

# Instrument Control Toolbox Release Notes

---

The “Instrument Control Toolbox 1.2 Release Notes” on page 1-1 describe the changes introduced in the latest version of the Instrument Control Toolbox. The following topics are discussed in these Release Notes:

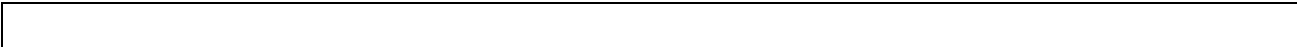
- “New Features” on page 1-2
- “Major Bug Fixes” on page 1-4
- “Upgrading from an Earlier Release” on page 1-6
- “Known Software and Documentation Problems” on page 1-7

The Instrument Control Toolbox Release Notes also provide information about the earlier versions of the product, in case you are upgrading from a version that was released prior to Release 12.1. If you are upgrading from a release earlier than Release 12.1, you should also see these sections:

- “Instrument Control Toolbox 1.1 Release Notes” on page 2-1
- “Instrument Control Toolbox 1.0 Release Notes” on page 3-1

## **Printing the Release Notes**

If you would like to print the Release Notes, you can link to a PDF version.



## Instrument Control Toolbox 1.2 Release Notes

1

<b>New Features</b> .....	1-2
New Objects for Networked Instruments .....	1-2
Configure Properties with the Property Inspector .....	1-2
Access Objects Using the Workspace Browser .....	1-2
Rohde & Schwarz Support .....	1-3
<b>Major Bug Fixes</b> .....	1-4
Memory Leaks .....	1-4
Serial Port Issues .....	1-4
<b>Vendor Driver Requirements and Limitations</b> .....	1-5
Incorrect Vendor Driver Version Returned By instrhwinfo ...	1-5
<b>Upgrading from an Earlier Release</b> .....	1-6
delete Function .....	1-6
Measurement Computing GPIB Boards .....	1-6
<b>Known Software and Documentation Problems</b> .....	1-7
Serial Port Conflict .....	1-7
Incorrect Vendor Driver Version Returned By instrhwinfo ...	1-7

## Instrument Control Toolbox 1.1 Release Notes

2

<b>New Features</b> .....	2-2
VXI Block and FIFO Read Operations .....	2-2
Freeing the Serial Port on Windows Platforms .....	2-2
Data Parsing .....	2-2
binblock Data Transfer .....	2-3
Tektronix VISA Support .....	2-3
Enhancements to Existing Functions and Properties .....	2-3

<b>Vendor Driver Requirements and Limitations</b> .....	<b>2-4</b>
Tektronix VISA Limitations .....	<b>2-4</b>
<b>Upgrading from an Earlier Release</b> .....	<b>2-5</b>
Events, Callbacks, and Function Handles .....	<b>2-5</b>

## **Instrument Control Toolbox 1.0 Release Notes**

# **3**

<b>Introduction to the Instrument Control Toolbox</b> .....	<b>3-2</b>
Supported Platforms .....	<b>3-2</b>
Object-Based Design .....	<b>3-2</b>
Reading and Writing Data .....	<b>3-3</b>
Events and Actions .....	<b>3-3</b>
Data Recording .....	<b>3-3</b>
Graphical Tools .....	<b>3-3</b>
<b>Vendor Driver Requirements and Limitations</b> .....	<b>3-4</b>
Driver Requirements .....	<b>3-4</b>
GPIB Driver Limitations .....	<b>3-5</b>
VISA Driver Limitations .....	<b>3-7</b>
<b>Known Software Problems</b> .....	<b>3-9</b>
Pin Status Event .....	<b>3-9</b>
Available Serial Ports .....	<b>3-9</b>
Accessing Serial Ports on Solaris .....	<b>3-10</b>

# Instrument Control Toolbox 1.2 Release Notes

---

<b>New Features</b> . . . . .	1-2
New Objects for Networked Instruments . . . . .	1-2
Configure Properties with the Property Inspector . . . . .	1-2
Access Objects Using the Workspace Browser . . . . .	1-2
Rohde & Schwarz Support . . . . .	1-3
<b>Major Bug Fixes</b> . . . . .	1-4
Memory Leaks . . . . .	1-4
Serial Port Issues . . . . .	1-4
<b>Vendor Driver Requirements and Limitations</b> . . . . .	1-5
Incorrect Vendor Driver Version Returned By instrhwinfo . . .	1-5
<b>Upgrading from an Earlier Release</b> . . . . .	1-6
delete Function . . . . .	1-6
Measurement Computing GPIB Boards . . . . .	1-6
<b>Known Software and Documentation Problems</b> . . . . .	1-7
Serial Port Conflict . . . . .	1-7
Incorrect Vendor Driver Version Returned By instrhwinfo . . .	1-7

## New Features

This section summarizes the new features and enhancements introduced in the Instrument Control Toolbox 1.2.

If you are upgrading from a release earlier than Release 12.1, then you should see “New Features” on page 2-2 in the Instrument Control Toolbox 1.1 Release Notes.

### New Objects for Networked Instruments

The toolbox now includes objects that enable you to control instruments that are available on a network (that is, they have a network address). Two new instrument objects are supported: the TCP/IP and UDP objects.

#### TCP/IP Object

The TCP/IP object allows you to control networked instruments using the TCP/IP protocol. For an example that uses the TCP/IP object, type

```
instrhelp tcpip
```

#### UDP Object

The UDP object allows you to control networked instruments using the UDP protocol. For an example that uses the UDP object, type

```
instrhelp udp
```

### Configure Properties with the Property Inspector

You can use the Property Inspector to configure property values for instrument objects. You open the Property Inspector with the `inspect` function.

```
g = gpib('ni',0,1);  
inspect(g)
```

### Access Objects Using the Workspace Browser

You can configure properties and view information associated with instrument objects via the MATLAB Workspace browser.

- 1** Launch the Property Inspector by double-clicking an instrument object variable.

**2** Right-click an instrument object variable and select the **Explore** menu to:

- Display summary information
- Invoke the Property Inspector
- Display hardware information
- List the functions and properties

## **Rohde & Schwarz Support**

You can now communicate with Rohde & Schwarz instruments over the network using Rohde & Schwarz's VISA passport by creating a VISA object. The `visa` creation function requires `ni` as the vendor, and `RSIB::remote_host::INSTR` as the resource name, where `remote_host` is the host name or IP address of the instrument.

## Major Bug Fixes

The Instrument Control Toolbox 1.2 includes several bug fixes made since Version 1.1. This section describes the particularly important Version 1.2 bug fixes.

### Memory Leaks

- Memory leaks no longer occur when reading data from an instrument.
- Memory leaks no longer occur when opening and closing the serial port connection.

### Serial Port Issues

- Serial port objects now work with the baud rates 128,000 and 256,000 along with all other standard baud rates listed in the BaudRate property help.
- MATLAB can now find serial ports if the MATLAB root directory contains an underscore.
- On Windows, serial port objects now recognize ports greater than COM4.
- On Solaris, there is no restriction on the number of times that the serial port can be opened.
- On Linux, there is a significant speed increase when writing data to the serial port.



## **Vendor Driver Requirements and Limitations**

Driver requirements and limitations associated with the vendors supported by the Instrument Control Toolbox 1.1 are described in “Vendor Driver Requirements and Limitations” on page 2-4.

### **Incorrect Vendor Driver Version Returned By instrhwinfo**

The vendor driver version returned by instrhwinfo to the VendorDriverVersion field is incorrect for Tektronix VISA and Agilent VISA.

## **Upgrading from an Earlier Release**

This section describes the upgrade issues involved in moving from the Instrument Control Toolbox 1.1 to Version 1.2.

If you are upgrading from a version earlier than 1.1, then you should see “Upgrading from an Earlier Release” on page 2-5 in the Instrument Control Toolbox 1.1 Release Notes.

### **delete Function**

If you delete an open instrument object, then `fclose` is automatically called and the object is closed before deletion.

### **Measurement Computing GPIB Boards**

The vendor name for Measurement Computing (ComputerBoards) hardware is now `mcc`. For this release, you can still use the previous vendor name (`cbi`), although this name will not be supported in future toolbox releases.

## **Known Software and Documentation Problems**

### **Serial Port Conflict**

The serial port is not locked by MATLAB so other applications or other instances of MATLAB can access the same serial port. This may result in a conflict.

### **Incorrect Vendor Driver Version Returned By instrhwinfo**

The vendor driver version returned by instrhwinfo to the VendorDriverVersion field is incorrect for Tektronix VISA and Agilent VISA.



# Instrument Control Toolbox 1.1 Release Notes

---

<b>New Features</b> . . . . .	2-2
VXI Block and FIFO Read Operations . . . . .	2-2
Freeing the Serial Port on Windows Platforms . . . . .	2-2
Data Parsing . . . . .	2-2
binblock Data Transfer . . . . .	2-3
Tektronix VISA Support . . . . .	2-3
Enhancements to Existing Functions and Properties . . . . .	2-3
<b>Vendor Driver Requirements and Limitations</b> . . . . .	2-4
Tektronix VISA Limitations . . . . .	2-4
<b>Upgrading from an Earlier Release</b> . . . . .	2-5
Events, Callbacks, and Function Handles . . . . .	2-5

## New Features

This section introduces the new features and enhancements in the Instrument Control Toolbox 1.1, added since the Instrument Control Toolbox 1.0 (Release 12.0). For a brief introduction to the product, refer to “Introduction to the Instrument Control Toolbox” on page 3-2.

### VXI Block and FIFO Read Operations

For VISA-VXI and VISA-GPIB-VXI objects, you can specify if the VXI register offset increments after data is transferred with the `MemoryIncrement` property.

You can configure `MemoryIncrement` to be `block` or `FIFO`. If `MemoryIncrement` is `block`, the `memread` and `memwrite` functions increment the offset after every read and write operation, and data is transferred from consecutive memory elements. If `MemoryIncrement` is `FIFO`, the `memread` and `memwrite` functions always read from or write to the same memory element. Note that the Instrument Control Toolbox 1.0 supported only block data transfer.

### Freeing the Serial Port on Windows Platforms

The serial port object uses the `javax.comm` package to communicate with the serial port. However, due to a memory leak in `javax.comm`, the serial port object is not released from memory. You can use the `freeserial` function to release the MATLAB hold on the serial port.

`freeserial` is necessary only on Windows platforms. You should use `freeserial` only if you need to connect to the serial port from another application after a serial port object has been connected to that port, and you do not want to exit MATLAB.

### Data Parsing

You can parse formatted data read from your instrument with the `scanstr` function. The data is parsed according to the specified delimiters, and is stored in a cell array as either a double or a string.

## binblock Data Transfer

The binary-block (binblock) format is defined as #<N><D><A> where:

- N specifies the number of digits D that follow.
- D specifies the number of data bytes A that follow.
- A is the data to written to the instrument.

You can transfer binblock data between MATLAB and your instrument with the `binblockread` and `binblockwrite` functions.

## Tektronix VISA Support

Tektronix VISA is supported for serial and GPIB interfaces.

## Enhancements to Existing Functions and Properties

### Terminator and EOSCharCode Properties

For serial port and VISA-serial objects, you can configure `Terminator` to a decimal value ranging from 0 to 127, to the equivalent ASCII character, or to empty ("). For serial port objects, you can also set `Terminator` to CR/LF or LF/CR.

For GPIB- and VXI-related objects, you can configure `EOSCharCode` to a decimal value ranging from 0 to 255, or to the equivalent ASCII character.

### Timer Events

Some timer events may not be processed if your system is significantly slowed or if the `TimerPeriod` value is too small. The minimum `TimerPeriod` value is now 0.01 second.

### VXI Register Read and Write Functions

The `memread`, `memwrite`, `memppeek` and `mempoke` functions now handle single precision values.

### CompareBits Property

For VISA-GPIB, VISA-VXI, and VISA-GPIB-VXI objects, the `CompareBits` property no longer exists since its only supported value is 8.

## Vendor Driver Requirements and Limitations

Driver requirements and limitations associated with the vendors supported by the Instrument Control Toolbox 1.0 are described in “Vendor Driver Requirements and Limitations” on page 3-4.

### Tektronix VISA Limitations

The Tektronix VISA driver has these known limitations:

- Only serial port and GPIB objects are supported.
- Asynchronous read and write operations for the VISA-GPIB object are not supported. Therefore, the following toolbox functionality is not supported:
  - The readasync function
  - The async flag for the fprintf and fwrite functions
  - The BytesAvailableFcn and OutputEmptyFcn properties

---

**Note** The driver limitations described in this section are restricted to the limitations directly associated with using the Instrument Control Toolbox.

---



## Upgrading from an Earlier Release

This section describes the upgrade issues involved in moving the Instrument Control Toolbox 1.0 (Release 12.0) to the Instrument Control Toolbox 1.1 (Release 12.1).

### **Events, Callbacks, and Function Handles**

Action properties and action functions are now referred to as callback properties and callback functions. This new terminology is reflected in new names for the associated properties and functions. The general rule for the name changes is to change “Action” to “Fcn” for properties, and “action” to “callback” for functions. For example, `TimerAction` has been renamed `TimerFcn`, and `instraction` has been renamed `instrcallback`.

Additionally, if you want to automatically pass the object and event information to the callback function, then you must specify the function as either a function handle or as a cell array. Note that you can also specify the callback function as a string. In this case, the callback is evaluated in the MATLAB workspace and no requirements are made on the function’s input arguments.



# Instrument Control Toolbox 1.0 Release Notes

---

<b>Introduction to the Instrument Control Toolbox</b> . . . . .	3-2
Supported Platforms . . . . .	3-2
Object-Based Design . . . . .	3-2
Reading and Writing Data . . . . .	3-3
Events and Actions . . . . .	3-3
Data Recording . . . . .	3-3
Graphical Tools . . . . .	3-3
<b>Vendor Driver Requirements and Limitations</b> . . . . .	3-4
Driver Requirements . . . . .	3-4
GPIB Driver Limitations . . . . .	3-5
VISA Driver Limitations . . . . .	3-7
<b>Known Software Problems</b> . . . . .	3-9
Pin Status Event . . . . .	3-9
Available Serial Ports . . . . .	3-9
Accessing Serial Ports on Solaris . . . . .	3-10

## Introduction to the Instrument Control Toolbox

The Instrument Control Toolbox 1.0 is a new product from The MathWorks that provides a framework for communicating with instruments that support the GPIB interface (IEEE-488, HP-IB), the VISA standard, and the serial port interface (RS-232, RS-422, and RS-485).

---

**Note** The toolbox extends the serial port features included with MATLAB.

---

### Supported Platforms

The supported platforms and associated toolbox features are given below.

Platform	Supported Toolbox Features
Microsoft Windows 98, Windows 2000, and Windows NT 4.0	All features
Sun Solaris	National Instruments GPIB
	National Instruments VISA
	Serial port
Linux	Serial port

### Object-Based Design

Based on MATLAB object technology, the Instrument Control Toolbox provides functions for creating objects directly associated with your instrument.

You can create objects for instruments that support the GPIB interface, the VISA standard, and serial port interface. For example, the following command creates the GPIB object `g` associated with a National Instruments GPIB controller with board index 0, and an instrument with primary address 1.

```
g = gpib('ni',0,1);
```

`g` contains all the necessary parameters for communicating with the instrument.

---

## Reading and Writing Data

Functions are available for transferring data between MATLAB and your instrument:

- The data can be binary (numerical) or text.
- Text data can be any command used by your instrument such as a command given by the Standard Commands for Programmable Instruments (SCPI) language.
- The transfer can be synchronous and block the command line, or asynchronous and not block the command line.

## Events and Actions

You can enhance the power and flexibility of your instrument control application by using events. An event occurs after a condition is met and may result in one or more actions. For example, you can use events to display a message, display data, or analyze data.

You configure actions through action properties and action functions. All event types have an associated action property. Action functions are M-file functions that you construct to suit your specific application needs.

Event types supported by the Instrument Control Toolbox include error events, timer events, and bytes-available events.

## Data Recording

Functions are provided for recording instrument commands, data, and events to one or more disk files. Recording this information to disk provides a permanent record of your instrument control session, and is an easy way to debug your application.

## Graphical Tools

Tools are provided that facilitate instrument control in an easy-to-use graphical environment. Using these tools, you can:

- Create and configure an instrument object
- Communicate with your instrument via text commands

## Vendor Driver Requirements and Limitations

The requirements and limitations for the vendor GPIB and VISA drivers supported by the Instrument Control Toolbox are described below.

---

**Note** The limitations described in this section are restricted to the limitations directly associated with using the Instrument Control Toolbox.

---

### Driver Requirements

You can use the Instrument Control Toolbox with the GPIB and VISA drivers listed below.

Interface	Vendor	Driver Requirements
GPIB	Agilent	SICL version H.012.02.00
	ComputerBoards	GPIB-32 version 2.12
	Capital Equipment Corporation	CEC-488 version 5.08
	IOTech	IOTech version 1.0
	Keithley	Keithley-488 version 5.08
	National Instruments	NI-488 version 1.41.7649
VISA	Agilent	Agilent-VISA version 1.1
	National Instruments	NI-VISA version 2.0

See the following sections for a description of:

- “GPIB Driver Limitations” on page 3-5
- “VISA Driver Limitations” on page 3-7

## **GPIB Driver Limitations**

### **Agilent Technologies**

The Agilent GPIB driver has these limitations:

- Asynchronous read and write operations are not supported. Therefore, Agilent GPIB objects do not support the following toolbox functionality:
  - The `readasync` function
  - The `async` flag for the `fprintf` and `fwrite` functions
  - `BytesAvailableAction` and `OutputEmptyAction` properties
- The End Or Identify (EOI) line is not asserted when the End-Of-String (EOS) character is written to the hardware. Therefore, when the `EOSMode` property is configured to write and the `EOIMode` property is configured to on, the EOI line is not asserted when the `EOSCharCode` property value is written to the hardware.
- All eight bits are used for the EOS comparison. Therefore, the only value supported by the `CompareBits` property is 8.
- A board index value of 0 is not supported.
- An error is not reported for an invalid primary address. Instead, the read and write operations will time out.

### **ComputerBoards**

The ComputerBoards GPIB driver does not support asynchronous notification for the completion of read and write operations. Therefore, ComputerBoards GPIB objects do not support the following toolbox functionality:

- The `readasync` function
- The `async` flag for the `fprintf` and `fwrite` functions
- The `BytesAvailableAction` and `OutputEmptyAction` properties

### Capital Equipment Corporation

The Capital Equipment Corporation (CEC) GPIB driver has these limitations:

- Asynchronous read and write operations are not supported. Therefore, CEC GPIB objects do not support the following toolbox functionality:
  - The `readasync` function
  - The `async` flag for the `fprintf` and `fwrite` functions
  - The `BytesAvailableAction` and `OutputEmptyAction` properties
- The `Handshake` and `Bus Management` line values are not provided. The `BusManagementStatus` and `HandshakeStatus` properties always return the line values as `on`.
- The `EOI` line is not asserted when the `EOS` character is written to the hardware. Therefore, when the `EOSMode` property is configured to `write` and the `EOIMode` property is configured to `on`, the `EOI` line is not asserted when the `EOSCharCode` property value is written to the hardware.
- All eight bits are used for the `EOS` comparison. Therefore, the only value supported by the `CompareBits` property is `8`.
- You should not simultaneously use a GPIB controller address of `0` and an instrument primary address of `0`.

### IOTech

The IOTech GPIB driver has these limitations:

- Asynchronous read and write operations are not supported. Therefore, IOTech GPIB objects do not support the following toolbox functionality:
  - The `readasync` function
  - The `async` flag for the `fprintf` and `fwrite` functions
  - The `BytesAvailableAction` and `OutputEmptyAction` properties.
- Incorrect values are returned for the `REN` and `IFC` bus management lines. The `BusManagementStatus` property always returns a value of `on` for the `RemoteEnable` and the `InterfaceClear` fields.
- The `EOI` line is not asserted when the `EOS` character is written to the hardware. Therefore, when the `EOSMode` property is configured to `write` and the `EOIMode` property is configured to `on`, the `EOI` line will not be asserted when the `EOSCharCode` property value is written to the hardware.



## Keithley

The Keithley GPIB driver has these limitations:

- Asynchronous read and write operations are not supported. Therefore, Keithley GPIB objects do not support the following toolbox functionality:
  - The readasync function
  - The async flag for the fprintf and fwrite functions
  - The BytesAvailableAction and OutputEmptyAction properties
- The Handshake and Bus Management line values are not provided. The BusManagementStatus and HandshakeStatus properties always return the line value as on.
- The EOI line is not asserted when the EOS character is written to the hardware. Therefore, when the EOSMode property is configured to write and the EOIMode property is configured to on, the EOI line will not be asserted when the EOSCharCode property value is written to the hardware.
- All eight bits are used for the EOS comparison. Therefore, the only value supported by the CompareBits property is 8.
- You should not simultaneously use a GPIB controller address of 0 and an instrument primary address of 0.

## VISA Driver Limitations

### Agilent Technologies

The Agilent VISA driver has these known limitations:

- Asynchronous read and write operations for GPIB, GPIB-VXI and VXI VISA objects are not supported. Therefore, the following toolbox functionality is not supported for these objects:
  - The readasync function
  - The async flag for the fprintf and fwrite functions
  - The BytesAvailableAction and OutputEmptyAction properties
- All eight bits are used for the EOS comparison. Therefore, the only value supported by the CompareBits property is 8.

### **National Instruments**

The National Instruments VISA driver uses all eight bits for the EOS comparison. Therefore, the only value supported by the CompareBits property is 8.

## Known Software Problems

This section updates the Instrument Control Toolbox documentation, reflecting known software problems.

### Pin Status Event

On the Microsoft Windows 98 platforms, a pin-status event is not produced when the ring indicator pin changes value.

### Available Serial Ports

#### Windows 98

On Windows 98 platforms, you can use only the COM1 through COM4 ports. Additionally, the `instrhwinfo` function always indicates that these ports exist. To find out the actual serial ports that are available, examine the `AvailableSerialPorts` field.

```
out = instrhwinfo('serial');
out.AvailableSerialPorts
out =
    'COM1 '
    'COM2 '
```

#### Solaris

On Solaris, all serial ports are returned by `instrhwinfo` – including ports that you cannot access. For example, `instrhwinfo` may return the following ports.

```
out = instrhwinfo('serial')
out.AvailableSerialPorts
ans =
    '/dev/term/a'
    '/dev/term/b'
    '/dev/cua/a'
    '/dev/cua/b'
```

However, only `/dev/term/a` and `/dev/term/b` are available for communicating with instruments.

## **Accessing Serial Ports on Solaris**

If you repeatedly open and close one or more serial ports on Solaris, MATLAB will become unresponsive. To minimize the chance of encountering this problem, you should:

- Use only one serial port on your platform.
- Connect the serial port object to your instrument once per MATLAB session.

Alternatively, if you have NI-VISA installed, you can completely avoid this problem by using a VISA-serial object.