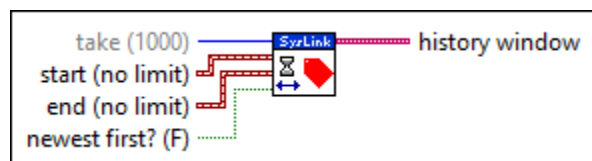
Defines the range of historical tag values to read with Read Tag History or Multi Read Tag History.

Configure Window (Decimated)

Defines the range of historical tag values to read with Read Tag History or Multi Read Tag History and decimates at set time intervals.

Configure Window (Undecimated)

Defines the range of historical tag values to read with Read Tag History or Multi Read Tag History. The default is to read the oldest 1,000 historical values.



take

Number of historical values requested from the server.

If this input is empty, Read Tag History returns the first 1,000 results by default.

start

The time that limits the returned historical values to those written at or after the given

time.

end

The time that limits the returned historical values to those written at or before the given time.

newest first?

A Boolean that determines whether to return the newest or oldest historical tag values first.

The default is False, which returns the oldest historical values first.

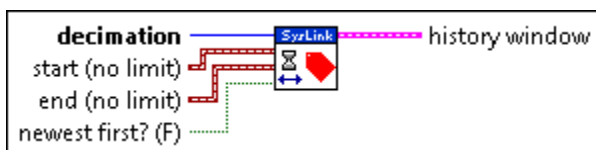
history window

The configured time range to read historical tag values.

Wire `history window` to Read Tag History or Multi Read History to define the historical values to return.

Configure Window (Decimated)

Defines the range of historical tag values to read with Read Tag History or Multi Read Tag History and decimates at set time intervals.



decimation

The approximate number of intervals to divide the historical tag values in the set time range.

start

The time that limits the returned historical values to those written at or after the given time.

end

The time that limits the returned historical values to those written at or before the given time.

newest first?

A Boolean that determines whether to return the newest or oldest historical tag values first.

The default is False, which returns the oldest historical values first.

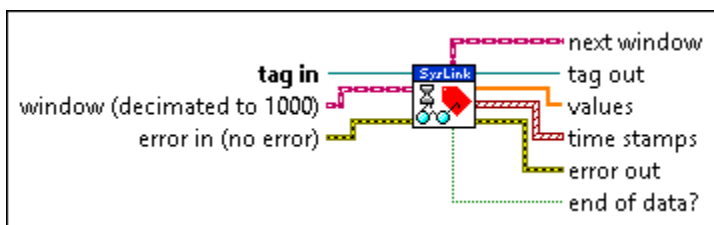
history window

The configured time range to read historical tag values.

Wire `history window` to Read Tag History or Multi Read History to define the historical values to return.

Read Tag History

Returns the historical values from a single tag. The data type of the tag must match the polymorphic instance.



tag in

Reference to a tag on the server.

Use Open Tag or Query Tags to obtain this reference.

window

Time range you specify to read historical tag values.

The default is set to read all values decimated to 1,000.

error in

Error conditions that occur before this VI runs. The VI responds to this input according to [standard error behavior](#).

next window

Time range you specify to read historical tag values.

The default is set to read all values decimated to 1,000.

tag out

Reference to a specific tag on SystemLink Server.

Use Open Tag or Query Tags to obtain this reference.

values

Data values of the tag history. Wire this output to a constant or control with the same data type as the polymorphic instance.

time stamps

Dates and times the server obtained the most recent historical data values.

error out

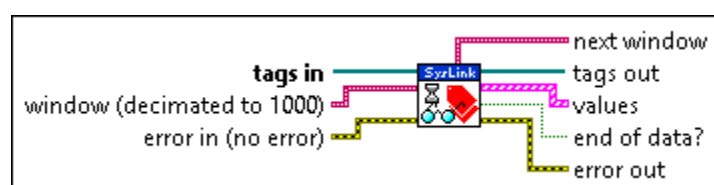
Error information. This VI produces this output according to [standard error behavior](#).

end of data?

Boolean that indicates whether the node read the last of the data for the tag.

Multi Read History

Returns the historical values from multiple tags in a single operation.



tags in

Reference to a specific tag or a group of tags on SystemLink Server.

Use Open Tag or Query Tags to obtain this reference.

window

Time range you specify to read historical tag values.

The default is set to read all values decimated to 1,000.

error in

Error conditions that occur before this VI runs. The VI responds to this input according to [standard error behavior](#).

next window

Time range you specify to read historical tag values.

The default is set to read all values decimated to 1,000.

tags out

Reference to a specific tag or group of tags on SystemLink Server.

Use Open Tag or Query Tags to obtain this reference.

values

Historical data values of the tag(s).

end of data?

Boolean that indicates whether the node read the last of the data for the tag.

error out

Error information. This VI produces this output according to [standard error behavior](#).

Configure Retention

Defines how long a tag retains historical values.

Configure Retention (Duration)

Creates a retention policy for historical tag values to expire after a set number of days they were written.

Configure Retention (Count)

Creates a retention policy to specify the maximum number of tags stored in tag history.

Configure Retention (None)

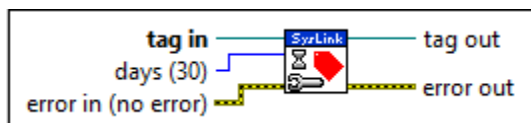
Creates a retention policy to never retain historical tag values.

Configure Retention (Permanent)

Creates a retention policy to retain all historical tag values.

Configure Retention (Duration)

Creates a retention policy for historical tag values to expire after a set number of days they were written.



tag in

Reference to a tag on the server.

Use Open Tag or Query Tags to obtain this reference.

days

The number of days tag values stay in the tag history before they expire.

The default is 30 days.

error in

Error conditions that occur before this VI runs. The VI responds to this input according to [standard error behavior](#).

tag out

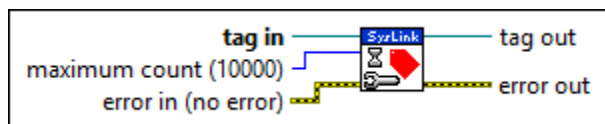
Reference to a specific tag on SystemLink Server.

error out

Error information. This VI produces this output according to [standard error behavior](#).

Configure Retention (Count)

Creates a retention policy to specify the maximum number of tags stored in tag history.



tag in

Reference to a tag on the server.

Use Open Tag or Query Tags to obtain this reference.

maximum count

The number of historical tag values to retain.

The default is 10,000.

error in

Error conditions that occur before this VI runs. The VI responds to this input according to [standard error behavior](#).

tag out

Reference to a specific tag on SystemLink Server.

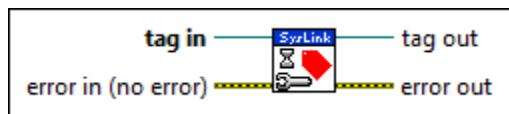
error out

Error information. This VI produces this output according to [standard error behavior](#).

Configure Retention (None)

Creates a retention policy to never retain historical tag values.

This is the default retention policy assigned to new tags.



tag in

Reference to a tag on the server.

Use Open Tag or Query Tags to obtain this reference.

error in

Error conditions that occur before this VI runs. The VI responds to this input according to [standard error behavior](#).

tag out

Reference to a specific tag on SystemLink Server.

Use Open Tag or Query Tags to obtain this reference.

error out

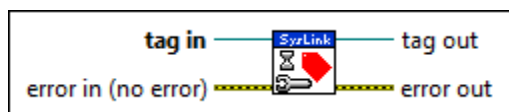
Error information. This VI produces this output according to [standard error behavior](#).

Configure Retention (Permanent)

Creates a retention policy to retain all historical tag values.



Note Any tag that uses this retention policy may use a large amount of storage space on SystemLink Server.



tag in

Reference to a tag on the server.

Use Open Tag or Query Tags to obtain this reference.

error in

Error conditions that occur before this VI runs. The VI responds to this input according to [standard error behavior](#).

tag out

Reference to a specific tag on SystemLink Server.

Use Open Tag or Query Tags to obtain this reference.

error out

Error information. This VI produces this output according to [standard error behavior](#).

Messages Functions

Communicate as a network queue using the publish-subscribe model. Use messages to send commands, status updates, and data between your systems and server.

Open

Creates a message session with the server.

Close

Terminates the message session with the server.

Publish

Writes a message to a topic, which immediately sends the message to subscribers of the topic.

Read

Returns a message from the queue.

Subscribe

Registers a session to receive messages published to a specific topic.

Unsubscribe

Unregisters a session from a specific topic to stop receiving messages.

Flush

Clears all queued messages associated with the session from the queue.

Message Properties

Provides access to properties of an open SystemLink message session.

Open

Creates a message session with the server.

Open (HTTP)

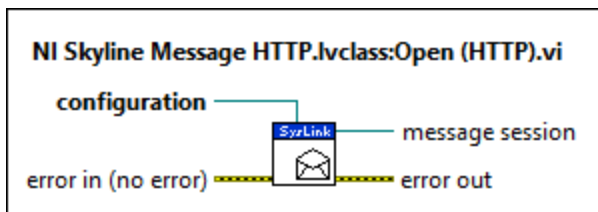
Creates an HTTP message session with the server based on the SystemLink configuration information.

Open (HTTP)

Creates an HTTP message session with the server based on the SystemLink configuration information.

Programming Patterns

- Sending Messages Between Systems



configuration

HTTP connection to the server.

If you leave this input unwired, this VI uses the default connection. It automatically retrieves your SystemLink Server credentials when you execute the application on a managed system or on the server.

error in

Error conditions that occur before this VI runs. The VI responds to this input according to [standard error behavior](#).

message session

HTTP message session created with the server.

error out

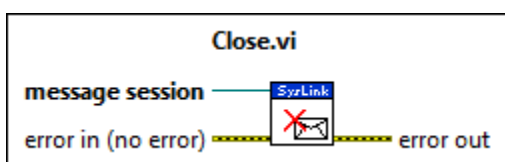
Error information. This VI produces this output according to [standard error behavior](#).

Close

Terminates the message session with the server.

Programming Patterns

- Sending Messages Between Systems



message session

Message session established with the server.

error in

Error conditions that occur before this VI runs. Unlike most VIs, this VI runs normally even if an error occurred before this VI runs.

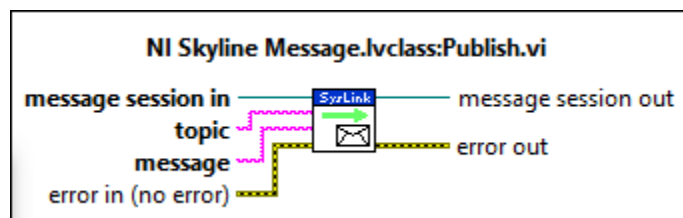
error out

Error information. This VI produces this output according to [standard error behavior](#).

Publish

Writes a message to a topic, which immediately sends the message to subscribers of the topic.

Subscribers of a topic receive every message published to the topic.



message session in

Message session established with the server.

message session

Message session established with the server.

topic

Subject to which endpoints publish or subscribe messages.

message

Content published for subscribers of a topic to read.

error in

Error conditions that occur before this VI runs. The VI responds to this input according to [standard error behavior](#).

message session out

Message session established with the server.

error out

Error information. This VI produces this output according to [standard error behavior](#).

Read

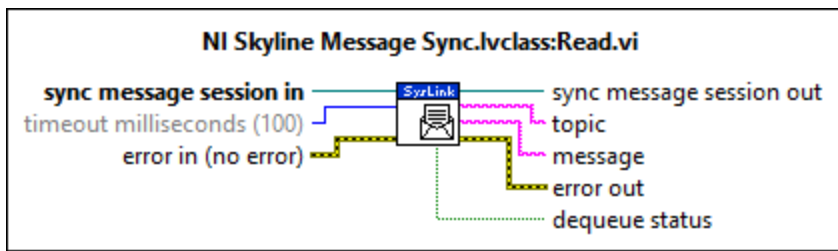
Returns a message from the queue.

To read a message, you must subscribe to the topic the message is published to. If you do not subscribe to the topic, you cannot receive the message.

If there are no queued messages, the call synchronously blocks up to the timeout specified or to the maximum timeout the web service defines, whichever is smaller. The SystemLink Server times out after 10 seconds.

Programming Patterns

- Sending Messages Between Systems



sync message session in

Synchronous message session created with the server. As a result, this VI does not apply to sessions created by Open (AMQP Async). If your message session is asynchronous, wire `sync message session in` to an event registration `refnum` on Flush Event Queue instead.

timeout milliseconds

Time, in milliseconds, that this VI waits for an enqueued message before timing out.

The default is 100.

error in

Error conditions that occur before this VI runs. The VI responds to this input according to [standard error behavior](#).

sync message session out

Synchronous message session created with the server.

topic

Channel to which endpoints publish or subscribe messages.

message

Content published for subscribers of a topic to read.

error out

Error information. This VI produces this output according to [standard error behavior](#).

dequeue status

Information about the current state of the VI. If True, the VI received an incoming message. If False, the VI timed out.

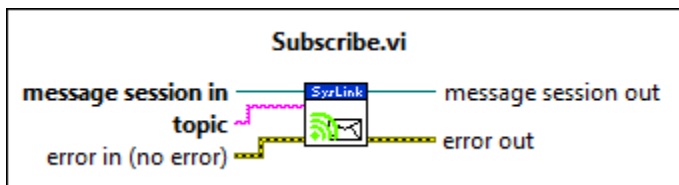
Subscribe

Registers a session to receive messages published to a specific topic.

Subscribers of a topic receive every message published to it. Wire several instances of this VI in a series to subscribe the multiple topics.

Programming Patterns

- Sending Messages Between Systems



message session

Message session established with the server.

topic

Subject to which endpoints publish or subscribe messages.

error in

Error conditions that occur before this VI runs. The VI responds to this input according to [standard error behavior](#).

message session out

Message session established with the server.

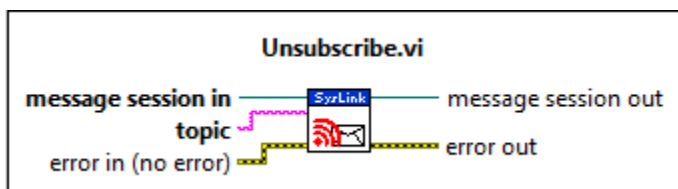
error out

Error information. This VI produces this output according to [standard error behavior](#).

Unsubscribe

Unregisters a session from a specific topic to stop receiving messages.

Endpoints continue to receive messages in the queue after you call Unsubscribe, if the application published them prior to you calling Unsubscribe.



message session

Message session established with the server.

topic

Subject to which endpoints publish or subscribe messages.

error in

Error conditions that occur before this VI runs. The VI responds to this input according to [standard error behavior](#).

message session out

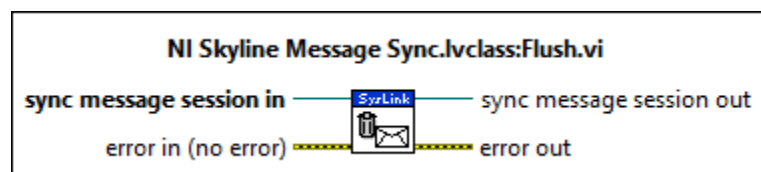
Message session established with the server.

error out

Error information. This VI produces this output according to [standard error behavior](#).

Flush

Clears all queued messages associated with the session from the queue.



sync message session in

Synchronous message session created with the server. As a result, this VI does not apply to sessions created by Open (AMQP Async). If your message session is asynchronous, wire `sync message session in` to an event registration `refnum` on Flush Event Queue instead.

error in

Error conditions that occur before this VI runs. The VI responds to this input according to [standard error behavior](#).

sync message session out

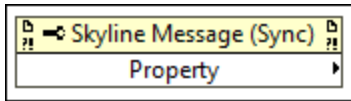
Synchronous message session created with the server.

error out

Error information. This VI produces this output according to [standard error behavior](#).

Message Properties

Provides access to properties of an open SystemLink message session.



reference

Control reference associated with the object whose properties you want to configure.

error in

Error conditions that occur before this VI runs. The VI responds to this input according to [standard error behavior](#).

reference out

Control reference specified by `reference`.

If you wrote any values to Message Properties, the value of reference out may differ from reference in.

error out

Error information. This VI produces this output according to [standard error behavior](#).

File Transfer Functions

Upload, update, delete, or download files using the File Transfer API.

[Send](#)

Sends a local file or data string to the server using the configuration you specify.

[Create Filter](#)

Creates or adds to an existing filter to query files on the server.

[Query Files](#)

Searches the server for files matching `filter`.

[Download](#)

Receives a file or an array of files from the server.

Open

Opens a reference to an existing file on the SystemLink file service.

Close

Terminates an open file reference.

Get Property

Returns the property `value` of the file that corresponds to the `key`.

Set Property

Writes the property `value` of the file that corresponds to the `key` and creates the property if it does not exist.

Update Metadata Properties

Modifies the metadata properties of a file.

Merge Metadata Properties

Updates metadata properties of a file after querying the server for the latest values.

File Properties

Provides information about an open file reference.

Delete

Removes a file or a group of files from the server.

Advanced Functions

Send files one data packet at a time, receive events when new files are available on the server, or receive file packets from the server asynchronously through events.

Send

Sends a local file or data string to the server using the configuration you specify.

File (HTTP)

Sends a local file to the SystemLink file service using the HTTP configuration.

Buffer (HTTP)

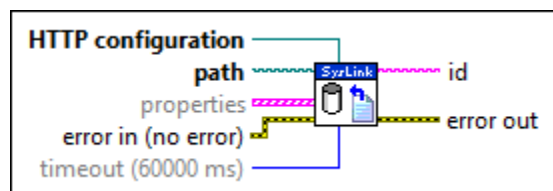
Uploads a buffer of data to create a new file on the SystemLink file service using the HTTP configuration.

File (HTTP)

Sends a local file to the SystemLink file service using the HTTP configuration.

Programming Patterns

- Transferring Files from Disk to the Server



HTTP configuration

HTTP connection to the server.

If you leave this input unwired, this VI uses the default connection and automatically retrieves your SystemLink Server credentials when you execute the application on a managed system or on the server.

path

Path to the file to send.

properties

Metadata about the file to send to the server.



Note The SystemLink file service uses the current file name to name the file on the server.

error in

Error conditions that occur before this VI runs. The VI responds to this input according to [standard error behavior](#).

timeout

Time, in milliseconds, that this VI waits for a file before timing out.

Default is 60,000 milliseconds.

id

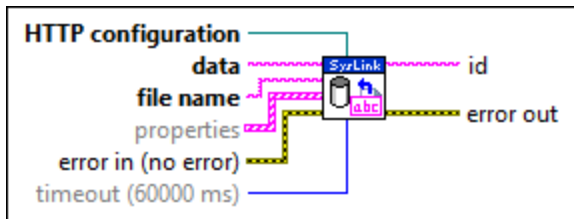
Unique file identifier the SystemLink file service generates to identify the file on the server.

error out

Error information. This VI produces this output according to [standard error behavior](#).

Buffer (HTTP)

Uploads a buffer of data to create a new file on the SystemLink file service using the HTTP configuration.



HTTP configuration

HTTP connection to the server.

If you leave this input unwired, this VI uses the default connection and automatically retrieves your SystemLink Server credentials when you execute the application on a managed system or on the server.

data

A string containing the contents of the file. When the SystemLink file service completes uploading the contents to the server, it creates the file.

file name

Name to give the file after the SystemLink file service uploads it to the server.

properties

Metadata about the file to send to the server.

error in

Error conditions that occur before this VI runs. The VI responds to this input according to [standard error behavior](#).

timeout

Time, in milliseconds, that this VI waits for a file before timing out.

Default is 60,000 milliseconds.

id

Unique file identifier the SystemLink file service generates to identify the file on the server.

error out

Error information. This VI produces this output according to [standard error behavior](#).

Create Filter

Creates or adds to an existing filter to query files on the server.

[Create Filter \(Extension\)](#)

Creates or adds a file extension query to an existing file query filter.

[Create Filter \(Name\)](#)

Creates or adds a file name query to an existing file query filter.

[Create Filter \(Property\)](#)

Creates or adds a property query to an existing file query filter.

[Create Filter \(Size\)](#)

Creates or adds a size query to an existing file query filter.

[Create Filter \(Time\)](#)

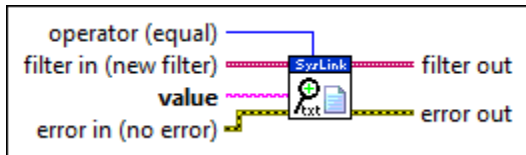
Creates or adds a timestamp query to an existing file query filter.

Create Filter (Extension)

Creates or adds a file extension query to an existing file query filter.

When you wire this VI to Query Files, the results only contain files that have a name property that matches the file extension query. If you provide multiple extension

queries, the last extension query overrides all preceding ones.



operator

Criteria for how to filter files on the SystemLink file service.

filter in

Query or queries you want to use find matching files on the SystemLink file service.

value

Metadata property to add.

error in

Error conditions that occur before this VI runs. The VI responds to this input according to [standard error behavior](#).

filter out

Updated query or queries to find matching files on the SystemLink file service.

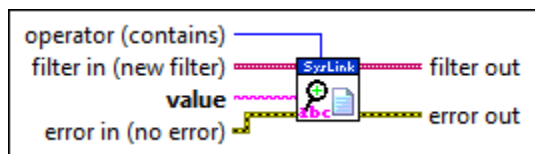
error out

Error information. This VI produces this output according to [standard error behavior](#).

Create Filter (Name)

Creates or adds a file name query to an existing file query filter.

When you wire this VI to Query Files, the results only contain files that have a name property that matches the file name query. If you specify multiple name queries, only files that match all queries are opened.



operator

Criteria for how to filter files on the SystemLink file service.

filter in

Query or queries you want to use find matching files on the SystemLink file service.

value

Metadata property to add.

error in

Error conditions that occur before this VI runs. The VI responds to this input according to [standard error behavior](#).

filter out

Updated query or queries to find matching files on the SystemLink file service.

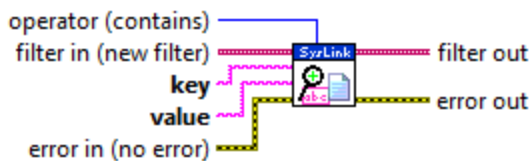
error out

Error information. This VI produces this output according to [standard error behavior](#).

Create Filter (Property Value)

Creates or adds a property query to an existing file query filter.

When you wire this VI to Query Files, the results contain only files that have property attributes that satisfy the query entry. If you specify multiple property queries for the same property, only files that match all queries open.



operator

Criteria for how to filter files on the SystemLink file service.

filter in

Query or queries you want to use find matching files on the SystemLink file service.

key

Unique identifier for the property you want to locate.

value

Metadata property to add.

error in

Error conditions that occur before this VI runs. The VI responds to this input according to [standard error behavior](#).

filter out

Updated query or queries to find matching files on the SystemLink file service.

error out

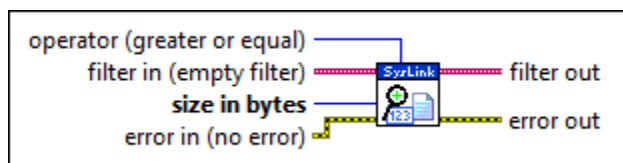
Error information. This VI produces this output according to [standard error behavior](#).

Create Filter (Size)

Creates or adds a size query to an existing file query filter.

When you wire this VI to Query Files, the results only contain files that have size

attributes that satisfy the query entry. If you provide multiple size queries with the same operator, the last size query overrides preceding ones.



operator

Criteria for how to filter files on the SystemLink file service.

filter in

Query or queries you want to use find matching files on the SystemLink file service.

size in bytes

Size, in bytes, of the file or files for which you want to query.

error in

Error conditions that occur before this VI runs. The VI responds to this input according to [standard error behavior](#).

filter out

Updated query or queries to find matching files on the SystemLink file service.

error out

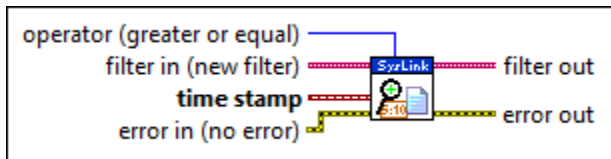
Error information. This VI produces this output according to [standard error behavior](#).

Create Filter (Time)

Creates or adds a timestamp query to an existing file query filter.

When you wire this VI to Query Files, the results only contain files that have timestamp

attributes that satisfy the query entry. If you provide multiple timestamp queries with the same operator, the last timestamp query provided overrides preceding ones.



operator

Criteria for how to filter files on the SystemLink file service.

filter in

Query or queries you want to use find matching files on the SystemLink file service.

time stamp

Date and time in which the file was created.

error in

Error conditions that occur before this VI runs. The VI responds to this input according to [standard error behavior](#).

filter out

Updated query or queries to find matching files on the SystemLink file service.

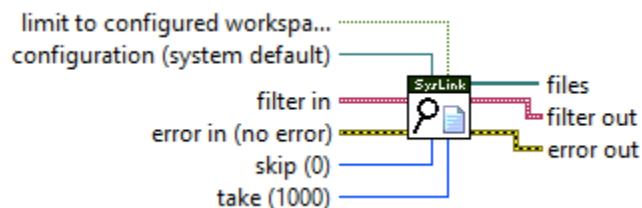
error out

Error information. This VI produces this output according to [standard error behavior](#).

Query Files

Searches the SystemLink file service for files matching `filter` using an HTTP configuration.

If you do not provide a `filter`, this VI opens references to every file on the service. The SystemLink file service limits the maximum number of results returned in a single query to 1,000 files.



HTTP configuration

HTTP connection to the server.

If you leave this input unwired, this VI uses the default connection and automatically retrieves your SystemLink Server credentials when you execute the application on a managed system or on the server.

filter in

Query or queries you want to use find matching files on the SystemLink file service.

limit to configured workspace?

Boolean that determines whether this VI queries only for tags in the workspace you specify in `configuration`. By default, the VI queries for tags from all accessible workspaces.

error in

Error conditions that occur before this VI runs. The VI responds to this input according to [standard error behavior](#).

files

Files returned from the SystemLink file service.

filter out

Query or queries use to return files from the SystemLink file service.

error out

Error information. This VI produces this output according to [standard error behavior](#).

Download

Receives a file or an array of files from the server.

[Download \(Single\)](#)

Receives a file synchronously from the SystemLink file service.

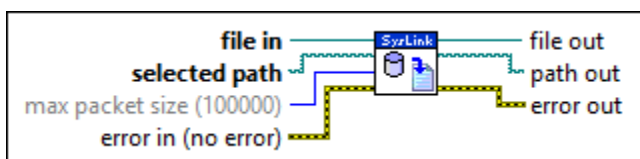
[Download \(Multiple\)](#)

Receives an array of files synchronously from the SystemLink file service.

Download (Single)

Receives a file synchronously from the SystemLink file service.

When you call this VI, it blocks additional calls until the file is downloaded or it returns an error.



file in

File on the SystemLink file service.

selected path

Fully qualified path to the file; for example, `C:\Users\Me\Downloads\myfile.txt`.

max packet size

Maximum size, in bytes, each data packet can use to download the file.

Default is 100,000 bytes.

error in

Error conditions that occur before this VI runs. The VI responds to this input according to [standard error behavior](#).

file out

File on the SystemLink file service.

path out

Location of the file.

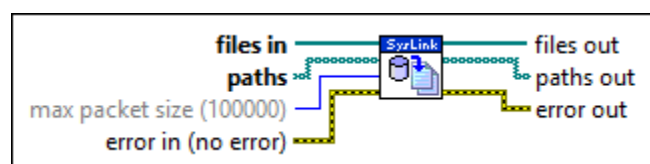
error out

Error information. This VI produces this output according to [standard error behavior](#).

Download (Multiple)

Receives an array of files synchronously from the SystemLink file service.

When you call this VI, it blocks additional calls until the files are downloaded or it returns an error.



files in

Files on the SystemLink file service.

paths

Fully qualified path to the file; for example, `C:\Users\Me\Downloads\myfile.txt`.

max packet size

Maximum size, in bytes, each data packet can use to download the file.

Default is 100,000 bytes.

error out

Error information. This VI produces this output according to [standard error behavior](#).

files out

Files on the SystemLink file service.

paths out

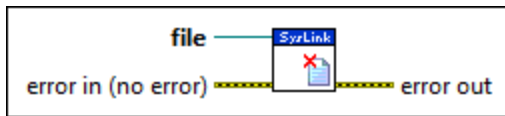
Location of the files.

error out

Error information. This VI produces this output according to [standard error behavior](#).

Close

Terminates an open file reference.



file

File on the SystemLink file service.

error in

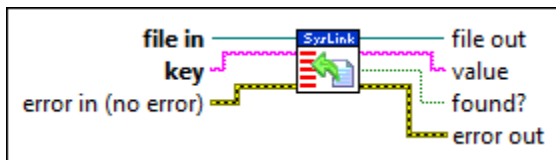
Error conditions that occur before this VI runs. The VI responds to this input according to [standard error behavior](#).

error out

Error information. This VI produces this output according to [standard error behavior](#).

Get Property

Returns the property `value` of the file that corresponds to the `key`.



file in

File on the SystemLink file service.

key

Unique identifier for the property you want to locate.

error in

Error conditions that occur before this VI runs. The VI responds to this input according to standard error behavior. For more information, refer to **Standard Error**

Behavior.**file out**

File on the SystemLink file service.

value

Property metadata returned.

found?

A Boolean that determines whether the property is located and returned.

error out

Error information. This VI produces this output according to standard error behavior. For more information, refer to ***Standard Error Behavior***.

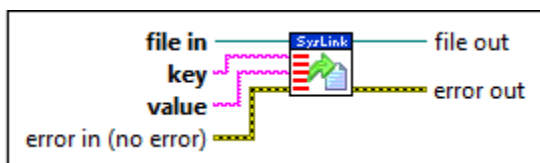
Related information:

- [Standard Error Behavior](#)

Set Property

Writes the property `value` of the file that corresponds to the `key` and creates the property if it does not exist.

You must wire this VI to Update Metadata Properties to reflect the updates on the SystemLink file service.



file in

File on the SystemLink file service.

key

Unique identifier for the property you want to locate.

value

Metadata property to add.

error in

Error conditions that occur before this VI runs. The VI responds to this input according to [standard error behavior](#).

file out

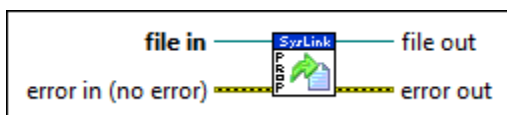
File on the SystemLink file service.

error out

Error information. This VI produces this output according to [standard error behavior](#).

Update Metadata Properties

Modifies the metadata properties of a file.

**file in**

File on the SystemLink file service.

error in

Error conditions that occur before this VI runs. The VI responds to this input according to [standard error behavior](#).

file out

File on the SystemLink file service.

error out

Error information. This VI produces this output according to [standard error behavior](#).

Merge Metadata Properties

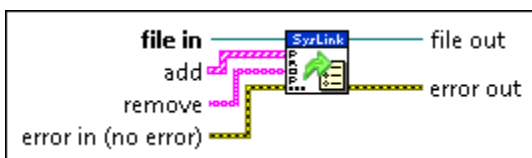
Updates metadata properties of a file after querying the server for the latest values.

Unlike Update Metadata Properties, Merge Metadata Properties verifies and merges properties according to the latest values available on the server. Therefore, use this VI to safely modify the metadata properties of a file.



Note

- When you use this VI to communicate to a server with SystemLink Server 19.5 or later installed, it provides the safe modification guarantee. However, if you use it with an earlier version of SystemLink Server, it acts similar to Update Metadata Properties.
- This VI only supports an HTTP configuration.



file in

File on the SystemLink file service.

add

Metadata properties you want to add to or overwrite on the file.

remove

Metadata properties you want to remove from the file.

error in

Error conditions that occur before this VI runs. The VI responds to this input according to [standard error behavior](#).

file out

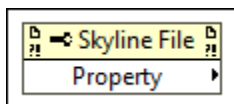
File on the SystemLink file service.

error out

Error information. This VI produces this output according to [standard error behavior](#).

Property Node

Provides information about an open file reference.

**reference**

Control reference associated with the object whose properties you want to configure.

error in

Error conditions that occur before this VI runs. The VI responds to this input according to [standard error behavior](#).

reference out

Control reference specified by `reference`. If you wrote values to this VI, the value of

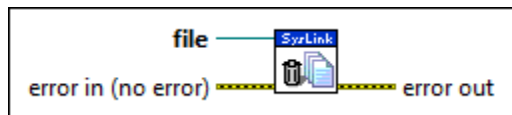
reference out may differ from reference in.

error out

Error information. This VI produces this output according to [standard error behavior](#).

Delete

Removes a file or a group of files from the server.



file

File on the SystemLink file service.

error in

Error conditions that occur before this VI runs. The VI responds to this input according to [standard error behavior](#).

error out

Error information. This VI produces this output according to [standard error behavior](#).

Advanced Functions

Send files one data packet at a time, receive events when new files are available on the server, or receive file packets from the server asynchronously through events.

[Create](#)

Creates a file on the SystemLink file service using an AMQP configuration.

[Send Packet](#)

Uploads a packet containing file data to the SystemLink file service.

Send Last Packet

Uploads the last packet of the file's data, which causes the SystemLink file service to sequentially write the packets to disk.

Close

Terminates an open file reference.

Register for Available File Events

Subscribes to the SystemLink file service to receive events when new files are available.

Open

Opens a reference to an existing file on the SystemLink file service.

Unregister for Available File Events

Unsubscribes from the SystemLink file service to stop receiving available file events.

Register for Packet Events

Subscribes to the SystemLink file service to receive events when file data packets are available during a file download.

Download (Async)

Sends a request to the SystemLink file service to transfer a file or group of files.

Unregister for Packet Events

Unsubscribes from the SystemLink file service to stop receiving events when file data packets are available.

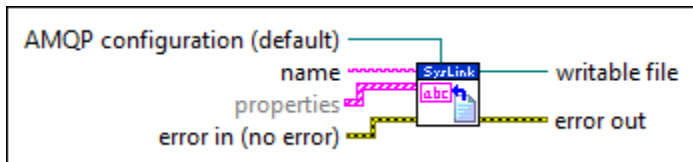
Create

Creates a file on the SystemLink file service.

Wire `writable file` to Send Packet to upload the file data. The file is unavailable on the server until Send Last Packet completes transferring the last data packet of the file.

Programming Patterns

- Transferring Files from Memory to the Server



configuration

Connection to the server.

If you leave this input unwired, this VI uses the default connection. It automatically retrieves your SystemLink Server credentials when you execute the application on a managed system or on the server.

name

Name assigned to the file.

properties

Key-value pairs that exist as metadata on a file.

error in

Error conditions that occur before this VI runs. The VI responds to this input according to [standard error behavior](#).

writable file

File you created.

error out

Error information. This VI produces this output according to [standard error behavior](#).

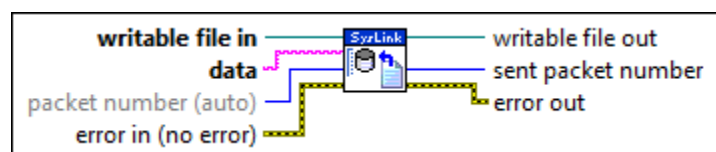
Send Packet

Uploads a packet containing file data to the SystemLink file service.

Use Send Last Packet to complete transmission of the file.

Programming Patterns

- Transferring Files from Memory to the Server



writable file in

File you want to upload to the SystemLink file service.

data

Information contained in file you can analyze and process.

packet number

A zero-indexed sequential number automatically assigned to the packet.



Note If you wire a number to the `packet number`, you must wire it for all future packets for the writable file reference.

error in

Error conditions that occur before this VI runs. The VI responds to this input according to [standard error behavior](#).

file out

File uploaded to the SystemLink file service. You cannot write to the file anymore.

sent packet number

A zero-indexed sequential number assigned to the packet.

error out

Error information. This VI produces this output according to [standard error behavior](#).

Send Last Packet

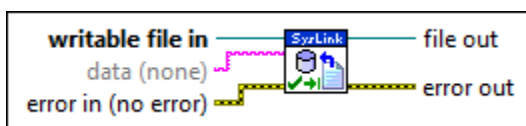
Uploads the last packet of the file's data, which causes the SystemLink file service to sequentially write the packets to disk.

If you wired a `packet number` to any of the Send Packet calls, the `packet number` is required for the writable file reference. In this instance, the `packet number` must be greater than the largest number previously sent and not previously used. Each number, from 0 to the largest number, is sent to the SystemLink file service once. Otherwise, the SystemLink file service rejects the file data. If you leave `packet number` unwired, the SystemLink file service automatically assigns a number to each data packet.

When you call this VI, you cannot write additional data to the file to upload.

Programming Patterns

- Transferring Files from Memory to the Server



writable file in

File you want to upload to the SystemLink file service.

data

Information contained in file you can analyze and process.

error in

Error conditions that occur before this VI runs. The VI responds to this input according to [standard error behavior](#).

file out

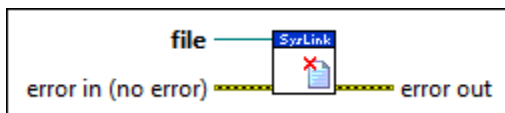
File uploaded to the SystemLink file service. You cannot write to the file anymore.

error out

Error information. This VI produces this output according to [standard error behavior](#).

Close

Terminates an open file reference.

**file**

File on the SystemLink file service.

error in

Error conditions that occur before this VI runs. The VI responds to this input according to [standard error behavior](#).

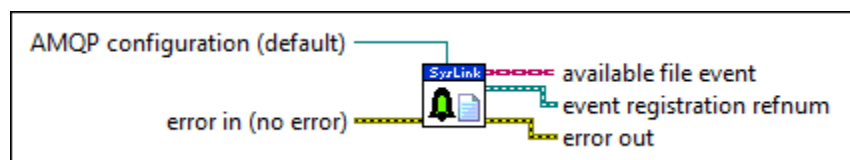
error out

Error information. This VI produces this output according to [standard error behavior](#).

Register for Available File Events

Subscribes to the SystemLink file service to receive events when new files are available.

Wire the `event registration refnum` to an event structure to listen for events.



configuration

Connection to the server.

error in

Error conditions that occur before this VI runs. The VI responds to this input according to [standard error behavior](#).

available file event

An event stating new files are available to upload or download.

event registration refnum

An event registration reference that listens for new file events on the server. Wire the `event registration refnum` to an event structure to listen for events.

error out

Error information. This VI produces this output according to [standard error behavior](#).

Open

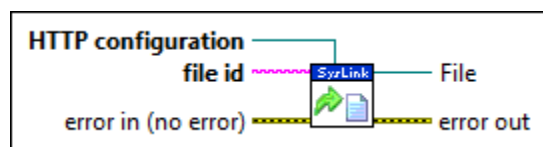
Opens a reference to an existing file on the SystemLink file service.

Open (HTTP)

Opens a reference to an existing file on the SystemLink file service using an HTTP configuration.

Open (HTTP)

Opens a reference to an existing file on the SystemLink file service using an HTTP configuration.



HTTP configuration

HTTP connection to the server.

If you leave this input unwired, this VI uses the default connection and automatically retrieves your SystemLink Server credentials when you execute the application on a managed system or on the server.

file id

A unique identifier of the file.

error in

Error conditions that occur before this VI runs. The VI responds to this input according to [standard error behavior](#).

file

The file on the SystemLink file service.

error out

Error information. This VI produces this output according to [standard error behavior](#).

Unregister for Available File Events

Unsubscribes from the SystemLink file service to stop receiving available file events.



available file event

An event stating new files are available to upload or download.

error in

Error conditions that occur before this VI runs. Unlike most VIs, this VI runs normally even if an error occurred before this VI runs.

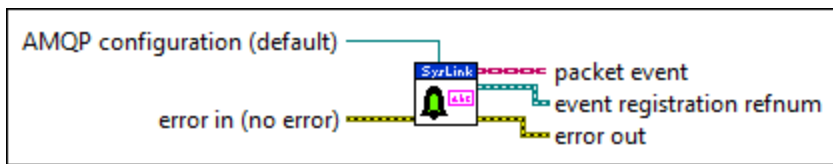
error out

Error information. This VI produces this output according to [standard error behavior](#).

Register for Packet Events

Subscribes to the SystemLink file service to receive events when file data packets are available during a file download.

Wire the `event_registration_refnum` to an event structure to listen for events.



configuration

Connection to the server.

error in

Error conditions that occur before this VI runs. The VI responds to this input according to [standard error behavior](#).

packet event

An event stating new packets are available to upload or download.

event registration refnum

An event registration reference that listens for new file events on the server. Wire the `event registration refnum` to an event structure to listen for events.

error out

Error information. This VI produces this output according to [standard error behavior](#).

Download (Async)

Sends a request to the SystemLink file service to transfer a file or group of files.

[Download \(Async\) \(Single\)](#)

Sends a request to the SystemLink file service to transfer a file.

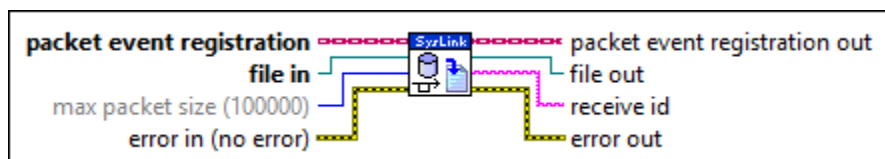
[Download \(Async\) \(Multiple\)](#)

Sends a request to the SystemLink file service to transfer a group of files.

Download (Async) (Single)

Sends a request to the SystemLink file service to transfer a file.

When the VI receives a packet through `packet event registration`, it triggers an event to download the packet. You must handle each packet's data in an event structure.



packet event registration

An event that occurs when new packets are available to upload or download.

file in

File on the SystemLink file service.

max packet size

Maximum size, in bytes, of a data packet.

error in

Error conditions that occur before this VI runs. The VI responds to this input according to [standard error behavior](#).

packet event registration out

An event that occurs when new packets are available to upload or download.

file out

File on the SystemLink file service.

receive id

Unique identifier of the download request, which is valid only for the duration of the download.

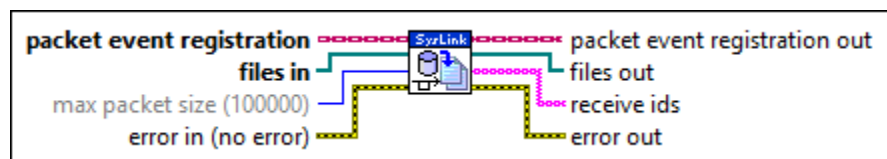
error out

Error information. This VI produces this output according to [standard error behavior](#).

Download (Async) (Multiple)

Sends a request to the SystemLink file service to transfer a group of files.

When the VI receives a packet through `packet event registration`, it triggers an event to download the packet. You must handle each packet's data in an event structure.



packet event registration

An event that occurs when new packets are available to upload or download.

files in

Files on the SystemLink file service.

max packet size

Maximum size, in bytes, of a data packet.

error in

Error conditions that occur before this VI runs. The VI responds to this input according

to [standard error behavior](#).

packet event registration out

An event that occurs when new packets are available to upload or download.

files out

Files on the SystemLink file service.

receive ids

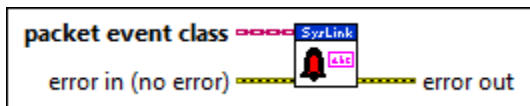
Unique identifier of the download requests, which are valid only for the duration of the downloads.

error out

Error information. This VI produces this output according to [standard error behavior](#).

Unregister for Packet Events

Unsubscribes from the SystemLink file service to stop receiving events when file data packets are available.



packet event class

An event stating new packets are available to upload or download.

error in

Error conditions that occur before this VI runs. Unlike most VIs, this VI runs normally even if an error occurred before this VI runs.

error out

Error information. This VI produces this output according to [standard error behavior](#).

Test Monitor Functions

Manage and communicate your test results and test steps between your test systems and SystemLink Server.

[Add Test Step Measurement](#)

Adds a measurement to a test step.

[Create Test Result](#)

Creates a test result or multiple test results on the server using the local reference from Initialize Test Result.

[Create Test Step](#)

Creates a test step or multiple test steps on the server using the local reference configured with Initialize Test Step.

[Delete Test Result](#)

Removes a test result or array of test results from the server.

[Delete Test Step](#)

Removes a test step or an array of test steps from the server.

[Generate Named Value](#)

Converts the `name` of the type definition and data provided into a `named value` object, which represents the inputs and outputs of a test step.

[Initialize Child Step from Step](#)

Configures a reference to a `new test step` for a test step reference on the server.

Initialize Test Result

Configures a local reference to a new test result on the server.

Initialize Test Step

Configures a local reference to a test step on the server.

Initialize Test Step from Result

Configures a reference to a child `test step` for a test result reference on the server.

Read Named Value Data

Returns the `name` and `value data` from the `named value` object, which represents the inputs and outputs of a test step.

Update Test Result

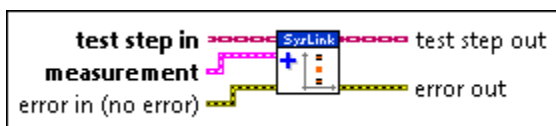
Modifies a test result or multiple test results on the server to match all the properties of the object supplied.

Update Test Step

Modifies a test step or multiple test steps on the server.

Add Test Step Measurement

Adds a measurement to a test step.



test step in

Test step reference on the server.

measurement

Test measurement to add to the test step.

error in

Error conditions that occur before this VI runs. The VI responds to this input according to [standard error behavior](#).

test step out

Test step reference on the server.

error out

Error information. This VI produces this output according to [standard error behavior](#).

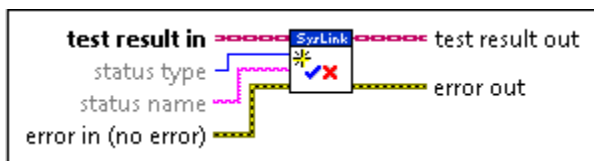
Create Test Result

Creates a test result or multiple test results on the server using the local reference from Initialize Test Result.

Test Monitor automatically generates a result ID and start time.

Programming Patterns

- Acquiring Test Results



test result in

Test result reference on the server.

status type

Execution state the test is in.

When you leave this input unwired, this VI uses the status associated with `test_step_in`. The default is Running.

status name

Name of the test execution state.

The name is used only when `status_type` is set to Custom.

error in

Error conditions that occur before this VI runs. The VI responds to this input according to [standard error behavior](#).

test result

Local test result reference on your machine. Wire this input to Create Test Result to create and add the test result to the server.

error out

Error information. This VI produces this output according to [standard error behavior](#).

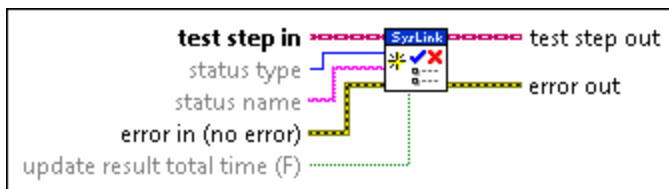
Create Test Step

Creates a test step or multiple test steps on the server using the local reference configured with Initialize Test Step.

The test result associated with the step must exist on the server before you create the step. Test Monitor automatically generates `step IDs` if you do not supply them.

Programming Patterns

- Acquiring Test Results



test step in

Test step reference on the server.

status type

Execution state the test is in.

When you leave this input unwired, this VI uses the status associated with `test_step_in`. The default is Running.

status name

Name of the test execution state.

The name is used only when `status_type` is set to Custom.

error in

Error conditions that occur before this VI runs. The VI responds to this input according to [standard error behavior](#).

update result total time

Boolean that determines whether to replace the automatically calculated time it takes to complete the test result.

If you want to define the total time, wire True to this input.

The default is False.

test step out

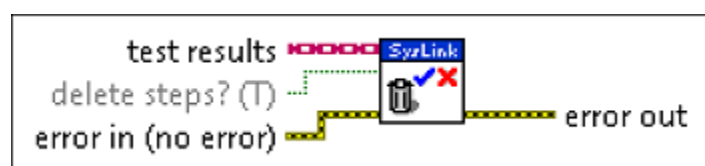
Test step reference on the server.

error out

Error information. This VI produces this output according to [standard error behavior](#).

Delete Test Result

Removes a test result or array of test results from the server.



test results

Test result references on the server.

delete steps?

Boolean that determines whether you want to delete the test steps associated with the test.

error in

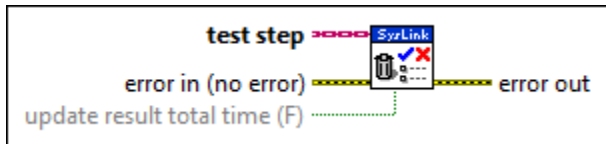
Error conditions that occur before this VI runs. The VI responds to this input according to [standard error behavior](#).

error out

Error information. This VI produces this output according to [standard error behavior](#).

Delete Test Step

Removes a test step or an array of test steps from the server.



test step in

Test step reference on the server.

error in

Error conditions that occur before this VI runs. The VI responds to this input according to [standard error behavior](#).

update result total time

Boolean that determines whether to replace the automatically calculated time it takes to complete the test result.

If you want to define the total time, wire True to this input.

The default is False.

error out

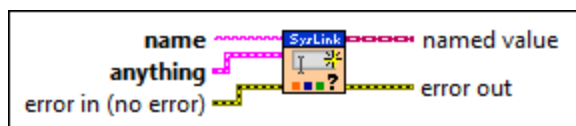
Error information. This VI produces this output according to [standard error behavior](#).

Generate Named Value

Converts the `name` of the type definition and data provided into a `named value` object, which represents the inputs and outputs of a test step. Wire the `named value` to a Property Node to set the inputs or outputs of the test step.

Programming Patterns

- Acquiring Test Results



name

Name of the type definition.

anything

Data you want to add to the test step. This parameter accepts arrays and clusters of Booleans, floating-point numbers, and strings. It can also accept an array of clusters, or a cluster of arrays, of these data types. Cluster elements may be named or unnamed, but not a combination of both. If cluster elements are named, then each name must be unique to that cluster. If you wire an unsupported data type, LabVIEW breaks the VI.

error in

Error conditions that occur before this VI runs. The VI responds to this input according to [standard error behavior](#).

named value

An object that represents the inputs and outputs of a test step.

error out

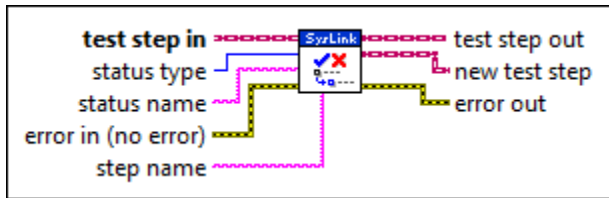
Error information. This VI produces this output according to [standard error behavior](#).

Initialize Child Step from Step

Configures a reference to a `new test step` for a test step reference on the server.

Wire Create Test Step to this VI to create the test step reference you need to configure with a child test step.

Subsequent calls that use the test step references modify the corresponding test step data on the server.



test step in

Test step reference on the server.

status type

Execution state the test is in.

When you leave this input unwired, this VI uses the status associated with `test_step_in`. The default is Running.

status name

Name of the test execution state.

The name is used only when `status_type` is set to Custom.

error in

Error conditions that occur before this VI runs. The VI responds to this input according to [standard error behavior](#).

step name

Name you assign to the test step.

test step out

Test step reference you added the `new_test_step` to.

new test step

Child test step of the test step reference on the server.

error out

Error information. This VI produces this output according to [standard error behavior](#).

Initialize Test Result

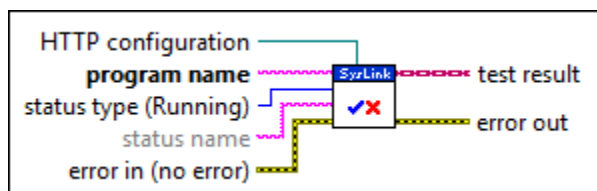
Configures a local reference to a new test result on the server.

Wire `test result` to Create Test Result to create a reference of it on the server.

Subsequent calls using the test result reference modify the corresponding test result data on the Test Monitor service.

Programming Patterns

- Acquiring Test Results



HTTP configuration

HTTP connection to the server.

If you leave this input unwired, this VI uses the default connection and automatically retrieves your SystemLink Server credentials when you execute the application on a managed system or on the server.

program name

Name of the test from which you want to obtain test results.

status type

Execution state the test is in.

When you leave this input unwired, this VI uses the status associated with `test_step_in`. The default is Running.

status name

Name of the test execution state.

The name is used only when `status_type` is set to Custom.

error in

Error conditions that occur before this VI runs. The VI responds to this input according to [standard error behavior](#).

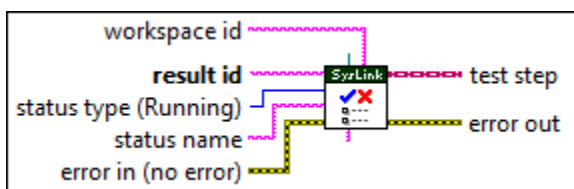
test result

Local test result reference on your machine. Wire this input to Create Test Result to create and add the test result to the server.

Initialize Test Step

Configures a local reference to a test step on the server.

Wire `test_step_out` to Create Test Step to create a reference of it on the server.



workspace id

ID of the workspace in which to add the new step.

result id

Unique identifier of the test result.

status type

Execution state the test is in.

When you leave this input unwired, this VI uses the status associated with `test_step_in`. The default is Running.

status name

Name of the test execution state.

The name is used only when `status_type` is set to Custom.

error in

Error conditions that occur before this VI runs. The VI responds to this input according to [standard error behavior](#).

step name

Name you assign to the test step.

test step

Local test step reference on your machine. You must wire this input to Create Test Step to create and add the test step to the SystemLink Test Monitor service.

error out

Error information. This VI produces this output according to [standard error behavior](#).

Initialize Test Step from Result

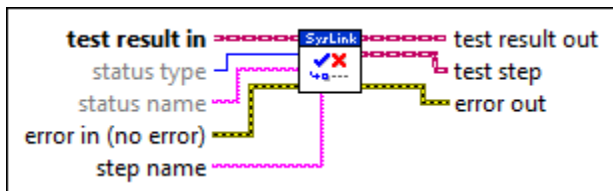
Configures a reference to a child `test_step` for a test result reference on the server. Wire Create Test Result to this VI to create the test result reference you need to

configure a child test step.

Subsequent calls that use the `test result` and `test step` modify the corresponding data on the server.

Programming Patterns

- Acquiring Test Results



test result in

Test result reference on the server.

status type

Execution state the test is in.

When you leave this input unwired, this VI uses the status associated with `test step in`. The default is Running.

status name

Name of the test execution state.

The name is used only when `status type` is set to Custom.

error in

Error conditions that occur before this VI runs. The VI responds to this input according to [standard error behavior](#).

step name

Name you assign to the test step.

test result

Local test result reference on your machine. Wire this input to Create Test Result to create and add the test result to the server.

test step

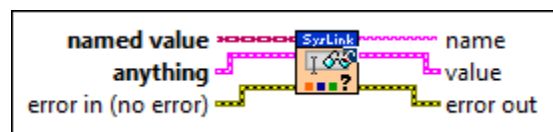
Test step reference on the server.

error out

Error information. This VI produces this output according to [standard error behavior](#).

Read Named Value Data

Returns the `name` and `value` data from the `named value` object, which represents the inputs and outputs of a test step.



named value

An object that represents the inputs and outputs of a test step.

anything

Data you want to add to the test step. This parameter accepts arrays and clusters of Booleans, floating-point numbers, and strings. It can also accept an array of clusters, or a cluster of arrays, of these data types. Cluster elements may be named or unnamed, but not a combination of both. If cluster elements are named, then each name must be unique to that cluster. If you wire an unsupported data type, LabVIEW

breaks the VI.

error in

Error conditions that occur before this VI runs. The VI responds to this input according to [standard error behavior](#).

name

Name of the type definition.

value

Updated data of the test step.

error out

Error information. This VI produces this output according to [standard error behavior](#).

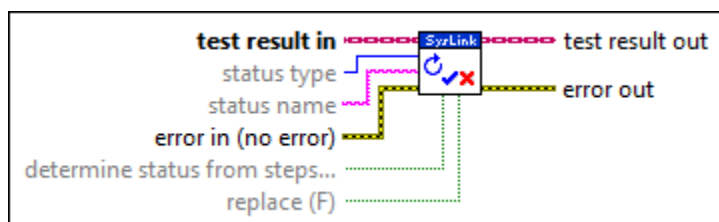
Update Test Result

Modifies a test result or multiple test results on the server to match all the properties of the object supplied.

Test results must exist already on the server before you can update them.

Programming Patterns

- Acquiring Test Results



test result in

Test result reference on the server.

status type

Execution state the test is in.

When you leave this input unwired, this VI uses the status associated with `test_step_in`. The default is Running.

status name

Name of the test execution state.

The name is used only when `status_type` is set to Custom.

error in

Error conditions that occur before this VI runs. The VI responds to this input according to [standard error behavior](#).

determine status from steps

Boolean that determines whether to automatically update the test result status based on the child steps of the result. Test Monitor orders statuses by their severity, from highest to lowest. The highest severity child step is then used to update the top-level result.

replace

A Boolean that determines whether to replace existing test result fields instead of merging them.

test result

Local test result reference on your machine. Wire this input to Create Test Result to create and add the test result to the server.

error out

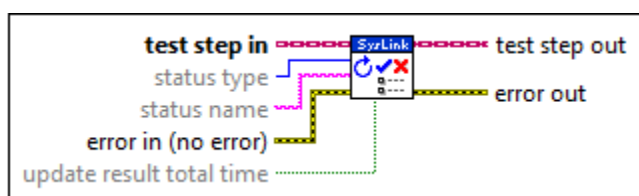
Error information. This VI produces this output according to [standard error behavior](#).

Update Test Step

Modifies a test step or multiple test steps on the server.

Programming Patterns

- Acquiring Test Results



test step in

Test step reference on the server.

status type

Execution state the test is in.

When you leave this input unwired, this VI uses the status associated with `test step in`. The default is Running.

status name

Name of the test execution state.

The name is used only when `status type` is set to Custom.

error in

Error conditions that occur before this VI runs. The VI responds to this input according

to [standard error behavior](#).

update result total time

Boolean that determines whether to replace the automatically calculated time it takes to complete the test result.

If you want to define the total time, wire True to this input.

The default is False.

test step out

Test step reference on the server.

error out

Error information. This VI produces this output according to [standard error behavior](#).

Utilities Functions

Obtain system names and minion IDs to assign tag paths and properties to your data.

[Get Minion ID](#)

Obtains the minion ID of the Salt system, if one is available.

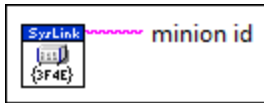
[Get System Name](#)

Obtains the name of the system, if one is available.

Get Minion ID

Obtains the minion ID of the Salt system, if one is available.

If a minion ID is not available, this VI returns an empty string.



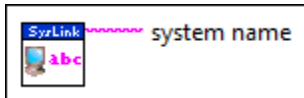
minion id

The unique identifier for the Salt minion.

Get System Name

Obtains the name of the system, if one is available.

If a system name is not available, this VI returns an empty string.



system name

The name of the system.

Alarm Functions

Create alarms to implement rules and manage conditions affecting your systems, tests, or assets.

Acknowledge Alarm

Acknowledges one or more active alarm instances on the server.

Add Alarm Notes

Adds one or more notes, or comments, to an alarm instance.

Alarm Properties

Reads and writes properties of an alarm instance or a notification strategy.

Build Alarm ID

Creates an `alarm_id` using the names of the system and application where the alarm originated.

Build Alarm Set Transition

Creates a `set_transition` to apply when an alarm condition is met.

Clear Alarm

Clears a set alarm instance on the server if the VI does not detect a condition to trigger the alarm.

Create Alarm Query Filter

Creates a filter to query the server for specific alarms.

Delete Alarm

Removes the alarm instance identified by `alarm` from the server.

Get Notification Strategies

Queries the server for all alarm notification strategies.

Query Alarms

Searches the server for alarm instances that match the `filter`.

Set Alarm

Triggers an alarm instance to notify subscribers of an issue or a state requiring attention when an application detects a condition.

Acknowledge Alarm

Acknowledges one or more active alarm instances on the server.

Acknowledge Alarm (Multiple)

Acknowledges a group of active alarm instances on the server with a single call.

Acknowledge Alarm (Single)

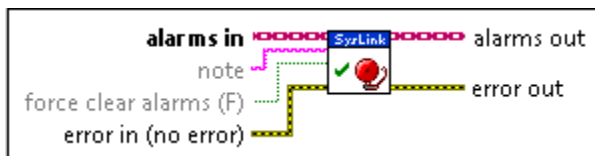
Acknowledges an active alarm instance on the server.

Acknowledge Alarm (Multiple)

Acknowledges a group of active alarm instances on the server with a single call.

An alarm instance remains active until you acknowledge and clear it. Typically, the application that sets the alarm clears it when the condition(s) that triggered the alarm no longer exist. An operator must acknowledge the alarm in the user interface or call this VI.

Use Query Alarms to find alarm instances you want to acknowledge.



alarms in

Group of active alarm instances you need to acknowledge.

note

Information you want to include about the alarm acknowledgement.

force clear alarms

A Boolean that determines whether to clear alarms without waiting for the application that set the alarms to do it.

The default is False.

error in

Error conditions that occur before this VI runs. The VI responds to this input according to [standard error behavior](#).

alarms out

Group of active alarm instances you acknowledged.

error out

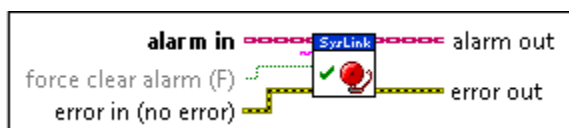
Error information. This VI produces this output according to [standard error behavior](#).

Acknowledge Alarm (Single)

Acknowledges an active alarm instance on the server.

An alarm instance remains active until you acknowledge and clear it. Typically, the application that sets the alarm clears it when the condition(s) that triggered the alarm no longer exist. An operator must acknowledge the alarm in the user interface or call this VI.

Use Query Alarms to find alarm instances you want to acknowledge.



alarm in

Active alarm instance you need to acknowledge.

note

Information you want to include about the alarm acknowledgement.

force clear alarms

A Boolean that determines whether to clear alarms without waiting for the application that set the alarms to do it.

The default is False.

error in

Error conditions that occur before this VI runs. The VI responds to this input according to [standard error behavior](#).

alarm out

Active alarm instance you acknowledged.

error out

Error information. This VI produces this output according to [standard error behavior](#).

Add Alarm Notes

Adds one or more notes, or comments, to an alarm instance.

[Add Alarm Notes \(Multiple\)](#)

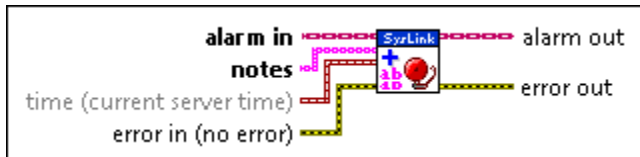
Add a group of notes to an alarm instance to record information about it.

[Add Alarm Notes \(Single\)](#)

Add a note to an alarm instance to record information about it.

[Add Alarm Notes \(Multiple\)](#)

Add a group of notes to an alarm instance to record information about it.



alarm in

Alarm instance to which you want to add notes about on the server.

notes

Information you want to record about the alarm instance, such as a description of an investigation into what caused the alarm.

time

Timestamp of when you created a note. Leave this input unwired to use the current time on the server.

error in

Error conditions that occur before this VI runs. The VI responds to this input according to [standard error behavior](#).

alarm out

Alarm instance to which you added notes about on the server.

error out

Error information. This VI produces this output according to [standard error behavior](#).

Add Alarm Notes (Single)

Add a note to an alarm instance to record information about it.



alarm in

Alarm instance to which you want to add a note about on the server.

note

Information you want to record about the alarm instance, such as a description of an investigation into what caused the alarm.

time

Timestamp of when you created a note. Leave this input unwired to use the current time on the server.

error in

Error conditions that occur before this VI runs. The VI responds to this input according to [standard error behavior](#).

alarm out

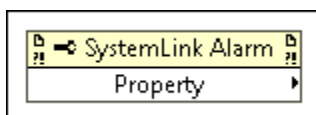
Alarm instance to which you added a note about on the server.

error out

Error information. This VI produces this output according to [standard error behavior](#).

Property Node

Reads and writes properties of an alarm instance or a notification strategy.



reference

Control reference associated with the alarm instance for which you want to set or get properties.

error in

Error conditions that occur before this VI runs. The VI responds to this input according to [standard error behavior](#).

reference out

Control reference specified by `reference`.

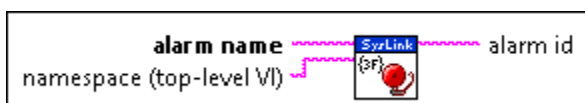
If you wrote any values to Alarm Properties, the value returned may differ from `reference`.

error out

Error information. This VI produces this output according to [standard error behavior](#).

Build Alarm ID

Creates an `alarm id` using the names of the system and application where the alarm originated.



alarm name

Name of the alarm triggered.

namespace

Name to overwrite the default namespace.

The namespace of the `alarm_id` defaults to the name of the top-level VI where the alarm triggered. If you want to override the default namespace, wire a name to the `namespace` input.

alarm_id

Unique identifier for a process or condition tracked by an alarm. The Build Alarm ID VI concatenates the names of the system and top-level VI where the rule triggered with the `alarm_name` into a path.

For example, an `alarm_id` may look like `<system>.<namespace>.<alarm_name>`.

Build Alarm Set Transition

Creates a `set_transition` to apply when an alarm condition is met.

Build Alarm Set Transition (Multiple)

Creates a `set_transition` to apply to multiple notification strategies when an alarm condition is met and the alarm changes in severity.

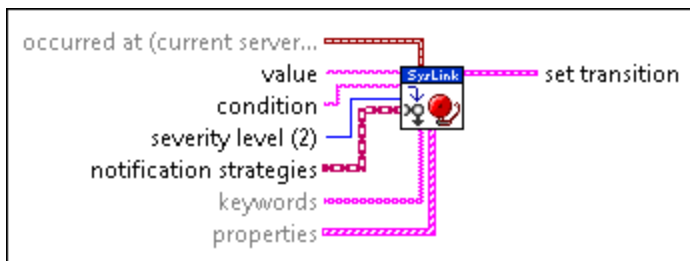
Build Alarm Set Transition (Single)

Creates a `set_transition` to apply to a notification strategy when an alarm condition is met and the alarm changes in severity.

Build Alarm Set Transition (Multiple)

Creates a `set_transition` to apply to multiple notification strategies when an alarm condition is met and the alarm changes in severity.

Wire `set_transition` to Set Alarm to create or update an alarm instance on the server.



occurred at

Timestamp of when a system or application triggered an alarm instance.

value

Data value that caused the application to set the alarm.

condition

Description of what caused the transition.

For example, `Greater than 90% CPU or Low disk space.`

severity level

Number that indicates how severe an alarm instance is. In the user interface, severity levels range from 1 to 4, from lowest to highest.

notification strategies

Plans established to notify the specified users with the correct information whenever an alarm triggers on a system or application.

keywords

Words or phrases associated with the transition.



Note NI SystemLink Web Interface does not show this information by default.

properties

Key-value pairs associated with the transition.



Note NI SystemLink Web Interface does not show this information by default.

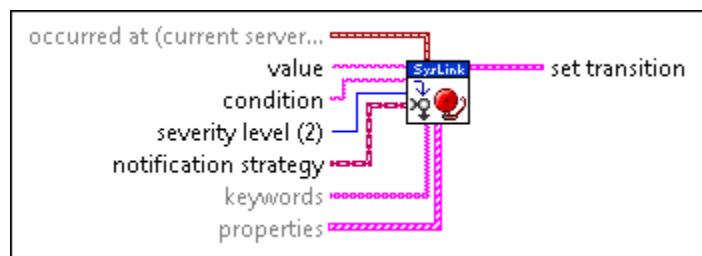
set transition

State change and severity level that meet a specific condition to create or update an alarm instance.

Build Alarm Set Transition (Single)

Creates a `set transition` to apply to a notification strategy when an alarm condition is met and the alarm changes in severity.

Wire `set transition` to Set Alarm to create or update an alarm instance on the server.



occurred at

Timestamp of when a system or application triggered an alarm instance.

value

Data value that caused the application to set the alarm.

condition

Description of what caused the transition.

For example, `Greater than 90% CPU or Low disk space`.

severity level

Number that indicates how severe an alarm instance is. In the user interface, severity levels range from 1 to 4, from lowest to highest.

notification strategy

Plan established to notify the specified users with the correct information whenever an alarm triggers on a system or application.

Wire `Get Notification Strategies` to this input.

keywords

Words or phrases associated with the transition.



Note NI SystemLink Web Interface does not show this information by default.

properties

Key-value pairs associated with the transition.



Note NI SystemLink Web Interface does not show this information by default.

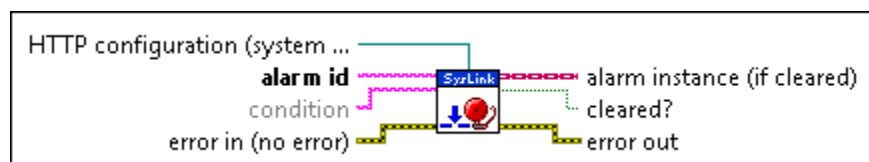
set transition

State change and severity level that meet a specific condition to create or update an alarm instance.

Clear Alarm

Clears a set alarm instance on the server if the VI does not detect a condition to trigger the alarm.

An alarm instance remains active until you acknowledge and clear it. Typically, the application that sets the alarm clears it when the condition(s) that triggered the alarm no longer exist.



HTTP configuration

HTTP connection to the server.

If you leave this input unwired, this VI uses the default connection and automatically retrieves your SystemLink Server credentials when you execute the application on a managed system or on the server.

alarm id

Unique identifier for a process or condition tracked by an alarm. The Build Alarm ID VI concatenates the names of the system and top-level VI where the rule triggered with the `alarm name` into a path.

For example, an `alarm id` may look like `<system>.<namespace>.<alarm name>`.

condition

Description of what caused the transition.

For example, `Greater than 90% CPU or Low disk space`.

error in

Error conditions that occur before this VI runs. The VI responds to this input according to [standard error behavior](#).

alarm instance

Cleared alarm instance, if an instance exists. If no instance exists, this output returns an empty class instance.

cleared?

Boolean that determines whether a set alarm instance exists. If no alarm instance exists, it returns False and `alarm instance` returns default data.

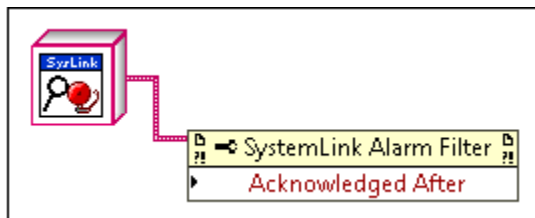
error out

Error information. This VI produces this output according to [standard error behavior](#).

Create Alarm Query Filter

Creates a filter to query the server for specific alarms.

This VI uses a Property Node to create the filter you want to use. If you want to add more than one property to the filter, expand the Property Node and set the properties.



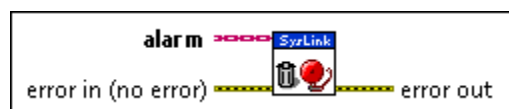
reference out

Filter to use to query the server for matching alarm instances.

Delete Alarm

Removes the alarm instance identified by `alarm` from the server.

Only delete an active alarm as a last resort. Instead, you should clear and acknowledge it or force clear and acknowledge it.



alarm

Alarm instance you want to delete on the server.

error in

Error conditions that occur before this VI runs. The VI responds to this input according to [standard error behavior](#).

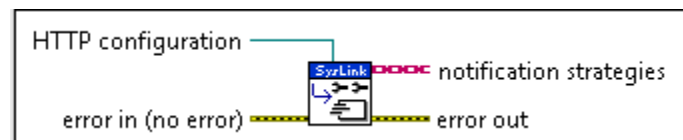
error out

Error information. This VI produces this output according to [standard error behavior](#).

Get Notification Strategies

Queries the server for all alarm notification strategies.

The `notification strategies` are then sent to Set Alarm when a condition is triggered. Refer to Specifying Notification Settings for Alarms for more information.



HTTP configuration

HTTP connection to the server.

If you leave this input unwired, this VI uses the default connection and automatically retrieves your SystemLink Server credentials when you execute the application on a managed system or on the server.

error in

Error conditions that occur before this VI runs. The VI responds to this input according to [standard error behavior](#).

notification strategies

Plans established to notify the specified users with the correct information whenever an alarm triggers on a system or application.

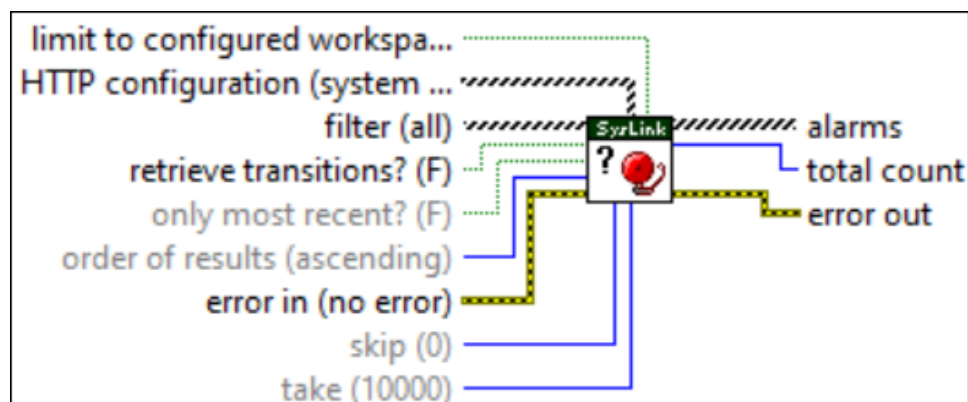
Wire a Property Node to this output to obtain information about a notification strategy, such as its name.

error out

Error information. This VI produces this output according to [standard error behavior](#).

Query Alarms

Searches the server for alarm instances that match the `filter`.



limit to configured workspace?

Boolean that determines whether this VI queries only for alarms in the workspace you specify in `configuration`. By default, the VI queries for alarms from all accessible workspaces.

HTTP configuration

HTTP connection to the server.

If you leave this input unwired, this VI uses the default connection and automatically retrieves your SystemLink Server credentials when you execute the application on a managed system or on the server.

filter

Properties and values used to find and return matching alarm instances.

Wire Create Alarm Query Filter to this input.

retrieve transitions?

Boolean that determines whether to obtain the transition history of an alarm.

The default is False.

only most recent?

Boolean that determines whether the server should return only the most recent instance of each `alarm id`.

order of results

Order in which the results are sorted based on the **updated at** field of an alarm instance.

The default value is ascending order.

Name	Description
Ascending	Sorts the alarm instances returned in least to greatest order.
Descending	Sorts the alarm instances returned in greatest to least order.

error in

Error conditions that occur before this VI runs. The VI responds to this input according to [standard error behavior](#).

skip

Number of alarm instances to skip on the server.

The default value is 0.

take

Number of alarm instances requested from the server.

If you leave `skip` and `take` empty, the query returns the first 1,000 results by default.

alarms

Alarm instances matching the `filter` returned from the server.

total count

Total number of alarm instances matching the `filter` before taking `skip` and `take` into account on the server.

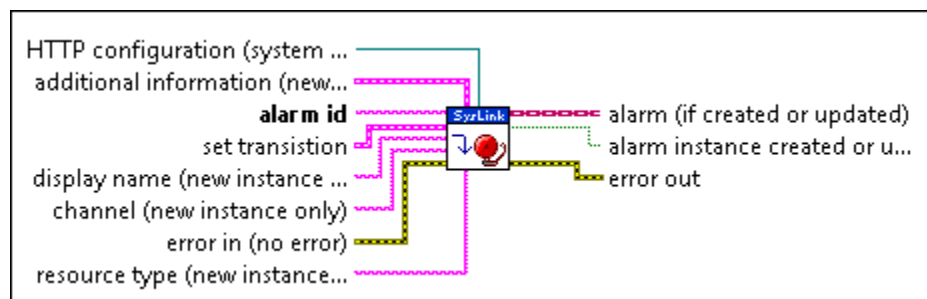
error out

Error information. This VI produces this output according to [standard error behavior](#).

Set Alarm

Triggers an alarm instance to notify subscribers of an issue or a state requiring attention when an application detects a condition.

Wire Clear Alarm after this VI to clear the alarms associated with the alarm instance.



HTTP configuration

HTTP connection to the server.

If you leave this input unwired, this VI uses the default connection and automatically retrieves your SystemLink Server credentials when you execute the application on a managed system or on the server.

additional information

Variety of data about the new alarm instance, such as the application or user who created it and keywords and properties associated with it.

This VI ignores this input when updating an existing alarm instance.

alarm id

Unique identifier for a process or condition tracked by an alarm. The Build Alarm ID VI concatenates the names of the system and top-level VI where the rule triggered with the `alarm name` into a path.

For example, an `alarm id` may look like `<system>.<namespace>.<alarm`

name>.

set transition

State change and severity level that meet a specific condition to create or update an alarm instance.

Wire Build Alarm Set Transition to this input.

display name

Title you want to give the alarm instance.

This VI ignores this input when updating an existing alarm instance.

channel

Resource associated with the alarm. For example, if an application sets an alarm because a wireless network went down, the `channel` could be the name of the network and the `resource type` classifies the alarm as the network.

This VI ignores this input when it updates an existing alarm instance.

error in

Error conditions that occur before this VI runs. The VI responds to this input according to [standard error behavior](#).

resource type

Type of resource associated with the alarm. For example, if an application sets an alarm because a wireless network went down, the `channel` may be the name of the network where the alarm triggered and the `resource type` may classify the alarm as the network.

This VI ignores this input when it updates an existing alarm instance.

alarm

Alarm instance created or updated.

alarm instance created or updated?

Boolean that determines whether an alarm instance with the provided `set transition` already exists. If an alarm instance already exists or the current severity level already occurred, it returns `False` and `alarm` returns default data.

error out

Error information. This VI produces this output according to [standard error behavior](#).

Asset Functions

Track how long and how often you use your assets in test systems or use tags to read and write data values between assets and SystemLink Server.

[Asset Utilization Functions](#)

Track how long and how often you use your assets in test systems.

[List All Assets](#)

Lists all assets you can use to build a tag path.

[Build Asset Tag Path](#)

Creates a tag path to publish data to the server based on an asset you specify.

Asset Utilization Functions

Track how long and how often you use your assets in test systems.

[Start Utilization](#)

Collects usage data about an asset, group of assets, or all assets when a test or measurement task starts.

End Utilization

Stops collecting usage data about an asset or group of assets when a test or measurement task finishes.



Note Asset Utilization functions only run on Windows systems.

Start Utilization

Collects usage data about an asset, group of assets, or all assets when a test or measurement task starts.

Start Utilization (All)

Collects usage data about every asset when a test or measurement task starts.

Start Utilization (Multiple)

Collects usage data about a group of assets when a test or measurement task starts.

Start Utilization (Single)

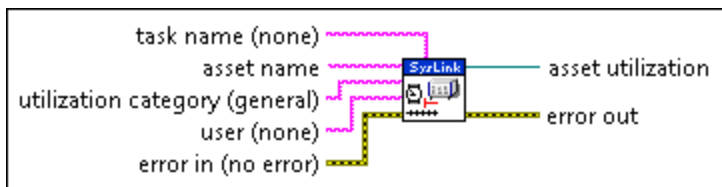
Collects usage data about an asset when a test or measurement task starts.

Start Utilization (All)

Collects usage data about every asset when a test or measurement task starts.



Note You can only run this VI on a managed system or on the server itself. If you run this VI on an unmanaged system, the data obtained will not transmit to the server. Instead, the data transfer will fail silently.



task name

Name of the test or measurement task you want the asset to perform.

asset name

Name of the device or devices.

If you do not wire this input, the VI collects usage data for all assets.

utilization category

Reason the `user` is using the asset.

The default is General.

user

Name of the person using the asset.

error in

Error conditions that occur before this VI runs. The VI responds to this input according to [standard error behavior](#).

asset utilization

Asset utilization reference on the server.

error out

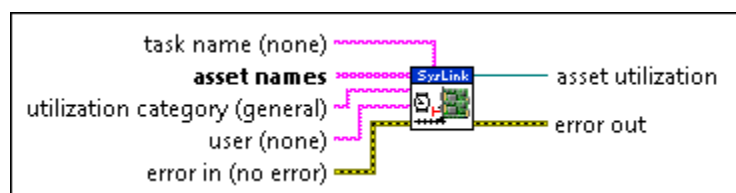
Error information. This VI produces this output according to [standard error behavior](#).

Start Utilization (Multiple)

Collects usage data about a group of assets when a test or measurement task starts.



Note You can only run this VI on a managed system or on the server itself. If you run this VI on an unmanaged system, the data obtained will not transmit to the server. Instead, the data transfer will fail silently.



task name

Name of the test or measurement task you want the asset to perform.

asset names

Names of the devices.

utilization category

Reason the `user` is using the asset.

The default is General.

user

Name of the person using the asset.

error in

Error conditions that occur before this VI runs. The VI responds to this input according to [standard error behavior](#).

asset utilization

Asset utilization task reference on the server.

error out

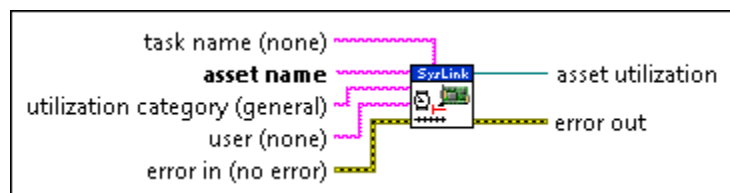
Error information. This VI produces this output according to [standard error behavior](#).

Start Utilization (Single)

Collects usage data about an asset when a test or measurement task starts.



Note You can only run this VI on a managed system or on the server itself. If you run this VI on an unmanaged system, the data obtained will not transmit to the server. Instead, the data transfer will fail silently.



task name

Name of the test or measurement task you want the asset to perform.

asset name

Name of the device.

utilization category

Reason the `user` is using the asset.

The default is General.

user

Name of the person using the asset.

error in

Error conditions that occur before this VI runs. The VI responds to this input according to [standard error behavior](#).

asset utilization

Asset utilization task reference on the server.

error out

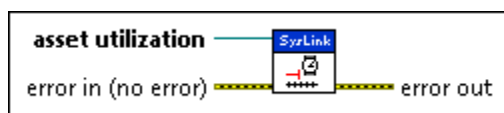
Error information. This VI produces this output according to [standard error behavior](#).

End Utilization

Stops collecting usage data about an asset or group of assets when a test or measurement task finishes.



Note You can only run this VI on a managed system or on the server itself. If you run this VI on an unmanaged system, the data obtained will not transmit to the server. Instead, the data transfer will fail silently.

**asset utilization**

Asset utilization task reference on the server.

error in

Error conditions that occur before this VI runs. The VI responds to this input according to [standard error behavior](#).

error out

Error information. This VI produces this output according to [standard error behavior](#).

List All Assets

List of all assets you can use to build a tag path.



error in

Error conditions that occur before this VI runs. Unlike most VIs, this VI runs normally even if an error occurred before this VI runs.

All Assets List

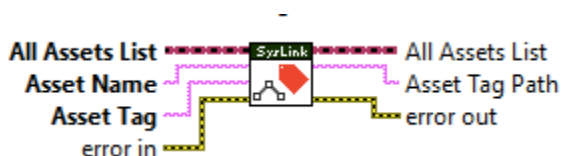
 List of all assets you can use to build a tag path.

error out

Error information. This VI produces this output according to [standard error behavior](#).

Build Asset Tag Path

Creates a tag path to publish data to the server based on an asset you specify.



All Assets List



List of all assets you can use to build a tag path.

Asset Name



Asset to build a tag path for.

Asset Tag



Name of the tag this VI creates.

error in

Error conditions that occur before this VI runs. The VI responds to this input according to [standard error behavior](#).

All Assets List



List of all assets you can use to build a tag path.

Asset Tag Path



Tag path to publish data from the asset to SystemLink.

error out

Error information. This VI produces this output according to [standard error behavior](#).

HTTP API Reference

Interact with SystemLink data services through the HTTP APIs of SystemLink Server and SystemLink Enterprise. You can access the API documentation in an OpenAPI document format.

SystemLink Server

You can access the HTTP API documentation for SystemLink Server in one of the following ways.

- Append `/niapis` to the address of your SystemLink server in your browser. For example, `http://localhost/niapis`.
- Append one of the following paths to the address of your SystemLink server to open SystemLink TDM documents:

API Documentation	URL to Append to Server URL	Example
◦ Upload Analysis Automation Procedures ◦ Create Manual Task	<code>/systemlink-tdm-offline-doc/AnalysisServer/UploadAnalysisProcedureCreateManualTask.html</code>	<code>http://localhost/AnalysisServer/UploadAnalysisPr</code>
◦ Run Manual Task ◦ Inspect Task Execution	<code>/systemlink-tdm-offline-doc/AnalysisServer/RunAndInspectManualTask.html</code>	<code>http://localhost/AnalysisServer/R</code>
Data Navigation Data Service HTTP API	<code>/systemlink-tdm-offline-doc/DataNavigatorDataService/DataNavigatorDataService.html</code>	<code>http://localhost/DataNavigatorDat DataNavigatorDat</code>

SystemLink Enterprise

You can access the HTTP API documentation for SystemLink Enterprise in one of the following ways.

- Navigate to the **Home** page, and click the API documentation link.
- Access the ***Online API Documentation***.

Related information:

- [Sharing Data Across Systems](#)
- [Online API Documentation](#)

Python API Reference

Use the `nisystemlink-clients` Python API to interact with SystemLink Server and SystemLink Enterprise HTTP APIs from Jupyter or from within other Python scripts.



Note For installation instructions and usage guidance, refer to the [nisystemlink-clients API documentation](#).

The following example code uses the SystemLink Python APIs:

- For SystemLink Server examples, refer to the `Jupyter` folder in the [SystemLink Server Examples GitHub repository](#).
- For SystemLink Enterprise examples, refer to the `Python Examples` folder in the [SystemLink Enterprise Examples GitHub repository](#).

Related information:

- [Analyzing Files with Jupyter Notebooks](#)
- [JupyterHub Documentation](#)
- [Papermill Documentation](#)

.NET API Reference

Use .NET APIs to leverage SystemLink services with C# commands.

Access .NET API documentation for SystemLink in one of the following ways:

- Navigate to [NuGet.org](#) to obtain the .NET API package and reference documentation.

.NET API Package	Description
Test Monitor	Manage and communicate your test results

.NET API Package	Description
	and steps between your systems and SystemLink Server.
File	Manage and transfer files between your systems and SystemLink Server.
Message	Send and receive status updates and commands between applications.
Tag	Transfer low-frequency test and measurement data between your systems and SystemLink Server.
Asset Management	Manage and communicate asset information relating to your systems and tests.

- Refer to the [SystemLink Client API GitHub repository](#) for .NET API documentation and examples.