

Data Acquisition Toolbox Quick Reference Guide

Getting Started

If you have a sound card installed, you can run the following code which collects one second of data

```
ai = analoginput('winsound')
addchannel(ai, 1)
set(ai, 'SampleRate', 8192);
start(ai)
data = getdata(ai);
plot(data)
```

The resources can assist you in getting started:

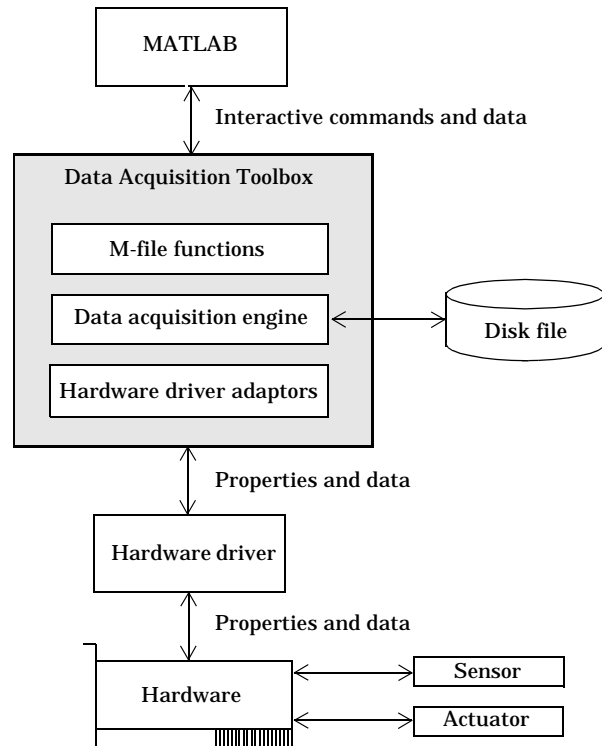
Type this...	To learn about...
info daq	General toolbox functionality
demos	Data acquisition applications and toolbox syntax

Toolbox Components

The Data Acquisition Toolbox consists of the three components listed below.

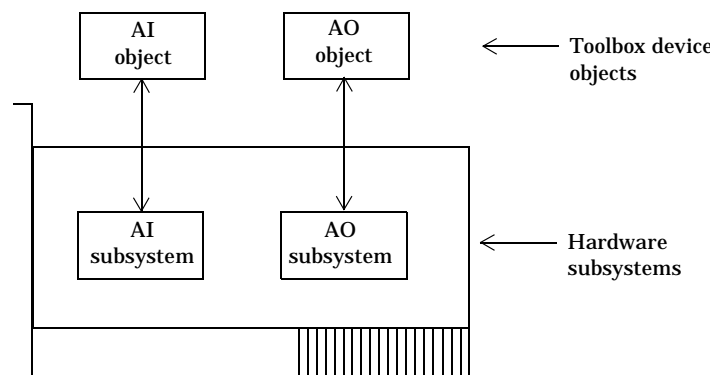
The toolbox component...	Allows you to...
M-files	Create toolbox objects, control your acquisition behavior, and evaluate your acquisition status and resources
engine	Store device objects and their property values, control the storage of acquired data, and control the synchronization of events
adaptors	Pass properties and data between the hardware driver and the engine

These components are shown below.



Device Objects

Device objects allow you to access specific hardware subsystems. The device objects supported by the toolbox are analog input (AI) and analog output (AO).



The Data Acquisition Session

A complete data acquisition session involves:

- 1 Initialization - Create a device object.
- 2 Configuration - Add channels and control acquisition behavior with properties.
- 3 Execution - Start the device object and acquire (AI) or send (AO) data.
- 4 Termination - Delete the device object.

Creating a Device Object

To create a device object, you must call the appropriate creation function. Creation functions are named for the device object they create:

Subsystem type	Creation function
Analog input	anal o g i n p u t (a d a p t o r , i d)
Analog output	anal o g o u t p u t (a d a p t o r , i d)

i d must be specified when using NI devices and multiple sound cards. The supported adaptors are:

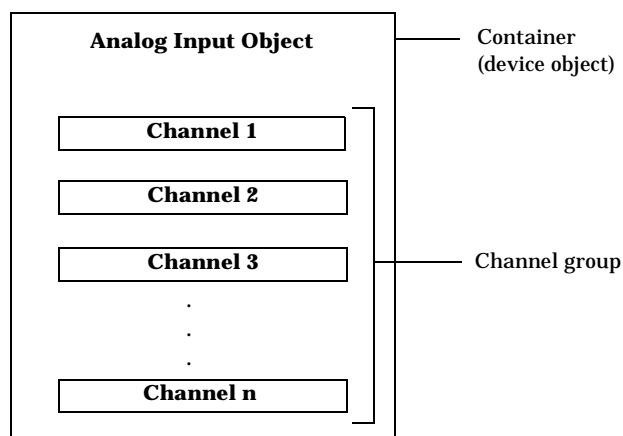
Hardware Type	Adaptor name
sound cards	wi n s o u n d
National Instruments (NI)	n i d a q

To create the analog input object ai for a sound card

```
ai = anal o g i n p u t ( ' w i n s o u n d ' );
```

Adding Channels to a Device Object

To perform any data acquisition task, add at least one channel to the device object. Think of a device object as a channel container and the channels added to the device object as a channel group.



To add channels to a device object, you must use the addchannel function, e.g.,

```
ch = addchannel ( ai , 1 : 2 );
```

Controlling Acquisition Behavior with Properties

Device object and channel properties allow you to control the behavior of your data acquisition application. General property information is listed below:

- Properties are not case sensitive.
- Property names can be abbreviated.
- set (ai) returns a list of all settable properties for ai .
- get (ai) returns a list of all the property values for ai .

Property Types

Toolbox properties are divided into two main types:

Property type	Description
Common - Base - Device-specific	These properties apply to every channel or line contained by a device object.
Channel - Base - Device-specific	These properties can be configured for individual channels or lines.

- Base properties appear for all supported hardware devices
- Device-specific properties appear only for the specific hardware you are using.
- set and get display alphabetically the base properties followed by the device-specific properties.

Property Syntax

You can obtain property information with three syntaxes: get/set notation, dot notation, or named index notation.

The get/set syntax is similar to the Handle Graphics® get/set syntax:

```
get ( ai , ' Sampl eRate' )
set ( ai , ' Sampl eRate' , 8000 );
```

The dot notation has the following syntax:

```
ai . Sampl eRate
ai . Sampl eRate = 8000;
```

The named index syntax has the following syntax and is used to reference channels and channel property information:

```
set ( ai . Chan nel ( 1 ) , ' Chan nel Name' , ' Chan 1' );
ai . Chan 1 . Uni tsRange;
ai . Chan 1 . Uni tsRange = [ 0 10 ];
```

where Name is the value of the Channel Name property of the channel that is being referenced.

Acquiring or Sending Data

To acquire (AI) or send (AO) data, you must:

- 1 Start the device object.
- 2 Issue a trigger.

Starting the Device Object

To start the device object, use the start command.

```
start(ai)
```

After the device object is started, the Running (AI) or Sending (AO) property is automatically set to On.

Issuing a Trigger

To log data to memory or a disk file, a trigger must occur. The supported trigger types are given below.

Trigger type	Description
Immediate	The trigger occurs just after start is issued. This is the default trigger type.
Manual	The trigger occurs after you manually issue the trigger function.
Software (AI only)	The trigger is issued when a signal satisfying the specified condition is detected. You must specify one or more channels as trigger sources.

After the trigger occurs, the Logging property is automatically set to On.

Stopping a Device Object

A device object stops when the requested data is acquired or sent (output), a runtime error occurs, or the stop command is issued.

```
stop(ai)
```

Managing Data

Previewing Data

While an analog input device object is running, acquired data can be previewed with the peekdata function.

```
data = peekdata(ai, data)
```

peekdata is a nonblocking function that returns execution control immediately to MATLAB and does not extract data from the engine.

Extracting Data

At any time after data is acquired, it can be extracted from the engine with the getdata function.

```
data = getdata(ai, data);
```

getdata is a blocking function that guarantees all requested samples are acquired without gaps.

Sending Data

To send (output) data, it must be queued in the engine. You can queue data in the engine with the putdata function.

```
putdata(ao, data)
```

Once data is queued, you can start the analog output object.

Logging Information to Disk

For analog input objects, you can log acquired data, events, device objects, and hardware information to a disk file using these properties:

```
set(ai, 'LoggingMode', 'Disk&Memory');  
set(ai, 'LogFileNames', 'data.daq');  
set(ai, 'LogToDiskMode', 'Index');
```

You can retrieve data, events, and object and hardware information from an existing log file using the daqread function. To retrieve all logged data:

```
data = daqread('data.daq')
```

To retrieve only object and hardware information:

```
daqinfo = daqread('data.daq', 'info');
```

Events and Action Functions

Unless an error occurs, all data acquisition sessions contain a start, trigger and stop event. You can access event information with the EventLog property.

```
Events = ai.EventLog  
EventTypes = {Events.Type}  
EventTypes =  
    'Start'    'Trigger'    'Stop'
```

When an event occurs, an action function (M-file) can be executed. You can specify the action function to be executed when an event occurs by including the name of an M-file as the value for the associated action property. For example,

```
set(ai, 'TriggerAction', 'daqaction');  
set(ai, 'RuntimeErrorAction', 'daqaction');  
set(ai, 'StopAction', 'daqaction');
```

configures ai so that daqaction.m is executed when a trigger, runtime error, or stop event occurs. To see how an action function is constructed, type

```
daqhelp daqaction
```

Deleting Device Objects

The `delete` command removes the specified device object from the MATLAB workspace and the engine.

```
delete(ai)
```

If more than one MATLAB variable points to a device object that was removed from the engine, these MATLAB variables will be invalid, though they may still exist in the MATLAB workspace. For example, if

```
ai1 = ai
```

and `ai` is deleted, the variable `ai1` still exists in the MATLAB workspace as an invalid device object. The variable `ai1` should be cleared from the workspace.

Clearing and Finding Device Objects

The `clear` command removes the specified device object from the MATLAB workspace but not from the engine.

```
clear ai
```

To find (retrieve) the device object, you must use the `daqfind` command.

```
a = daqfind  
ai = a{1}
```

Saving and Loading Device Objects

You can use the `save` command to save a device object to a MAT-file.

```
save ai
```

You can use the `load` command to load a device object into the MATLAB workspace.

```
load ai
```

You can convert a device object to equivalent MATLAB code with the `obj2code` command.

```
obj2code(ai, 'ai_save1')
```

You can recreate the device object by running the M-file.

```
ai1 = ai_save1
```

Getting Information and Help

You can obtain information or help about installed hardware, driver adaptors, device objects, functions, or properties using the functions shown below.

Functions	Description
<code>daqhelp</code>	Display help for device objects, constructors, functions, and properties.
<code>daqhwinfo</code>	Display data acquisition hardware information.

Functions	Description
<code>ni daq</code>	Display information about National Instruments adaptor
<code>propinfo</code>	Return information for device object or channel properties.
<code>wi nsound</code>	Display information for sound card adaptor

PDF and HTML versions of the *Data Acquisition Toolbox User's Guide* are available through the Help Desk. To invoke the Help Desk, type

```
helpdesk
```

Toolbox Functions

Creating Device Objects		AI	AO
analoginput	Create an analog input object.	✓	
analogoutput	Create an analog output object.		✓
digitalio	Create a digital I/O object.		

Configuring Device Objects		AI	AO
addchannel	Add hardware channels to an analog input or analog output object.	✓	✓
get	Return property values.	✓	✓
set	Configure or display property values.	✓	✓
setverify	Set and return the specified property value.	✓	✓

Executing the Device Object		AI	AO
start	Start the execution of a device object.	✓	✓
stop	Stop the execution of a device object.	✓	✓
trigger	Manually execute a trigger.	✓	✓

Working with Data		AI	AO
daqread	Read a Data Acquisition Toolbox (.daq) file.	✓	
flushdata	Remove data from the data acquisition engine.	✓	✓
getdata	Return data, time, and event information to the MATLAB workspace for an analog input.	✓	
getsample	Return one sample for each analog input channel group member or digital I/O line group member.	✓	
peekdata	Immediately return data to the MATLAB workspace for an analog input object.	✓	
putdata	Queues data in the engine for eventual output to the D/A hardware.		✓
putsample	Send one sample for each analog output channel group member or digital I/O line group member.		✓

Getting Information and Help		AI	AO
daqhelp	Display help for device objects, constructors, functions, and properties.	✓	✓
daqhwinfo	Display data acquisition hardware information.	✓	✓

Getting Information and Help		AI	AO
ni daq	Display information about National Instruments adaptor	✓	✓
propinfo	Return property information for device objects, channels, or lines.	✓	✓
winsound	Display information for sound card adaptor	✓	✓

General Purpose		AI	AO
clear	Clear device objects from the MATLAB workspace.	✓	✓
daqaction	Display event information for specified event	✓	✓
daqfind	Return device objects, channels, or lines from the data acquisition engine to the workspace.	✓	✓
daqmem	Allocate or display memory for one or more device objects.	✓	✓
daqregister	Register or unregister a Data Acquisition Toolbox adaptor.	✓	✓
daqreset	Remove data acquisition objects and .dll files from memory.	✓	✓
delete	Delete device objects, channels, or lines.	✓	✓
disp	Display device object, channel, and line information.	✓	✓
ischannel	Check for channels.	✓	✓
isline	Check for lines		
isvalid	Check for valid device objects, channels, or lines.	✓	✓
length	Determine length of data acquisition objects.	✓	✓
load	Load device objects into the MATLAB workspace	✓	✓
makenames	Generate a list of channel or line names.	✓	✓
obj2code	Convert device objects, channels, or lines to MATLAB code and save to disk.	✓	✓
save	Save device objects to a MAT-file.	✓	✓
size	Determine size of data acquisition objects.	✓	✓

Analog Input Properties

Common Properties

Common properties apply to all channels contained by the analog input object.

Analog Input Basic Setup Properties

Channel	Contains all the channels associated with the device object as created with addchannel
SamplesPerTrigger	Specifies the number of samples to acquire per channel for each trigger issued
SampleRate	Specifies the samples per second of the analog input hardware device
TriggerType	Specifies the type of trigger to be issued

Analog Input Logging Properties

LogFileNames	Specifies the name of the disk file that data and events are written to
LoggingMode	Specifies where data is logged
LogToDiskMode	Specifies if data and events are saved in one disk file or multiple disk files

Analog Input Trigger Properties

TriggerChannel	Contains the hardware channels serving as trigger sources
TriggerCondition	Specifies the condition to be satisfied before a trigger is issued
TriggerConditionValue	Specifies the value of a trigger condition
TriggerDelay	Specifies the delay value for data logging
TriggerDelayUnits	Specifies the trigger delay as samples or seconds
TriggerRepeat	Specifies the number of times to repeat the trigger
TriggerTime	Indicates the observed absolute times when triggers occurred

Analog Input Status Properties

Logging	Indicates whether data is logged to memory and/or a disk file
Running	Indicates whether the device object and data acquisition engine are running
SamplesAcquired	Indicates the number of samples acquired per channel
SamplesAvailable	Indicates the number of samples available per channel in the data acquisition engine

Analog Input Hardware Configuration Properties

Channel Skew	Specifies the time, in seconds, between consecutive scanned hardware channels
Channel SkewMode	Specifies the hardware-specific channel skew
ClockSource	Specifies which clock is used to govern the acquisition rate
InputType	Specifies the hardware channel configuration, e.g., single-ended, differential, or AC-Coupled

Analog Input Action Properties

DataMissedAction	Specifies the M-file to be executed when data is missed
InputOverRangeAction	Specifies the M-file to be executed when an input channel exceeds its valid range
RuntimeErrorAction	Specifies the M-file to be executed when a runtime error occurs
SamplesAcquiredAction	Specifies the M-file to be executed every time the number of samples specified by SamplesAcquiredActionCount are acquired
SamplesAcquiredActionCount	Specifies the number of samples to acquire for each channel group member before the M-file specified by SamplesAcquiredAction is executed
StartAction	Specifies the M-file to be executed just before the hardware device and data acquisition engine start
StopAction	Specifies the M-file to be executed just after the hardware device and data acquisition engine stop
TimerAction	Specifies the M-file to be executed after the time specified by TimerPeriod has passed
TimerPeriod	Specifies the time, in seconds, between calls to TimerAction
TriggerAction	Specifies the M-file to be executed just after a trigger occurs

Analog Input General Purpose Properties

BufferingConfig	Specifies the per-channel allocated memory
BufferingMode	Specifies if memory is automatically by the engine or manually
EventLog	Contains information related to certain events that occur for a device object
Name	Contains the name of the device object
Tag	Stores a device object label
Timeout	Specifies the additional time to wait for a data block to fill
Type	Indicates the device object type
UserData	Stores data that you want to associate with the device object

Channel Properties

Channel properties can be set on a per-channel basis.

Analog Input Channel Properties	
Channel Name	Specifies a descriptive channel name
HwChannel	Specifies the hardware channel ID
Index	Indicates the MATLAB index number of a channel
InputRange	Specifies the range of data input to the analog input hardware device
Parent	Contains the parent (device) object of the channel
SensorRange	Specifies the range of data you expect from your sensor
Units	Specifies the engineering units
UnitsRange	Specifies the engineering units range of the data input into MATLAB

Analog Output Properties

Common Properties

Common properties apply to all channels contained by the analog output object.

Analog Output Basic Setup Properties

Channel	Contains all the channels associated with the device object as created with addchannel .
SampleRate	Specifies the sampling rate of the analog output device
TriggerType	Specifies the type of trigger to be issued

Analog Output Trigger Properties

TriggerTime	Indicates the observed times when triggers occurred
-------------	-----------------------------------------------------

Analog Output Status Properties

Running	Indicates whether the hardware device and data acquisition engine are running
SamplesAvailable	Indicates the number of samples available per channel in the data acquisition engine
SamplesOutput	Indicates the number of samples per channel that have been sent to the hardware device
Sending	Indicates data is being output to the hardware device

Analog Output Hardware Configuration Properties

OutOfDataMode	Specifies the last value to write to the analog output device when data is no longer being sent.
ClockSource	Specifies which clock is used to govern the acquisition rate

Analog Output Action Properties

OutOfDataAction	Specifies the M-file to be executed when data is no longer sent to the analog output device
RuntimeErrorAction	Specifies the M-file to be executed when a runtime error has occurred
SamplesOutputAction	Specifies the M-file to be executed every time the number of samples specified by SamplesOutputActionCount are sent
SamplesOutputActionCount	Specifies the number of samples to output before the function specified by SamplesOutputAction is executed

Analog Output Action Properties

StartAction	Specifies the M-file to be executed just before the hardware device and data acquisition engine start
StopAction	Specifies the M-file to be executed just after the hardware device and data acquisition engine stop
TimerAction	Specifies the M-file to be executed after the time specified by TimerPeriod has passed
TimerPeriod	Specifies the time, in seconds, between TimerAction callbacks
TriggerAction	Specifies the M-file to be executed just after a trigger occurs

Analog Output General Purpose Properties

BufferingConfig	Specifies the per-channel allocated memory
BufferingMode	Specifies if memory is automatically by the engine or manually
EventLog	Contains information related to certain events
MaxSamplesQueued	Specifies the maximum number of samples allowed in the analog output queue
Name	Contains the name of the device object
RepeatOutput	Specifies the number of times to send queued data to the analog output device
Tag	Stores a device object label
Timeout	Specifies the additional time to wait for a data block to be output
Type	Indicates the device object type
UserData	Stores data that you want to associate with the device object

Channel Properties

Channel properties can be set on a per-channel basis.

Analog Output General Properties

Channel Name	Specifies a descriptive channel name
DefaultChannelValue	Specifies the value to write to the analog output hardware device when data has stopped being sent
HwChannel	Specifies the hardware channel ID
Index	Indicates the MATLAB index number of a channel
OutputRange	Specifies the range of the D/A hardware
Parent	Contains the parent (device) object of the channel
Units	Specifies the engineering units
UnitsRange	Specifies the engineering units range of the data
