MS05P/MS10P/MS10E Platform

User Guide

Copyright © 2011 JGR Optics Inc. All rights reserved. No part of this publication may be reproduced, stored in a retrieval system or transmitted in any form, be it electronically, mechanically, or by any other means such as photocopying, recording or otherwise, without the prior written permission of JGR Optics Inc. (JGR).

Information provided by JGR is believed to be accurate and reliable. However, no responsibility is assumed by JGR for its use nor for any infringements of patents or other rights of third parties that may result from its use. No license is granted by implication or otherwise under any patent rights of JGR.

JGR's Commerce And Government Entities (CAGE) code under the North Atlantic Treaty Organization (NATO) is 0L8C3.

The information contained in this publication is subject to change without notice.

Trademarks

JGR's trademarks have been identified as such. However, the presence or absence of such identification does not affect the legal status of any trademark.

Units of Measurement

Units of measurement in this publication conform to SI standards and practices.

Version number: 1.0.0

End-User License Agreement

You have acquired a device ("DEVICE") that includes software licensed by JGR Optics Inc. (JGR) from an affiliate of Microsoft Corporation ("MS"). Those installed software products of MS origin, as well as associated media, printed materials, and "online" or electronic documentation ("SOFTWARE") are protected by international intellectual property laws and treaties. Manufacturer, MS and its suppliers (including Microsoft Corporation) own the title, copyright, and other intellectual property rights in the SOFTWARE. The SOFTWARE is licensed, not sold. All rights reserved.

This EULA is valid and grants the end-user rights ONLY if the SOFTWARE is genuine and a genuine Certificate of Authenticity for the SOFTWARE is included. For more information on identifying whether your software is genuine, please see http://www.microsoft.com/piracy/howtotell.

IF YOU DO NOT AGREE TO THIS END USER LICENSE AGREEMENT ("EULA"), DO NOT USE THE DEVICE OR COPY THE SOFTWARE. INSTEAD, PROMPTLY CONTACT JGR FOR INSTRUCTIONS ON RETURN OF THE UNUSED DEVICE(S) FOR A REFUND. **ANY USE OF THE SOFTWARE, INCLUDING BUT NOT LIMITED TO USE ON THE DEVICE, WILL CONSTITUTE YOUR AGREEMENT TO THIS EULA (OR RATIFICATION OF ANY PREVIOUS CONSENT)**.

GRANT OF SOFTWARE LICENSE. This EULA grants you the following license:

- You may use the SOFTWARE only on the DEVICE.
- **Restricted Functionality.** You are licensed to use the SOFTWARE to provide only the limited functionality (specific tasks or processes) for which the DEVICE has been designed and marketed by JGR. This license specifically prohibits any other use of the software programs or functions, or inclusion of additional software programs or functions that do not directly support the limited functionality on the DEVICE. Notwithstanding the foregoing, you may install or enable on a DEVICE, systems utilities, resource management or similar software solely for the purpose of administration, performance enhancement and/or preventive maintenance of the DEVICE.
- If you use the DEVICE to access or utilize the services or functionality of Microsoft Windows Server products (such as Microsoft Windows Server 2003), or use the DEVICE to permit workstation or computing devices to access or utilize the services or functionality of Microsoft Windows Server products, you may be required to obtain a Client Access License for the DEVICE and/or each such workstation or computing device. Please refer to the end user license agreement for your Microsoft Windows Server product for additional information.
- **NOT FAULT TOLERANT.** THE SOFTWARE IS NOT FAULT TOLERANT. JGR HAS INDEPENDENTLY DETERMINED HOW TO USE THE SOFTWARE IN THE DEVICE, AND MS HAS RELIED UPON JGR TO CONDUCT SUFFICIENT TESTING TO DETERMINE THAT THE SOFTWARE IS SUITABLE FOR SUCH USE.
- **NO WARRANTIES FOR THE SOFTWARE.** THE SOFTWARE is provided "AS IS" and with all faults. THE ENTIRE RISK AS TO SATISFACTORY QUALITY, PERFORMANCE, ACCURACY, AND EFFORT (INCLUDING LACK OF NEGLIGENCE) IS WITH YOU. ALSO, THERE IS NO WARRANTY AGAINST INTERFERENCE WITH YOUR ENJOYMENT OF THE SOFTWARE OR AGAINST INFRINGEMENT. **IF YOU HAVE RECEIVED ANY WARRANTIES REGARDING THE DEVICE OR THE SOFTWARE, THOSE WARRANTIES DO NOT ORIGINATE FROM, AND ARE NOT BINDING ON, MS.**
- No Liability for Certain Damages. **EXCEPT AS PROHIBITED BY LAW, MS SHALL HAVE NO LIABILITY FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL OR INCIDENTAL DAMAGES ARISING FROM OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THE SOFTWARE. THIS LIMITATION SHALL APPLY EVEN IF ANY REMEDY FAILS OF ITS ESSENTIAL PURPOSE. IN NO EVENT SHALL MS BE LIABLE FOR ANY AMOUNT IN EXCESS OF U.S. TWO HUNDRED FIFTY DOLLARS (U.S.$250.00).**
- **Restricted Uses.** The SOFTWARE is not designed or intended for use or resale in hazardous environments requiring fail-safe performance, such as in the operation of nuclear facilities, aircraft navigation or communication systems, air traffic control, or other devices or systems in which a malfunction of the SOFTWARE would result in foreseeable risk of injury or death to the operator of the device or system, or to others.
- **Limitations on Reverse Engineering, Decompilation, and Disassembly.** You may not reverse engineer, decompile, or disassemble the SOFTWARE, except and only to the extent that such activity is expressly permitted by applicable law notwithstanding this limitation.
- **SOFTWARE as a Component of the DEVICE - Transfer.** This license may not be shared, transferred to or used concurrently on different computers. The SOFTWARE is licensed with the DEVICE as a single integrated product and may only be used with the DEVICE. If the SOFTWARE is not accompanied by a DEVICE, you may not use the SOFTWARE. You may permanently transfer all of your rights under this EULA only as part of a permanent sale or transfer of the DEVICE, provided you retain no copies of the SOFTWARE. If the SOFTWARE is an upgrade, any transfer must also include all prior versions of the SOFTWARE. This transfer must also include the Certificate of Authenticity label. The transfer may not be an indirect transfer, such as a consignment. Prior to the transfer, the end user receiving the SOFTWARE must agree to all the EULA terms.
- **Consent to Use of Data.** You agree that MS, Microsoft Corporation and their affiliates may collect and use technical information gathered in any manner as part of product support services related to the SOFTWARE. MS, Microsoft Corporation and their affiliates may use this information solely to improve their products or to provide customized services or technologies to you. MS, Microsoft Corporation and their affiliates may disclose this information to others, but not in a form that personally identifies you.

- **Internet Gaming/Update Features.** If the SOFTWARE provides, and you choose to utilize, the Internet gaming or update features within the SOFTWARE, it is necessary to use certain computer system, hardware, and software information to implement the features. By using these features, you explicitly authorize MS, Microsoft Corporation and/or their designated agent to use this information solely to improve their products or to provide customized services or technologies to you. MS or Microsoft Corporation may disclose this information to others, but not in a form that personally identifies you.
- **Internet-Based Services Components.** The SOFTWARE may contain components that enable and facilitate the use of certain Internet-based services. You acknowledge and agree that MS, Microsoft Corporation or their affiliates may automatically check the version of the SOFTWARE and/or its components that you are utilizing and may provide upgrades or supplements to the SOFTWARE that may be automatically downloaded to your DEVICE. Microsoft Corporation or their affiliates do not use these features to collect any information that will be used to identify you or contact you. For more information about these features, please see the privacy statement at http://go.microsoft.com/fwlink/?LinkId=25243.
- **Links to Third Party Sites.** You may link to third party sites through the use of the SOFTWARE. The third party sites are not under the control of MS or Microsoft Corporation, and MS or Microsoft are not responsible for the contents of any third party sites, any links contained in third party sites, or any changes or updates to third party sites. MS or Microsoft Corporation is not responsible for webcasting or any other form of transmission received from any third party sites. MS or Microsoft Corporation are providing these links to third party sites to you only as a convenience, and the inclusion of any link does not imply an endorsement by MS or Microsoft Corporation of the third party site.
- **Notice Regarding Security.** To help protect against breaches of security and malicious software, periodically back up your data and system information, use security features such as firewalls, and install and use security updates.
- **No Rental/Commercial Hosting.** You may not rent, lease, lend or provide commercial hosting services with the SOFTWARE to others.
- **Separation of Components.** The SOFTWARE is licensed as a single product. Its component parts may not be separated for use on more than one computer.
- **Additional Software/Services.** This EULA applies to updates, supplements, add-on components, product support services, or Internet-based services components ("Supplemental Components"), of the SOFTWARE that you may obtain from JGR, MS, Microsoft Corporation or their subsidiaries after the date you obtain your initial copy of the SOFTWARE, unless you accept updated terms or another agreement governs. If other terms are not provided along with such Supplemental Components and the Supplemental Components are provided to you by MS, Microsoft Corporation or their subsidiaries then you will be licensed by such entity under the same terms and conditions of this EULA, except that (i) MS, Microsoft Corporation or their subsidiaries providing the Supplemental Components will be the licensor with respect to such Supplemental Components in lieu of the "COMPANY" for the purposes of the EULA, and (ii) TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, THE SUPPLEMENTAL COMPONENTS AND ANY (IF ANY) SUPPORT SERVICES RELATED TO THE SUPPLEMENTAL COMPONENTS ARE PROVIDED AS IS AND WITH ALL FAULTS. ALL OTHER DISCLAIMERS, LIMITATION OF DAMAGES, AND SPECIAL PROVISIONS PROVIDED BELOW AND/OR OTHERWISE WITH THE SOFTWARE SHALL APPLY TO SUCH SUPPLEMENTAL COMPONENTS. MS, Microsoft Corporation or their subsidiaries reserve the right to discontinue any Internet-based services provided to you or made available to you through the use of the SOFTWARE.
- **Recovery Media.** If SOFTWARE is provided by JGR on separate media and labeled "Recovery Media" you may use the Recovery Media solely to restore or reinstall the SOFTWARE originally installed on the DEVICE.
- **Backup Copy.** You may make one (1) backup copy of the SOFTWARE. You may use this backup copy solely for your archival purposes and to reinstall the SOFTWARE on the DEVICE. Except as expressly provided in this EULA or by local law, you may not otherwise make copies of the SOFTWARE, including the printed materials accompanying the SOFTWARE. You may not loan, rent, lend or otherwise transfer the backup copy to another user.
- **End User Proof of License.** If you acquired the SOFTWARE on a DEVICE, or on a compact disc or other media, a genuine Microsoft "Proof of License"/Certificate of Authenticity label with a genuine copy of the SOFTWARE identifies a licensed copy of the SOFTWARE. To be valid, the label must be affixed to the DEVICE, or appear on [COMPANY's] software packaging. If you receive the label separately other than from JGR, it is invalid. You should keep the label on the DEVICE or packaging to prove that you are licensed to use the SOFTWARE.
- **Product Support.** Product support for the SOFTWARE is not provided by MS, Microsoft Corporation, or their affiliates or subsidiaries. For product support, please refer to JGR support number provided in the documentation for the DEVICE. Should you have any questions concerning this EULA, or if you desire to contact JGR for any other reason, please refer to the address provided in the documentation for the DEVICE.
- **Termination.** Without prejudice to any other rights, JGR may terminate this EULA if you fail to comply with the terms and conditions of this EULA. In such event, you must destroy all copies of the SOFTWARE and all of its component parts.
- **EXPORT RESTRICTIONS.** You acknowledge that SOFTWARE is subject to U.S. and European Union export jurisdiction. You agree to comply with all applicable international and national laws that apply to the SOFTWARE, including the U.S. Export Administration Regulations, as well as end-user, end-use and destination restrictions issued by U.S. and other governments. For additional information see http://www.microsoft.com/exporting/.

Contents

Certification Information ix

1 Introducing the MS Platform 1

Main Features 1
Available Models 2
Understanding MS Platform LEDs (MS05P) 7
Understanding MS Platform LEDs (MS10P) 8
Understanding MS Platform Hardware Buttons and Selection Dial (MS05P) 9
Locating MS Platform Synchronization Ports 10
Hard Drive Structure 11
Conventions 12

2 Safety Information 13

Laser Safety Information 13
Electrical Safety Information 14

3 Getting Started with Your MS Platform 17

Installing Your Controller and Expansion Units in a Rackmount 18
Connecting One or More Expansion Units to a Controller Unit 22
Inserting and Removing Test Modules 25
Safeguarding Units with the Interlock 30
Turning the Platform On and Off 32
Installing or Upgrading MS Manager 33
Accessing MS Manager 34
Exiting MS Manager 35

4 Setting Up Your MS Platform 37
Calibrating Your Touchscreen (MS05P) 37
Installing an External Monitor 38
Adjusting the Display and Sound 40
Deactivating the LED Display 41
Selecting Regional and Language Options 42
Changing the Backlight Level (MS05P) 43
Installing an External Keyboard 43
Activating and Selecting the On-Screen Keyboard Type 44
Installing a Mouse 46
Installing a Tape, CD, or DVD Player 47
Installing a Headset or a Speaker 48
Installing a Microphone 49
Installing a Printer 50
Adjusting the Date and Time 51
Selecting the Startup Interface 52
Customizing Access Levels 53
Setting Up Auto Logon for Windows XP 55
Setting Communication Parameters 57
Adjusting the Fan Speed 57
Saving and Opening Configurations 59
Saving the Configuration on Exiting 61
Changing Unit and Module Identification 62
Reverting to Factory Settings 64
Accessing Your MS Platform Remotely 65

5 Using the MS Manager Interface 79
Accessing Current Modules 79
Creating an Integrated Data Display 83
Locating Units 85
Accessing Utilities from MS Manager 86

6 Preparing for Automation or Remote Control 87
Configuring Your Platform as a GPIB Controller 91
Configuring Your Platform as a GPIB Device 93
Preparing Hardware for GPIB Control 98
Linking Units with the GPIB Port 100
Linking Units with the Ethernet Port 100
Linking Units with the Serial Port 101
Getting Optimum Performance from Your Platform 101
Changing Communication Settings 104
Preparing for Remote Control with DCOM 108

7 Using MS Products in an Automated Test Environment 143
Managing the States of Your MS Platform 144
Message Management 148
Standard Status Data Structure 151
SCPI Command Structure 155
Consulting Data Types 159
Writing Remote Control Code 159
Error Message Format 161
Working with JGR COM Objects 162
Monitoring Remote Commands 163

8 Controlling Modules Remotely 167

9 Maintenance 173
Cleaning the Touchscreen (MS05P) 173
Replacing the Fuse 173
Recycling and Disposal (Applies to European Union Only) 174

10 Troubleshooting 175
Solving Common Problems 175
Removing MS Manager Manually 176
Viewing Online Documentation 178
Customer Support and Technical Assistance 179
Viewing System Information 180
Transportation 184

A SCPI-Based Errors 185

B Data Types 201
Applicable Data Types for Input—IEEE 488.2 202
Applicable Data Types for Output —IEEE 488.2 211
Applicable Data Types for Input—SCPI 221
Special Numeric Values Received on Output 222

C IEEE 488.2 and Specific Commands 223
IEEE 488.2 Commands–Quick Reference 223
IEEE 488.2 Required Commands 224
Specific Commands—Quick Reference 244
Specific Commands 245

D COM Properties and Events 263
ActiveX (COM/DCOM)—Quick Reference 264
Properties 265
Events 272

E Communicating Through TCP/IP over Telnet 273
Introducing TCP/IP over Telnet 273
Features 273
Getting Started with TCP/IP over Telnet 274
Executing SCPI Commands Over Telnet 276
Examples for Releasing Modules 280
Internal Commands of the TCP/IP over Telnet Protocol 283

Index 287

Certification Information

FCC Information

Electronic test equipment is exempt from Part 15 compliance (FCC) in the United States. However, compliance verification tests are systematically performed on most JGR equipment.

CE Information

Electronic test equipment is subject to the EMC Directive in the European Union. The EN61326 standard prescribes both emission and immunity requirements for laboratory, measurement, and control equipment. This unit has undergone extensive testing according to the European Union Directive and Standards.

CSA Information

This unit is certified by the CSA (certificate number 162451) and was evaluated according to applicable CSA and UL standards (as confirmed by "C-US" mark) as well as applicable IEC standards for use in Canada, the United States, and other countries.

IMPORTANT

Use of shielded remote I/O cables, with properly grounded shields and metal connectors, is recommended in order to reduce radio frequency interference that may emanate from these cables.

1 Introducing the MS Platform

Main Features

The MS Platform is a platform designed for scientific and industrial applications. It is composed of one MS05P or MS10P controller unit and up to nine optional MS10E expansion units.

Both controller and expansion units are designed to work with MS modules.

The MS Platform runs under Microsoft Windows XP with the MS Manager software. Your controller unit features several possibilities, such as trigger and remote control ports, interlock terminals, as well as standard computer ports.

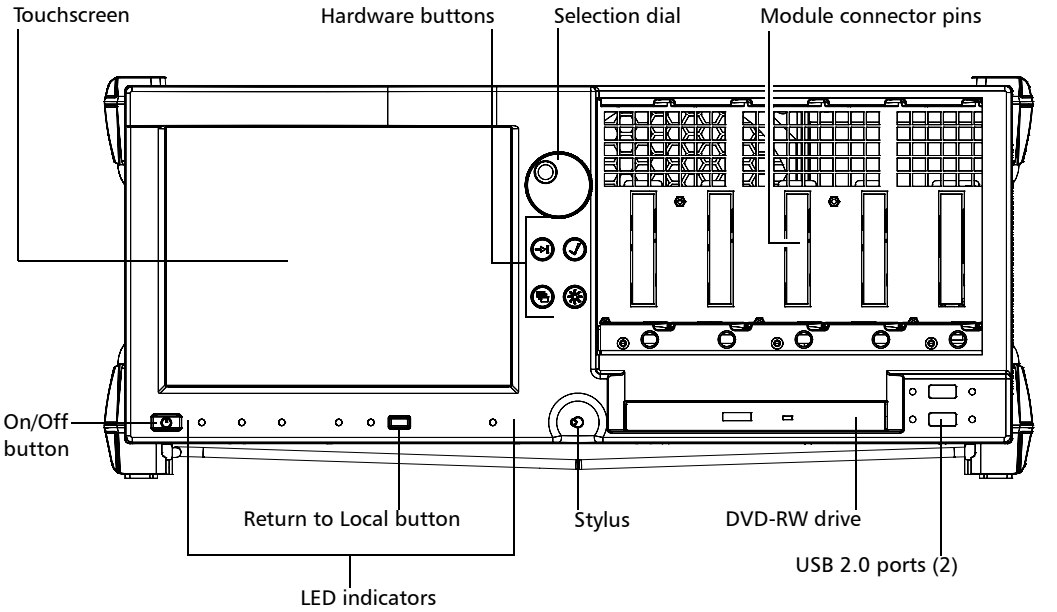
The MS Platform supports local control (via the MS Manager software) and remote control (through GPIB, RS-232 or Ethernet TCP/IP—using SCPI commands).

Available Models

Below are diagrams to locate the components on your controller and expansion units.

MS05P Controller Unit

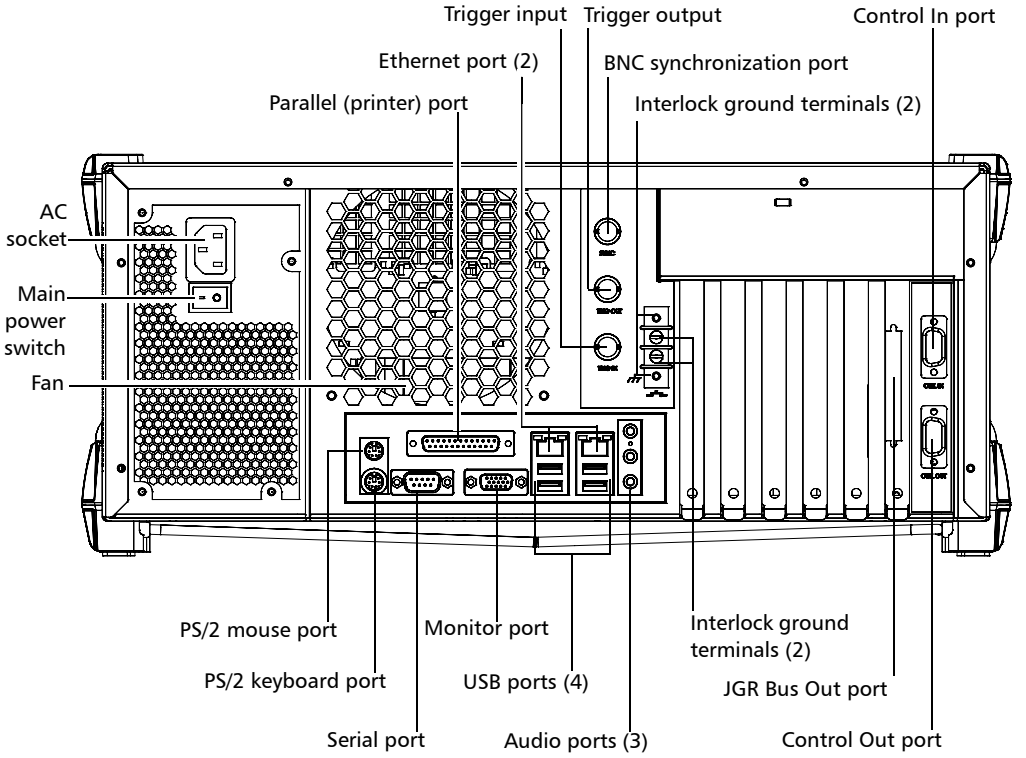
The front of the MS05P controller unit is where you perform basic operations such as turning your unit on or off, consulting LED indicators, and using data-related peripherals (for example: the USB ports).



For details on:

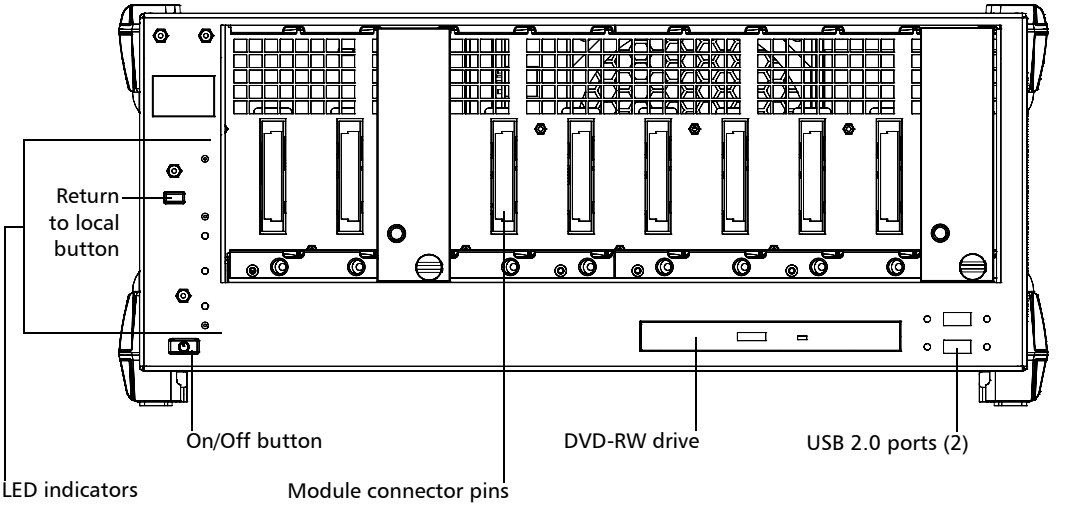
- System hardware buttons and selection dial, see *Understanding MS Platform Hardware Buttons and Selection Dial (MS05P)* on page 9.
- LED indicators, see *Understanding MS Platform LEDs (MS05P)* on page 7.

The back of the MS05P controller unit is where you find the peripheral connectors, as well as the power outlets and connectors to link expansion units to your controller unit.



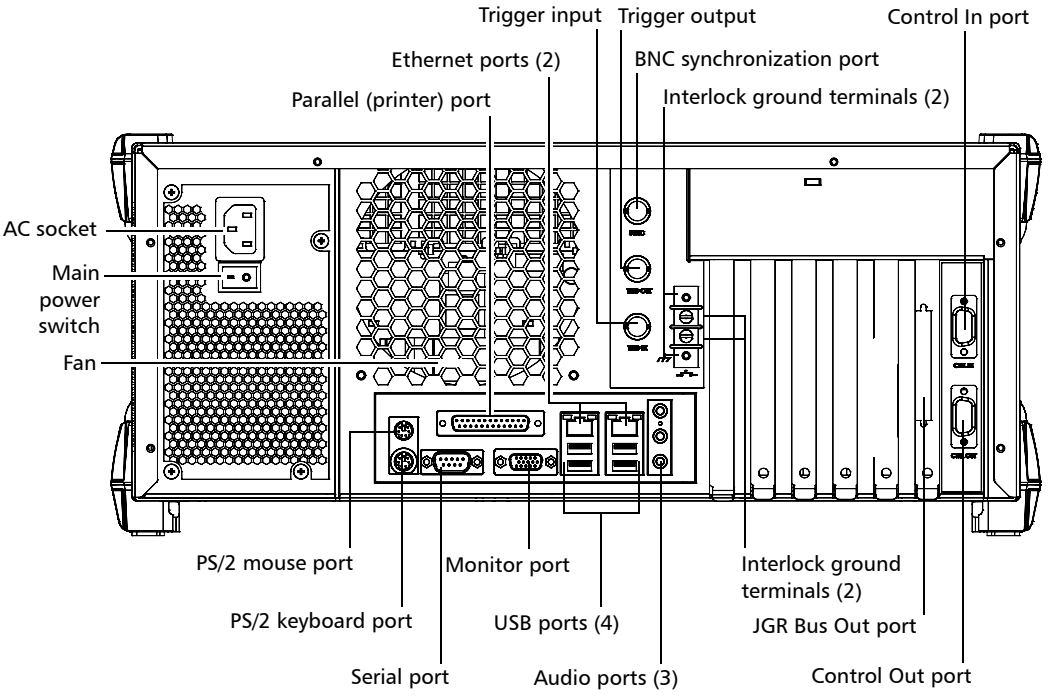
MS10P Controller Unit

The front of the MS10P controller unit is where you perform basic operations such as turning your unit on or off, consulting LED indicators and using data-related peripherals (for example: the USB ports and the DVD-RW drive).



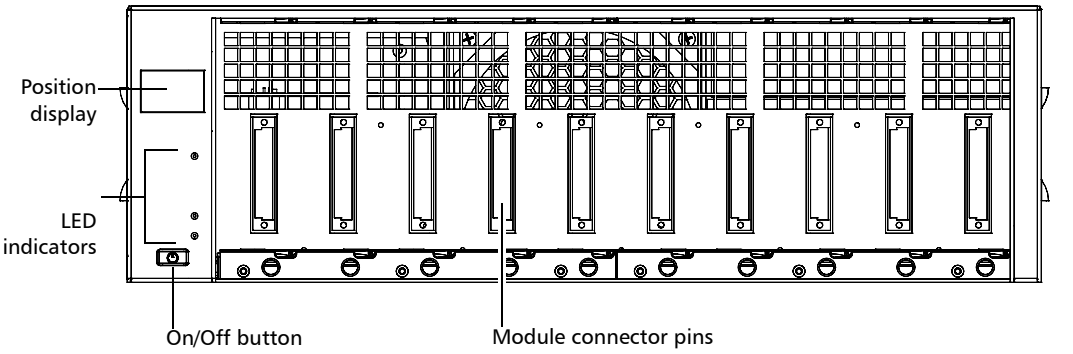
For details on LED indicators, see *Understanding MS Platform LEDs (MS10P)* on page 8

The back of the MS10P controller unit is where you find the peripheral connectors, as well as the power outlets and connectors to link expansion units to your controller unit.

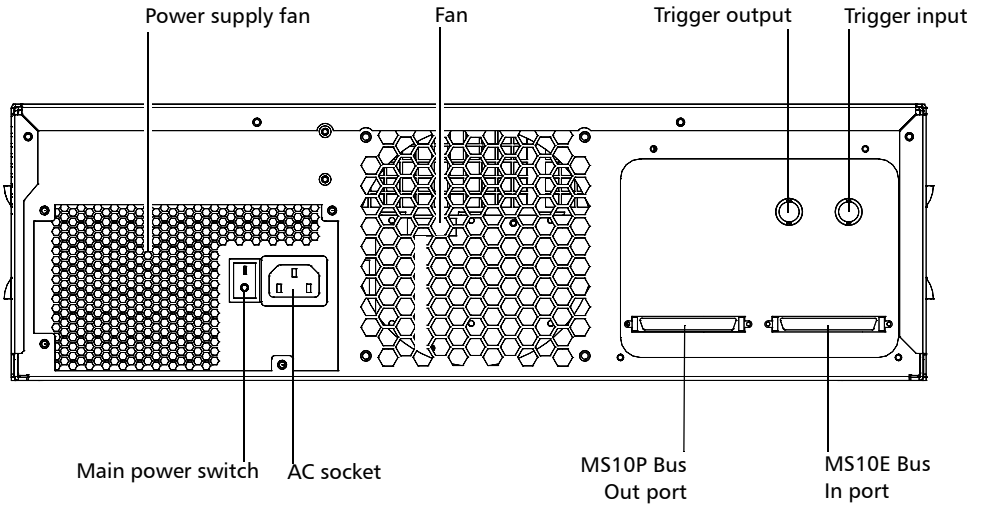


MS10E Expansion Unit

The front of the MS10E expansion unit is where you perform basic operations such as turning your unit on or off and consulting LED indicators.



The back of the MS10E expansion unit is where you find the power outlets and connectors to link your expansion units to your controller unit.



Understanding MS Platform LEDs (MS05P)

The LEDs on your MS Platform help you determine the current status of your unit.

Alarm indicator LED

When lit on the controller unit, it indicates a problem.

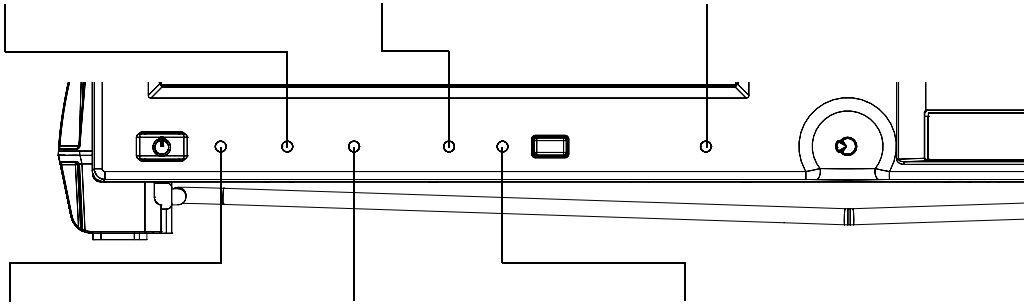
When lit on an expansion unit, the latter is not linked to the controller directly or through a chain.

Lock LED

When lit, the unit is in Lockout mode; you can only disable this mode by sending the corresponding remote command.

Interlock LED

When lit, it indicates that the Interlock is open (no current is passing through) and that the lasers in your unit are deactivated.



On/Off LED

When lit, it indicates that the unit is currently on.

Hard drive activity LED

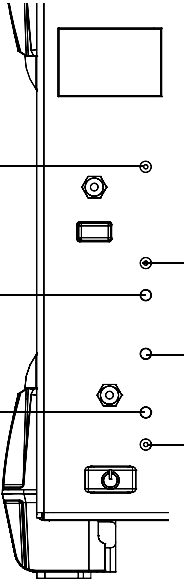
When blinking, it indicates current hard drive activity.

Remote LED

When lit, the unit is in Remote mode and the local peripherals such as the keyboard, selection dial and mouse are disabled.

Understanding MS Platform LEDs (MS10P)

The LEDs on your MS Platform help you determine the current status of your unit.



Interlock LED

When lit, it indicates that the Interlock is open (no current is passing through) and that the lasers present in your unit are deactivated.

Remote LED

When lit, the unit is in Remote mode and the keyboard and mouse are disabled.

Lock LED

When lit, the unit is in Lockout mode; you can only disable this mode by sending the corresponding remote command.

Hard drive activity LED

When blinking, it indicates current hard drive activity.

On/Off LED

When lit, it indicates that the unit is currently on.

Alarm indicator LED

When lit on the controller unit, it indicates a problem.
When lit on an expansion unit, the latter is not linked to the controller directly or through a chain.

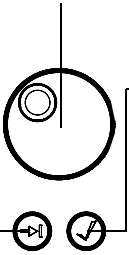
Understanding MS Platform Hardware Buttons and Selection Dial (MS05P)

The hardware buttons and selection dial on your MS Platform allow you to perform various tasks.

Selection Dial

Turn it to scroll up or down in lists, or to move slider.

Push it to select the item you have highlighted in the list.



Tab

This button allows you to switch between fields on an interface, or between the applications brought by the Program Switcher button. It is the equivalent of pressing the Tab key on a keyboard.

Enter

This button allows you to select the highlighted element on the screen. It is the equivalent of using the Enter key on a keyboard.

Program Switcher

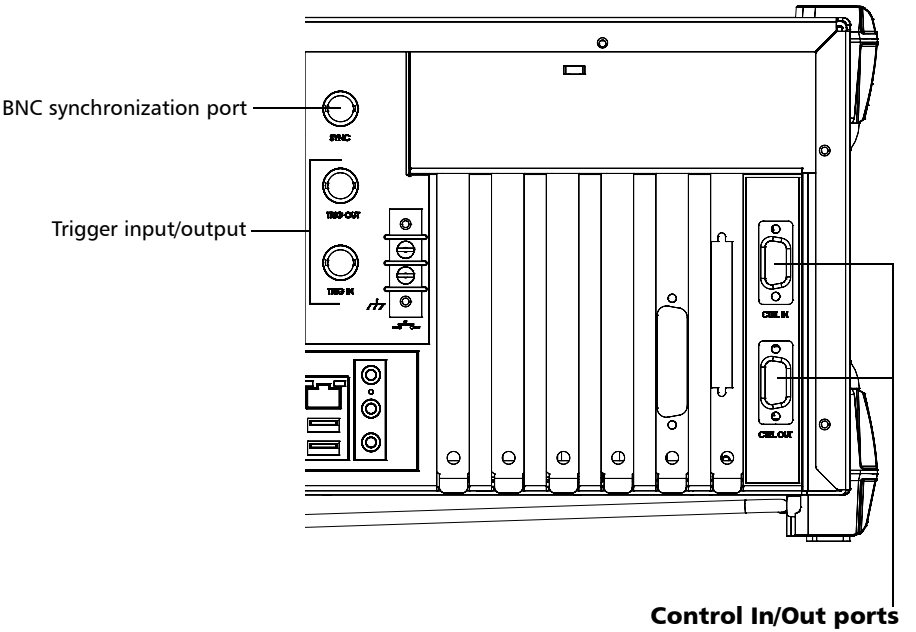
This button allows you to access a list of currently running applications. It is the equivalent of pressing the Alt-Tab combination on a keyboard.

Backlight

This button allows you to set the display brightness level.

Locating MS Platform Synchronization Ports

The synchronization and trigger ports are not currently supported by the MS Platform. The ports located at the back of the unit are reserved for future use only.



Control In/Out ports

Used to connect specific JGR units together.
Reserved for JGR.

Hard Drive Structure

Your MS Platform uses several hard drive partitions for specific purposes.

Partition	Contents
C	Operating System, JGR applications
D	User/JGR data, JGR application configurations
E	Utilities, MS Platform backup

Note: *The drive F corresponds to the DVD-RW drive.*

Note: *Contact JGR's Technical Support for further help if you need to use the MS Platform ghost.*

Conventions

Before using the product described in this manual, you should understand the following conventions:

WARNING

Indicates a potentially hazardous situation which, if not avoided, could result in *death or serious injury*. Do not proceed unless you understand and meet the required conditions.

CAUTION

Indicates a potentially hazardous situation which, if not avoided, may result in *minor or moderate injury*. Do not proceed unless you understand and meet the required conditions.

CAUTION

Indicates a potentially hazardous situation which, if not avoided, may result in *component damage*. Do not proceed unless you understand and meet the required conditions.

IMPORTANT

Refers to information about this product you should not overlook.

2 Safety Information

Laser Safety Information

Your MS Platform does not include laser components in itself. However, your modules may do so. Refer to your modules' user guides for further laser safety details and instructions.

WARNING

Do not install or terminate fibers while a light source is active. Never look directly into a live fiber and ensure that your eyes are protected at all times.

WARNING

Use of controls, adjustments and procedures for operation and maintenance other than those specified herein may result in hazardous radiation exposure or impair the protection provided by this unit.

Electrical Safety Information

This unit uses an international safety standard three-wire power cable. This cable serves as a ground when connected to an appropriate AC power outlet.

Note: *If you need to ensure that the unit is completely turned off, disconnect the power cable.*

WARNING

- **Insert the power cable plug into a power outlet with a protective ground contact. Do not use an extension cord without a protective conductor.**
- **Before turning on the unit, connect all grounding terminals, extension cords and devices to a protective ground via a ground socket. Any interruption of the protective grounding is a potential shock hazard and may cause personal injury. Whenever the ground protection is impaired, do not use the unit and secure it against any accidental operation.**
- **Do not tamper with the protective ground terminal.**

The color coding used in the electric cable depends on the cable. New plugs should meet the local safety requirements and include:

- adequate load-carrying capacity
- ground connection
- cable clamp

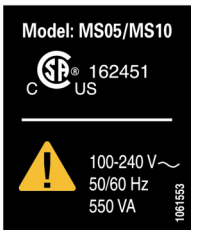
WARNING

- **Use this unit indoors only.**
- **Position the unit so that the air can circulate freely around it.**
- **Do not remove unit covers during operation.**
- **Operation of any electrical instrument around flammable gases or fumes constitutes a major safety hazard.**
- **To avoid electrical shock, do not operate the unit if any part of the outer surface (covers, panels, etc.) is damaged.**
- **Only authorized personnel should carry out adjustments, maintenance or repair of opened units under voltage. A person qualified in first aid must also be present. Do not replace any components while power cable is connected.**
- **Capacitors inside the unit may be charged even if the unit has been disconnected from its electrical supply.**

Equipment Ratings	
Temperature ➤ Operation ➤ Storage	 0 °C to 40 °C (32 °F to 104 °F) -40 °C to 70 °C (-40 °F to 158 °F)
Relative humidity[1]	0 % to 80 % non-condensing
Maximum operation altitude	3000 m (9843 ft)
Pollution degree	2
Overvoltage category	II
Power supply rating[2]	100 V to 240 V (50 Hz/60 Hz) maximum input power 550 VA

1. Measured in 0 °C to 31 °C (32 °F to 87.8 °F) range, decreasing linearly to 50 % at 40 °C (104 °F).
2. Not exceeding ± 10 % of the nominal voltage.

The following label is located on the back panel of the unit:

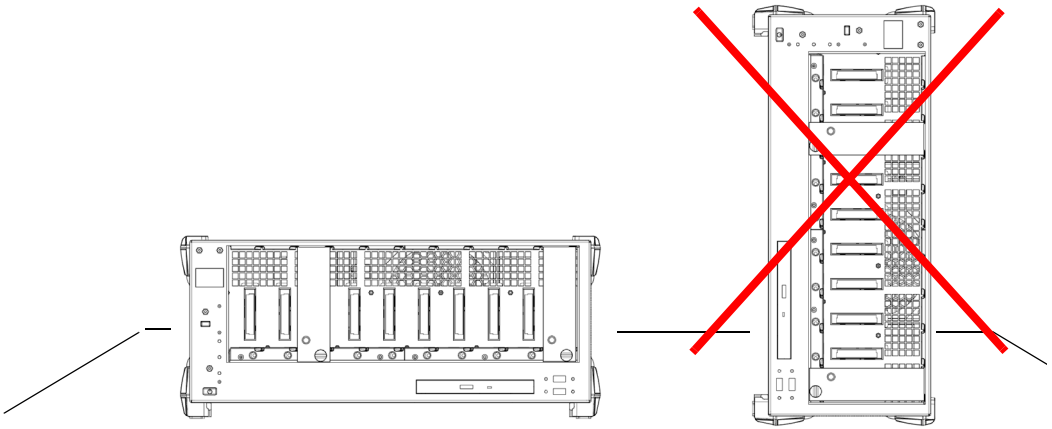


3 Getting Started with Your MS Platform

You can either use your MS Platform as a benchtop unit, or you can install your units in a rackmount.

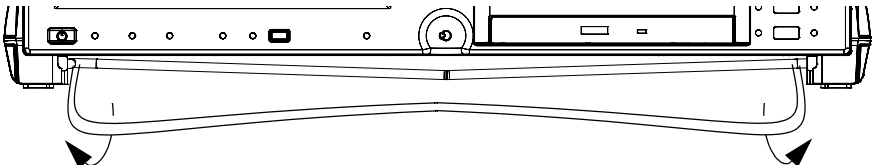
CAUTION

If you are using your MS Platform as a benchtop unit, do not turn it on its side. The platforms and modules were not designed to function in this position and, therefore, would either be damaged or provide inaccurate measurements.



Yes **No**

You can adjust the angle of your MS05P controller unit in order to have a better view of the screen. Use the stand located under the front part of the unit and pull it down and forward.



Installing Your Controller and Expansion Units in a Rackmount

Your controller and expansion units were designed to be installed in 19-inch rackmounts only. They will not fit into 23-inch racks.

Each unit has a side fan and a rear cooling fan. Mount or position your units so that air can circulate freely around them. When operating the units, choose a location that provides at least

- 75 mm (3 in.) of rear clearance
- 25 mm (1 in.) of side clearance (left and right)

CAUTION

Failure to provide adequate cooling clearance may result in an excessive internal temperature, thus reducing the reliability of your units.

Note: *There is no need to leave an empty slot between units when you stack them one on top of the other inside a rackmount.*

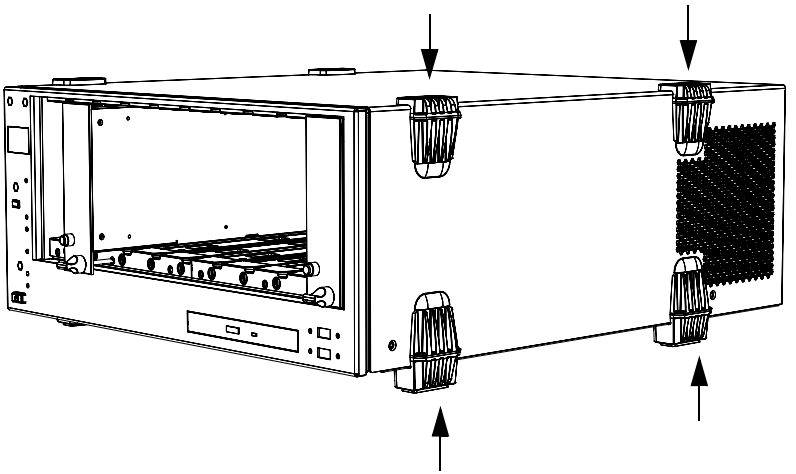
Install your controller and expansion units into a rackmount kit before inserting them into a rack.

To prepare your unit for rackmount use:

1. Ensure that your unit is turned off.
2. Remove the bumpers of your controller unit by unscrewing the top and bottom screws.

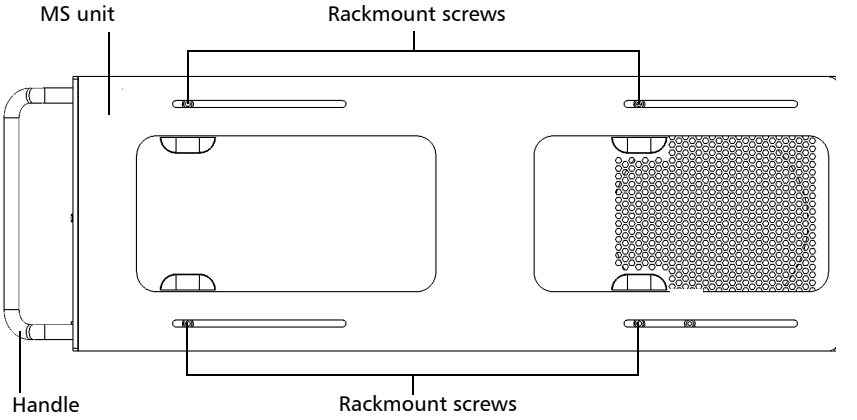
CAUTION

You should remove all your modules before installing your unit into a rackmount. Otherwise, they could be damaged or their calibration could be affected.

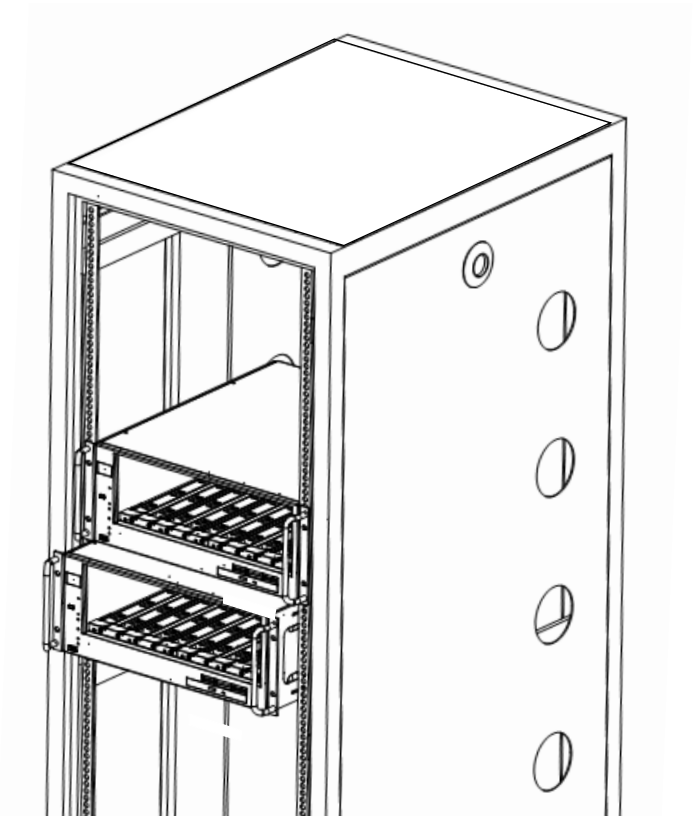


3. Assemble the rackmount kit panels using the supplied screws.

The screws used for the bumpers will not work to secure the rackmount kit panels. You must use the screws supplied with the kit.



4. Insert the rackmounted unit into the cabinet as shown below and secure it in place using the hardware supplied with the rack.



5. Repeat steps 1 through 3 for each unit being installed in the rackmount.

If you want to move a unit forward or backward in the rackmount, untighten the screws on each side of the rackmount kit slightly and slide the unit forward or backward. Retighten the screws afterwards.

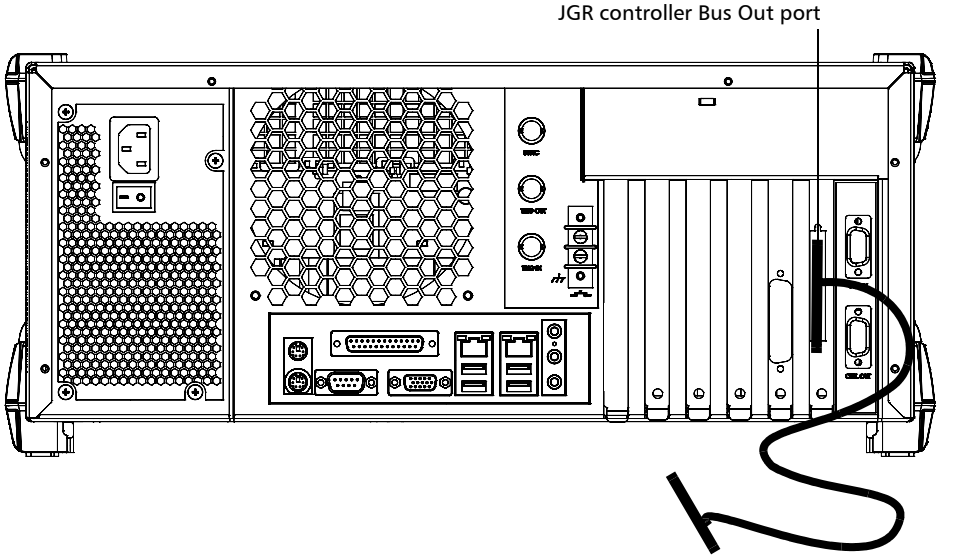
Connecting One or More Expansion Units to a Controller Unit

It is possible to connect expansion units to your controller unit. You can either stack your units on a desk or install them in a rackmount (see *Installing Your Controller and Expansion Units in a Rackmount* on page 18 for instructions).

Note: *As your expansion units are daisy chained, the position is indicated by a number displayed on the unit's position display.*

To connect an Expansion Unit to a Controller Unit:

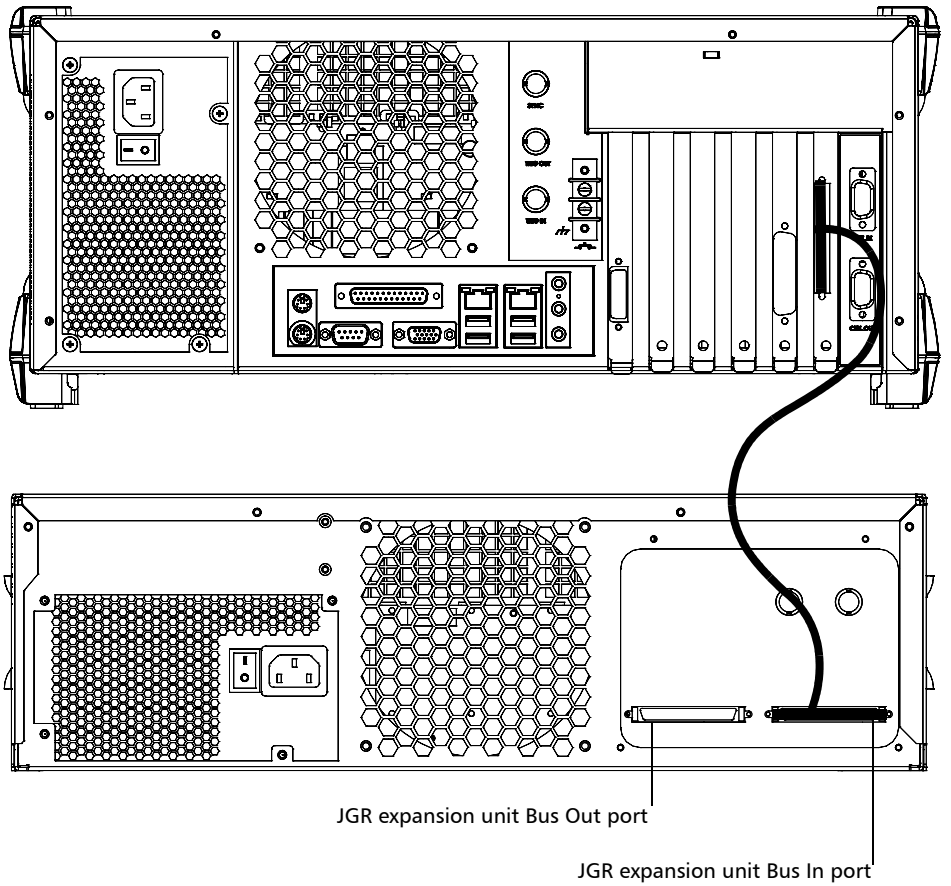
1. Connect the provided cable to the JGR Bus Out port located at the back of your controller unit.



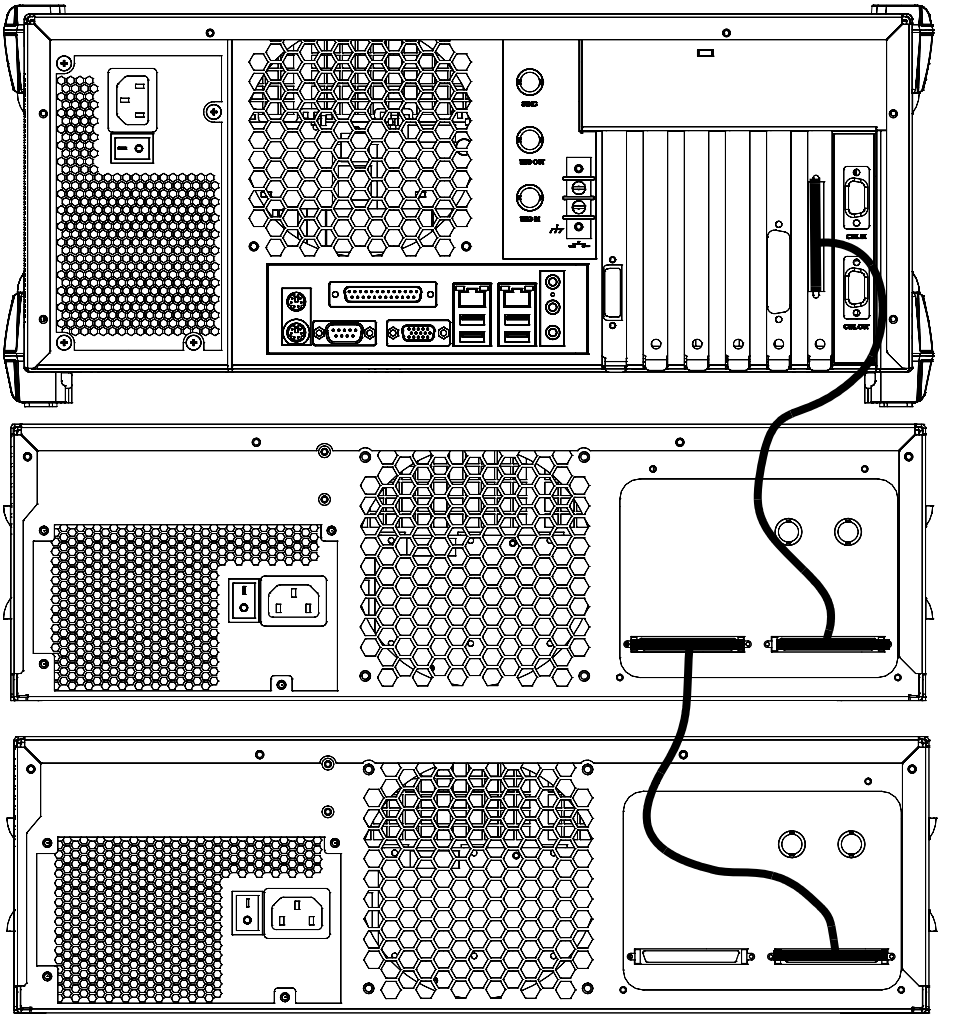
IMPORTANT

To ensure that your cable is properly connected, firmly push the MS interface cable into the port until it clicks twice. An improper connection might cause operation problems.

2. Connect the other end of the cable to the JGR Bus In port located at the back of your expansion unit.



3. To connect additional MS10E expansion units to your system, use another cable to connect the JGR Bus Out port of the first unit to the JGR Bus In port of the second unit.



4. Repeat the previous steps for each additional unit you want to connect.

Inserting and Removing Test Modules

CAUTION

Never insert or remove a module while the controller unit and its expansion units are turned on. This will result in immediate and irreparable damage to both the module and unit.

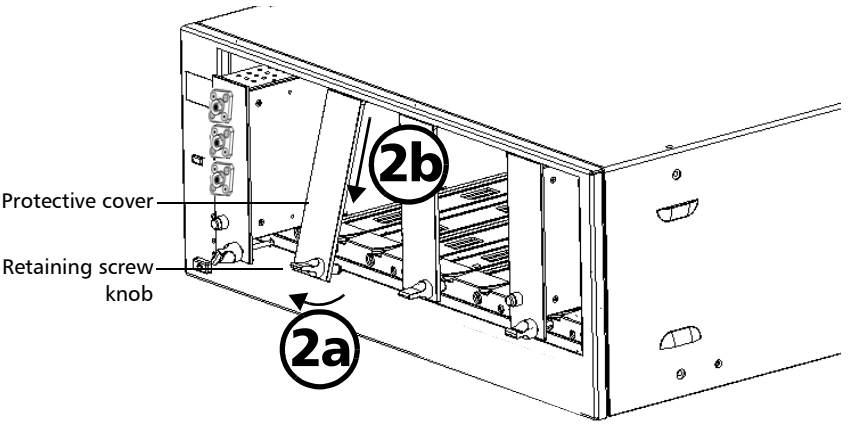
To insert a module into the controller or expansion unit:

1. Exit MS Manager and turn off all your units.

2. Remove the protective cover from the desired unused module slot.

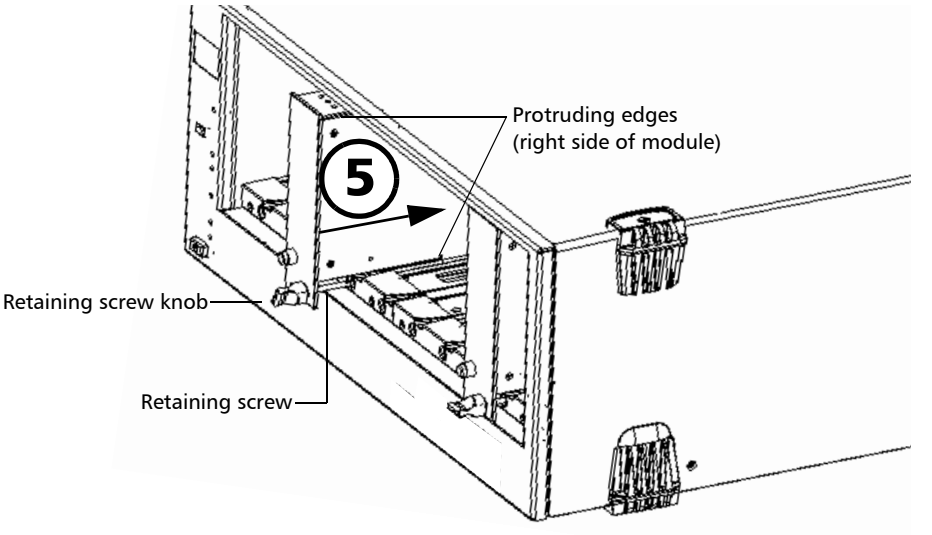
2a. Pull the retaining screw knob firmly towards you and release the bottom of the cover.

2b. Gently pull the top of the protective cover downwards, to remove it from the unit grooves.



3. Position the module so that its front panel is facing you and the top and bottom protruding edges are to your right.

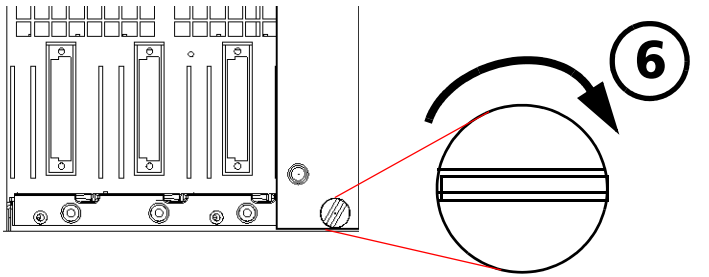
4. Insert the protruding edges of the module into the grooves of the unit's module slot.



5. Push the module all the way to the back of the slot, until the retaining screw makes contact with the unit casing.

6. While applying slight pressure to the module, turn the retaining screw knob (located at the bottom of the panel) clockwise until the knob is horizontal.

This will secure the module into its "seated" position.



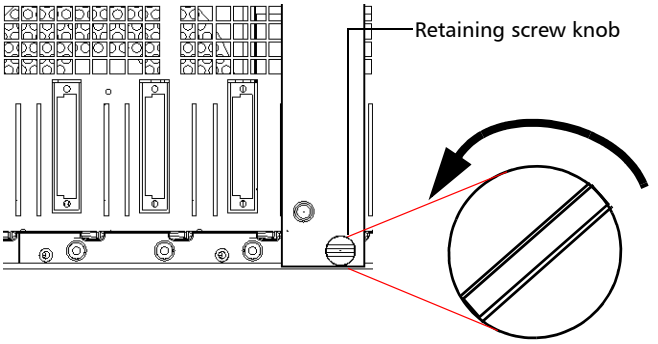
The module is correctly inserted when its front panel is flush with the front panel of the controller or expansion unit.

When you turn on the controller unit, the startup sequence will automatically detect your module.

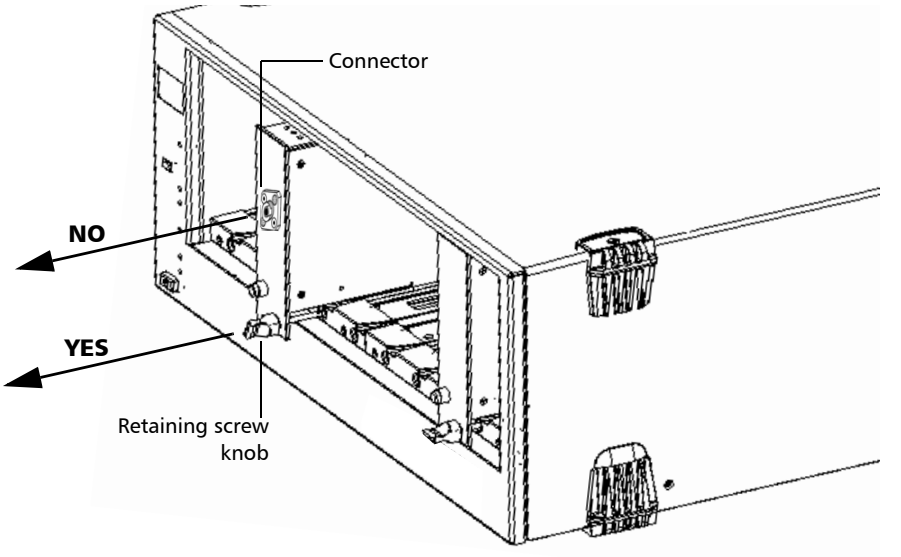
To remove a module from your controller or expansion unit:

1. While pulling gently on the knob, turn it counterclockwise until it stops.

The module will slowly be released from the slot.



2. Place your fingers underneath the module or hold it by the retaining screw knob *(NOT by the connector)* and pull it out.



CAUTION

Pulling out a module by a connector could seriously damage both the module and connector. Always pull out a module by the retaining screw knob.

3. Cover empty slots with the supplied protective covers.

3a. Slide the top of the protective cover into the upper grooves of the unit.

3b. Snap the cover into place by pushing the retaining screw knob.

CAUTION

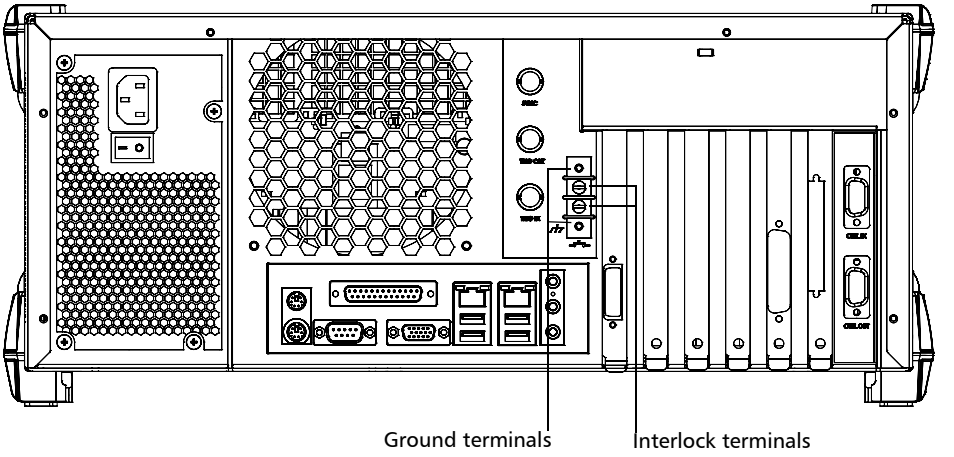
Failure to reinstall protective covers over empty slots will result in ventilation problems.

Use the same precautions regarding module protective covers as you would for your MS modules.

Safeguarding Units with the Interlock

JGR's more powerful and potentially harmful modules, such as laser sources, are equipped with an interlock system that prevents accidental power ups.

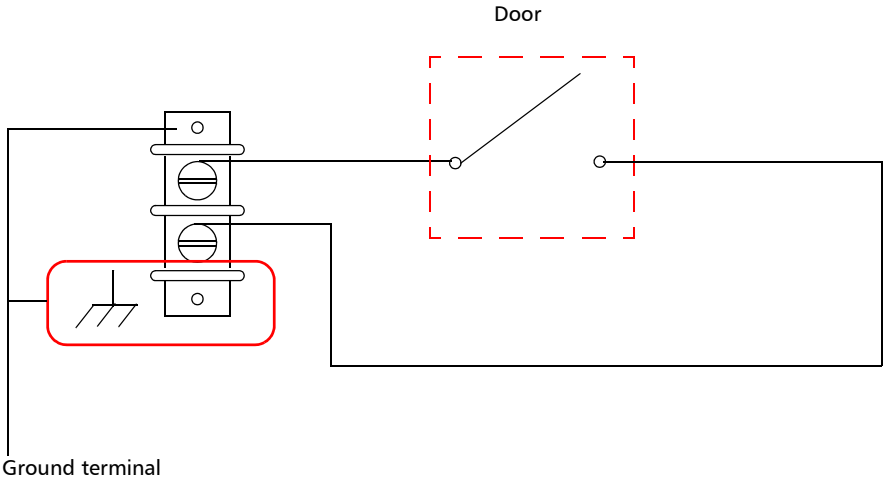
If your module is equipped with an interlock system, you might have to link the terminals, located at the back of your MS10P controller unit, with the provided connectors.



You can connect your interlock terminals to an external safety device using either your own connectors or the connectors provided by JGR.

Note: *If you are using shielded cables to connect your interlock terminals, use either ground terminal to secure your connection.*

If you use modules featuring an interlock, these terminals must communicate, whether it is directly or via the external security device. Otherwise, your modules will not function properly.



The output voltage is 5 V. Do not apply any voltage to the interlock terminals, as only a mechanical contact is needed.

IMPORTANT

Some modules have an interlock located on the front panel. Use the front interlock instead of the one located at the back of your controller unit, as the latter will not offer you any protection.

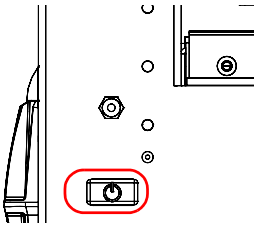
For more information about the type of interlock system to use with the module you are working with, refer to the corresponding user guide.

Turning the Platform On and Off

CAUTION

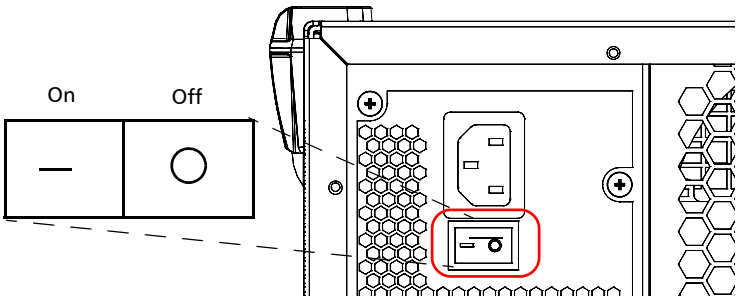
If the controller unit is stored at a temperature outside of the specified operating temperature range, let the controller unit reach operating temperature before turning it on.

To turn your controller or expansion unit on, use the button located on the lower left-hand part at the front of the unit. To ensure that your MS controller unit detects the expansion units, use a proper starting sequence. First turn on the expansion units, and then turn on your MS controller unit. Failure to use proper starting sequence may lead to malfunction.



If you have more than one expansion units linked together, the order in which you turn them on does not matter.

Ensure to turn on the main power switch located at the back of each unit.



Installing or Upgrading MS Manager

If you purchased an MS Platform, the MS Manager software will have been preinstalled and configured at the factory.

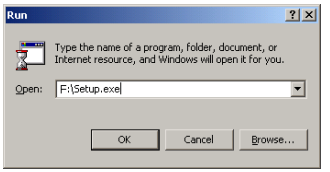
If you ever need to reinstall or upgrade your software, use the CD-ROM that came with your product or upgrade kit and follow the instructions below.

To install or upgrade MS Manager:

1. Insert the MS Manager installation CD in the CD-ROM drive.

2. The installation process should start automatically. If not, start it manually as follows:

2a. On the Windows taskbar, click **Start** and select **Run**.



2b. In the **Open** box, type *F:\setup.exe* (the drive letter may vary according to the location of your CD-ROM drive).

2c. Click **OK** to start the InstallShield Wizard and follow the on-screen instructions.

The installation program automatically reinstalls or updates all module applications so you do not have to do it for each individually.

Accessing MS Manager

When your MS Platform comes out of the factory, it is set to start automatically with Windows XP.

However, if you want to start it automatically with MS Manager, see *Selecting the Startup Interface* on page 52.

To access MS Manager when in the Windows environment:

➤ Click the **MS Manager** icon on your desktop.

OR

➤ On the **Start** menu, select **All Programs** > **JGR** > **MS Manager**.

Exiting MS Manager

To exit MS Manager:

1. Click the **Exit** button, located on the lower right-hand side of the main window.

2. Select the appropriate option among these three choices:

- **Exit MS Manager**: to close MS Manager and return to Windows.
- **Log off Windows**: to close MS Manager and log off your Windows session so another user may log on.
- **Turn off System**: to completely shut down the MS controller unit.

3. Click **OK** to confirm your choice, or **Cancel** to return to MS Manager.

IMPORTANT

To ensure proper operation of your MS Platform, follow this shut down procedure:

- **First, turn off your MS controller, and then**
- **Turn off all expansion units.**

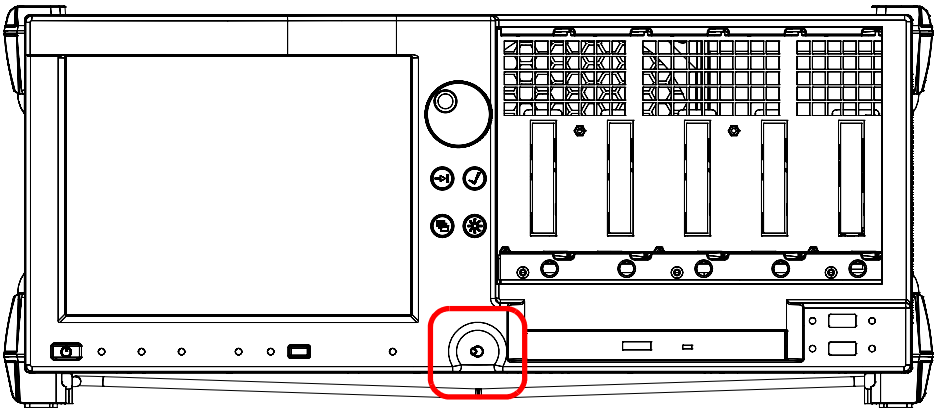
Note: *Expansion units are shut down manually by pressing on the On/Off button located on the front lower left-hand part of each unit.*

4 Setting Up Your MS Platform

Calibrating Your Touchscreen (MS05P)

The touchscreen simplifies and accelerates testing procedures by providing immediate access to commands. It detects the position of the finger or any other blunt pointing device used to activate a command, a function, or a button.

The MS05P controller unit comes equipped with a stylus to use with your touchscreen. The stylus is located in the lower right corner of the screen. Pull it out of its storage location to use it.



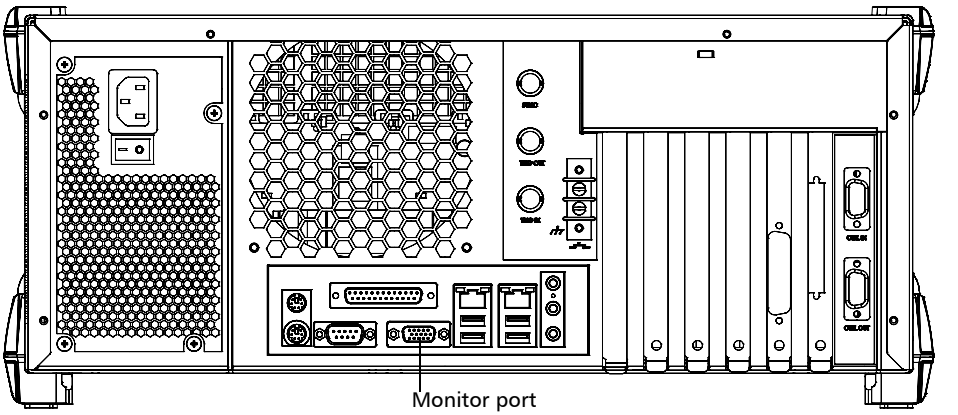
You can customize the touchscreen behavior to improve response to touches. To better see what is on the screen, you can also change the brightness of the screen backlight, as explained in *Changing the Backlight Level (MS05P)* on page 43.

When you receive your MS05P controller unit, calibrate the touchscreen to ensure that it behaves in the way that suits you best.

To calibrate your touchscreen, start **Windows**. On the taskbar, click **Start** and then select **All Programs > UPDD > Calibrate**. Follow the calibration procedure described on the corresponding tab.

Installing an External Monitor

A controller unit can be connected to an external monitor. Connect your monitor to the standard computer connector located at the back of the unit.



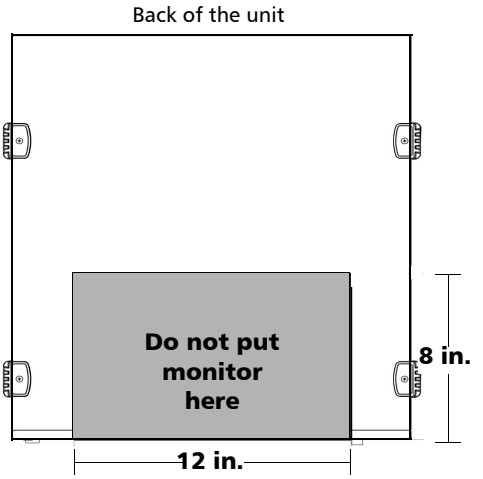
Note: *Use the Windows Control Panel to configure the display settings of your external monitor.*

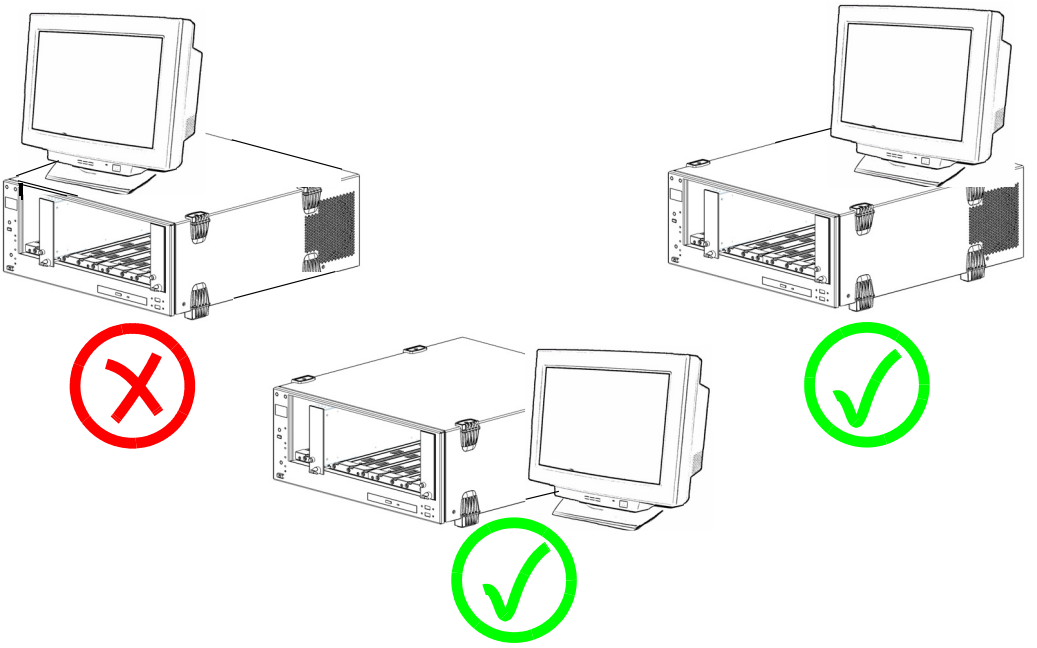
CAUTION

Do not put the monitor directly on the front part of your units. This would press on the front opening, thus preventing you from inserting or removing modules correctly.

Ensure that the monitor:

- **is at least 8 inches away from the front of the unit;**
- **does not rest on the 12-inch wide area across the front opening.**


Back of the unit
Do not put monitor here
8 in.
12 in.
Front of the unit

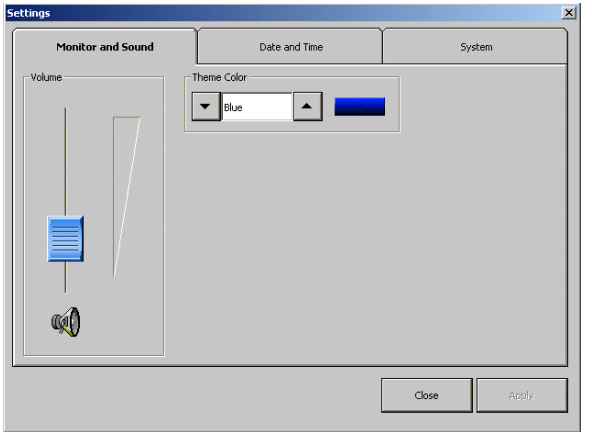


Adjusting the Display and Sound

Different working environments call for different display and sound settings.

To adjust the display and sound:

1. From the main window, select the **Utilities** function tab.
2. Click **Settings**.
3. Select the **Monitor and Sound** tab.



4. To adjust the volume of the unit, use the **Volume** slider.
5. To change the theme color for your unit, use the up and down arrow buttons under **Theme Color**.
6. Click **Apply** to use the new settings.

   OR

   Click **Close** to exit without using the new settings.

Deactivating the LED Display

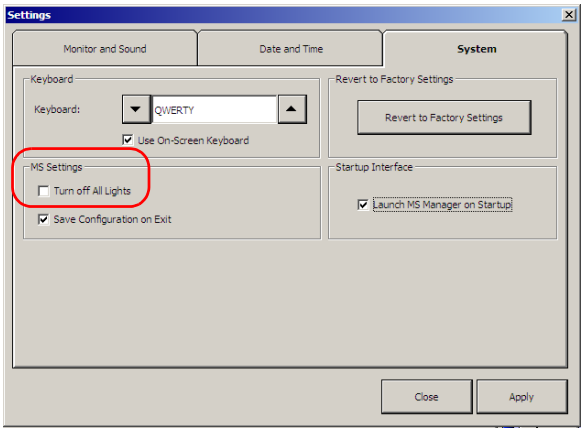
Note: *This feature is not available in offline mode.*

Your testing environment may require total darkness and you do not want any LED or light activity that could compromise your test results.

Note: *The backlight does not automatically turn off. For information on backlight adjustment, see* Changing the Backlight Level (MS05P) *on page 43.*

To deactivate the LED display:

1. Select the **Utilities** function tab.
2. Click **Settings**.
3. On the **System** tab, under MS **Settings**, select **Turn off All Lights**.



4. Click **Apply** to use the new settings.

   OR

   Click **Close** to exit without using the new settings.

Selecting Regional and Language Options

The number formatting of data processed by MS Manager may be affected by Windows regional settings. To ensure proper functioning of your MS Manager software, select the US English regional settings.

To set regional language settings:

1. On the Windows **Start** menu, point to **Settings** and then click **Control Panel**.
2. Double-click the **Regional and Language Options** icon.
3. On the **Regional Options** tab, click the arrow to view the country list, and select **English (United States)**.
4. Click **OK** to confirm your selection.

Changing the Backlight Level (MS05P)

The backlight has four brightness levels: Off, Low, Medium, and High. Pressing on the front of your unit enables you to toggle between these levels.

Installing an External Keyboard

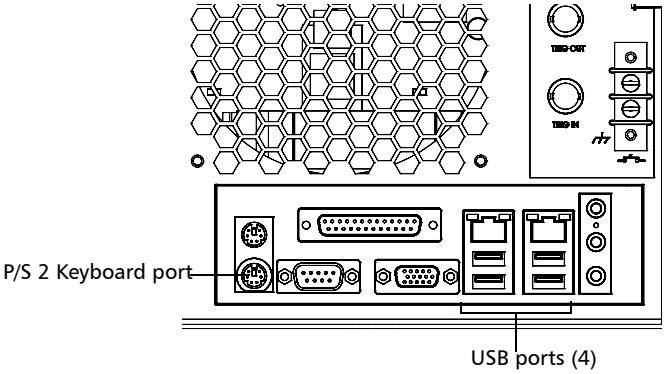
Your MS Platform can be used with either a USB or a PS/2 keyboard.

To connect a keyboard to the unit:

Use any of the USB ports located both at the front and back of the unit.

OR

Use the PS/2 keyboard port located at the back of the unit.



Refer to your keyboard's documentation for instructions on how to install the corresponding drivers, if needed.

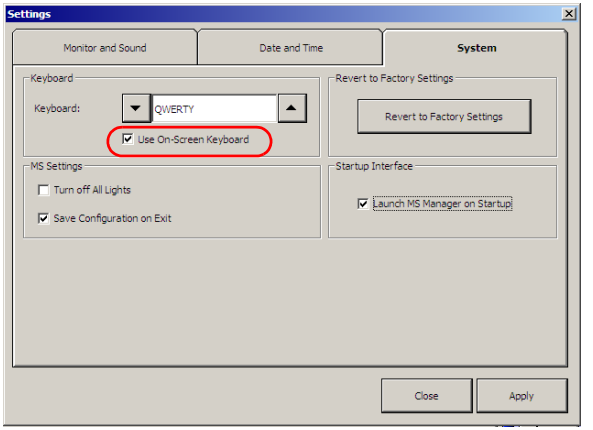
Activating and Selecting the On-Screen Keyboard Type

You might prefer to use the on-screen keyboard rather than an external one.

Note: *The on-screen keyboard is only available with MS Manager and its related applications. It does not function in Windows XP itself.*

To activate the on-screen keyboard:

1. In the main window, select the **Utilities** function tab.
2. Click **Settings**.
3. Select the **System** tab.



4. Under keyboard, select the **Use On-Screen Keyboard** option box.

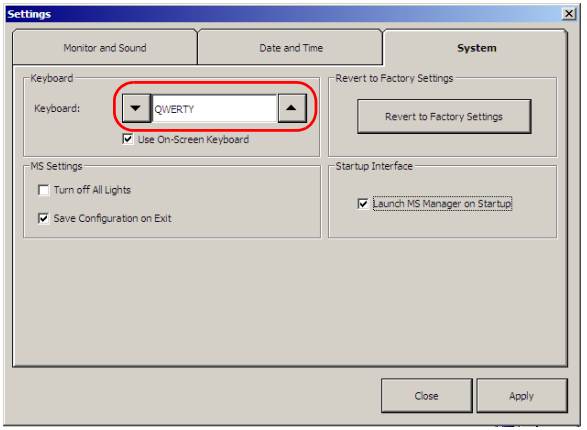
The next time you need to enter data, a keyboard appears and you can type using your mouse.

MS Manager was designed to adapt to various on-screen keyboard types according to your preferences.

Note: *These settings do not affect your external keyboard.*

To select the keyboard type:

1. In the main window, select the **Utilities** function tab.
2. Click **Settings**.
3. Select the **System** tab.



4. Under **Keyboard**, use the up and down arrow buttons to select your keyboard type in the **Keyboard** box.
5. Click **Apply** to use the new settings.

   OR

   Click **Close** to exit without using the new settings.

Installing a Mouse

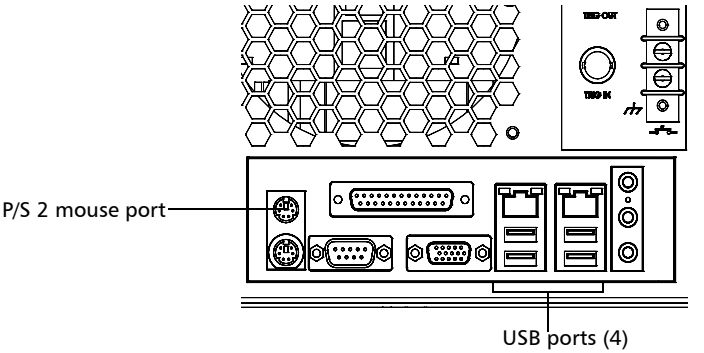
Your MS Platform can be used with either a USB or PS/2 mouse.

To connect a mouse to the unit:

Use any of the USB ports located both at the front and back of the unit.

OR

Use the PS/2 mouse port located at the back of the unit.



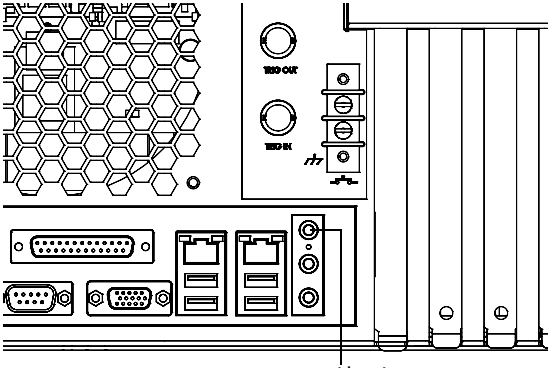
Refer to your mouse's documentation for instructions on how to install the corresponding drivers, if needed.

Installing a Tape, CD, or DVD Player

Your MS Platform can be used with a tape, CD, or DVD player.

To connect a tape, CD, or DVD player to the unit:

Use the Line In port located at the back of the unit.

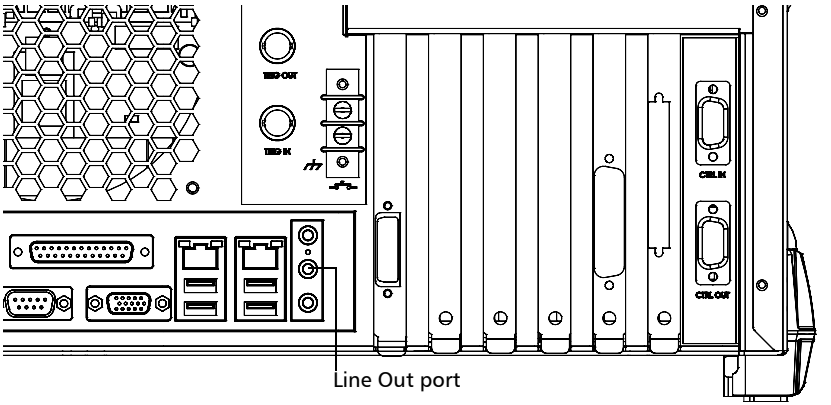


Installing a Headset or a Speaker

Your MS Platform can be used with either a headset or speaker.

To connect a headset or speaker to the unit:

Use the Line Out port located at the back of the unit.

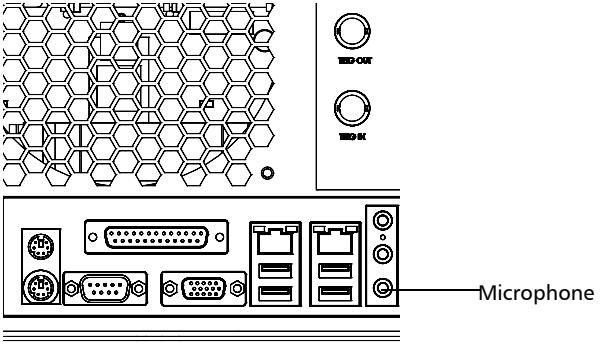


Installing a Microphone

Your MS Platform can be used with a microphone.

To connect a microphone to the unit:

Use the microphone port located at the back of the unit.



Installing a Printer

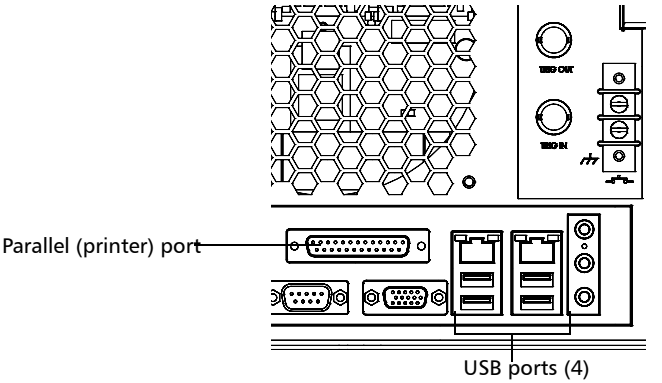
Your MS Platform can be used with a printer.

To connect a printer to the unit:

Use any of the USB ports located both at the front and back of the unit.

OR

Use the parallel port located at the back of the unit.



Refer to your printer's documentation for instructions on how to install the corresponding drivers.

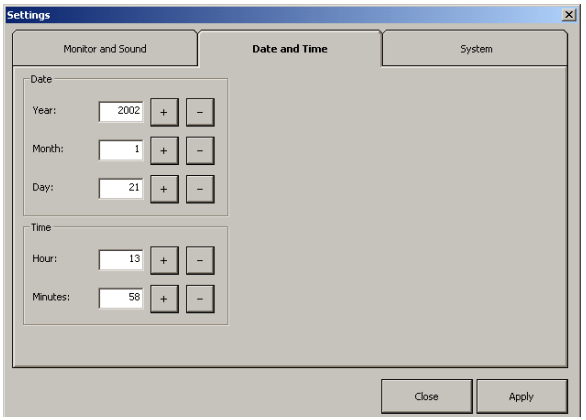
Adjusting the Date and Time

You might need to set the time according to your location or time zone.

Note: *This function is linked to Windows, so changing a date or time in MS Manager changes the date or time in your whole system. JGR recommends using Windows Control Panel to verify the current time zone and change it if necessary.*

To adjust the date and time:

1. In the main window, select the **Utilities** function tab.
2. Click **Settings**.
3. Select the **Date and Time** tab.



4. Under **Date** and **Time**, use the **+** and **-** buttons next to the item you want to change.
5. Click **Apply** to use the new settings.

   OR

   Click **Close** to close the window without using the new settings.

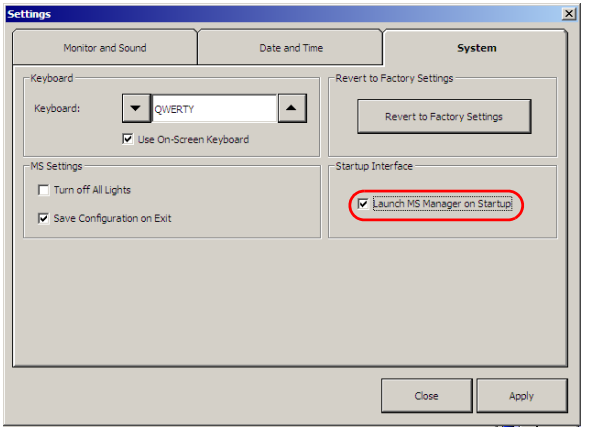
Note: *You must have the appropriate administrative rights to set this option.*

Selecting the Startup Interface

You can choose whether or not MS Manager automatically starts at startup.

To select the startup interface:

1. Select the **Utilities** function tab.
2. Click **Settings**.
3. On the **System** tab, select the **Launch MS Manager on Startup** check box.



4. Click **Apply** to confirm your choice, then **Close** to return to the **Utilities** function tab.

Your choice of interface will be used on your next startup. If the option is not selected, the system will start with Windows XP.

Customizing Access Levels

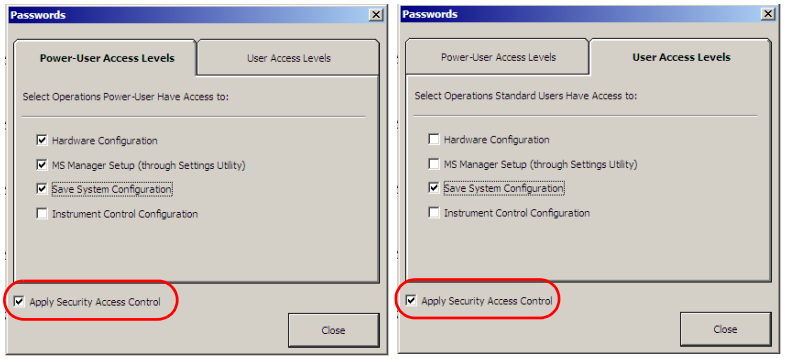
You cannot set passwords in MS Manager itself. It uses the same security levels and accesses as Windows XP. This means that if you are logged in Windows XP as an Administrator, you have access to everything on MS Manager.

Your MS Platform is set at the factory to automatically start MS Manager in Administrator level mode. If you want to set passwords for other user types, refer to the Windows XP online help, under “Passwords”.

Power-User and Standard-User levels have limited access to MS Manager functions, while Administrator level users can give specific access rights to power users and standard users.

To customize access levels:

1. In the main window, select the **Utilities** function tab.
2. Click **Passwords**.
3. Ensure that **Apply Security Access Control** is selected.



4. Click the tab representing the user level for which you want to set accesses.

5. Select the items you want the users to access by checking the corresponding boxes.
   - **Hardware Configuration**: to rename hardware
   - **MS Manager Setup**: to change items in the **Settings** section of the **Utilities** function tab
   - **Save System Configuration**: to save configurations on your MS Platform
   - **SCPI Manager**: to use or modify remote control commands
6. Click **Close** to accept your changes.

The changes will take effect upon the next logon.

Note: *In order for the access levels to take effect, you must select the **Apply Security Access Control** check box in the **Passwords** dialog box. If the box is cleared, MS Manager operates in Administrator level.*

Setting Up Auto Logon for Windows XP

You can set Windows XP to automatically log on when you start your unit.

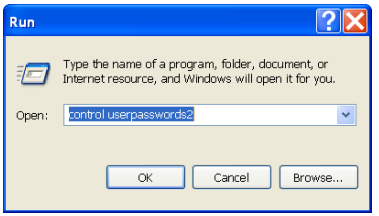
Note: *Keep in mind that you should not enable auto logon if you need privacy for any files on your computer. If you need to maintain privacy, create separate user accounts for every user, and have each user protect their account with a password. For more information, refer to the Windows XP online help, under "User Account".*

To automatically log on to Windows XP upon startup:

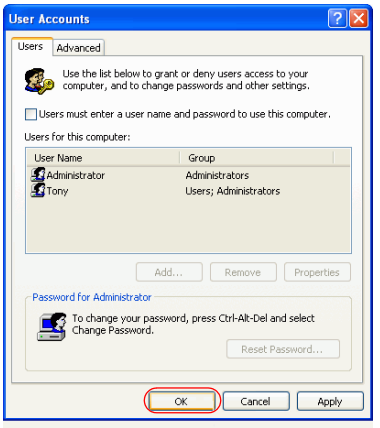
1. Start Windows.
2. On the taskbar, select **Start**, then **Run**.

Note: *Remember your user name at the top of the **start** menu, because you will need it later.*

3. In the **Open** box, type *control userpasswords2*, then select **OK**.

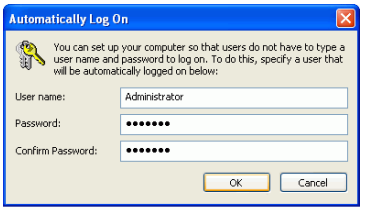


4. Clear the **Users must enter a user name and password to use this computer** option box. Select **OK**.



5. Select the **User name** box, type the user name that was displayed on the **start** menu in step 1. If you normally type a password to log on, type your password in both the **Password** and **Confirm Password** boxes. Otherwise, leave these boxes blank.

6. Select **OK**.



Auto logon now allows anyone to start and use your computer without typing a password.

Setting Communication Parameters

For information on communication settings for remote control, see *Preparing for Automation or Remote Control* on page 87.

Adjusting the Fan Speed

By default, the fan speed of the MS controller unit and expansion units is set to maximum. If you prefer, the application can also adjust the fan speed to a safe minimum. In this case, it determines the most appropriate fan speed according to the type and number of modules.

Note: *Fan speed is the same for the MS controller unit and its expansion units.*

Note: *Depending on your test configuration, it is possible that the minimum safe value corresponds to full speed.*

Note: *The specifications of the MS modules are based on maximum fan speed. Using a lower fan speed can increase the measurement uncertainty. To ensure optimal performance from your unit, use maximum fan speed.*

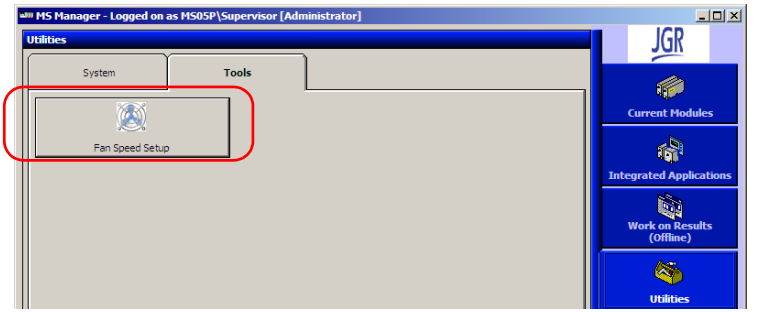
CAUTION

Before inserting or removing modules, always ensure that ALL controller and expansion units are turned off, not only the unit in which (or from which) you want to insert (or remove) modules.

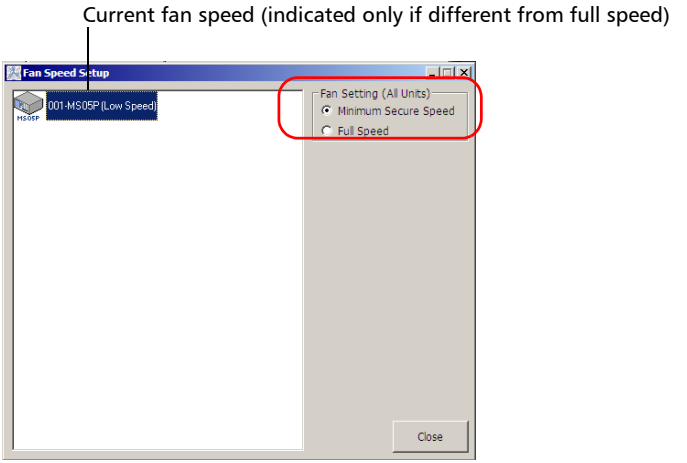
Otherwise, the application is not able to determine the appropriate fan speed and your modules will be irreparably damaged.

To adjust the fan speed:

1. In the main window, select the **Utilities** function tab.
2. Select the **Tools** tab, then **Fan Speed Setup**.



3. Select the desired option.



4. Click **Close** to save and apply your fan speed selection.

Saving and Opening Configurations

It is possible to save as many configurations as you want. This is particularly useful if users need to work with different configurations.

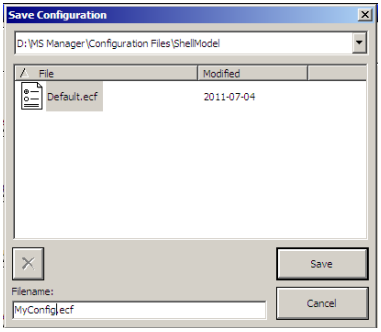
The configuration file includes

- active modules
- settings under the **Settings** button

Once a configuration is saved, you can use it on any other MS10P controller unit providing you are using the same type of modules in both.

To save a configuration:

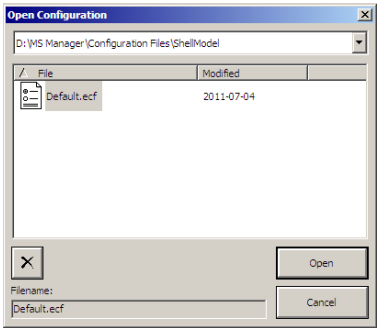
1. Select the **Utilities** function tab, and then select the **System** tab.
2. Click **Save Configuration**.



3. In the list at the top of the **Save Configuration** dialog box, select a folder in which to save your configurations.
4. In the **Filename** box, type the configuration name and click **Save**.

To load a configuration:

1. Select the **Utilities** function tab, and then select the **System** tab
2. Click **Load Configuration**.



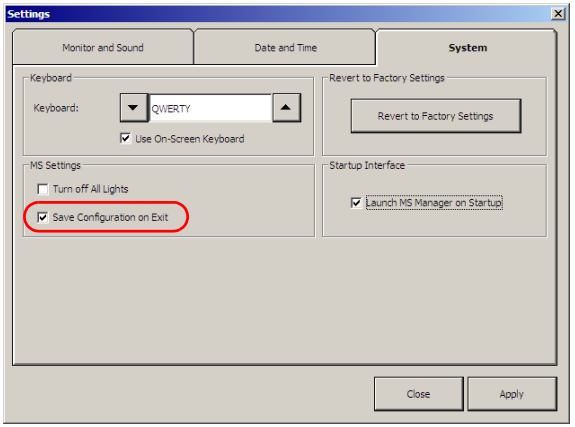
3. In the list at the top of the **Open Configuration** dialog box, select the configuration to use and click **Open**.

Saving the Configuration on Exiting

MS Manager offers you the possibility of saving your configuration as you exit. This allows you to continue where you left off when starting a new session.

To save your configuration when you exit MS Manager:

1. Select the **Utilities** function tab.
2. Click **Settings**.
3. On the **System** tab, select the **Save Configuration on Exit** check box.



4. Click **Apply** to confirm your choice, then **Close** to return to the **Utilities** function tab.

Each time you exit MS Manager using the **Exit** button, the current configuration is saved and recalled upon a new logon.

Note: *This function does not work if you log off Windows using the **Start** button while MS Manager is still running. You must first close MS Manager to save the configuration.*

Changing Unit and Module Identification

Note: *This feature is not available in Offline mode.*

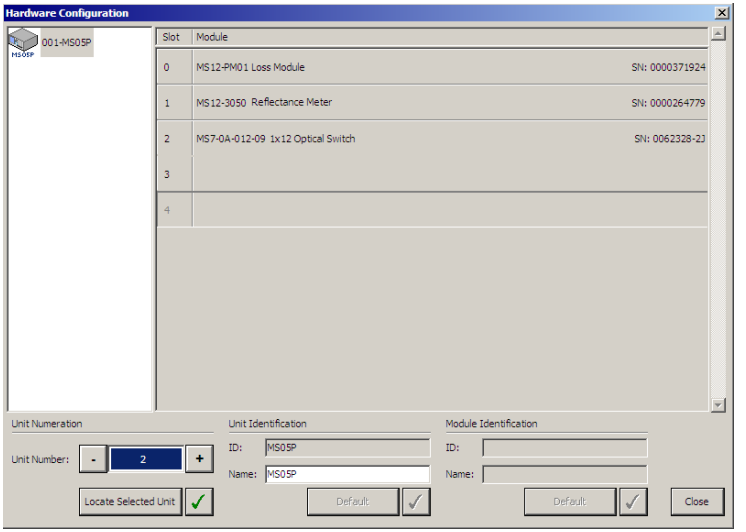
You can change the identification of your units and modules for easier hardware management.

To change the identification for your unit or module:

1. Select the **Utilities** function tab.

2. Click **Hardware Configuration**.

MS Manager prompts you to close all applications before changing the hardware configuration. Click **Yes** to proceed.



3. Click the unit or module you want to configure.

It turns white to indicate that it is selected.

4. Change the unit or module name and number as desired, using the corresponding fields. Click next to the field to enter your new settings.
5. To revert to the default identification of the unit or module, click the corresponding **Default** button.
6. Click **Close** to return to MS Manager.

Reverting to Factory Settings

You might need to revert to the factory settings, such as keyboard configuration, theme color or startup interface.

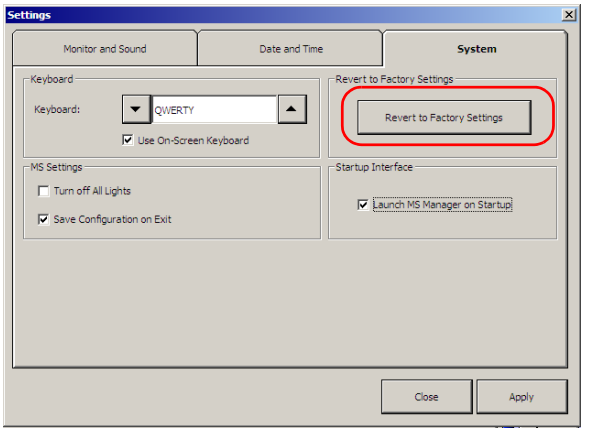
IMPORTANT

This action cannot be undone.

Note: *You can only use the* Revert to Factory *function if you are logged on as an Administrator level user. The button is unavailable for the other user levels.*

To revert to the factory settings:

1. Select the **Utilities** function tab.
2. Click **Settings**.



3. On the **System** tab, click the **Revert to Factory Settings** button.

Note: *These changes only affect parameters in the* ***Settings*** *dialog box (****Monitor and Sound****,* ***Date and Time****, and* ***System*** *tabs).*

Accessing Your MS Platform Remotely

You can access your unit remotely from a computer using either the Remote Desktop Connection or TightVNC Client applications.

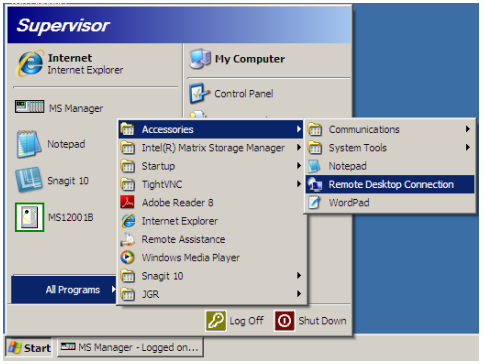
This could be particularly useful if you do not intend to perform automation tasks on your platform. If you prefer to perform automation tasks on your platform and modules, see *Using MS Products in an Automated Test Environment* on page 143.

The table below presents the differences between Remote Desktop Connection and TightVNC Client applications.

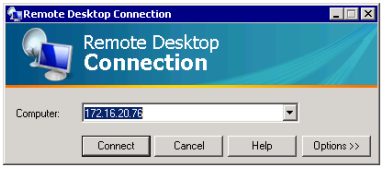
Characteristic	Remote Desktop Connection	TightVNC Client
Type of connection	Direct between the unit and the computer; only one user can be connected to the unit at a time.	Not exclusive; several users can be connected to the unit at the same time (sharing the same session).
Windows user rights	Taken into account.	Not taken into account.
Password-protected	Yes; mandatory. The user name and password are the same as those used to connect to the unit. By default, all the accounts with administrator and supervisor rights can use Remote Desktop Connection. If you want the operators to be able to use Remote Desktop as well, you must specifically grant them access.	Yes; mandatory. The password is defined on the TightVNC Server, the first time you start the server. This password is not related to the one used to connect to the unit. By default, all people that use TightVNC Client will enter the same password (as defined on the server). Each user to whom you provide the password will be able to connect to the unit via TightVNC.

To access your MS Platform remotely:

1. Start Windows.
2. On the taskbar, press **Start**, then select **All Programs**.
3. Select **Accessories**, then **Remote Desktop Connection**.

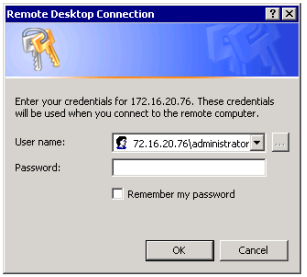


4. In the **Connecting to Remote Desktop** window, in the **Computer** list, type the TCP/IP address of the computer you want to access.



5. Click **Connect**.

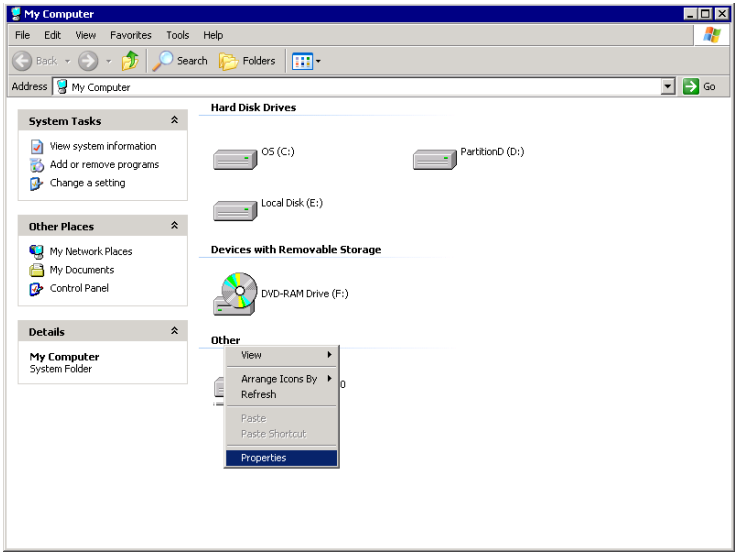
6. In **Connecting to Remote Desktop**, in the **User Name** menu, type your user name.



7. In the **User Name** menu, type your user name.
8. In the **Password** box, type your password, and then click **OK**.

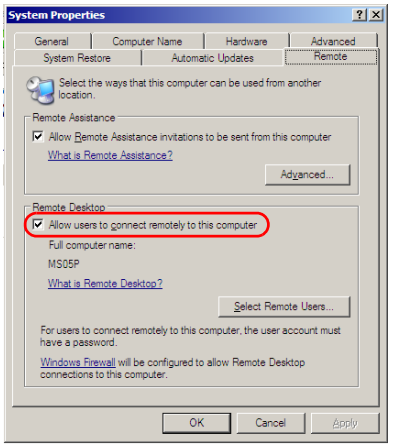
To allow an operator to access the Remote Desktop Connection:

1. On the taskbar, click **Start**, and then select **My Computer**.
2. Right-click **Other** or the appropriate disk drive, and then select **Properties**.

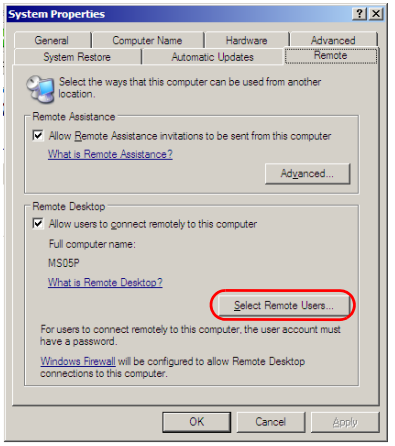


3. In the **System Properties** dialog box, click the **Remote** tab.

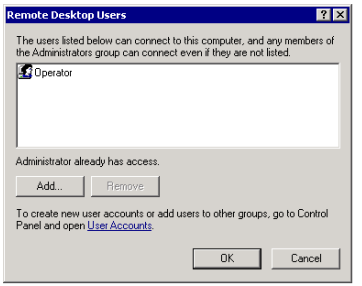
4. Under **Remote Desktop**, select **Allow users to connect remotely to this computer**.



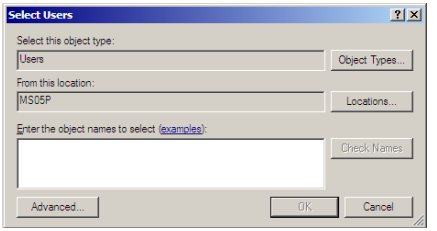
5. Click **Select Remote Users**.



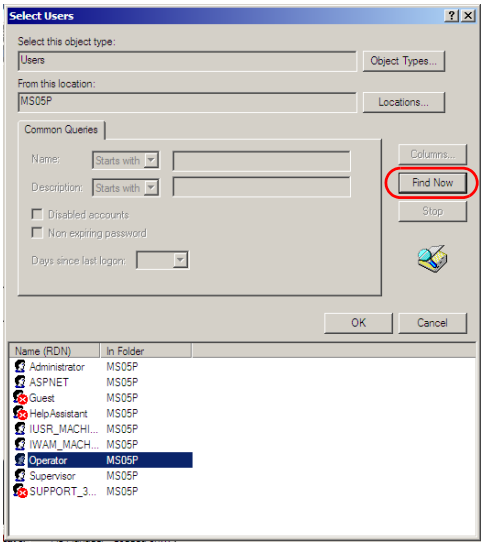
6. In the **Remote Desktop Users** dialog box, click **Add**.



7. In the **Select Users** dialog box, click **Advanced**.

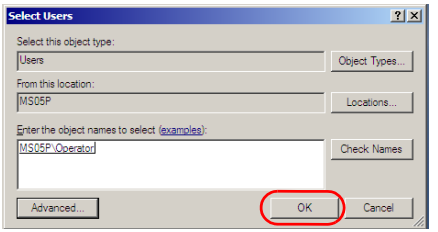


8. Click **Find Now** to let the system find and display the list of users.



9. Select the user named *Operator*, and then click **OK**.

10. In the list of users, select the user you have just added, and then click **OK**.



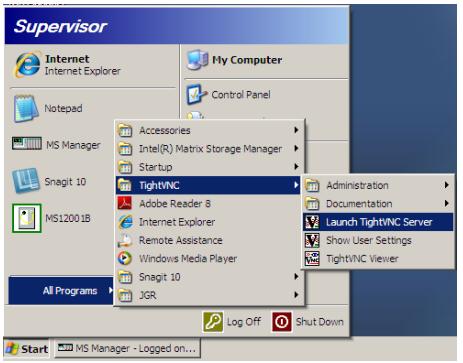
11. In the **Remote Desktop Users** dialog box, click **OK**.

Configuring the TightVNC Server

The TightVNC Server corresponds to the server that will be accessed and is the only component already installed on the system.

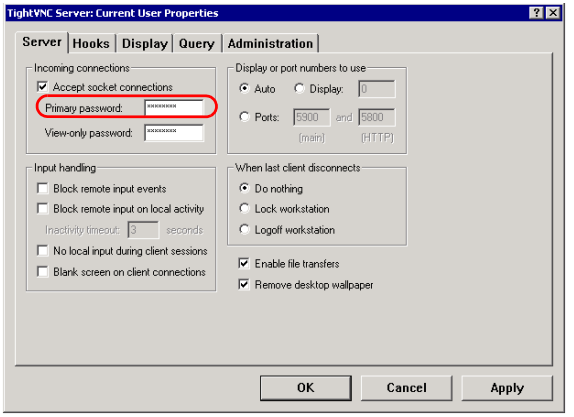
To configure the TightVNC Server:

1. Start Windows.
2. On the taskbar, click **Start**, and then select **All Programs**.
3. Select **TightVNC**, and then **Launch TightVNC Server**.



4. In the **Windows Security Alert** dialog box, click **Unblock**.

5. Under the **Incoming Connections**, select the **Server** tab.
6. In the **Primary password** box, type the appropriate primary password.



7. In the **View-only password** box, type your password.

Note: *The two passwords are independent of each other. They do not have to be identical.*

8. Click **Apply**, and then **OK**.

IMPORTANT

Once the VNC server is installed and configured, you should use the same network adaptor and not change it if you reconnect it. The VNC server must use the same network adaptor on the platform to be able to operate properly.

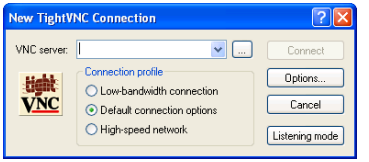
Installing the TightVNC Viewer on a Client Computer

The TightVNC Viewer is used to control another computer or the MS Platform. Since the TightVNC Viewer is not already installed on your unit you can download it by using the software provided on the Internet.

Note: *JGR recommends to make a shortcut on your desktop once your TightVNC Viewer is downloaded.*

To start the TightVNC Viewer on a client computer:

1. Click the TightVNC Viewer shortcut on your desktop.
2. In the **VNC Server** list, type the IP address of the MS Platform to which you want to connect.



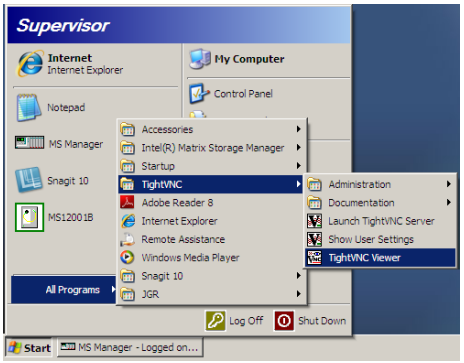
3. Click **Connect**.

Connecting to the Unit with TightVNC Viewer

Once the TightVNC Viewer is downloaded on your computer or MS Platform, you can access the other unit remotely.

To connect to the unit from the MS Platform:

1. Start Windows.
2. On the taskbar, click **Start**, and then select **All Programs**.
3. In the **TightVNC** submenu, select **TightVNC Viewer**.



4. In the **VNC Server** list, type the IP address of the MS or of the working station to which you want to connect.

5. Click **Connect**.

Using a Web Server as a Viewer

You can access to the TightVNC Server via a Web browser such as Microsoft Internet Explorer.

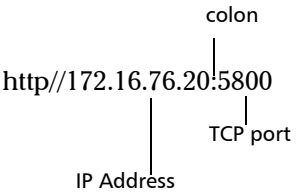
The TightVNC server contains a small Web server. When you connect to it by using your Internet search engine, the Java applet for the viewer is downloaded automatically. It allows you to access to the remote desktop.

IMPORTANT

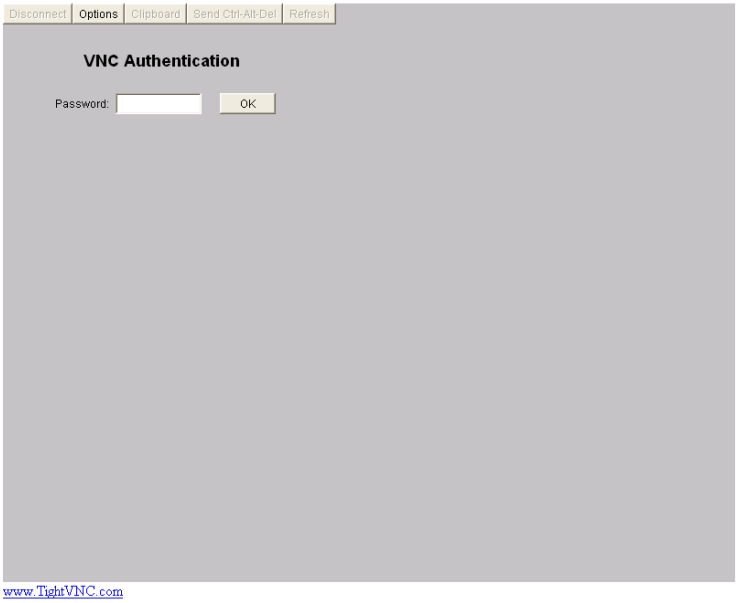
Your Web browser must support Java applets. In addition, you should not use a proxy in order to let the java applets access directly the remote server.

To connect remotely by using a Web browser:

1. From the computer Web browser, enter an appropriate IP address, following the example below:



2. In the **VNC Authentification** dialog box, in the **Password** box, type your password.



3. Click **OK** to access the MS Platform.

5 Using the MS Manager Interface

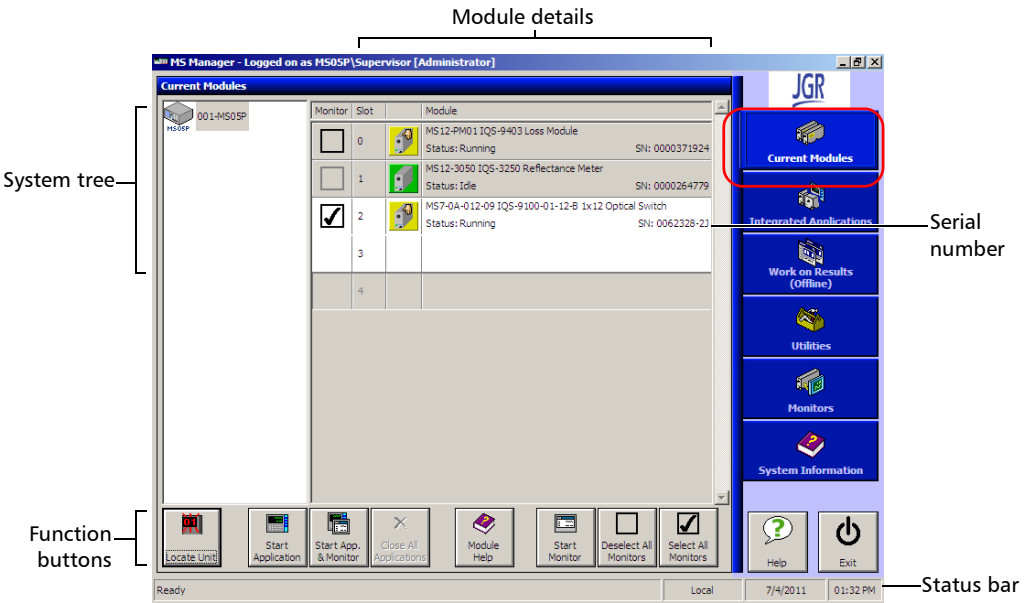
This section helps you to work with the MS Manager interface.

Accessing Current Modules

Note: *This feature is not available in Offline mode.*

If you did not select the **Save Configuration on Exit** option on the **Systems** tab of the **Settings** dialog box, the **Current Modules** function tab appears each time you start a new MS Manager session.

The **Current Modules** function tab is divided into four sections.



The system tree section allows you to see your units, as they are connected. The MS10P controller unit appears on top, with any connected MS10E expansion unit underneath.

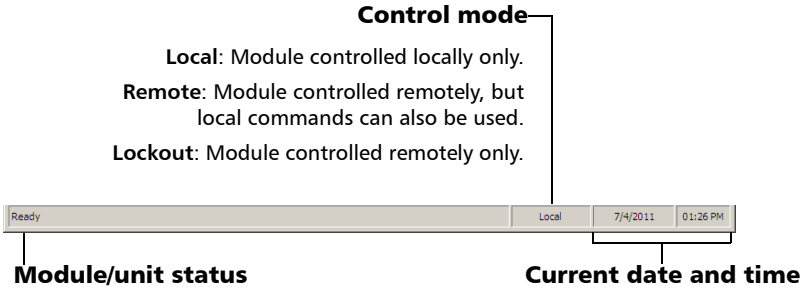
The name, current status and serial number of each module installed on the unit you have selected in the system tree appear besides the module's icon.

- A green icon means that the module is ready for you to use.
- A yellow icon means that your module is in use.
- If the icon bears a question mark, your controller or expansion unit does not recognize the module you have installed.

The function buttons section allows you to perform tasks on the modules currently in your system, tasks such as starting a single-module application.

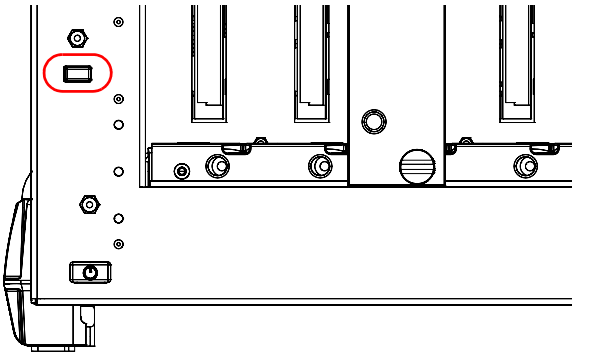
Status Bar

The status bar, located at the bottom of the main window, identifies the operational status of the MS Platform.



For more information about automating or remotely controlling the MS Platform, see the corresponding sections in this user guide.

If you are currently in **Remote** mode and wish to revert to **Local** mode, press on the LOCAL button on the front of your controller unit.



This button is ineffective if you are in **Lockout** mode.

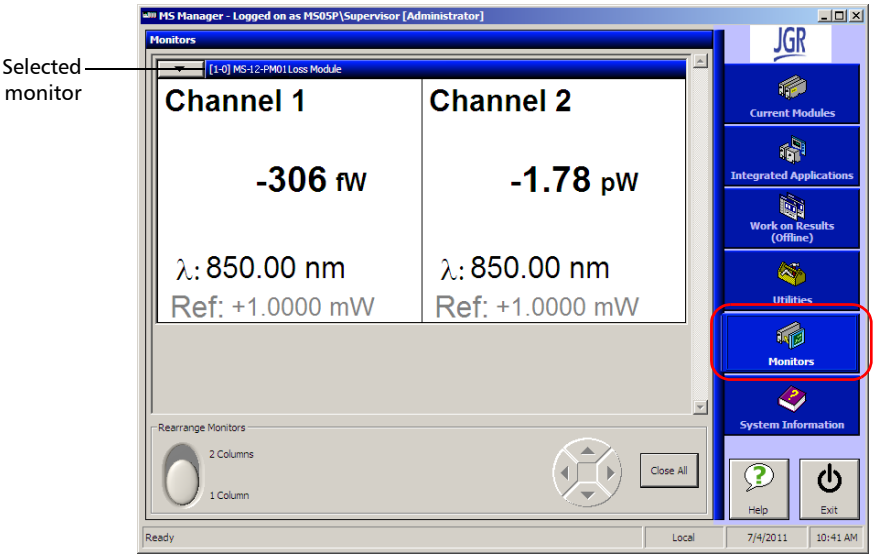
For detailed instructions on how to start the single-module application related to your specific MS module, refer to the corresponding user guide.

Creating an Integrated Data Display

Note: *This feature is not available in Offline mode.*

When using one or more MS modules in a test setup, you can view module data and status using monitor windows in MS Manager. To do so, select the **Monitors** function tab.

Monitor windows display basic data about modules. Using a combination of resizable monitor windows allows you to create an integrated data display.



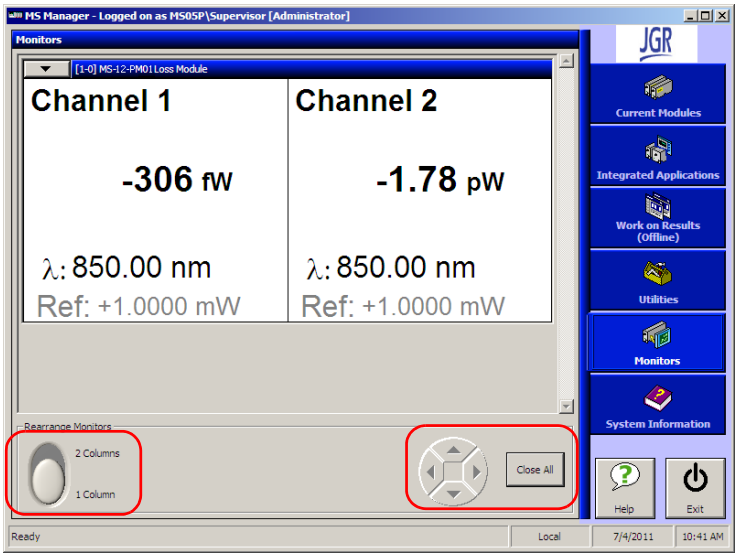
The **Monitors** function tab displays up to four complete monitor windows at a time, even if you are monitoring more than four applications.

You can customize the display so that the windows you need to see the most often are within easy view.

To move monitor windows:

1. On the monitor window you want to move, click the title bar once to select it. The window title bar changes color to indicate that it is selected.
2. Move the monitor window using the arrow buttons. The window either moves to the empty space in the chosen direction, or swaps with the window currently there.

You can also display only one column of monitor windows instead of two by using the button located on the lower left-hand corner of the window.



If you want to close all monitor windows, use the **Close All** button located on the lower right-hand part of the **Monitors** function tab.

Note: *Closing all monitor windows using the **Close All** button* does not *close the corresponding module applications.*

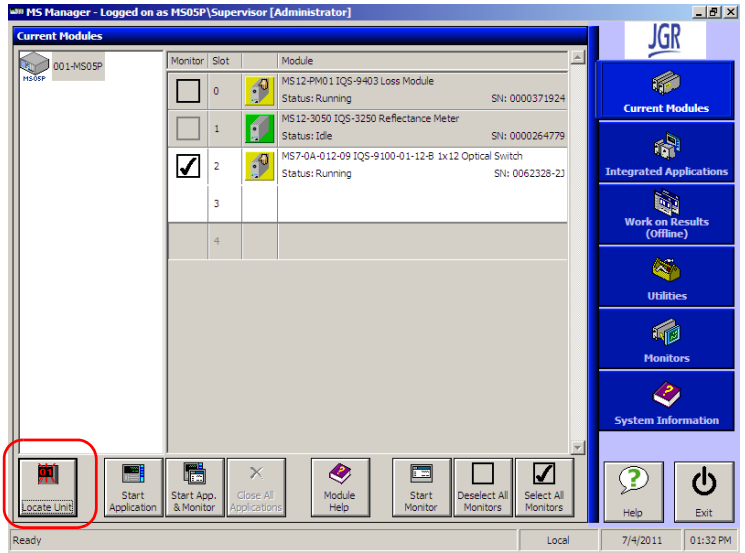
For instructions on how to use the monitor window features of your MS module, refer to the module's user guide.

Locating Units

Note: *This feature is not available in Offline mode.*

Since you can connect many MS10E expansion units to your MS10P controller unit, it may become difficult to find a particular module.

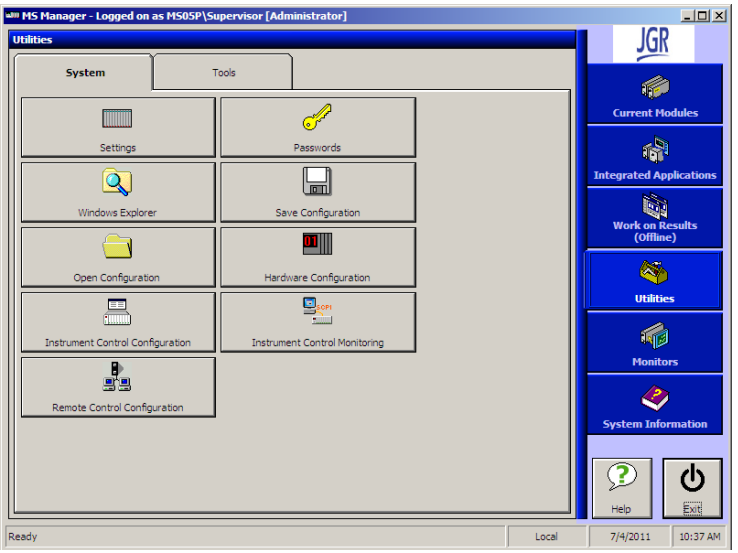
To help you locate units, click **Locate Unit** in the **Current Modules** function tab. The corresponding module position display blinks for ten seconds, allowing you to locate your module.



The first two digits indicate its position in the chain, while the third digit indicates the unit's number. You can personalize the unit numbers as explained in *Changing Unit and Module Identification* on page 62.

Accessing Utilities from MS Manager

Various utilities are located on the **Utilities** function tab.



To access Windows utilities other than Windows Explorer, do so directly from Windows.

6 Preparing for Automation or Remote Control

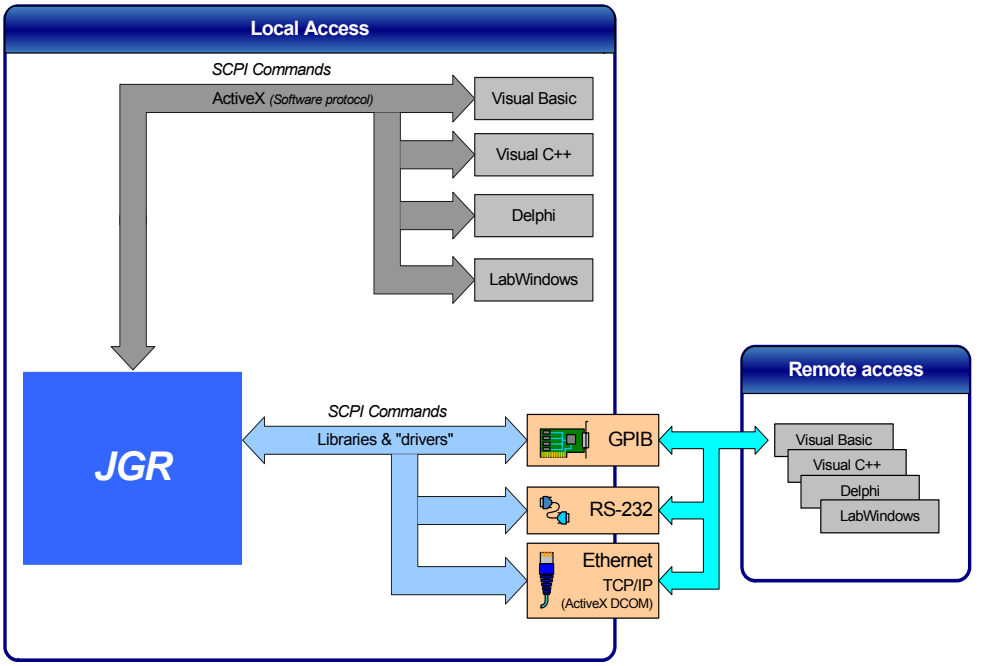
The MS Platform was designed to meet the requirements of automation and to facilitate its integration with your test environment.

An MS Platform always requires a controller unit (MS05P or MS10P) to drive the instruments.

Since MS controller units are built around an Intel processor and are based on Windows architecture, such a controller can:

- act as the system controller of an automated test setup (possibly including third-party instruments)
- control another MS controller over an Ethernet network
- act as a PXI controller (by integrating special third-party PCI cards)

JGR supplies commands that follow the guidelines determined by the SCPI consortium. JGR also supplies COM properties and events allowing you to build your own application.The instruments can be controlled either locally or remotely via the following technologies:



The choice of a technology will depend on your particular needs.

Control	Technology	Characteristics
Local	ActiveX (COM)	➤ Allows you to develop an application that will run locally on the MS controller unit within Windows XP ➤ Best approach when speed is your top priority (no physical connection that slows down the process) ➤ Supported by most development software ➤ Lower cost
Remote	ActiveX (DCOM) (Ethernet, TCP/IP)	➤ MS controller unit can be directly connected to a Local Area Network (LAN) or Wide Area Network (WAN) via its 10/100/1000 Base-T interface ➤ No need for a GPIB card ➤ Allows the sharing of network resources ➤ Allows you to develop computer-based applications to directly communicate with the MS controller unit
Remote	GPIB	➤ MS controller unit must be equipped with the GPIB option ➤ MS controller unit can be used as a device in a GPIB chain (it meets the IEEE 488.2 standard) ➤ A single GPIB address allows you to control up to 100 logical optical instruments ➤ Most commonly used interface to simultaneously control many instruments
Remote	RS-232	➤ Null-modem cable required to establish connection between the computer and the MS controller unit ➤ For increased speed and performance, run the application locally on the MS controller unit through ActiveX instead of using RS-232

Control	Technology	Characteristics
Remote	Telnet (Ethernet, TCP/IP)	➤ MS controller unit can be directly connected to a Local Area Network (LAN) or Wide Area Network (WAN) via its 10/100/1000 Base-T interface ➤ No need for a GPIB card ➤ Allows the sharing of network resources ➤ Allows you to develop computer-based applications very easily to directly communicate with the MS controller unit

For more information on programming aspects, see the section on using your product in an automated test environment.

Configuring Your Platform as a GPIB Controller

At time of purchase, your MS controller unit is configured to be used as a GPIB device. However, it is possible to configure it as a GPIB controller.

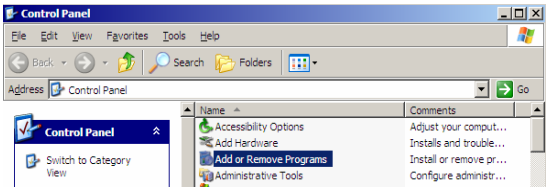
IMPORTANT

Do not remove the GPIB card from the MS controller unit.

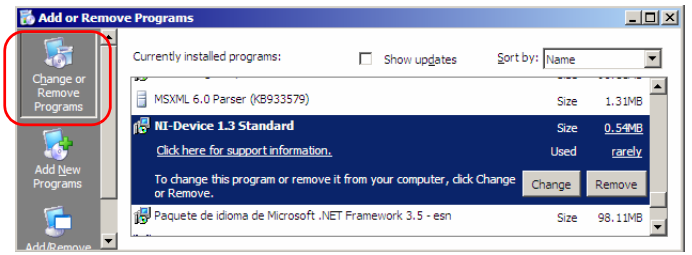
To configure your MS controller unit as a GPIB controller:

1. Remove the **NI-Device 1.3 Standard** software as follows:

1a. From the **Control Panel**, double-click the **Add or Remove Programs** icon.



1b. Click the **Change or Remove Programs** function tab.



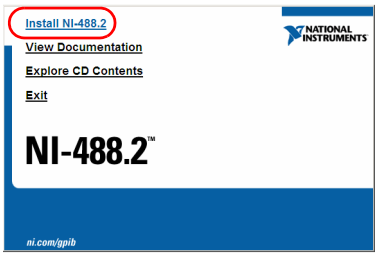
1c. From the list of currently installed programs, select **NI-Device 1.3 Standard** and click **Remove**.

1d. When Windows prompts you, click **Yes** to confirm the removal of the **NI-Device 1.3 Standard**.

2. Install the **NI-488.2** software as follows:

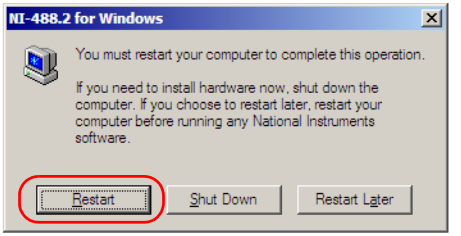
2a. Insert the **National Instruments** installation CD in the CD-ROM drive.

2b. Click **Install NI-488.2** to start the installation.



2c. Click **Next** until you reach the end, and then click **Finish**.

3. Click **Restart** to complete the installation of the **NI-488.2** software on your MS controller unit.



If the operating system does not automatically find the driver, you can locate it by typing this path in your browser: *C:\Program_Files\National_Intruments\NI-488.2\Driver 32*.

The MS controller unit can now be used as a GPIB controller.

Configuring Your Platform as a GPIB Device

You can configure your MS controller unit as a simple GPIB device if it has been previously configured as a GPIB controller. However, you must use a GPIB card made by National Instruments in your controller unit.

IMPORTANT

If you have not previously selected the GPIB option and now wish to purchase a GPIB card, contact JGR. Purchasing your new GPIB card from JGR will ensure compatibility with your controller unit.

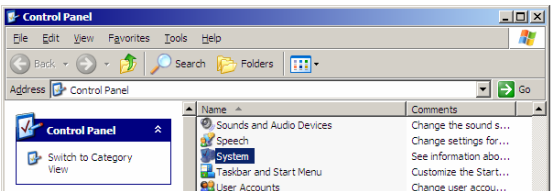
IMPORTANT

Do not remove the GPIB card from the MS controller unit.

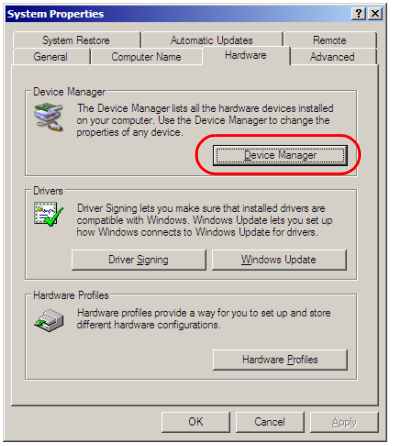
To configure your MS controller unit to a GPIB device:

1. Uninstall the GPIB card as follows:

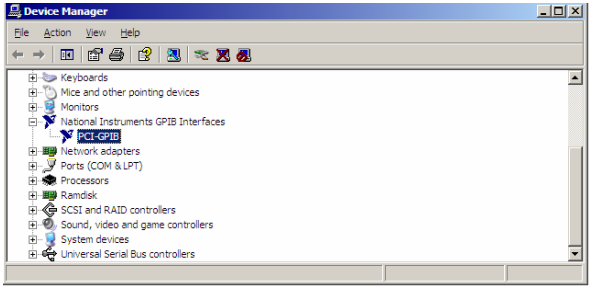
1a. On the Windows taskbar, click **Start** and select **Settings** > **Control Panel** > **System**.



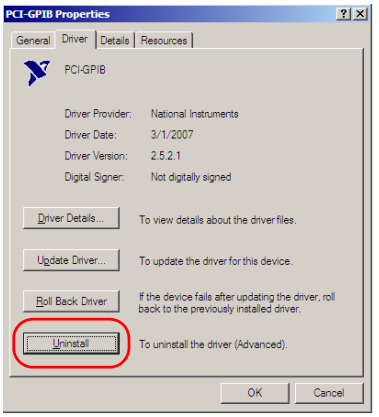
1b. From the **System Properties** dialog box, click the **Hardware** tab, and then click **Device Manager**.



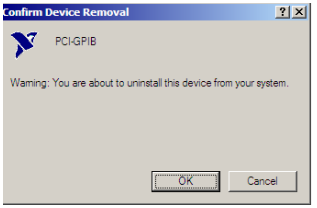
1c. From **Device Manager**, scroll down to **National Instruments GPIB Interfaces**, and then select **PCI-GPIB**.



1d. From **PCI-GPIB Properties**, select **Uninstall**.



1e. Click **OK** to confirm that you want to uninstall the device.

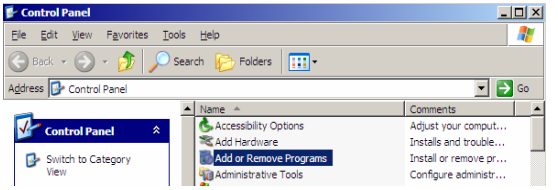


All corresponding hardware is removed automatically.

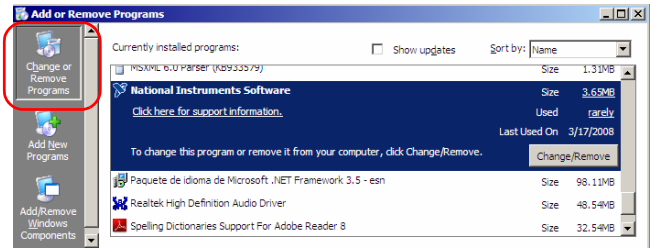
1f. Click **OK** to close the dialog box.

2. Remove the **NI-488.2** software as follows:

2a. From the **Control Panel**, double-click the **Add or Remove Programs** icon.

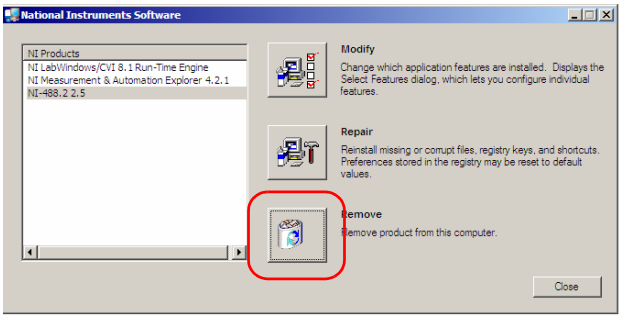


2b. Click the **Change or Remove Programs** function tab.



2c. From the list of currently installed programs, select **National Instruments Software** and click **Change/Remove**.

2d. From the **National Instruments Software** dialog box, select **NI-488.2**, and then click **Remove**.



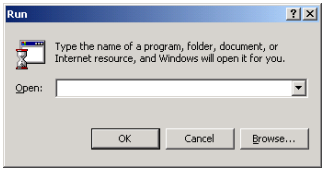
2e. When the application prompts you, click **Yes** to confirm that you want to remove the software.

2f. When you are prompted, click **Yes** to restart your computer. This will complete the software removal operation.

3. Install the NI-Device software as follows:

3a. Insert the JGR installation CD (*not the National Instruments CD*) in the CD-ROM drive.

3b. On the Windows taskbar, click **Start,** and then select **Run**.



3c. Click the **Browse** button to find the *GPIB Device* folder on the CD-ROM, and then select *setup.exe*.

3d. Click **OK** to start the installation.

3e. Click **Next** until you reach the end, and then click **Finish**.

4. Restart the MS controller unit.

The MS controller unit can now be used as a GPIB device.

Preparing Hardware for GPIB Control

If you intend to use GPIB to remotely control your instruments, your controller must have been equipped with the optional GPIB card.

To ensure the optimum efficiency of your system, JGR recommends that you follow these restrictions:

For the IEEE 488.1 protocol:

- Maximum of 15 devices physically connected to each GPIB bus.
- Maximum separation of 4 m between two devices and an average separation of 2 m over the entire range (bus).
- Maximum total cable length used in the system is 20 m.
- At least two-thirds of the connected devices must be turned on.

For the HS488 protocol:

- Maximum of 15 devices physically connected to each GPIB bus.
- Maximum total cable length used in the system is 15 m (that is, 1 m per device physically connected to the GPIB bus).
- All connected devices must be turned on.
- For each meter of cable used in the system, there must be a device connected to it or an equivalent device load.

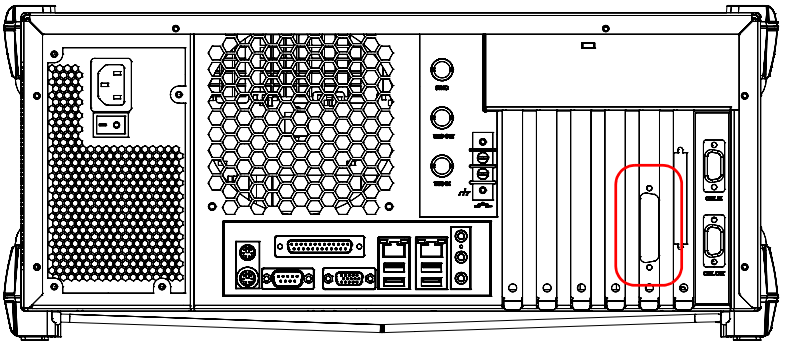
The MS Platform configuration complies with the IEEE 488.1 (also known as IEC60625.1) and the IEEE 488.2 (also known as IEC60625.2) standards to the extent shown in the following table.

Mnemonic	Function
SHE1[1]	Complete source handshake extended capability
AHE1[2]	Complete acceptor handshake extended capability
T6	Basic talker with serial poll; unaddressed to talk if addressed to listen
L4	Basic listener; unaddressed to listen if addressed to talk
SR1	Complete service request capability
RL2	No local lockout capability
PP0	No parallel poll capability
DC1	Complete Device Clear capability
DT0	No Device Trigger capability
C0	No controller capability
E2	Three-state driver capability

1. SHE1 corresponds to the extended capability of SH1, defined in the IEEE 488.1 standard. This extended capability is made possible by the device capability to use the HS488 high-speed protocol.
2. AHE1 corresponds to the extended capability of AH1, defined in the IEEE 488.1 standard. This extended capability is made possible by the device capability to use the HS488 high-speed protocol.

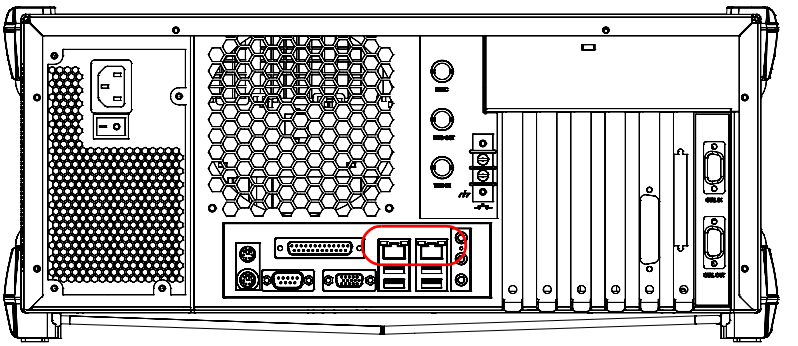
Linking Units with the GPIB Port

If your MS Platform is equipped with a GPIB card, use the provided cable to link it to the other unit with which you want to perform remote control. For more information about your GPIB card, refer to the documentation provided with it.



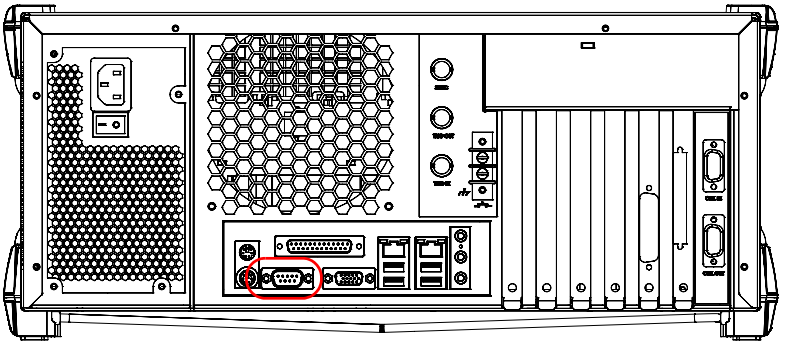
Linking Units with the Ethernet Port

Your MS Platform is equipped with an Ethernet portto send and receive data. Refer to the Windows documentation for information about Ethernet port settings and possibilities.



Linking Units with the Serial Port

Your MS Platform is equipped with a serial (RS-232) port to send and receive data. Refer to the Microsoft Windows documentation for information about serial port settings and possibilities.



Getting Optimum Performance from Your Platform

Several factors influence the data transfer rate of an MS controller unit. The information presented hereafter will help you get the best transfer rate possible.

➤ *Protocol (GPIB control only)*: You can use the IEEE 488.1 (standard) or the HS488 (high-speed) protocol. Both protocols can co-exist in the same system; communication between devices can then be achieved using the IEEE 488.1 protocol.

HS488 is a non-interlocked handshake protocol that allows data transmission between devices at higher rates (six to seven times faster for small systems) than those possible with the IEEE 488.1 protocol. Higher transfer rates are particularly interesting in systems containing devices that return long data blocks. However, the transfer rate is affected by the physical limitations of the cables used in the system.

For information on cabling, see *Preparing Hardware for GPIB Control* on page 98.

➤ *Bus timing (GPIB control only)*: Required if you intend to work with IEEE 488.1, for it affects the handshake rate of this protocol. It indicates the minimum amount of time a GPIB controller waits, once data is placed on the bus, before setting the data valid bus line (DAV) to true.

The possible bus timings are:

Mode	Bus timing delay
Normal	1100 ns
High speed	500 ns
Very high speed	350 ns

If your setup supports it, select the very-high-speed timing to get the best performance.

➤ *Output unit (GPIB, RS-232, ActiveX and TCP/IP):* The MS controller unit can return results in

- ➤ linear units (for example, watts)
- ➤ log units (for example, dBm)

Since internal units are linear, you will get optimal performance by using linear units for output (no need for an internal conversion to log).

Note: *You must make the choice of output unit for each instrument offering such a feature. Refer to the user guide of each optical instrument for a list of available commands and queries.*

➤ *Output format (GPIB, RS-232, ActiveX and TCP/IP)*: The MS controller unit provides the following output formats for measurement results:

- ➤ ASCii
- ➤ PACKed

Generally, the PACKed format allows to pass three to four times more information than the ASCii format for the same transfer rate. Often, the PACKed format is also more efficient since it reduces the MS controller unit CPU work load (no need for an internal conversion to ASCII format).

Note: *The PACKed format will only be applied to <DEFINITE LENGTH ARBITRARY BLOCK RESPONSE DATA> and <INDEFINITE LENGTH ARBITRARY BLOCK RESPONSE DATA>.*

Note: *The choice of data format cannot be made directly via the* MS *Manager software.*

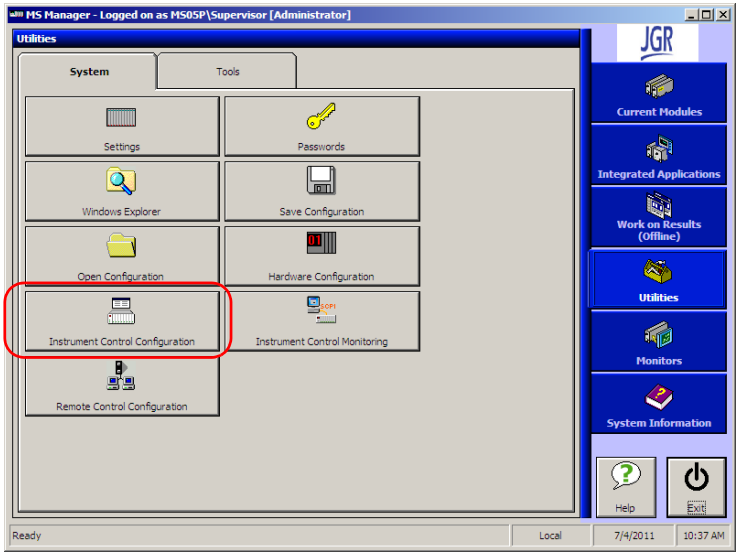
For more information on how to set the output format and data types, see *:FORMat[:DATA]*(IEEE 488.2 and specific commands appendix), *Read* and *ReadBinary* (COM properties and events appendix), and the data types appendix.

Changing Communication Settings

Communication settings cannot be modified without turning on your MS Platform and starting MS Manager.

To change communication settings:

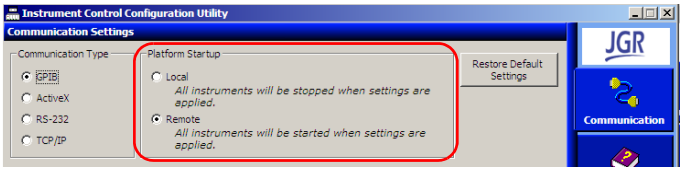
1. From the **Utilities** function tab, select **Instrument Control Configuration**.



2. Under **Platform Startup**, determine whether your module applications will be started locally or remotely by clicking the corresponding option.

   ➤ If **Local** mode is selected, you will not be able to send remote commands to your MS controller or expansion units.

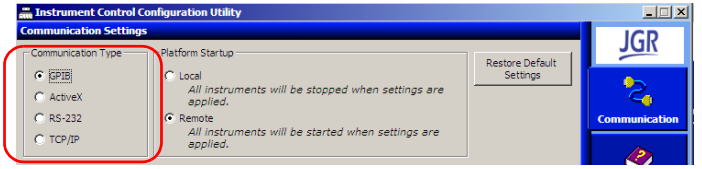
   ➤ If **Remote** mode is selected, all modules in your MS controller and expansion units will be initialized upon startup so you are ready to send remote commands.



If you selected **Local** mode, you can go directly to step 5.

3. Under **Communication Type**, select **GPIB**, **ActiveX**, **RS-232**, or **TCP/IP**.

   For more information on the choice of a particular type, see the table on page 88.



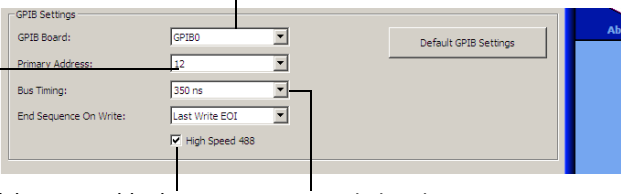
Note: *If the selected communication type does not match the protocol that will actually be used, an error message will be displayed when attempting to control the instruments.*

4. According to the communication type selected at step 3, customize the corresponding parameters as shown below.

➤ For GPIB

MS controller and expansion units can have up to four GPIB cards. Default value is GPIB0.

The selected address must not already be used by the GPIB bus controller or any other device connected to the GPIB bus. Default value is 12.



Select the check box to enable the high-speed mode.

Bus timing (in ns)

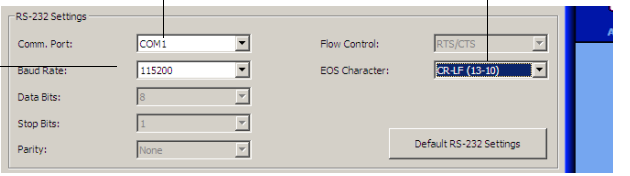
For information on bus timing or on high-speed mode, see the section pertaining to obtaining an optimum performance from your unit.

➤ For RS-232

End-of-string (EOS) character—used to indicate the end of a data string (when sending or receiving data).

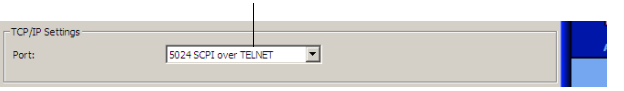
Serial port—used to connect the RS-232 cable.

Determines the speed at which data is sent between the MS Platform and a computer, in bits per second (b/s).



➤ For TCP/IP

TCP/IP port 5024—used to connect over Telnet.

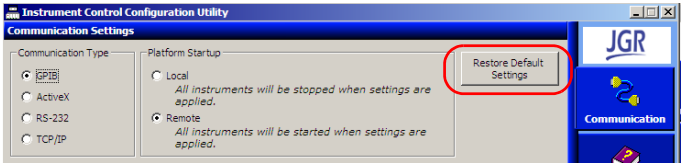


For information on communicating with TCP/IP over Telnet, see the section pertaining to communication through TCP/IP over Telnet.

5. Click **Apply** to confirm your changes.
6. To close the **Communication Settings** function tab, click **Exit**.

To revert to default general settings:

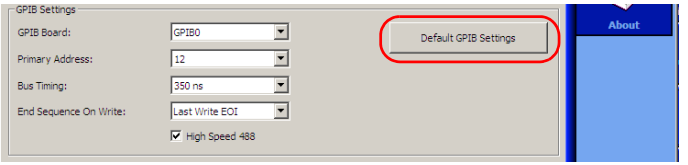
1. Click the **Restore Default Settings** button.



2. Click **Apply** to confirm your changes.

To revert to default GPIB settings:

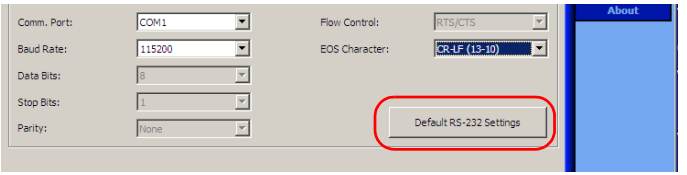
1. Click the **Default GPIB Settings** button.



2. Click **Apply** to confirm your changes.

To revert to default RS-232 settings:

1. Click the **Default RS-232 Settings** button.



2. Click **Apply** to confirm your changes.

Preparing for Remote Control with DCOM

DCOM technology allows to control devices and optical instruments via Ethernet. The JGR IcSCPIAccess Class component provided with your MS Platform acts as a communication link between a client application and JGR's Instrument Control.

DCOM ensures communication between the client application and Instrument Control via your local network. Since each network has its own configuration, you need to be familiar with network security, users, groups, domain management, etc. Basic programming skills are also required to work with DCOM. For more information, you can refer to the Microsoft MSDN Help feature, which provides exhaustive technical documentation on all DCOM issues.

The example presented in the following pages illustrates how to make the JGR IcSCPIAccess Class component available to all users of a local network under Windows XP. The example provided below is for guidance only; it may not work properly with all networks and interfaces may slightly differ depending on the operating system used.

To enable DCOM access to your MS Platform, you must:

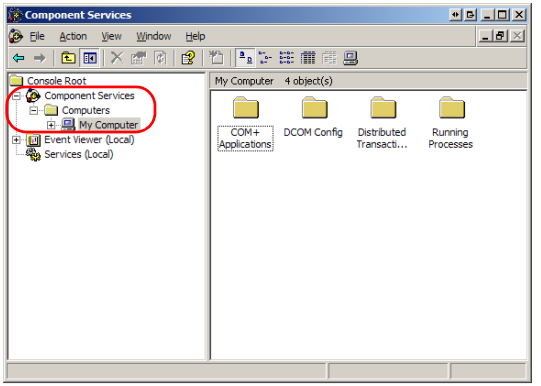
- set the general security parameters
- customize the specific security parameters
- register callback events.

Setting the General Security Parameters

Note: *To run DCOMCNFG.EXE, you need administrator access rights.*

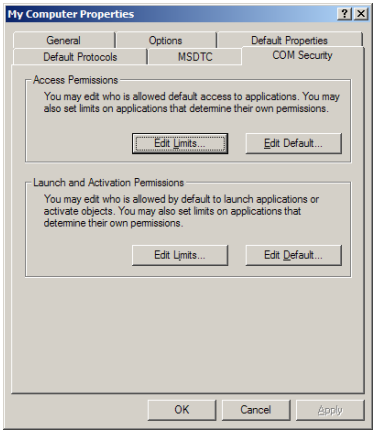
To set the general security parameters:

1. Start Windows.
2. On the taskbar, click **Start** and select **Run**.
3. In the **Open** box, type *DCOMCNFG.EXE* and click **OK**.
4. In the **Component Services** dialog box, go to **Console Root** > **Component Services** > **Computers**.



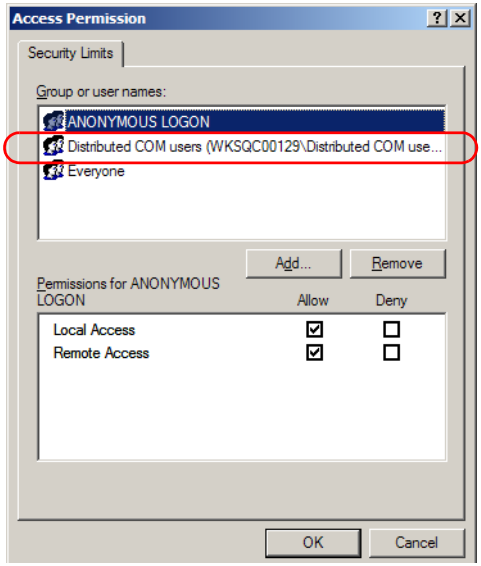
5. In the **Computers** dialog box, right-click **My Computer**, and then select **Properties**.

6. In the **My Computer Properties** dialog box, click the **COM Security** tab.



7. Under **Access Permissions**, click **Edit Limits**.

8. In the **Access Permission** dialog box, ensure that the **Distributed COM Users** group appears in the **Group or user names** list.



You can now allow users to access general DCOM services on your MS Platform. You can either:

➤ Add a user to the **Distributed COM Users** group (refer to Microsoft help).

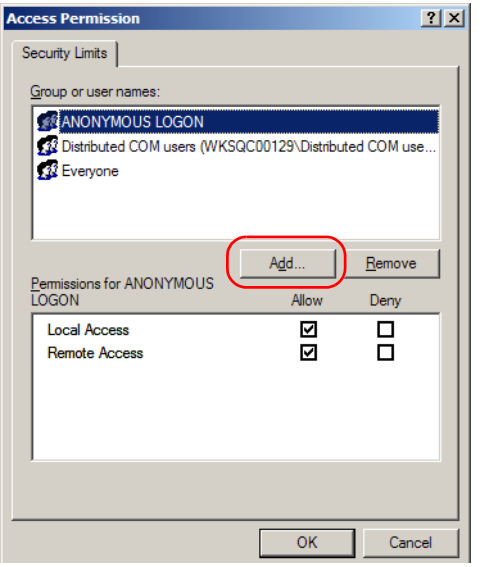
OR

➤ Add a user explicitly under both, **Access Permission** and **Launch Permission** tabs (see procedure below).

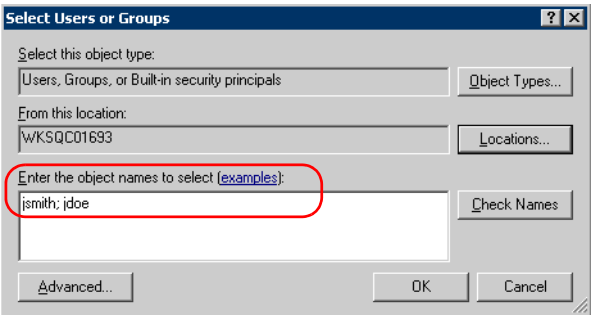
Note: *If you add a user explicitly, ensure to give **Remote Access** rights to the new user.*

To add a user explicitly:

1. In the **My Computer Properties** dialog box, click the **COM Security** tab.
2. Under **Access Permissions**, click **Edit Limits**.
3. In the **Access Permission** dialog box, click **Add**.



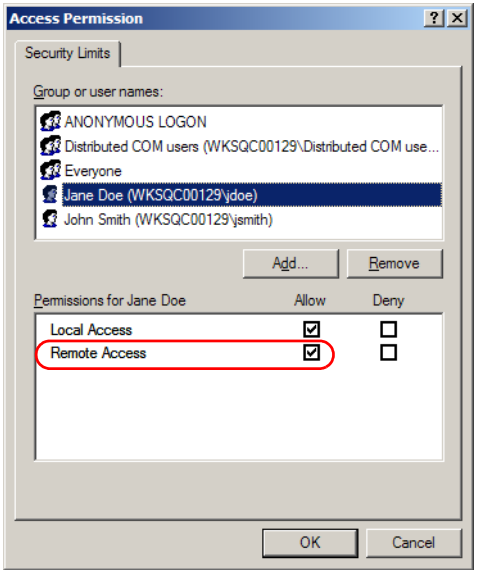
4. In the **Select Users or Groups** dialog box, under **Enter the object names to select**, type the name of the user to whom you want to give access rights.



5. Click **OK**.

6. Confirm the newly added user has **Remote Access** permission as follows:

6a. In the **Access Permission** dialog box, select the name of the new user.

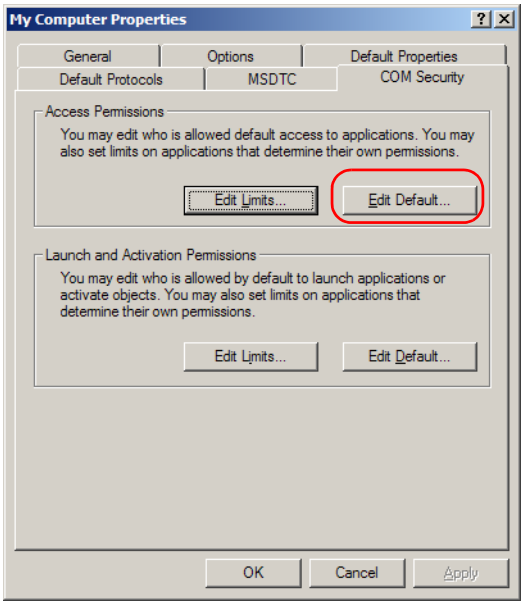


6b. Under **Permissions for** (new user), ensure **Allow** is selected for **Remote Access**.

6c. Click **OK**.

7. In the **My Computer Properties** dialog box, click the **COM Security** tab.

8. Under **Access Permission**, click **Edit Default.**

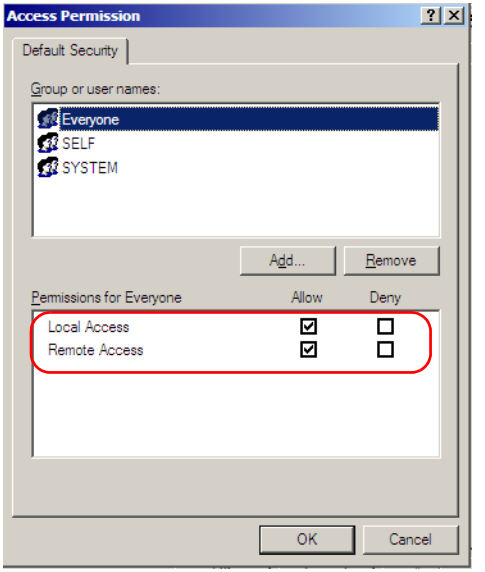


9. In the **Access Permission** dialog box, ensure the Everyone user appears in the **Group or user names** list.

10. Confirm the Everyone user has remote access permission as follows:

10a. In the **Access Permission** dialog box, select the Everyone user.

10b. Under **Permissions for Everyone**, ensure **Allow** is selected for both **Local Access** and **Remote Access**.

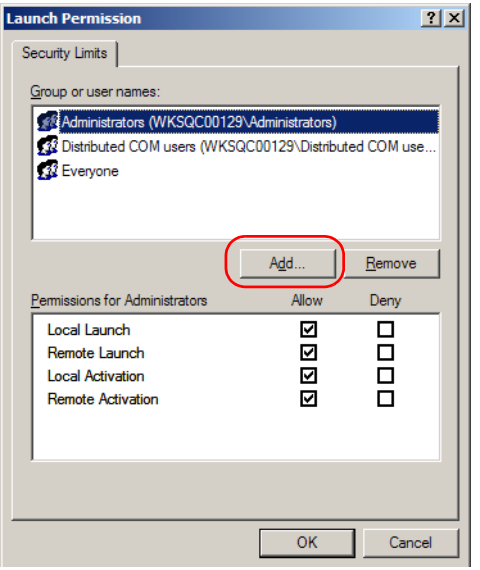


10c. Click **OK**.

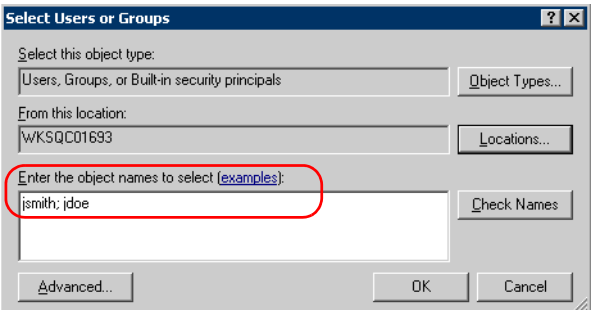
11. In the **My Computer Properties** dialog box, click the **COM Security** tab.

12. Under **Access Permissions**, click **Edit Limits**.

13. In the **Launch Permission** dialog box, click **Add**.



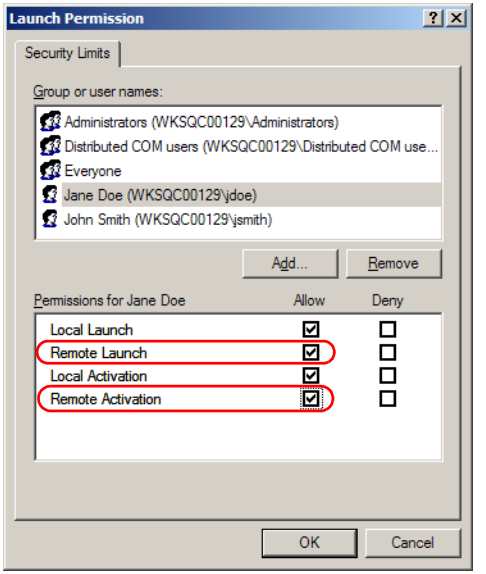
14. In the **Select Users or Groups** dialog box, under **Enter the object names to select**, type the name of the user to whom you want to give start and activation access rights.



15. Click **OK**.

16. Confirm the newly added user has **Remote Launch and Activation** permission as follows:

16a. In the **Launch Permission** dialog box, select the name of the new user.



16b. Under **Permissions for** (new user), ensure **Allow** is selected for both **Remote Launch** and **Remote Activation**.

16c. Click **OK**.

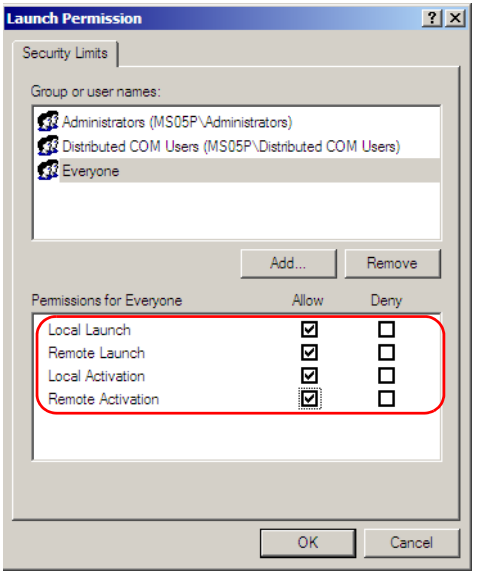
17. In the **My Computer Properties** dialog box, click the **COM Security** tab.

18. Under **Launch and Activation Permissions**, click **Edit Default**.

19. Confirm that the ANONYMOUS LOGON and Everyone users have the permission for **Local Launch**, **Remote Launch**, **Local Activation** and **Remote Activation**:

19a. In the **Launch Permission** dialog box, select Everyone.

19b. Under **Permission for Everyone**, ensure **Allow** is selected for **Local Launch**, **Remote Launch**, **Local Activation** and **Remote Activation**.



19c. Click **OK**.

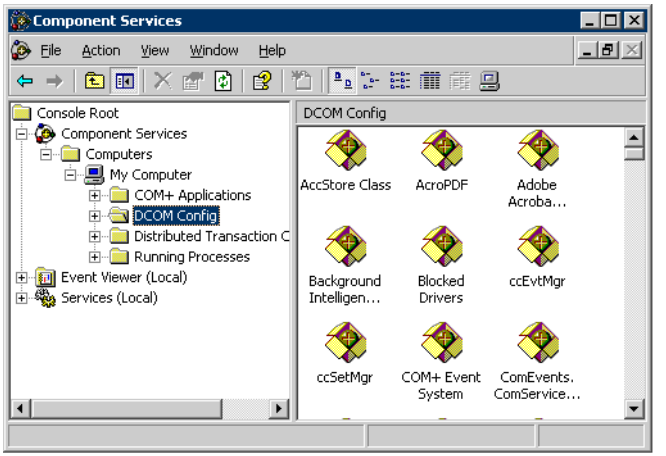
19d. In the **Launch Permission** dialog box, select the ANONYMOUS LOGON user, and then repeat steps 19b and 19c.

Customizing the Specific Security Parameters

Once you have defined the general security parameters, you can define the specific security parameters.

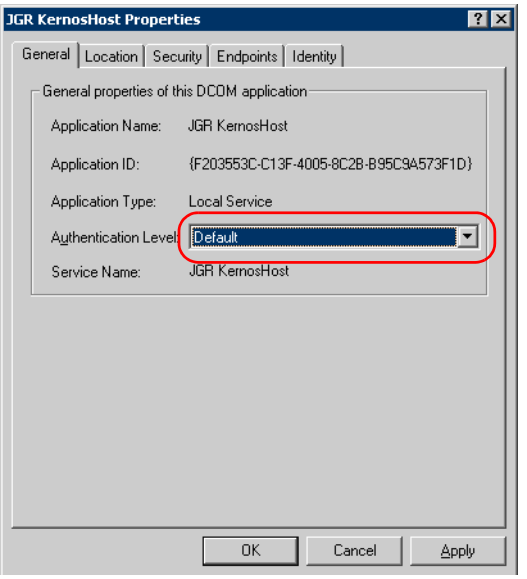
To customize the specific security parameters:

1. From the **Component Services** dialog box, select: **Console Root> Component Services> Computers> My Computer> DCOM Config** to show the contents of the **DCOM Config** folder.

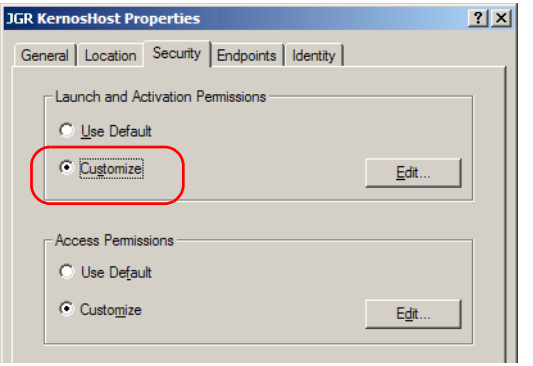


2. From **DCOM Config**, right-click **JGR KernosHost**, and select **Properties**.
3. Click the **General** tab.

4. In the **Authentication Level** list, select **Default**.



5. In the **JGR KernosHost Properties** dialog box, click the **Security** tab.
6. Under **Launch and Activation Permissions**, select **Customize**, and then click **Edit** to edit the list of allowed users.

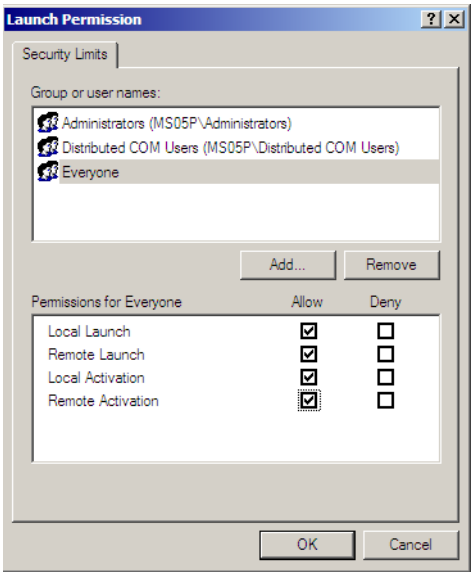


IMPORTANT

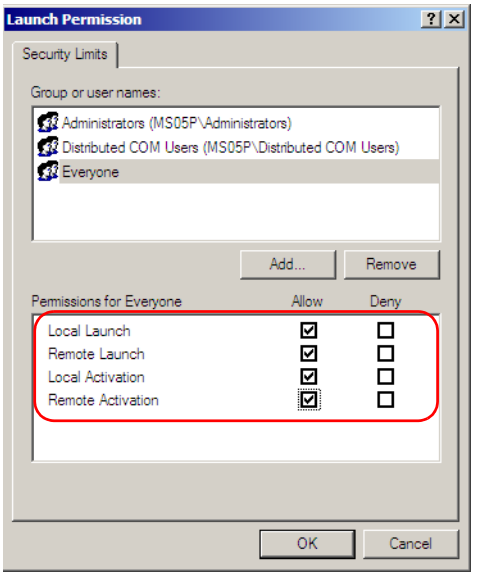
Customizing access rights sets both remote AND local permissions. As a result, you will have to specify every user who must have local access to the system (see *Setting the General Security Parameters* on page 109).

If you do not specify local access rights, no user will be able to access JGR KernosHost and, therefore, no user will be able to start MS Manager.

7. In the **Launch Permission** dialog box, select the Everyone user.

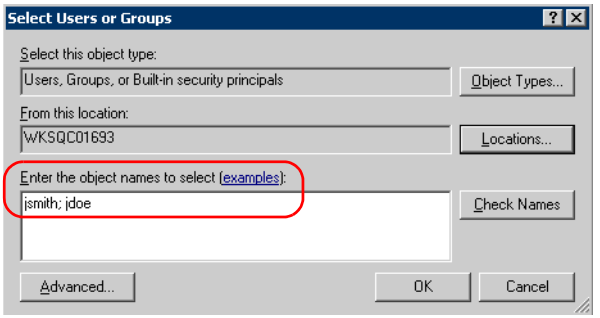


8. Under the **Permissions for Everyone**, ensure **Allow** is selected for **Local Launch**, **Remote Launch**, **Local Activation** and **Remote Activation**, and then click **OK**.



9. In the **Launch Permission** dialog box, click **Add**.

10. In the **Select Users or Groups** dialog box, under **Enter the object names to select**, type the name of the user to whom you want to give start and activation permissions for remote access.

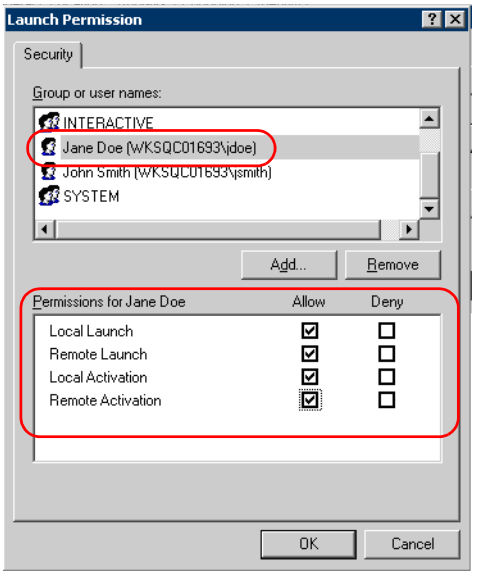


11. Click **OK**.

12. In the **Launch Permission** dialog box, select a user.

13. To allow this user to start and activate the MS Platform remotely, select **Allow** for all four permission choices.

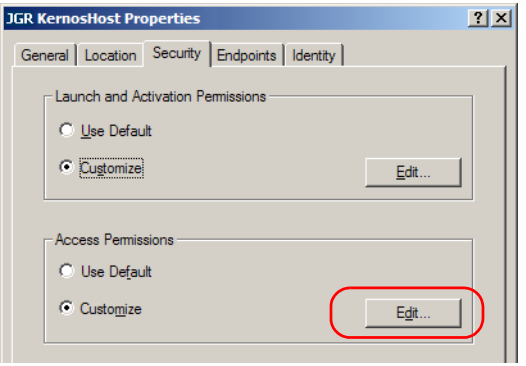
14. Repeat steps 12 and 13 for each newly added user.



15. Click **OK**.

16. In the **JGR KernosHost Properties** dialog box, click the **Security** tab.

17. Under **Access Permissions**, select **Customize**, and click **Edit** to edit the list of allowed users.



IMPORTANT

Customizing access rights sets both remote AND local permissions. As a result, you will have to specify every user who must have local access to the system (see *Setting the General Security Parameters* on page 109).

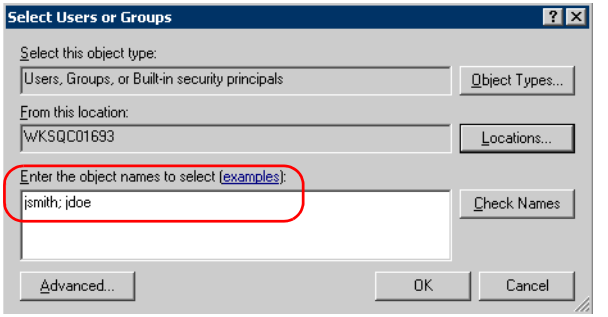
If you do not specify local access rights, no user will be able to access JGR KernosHost and, therefore, no user will be able to start MS Manager.

18. In the **Access Permission** dialog box, select **Everyone**.

19. Under the **Permissions for** (Everyone), ensure **Allow** is selected for **Local Access** and **Remote Access**, and then click **OK**

20. In the **Access Permission** dialog box, click **Add**.

21. In the **Select Users or Groups** dialog box, under **Enter the object names to select**, type the name of the user to whom you want to give access permissions for remote access.

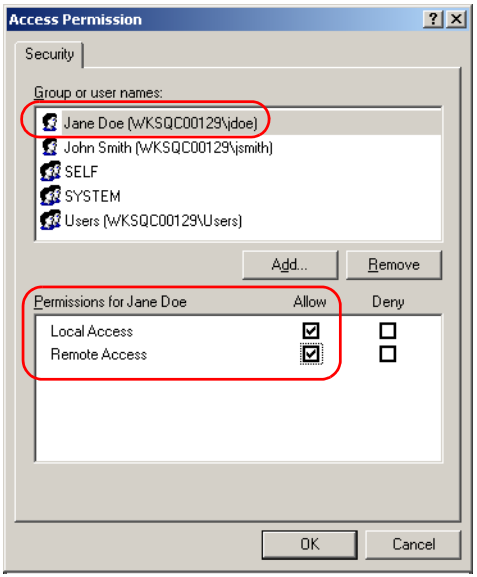


22. Click **OK**.

23. In the **Access Permission** dialog box, select a user.

24. To allow this user to access the MS Platform remotely, select **Allow** for both permission choices.

25. Repeat steps 23 and 24 for each newly added user.



Note: *You can also deny connection permission for specific users.*

26. Click **OK** to close the **Access Permission** dialog box.

27. Click **OK** to close the **JGR KernosHost Properties** dialog box.

28. Restart your MS Platform.

The JGR IcSCPIAccess Class component, located on your MS Platform, can now be accessed with DCOM.

Note: *To run DCOMCNFG.EXE, you need Administrator access rights.*

Preparing Client Computer for DCOM Operation

Prior to enabling DCOM operation on the client computer, you need to ensure the IcSCPIActiveX.dll file and the relevant communications port are installed. Otherwise, you will not be able to enable DCOM on the client computer.

To register the IcSCPIActiveX.dll file on the client computer:

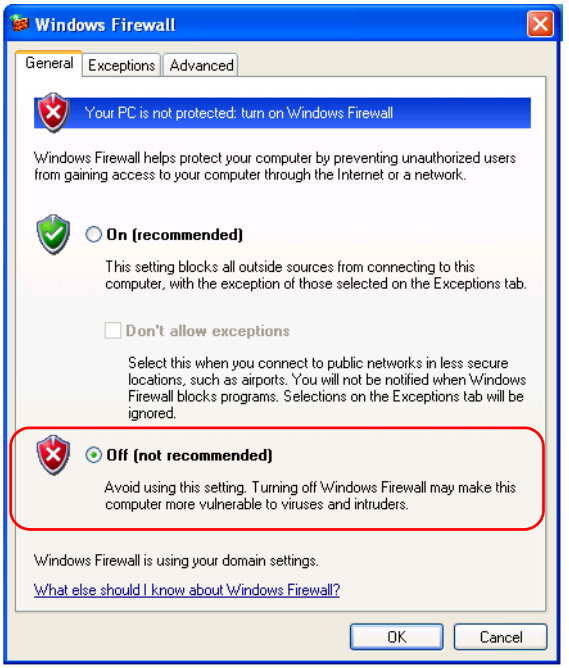
1. In C:\Program Files, select: Common Files>JGR>Instrument Control, and then select the IcSCPIActiveX.dll file.
2. For Windows XP, copy the file to C:\Windows\System32 folder on the client computer.

   OR

   For Windows 2000, copy the file to C:\WINNT\System32 folder on the client computer.
3. On the client computer, click **Start**, and then select **Run**.
4. In the **Open** box, type "regsvr32 IcSCPIActiveX.dll" to register the IcSCPIActiveX.dll file.

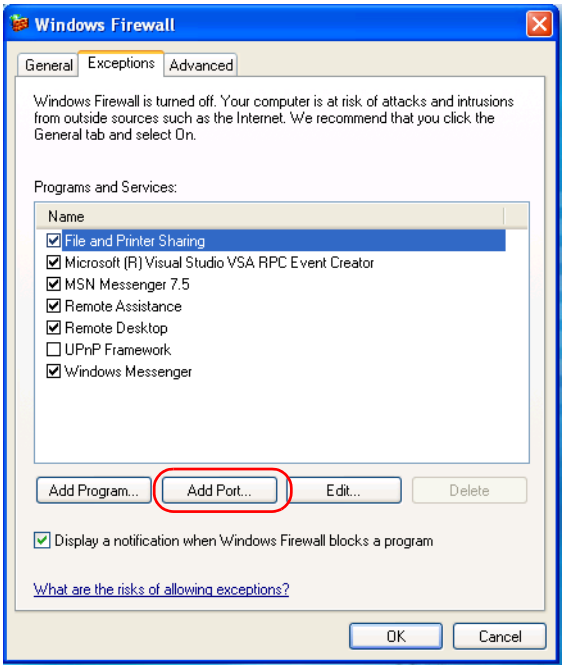
To add a communications port number on the client computer:

1. For Windows XP and Windows Vista, on the Control Panel, select **Windows Firewall**, and then select **Off (not recommended)** to disable the firewall.

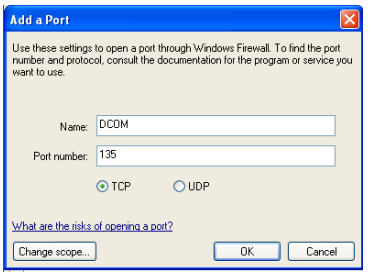


2. Select the **Exceptions** tab.

3. Click **Add Port**.



4. In the **Name** box, type "DCOM".
5. In the **Port number** box, type "135", and then click **OK**.



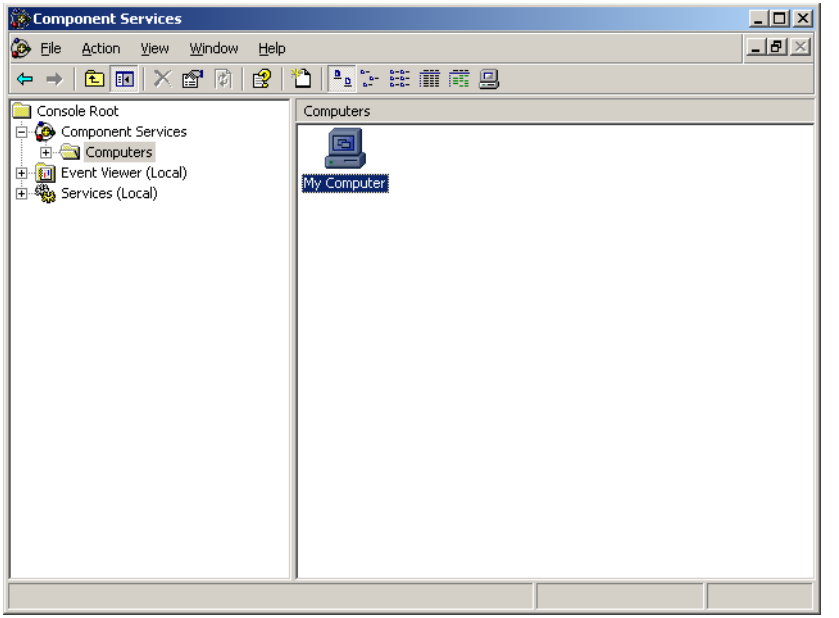
Enabling DCOM on Client Computer

If you want to subscribe to JGR IcSCPIAccess Class component events, you need to set security parameters on the client computer.

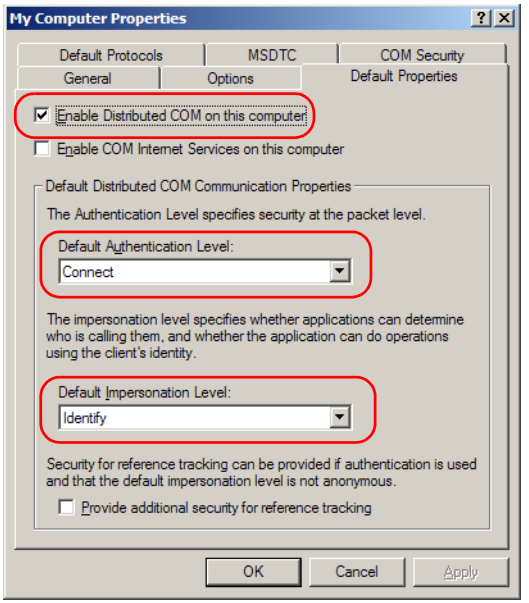
To enable DCOM on the client computer:

1. Start **Windows**, on the taskbar, click **Start** and select **Run**.
2. In the **Open** box, type “DCOMCNFG.EXE” and click **OK**.
3. Click **OK** to open the **Component Services** dialog box.

4. In the **Component Services** dialog box, select: **Console Root**> **Component Services**> **Computers** to show available computers.

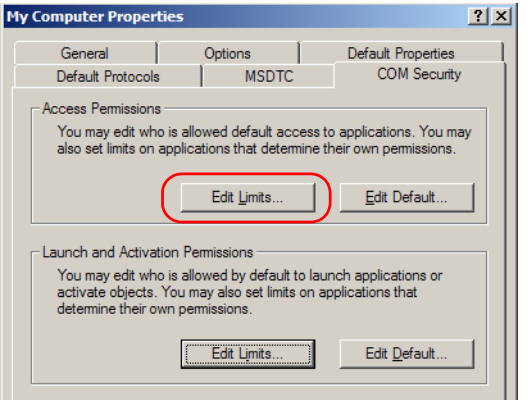


5. Right-click **My Computer**, and then select **Properties**.
6. In the **My Computer Properties** dialog box, click the **Default Properties** tab.
7. Select **Enable Distributed COM on this computer**.



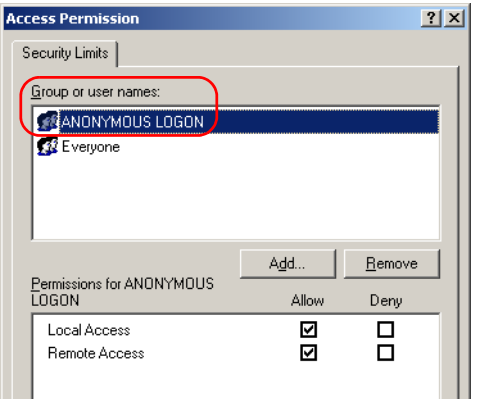
8. Under **Default Distributed COM Communication Properties**, in the **Default Authentication Level** list, select **Connect**.
9. In the **Default Impersonation Level** list, select **Identify**.
10. Click **OK**.

11. Click the **COM Security** tab and, under **Access Permissions**, click **Edit Limits**.



12. In the Access Permission dialog box, ensure that, for **ANONYMOUS LOGON**, local and remote accesses are allowed.

If **ANONYMOUS LOGON is not listed under Group or user names**, click **Add** to add it.



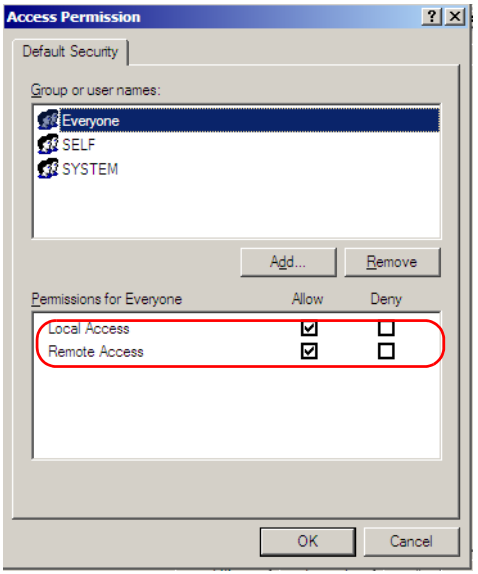
13. In the **My Computer Properties** dialog box, click the **COM Security** tab.

14. Under **Access Permissions**, click **Edit Default**.

15. Confirm the ANONYMOUS LOGON and Everyone users have **Remote Launch and Activation** permission as follows:

15a. In the **Access Permission** dialog box, select Everyone.

15b. Under **Permission for Everyone**, ensure **Allow** is selected for **Local Access** and **Remote Access**.

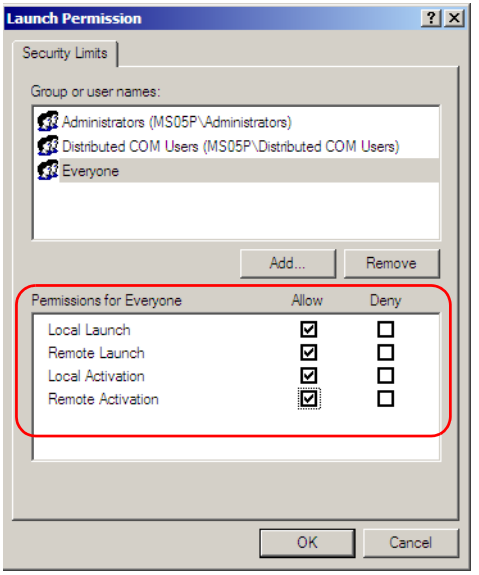


15c. Click **OK**.

15d. Repeat steps 15a to 15c for the ANONYMOUS LOGON user.

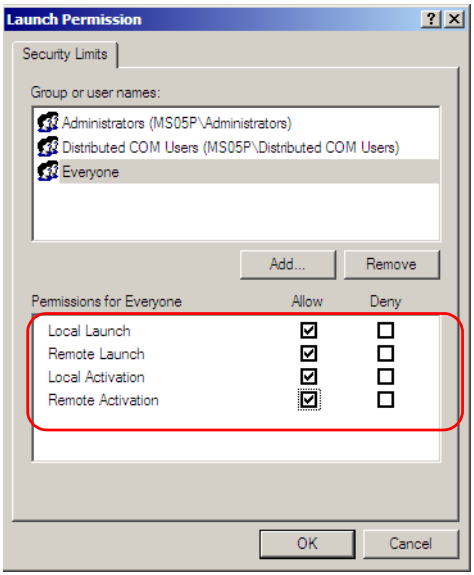
16. Click the **COM Security** tab and, under **Launch and Activation Permissions**, click **Edit Limits**.

17. To allow ANONYMOUS LOGON and Everyone to start and activate the platform, select **Allow** for all four permission choices.



18. In the **My Computer Properties** dialog box, click the **COM Security** tab and, under **Launch and Activation Permissions**, click **Edit Default**.

19. For ANONYMOUS LOGON and Everyone, ensure **Allow** is selected for **Local Launch**, **Remote Launch**, **Local Activation** and **Remote Activation**, and then click **OK**.



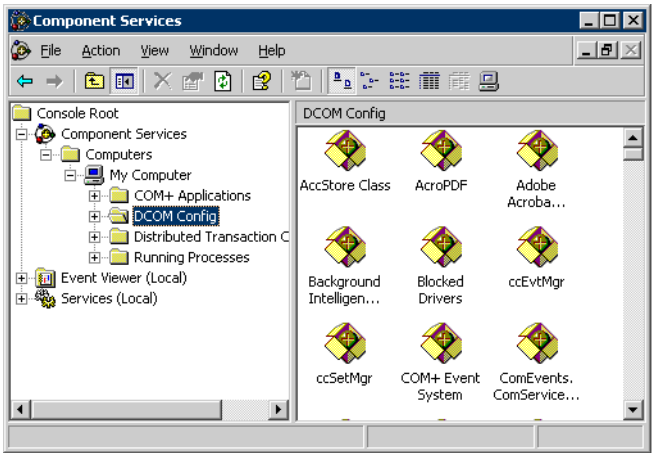
For more information on enabling events with DCOM, refer to *AppId Key* in MSDN Documentation.

Disabling DCOM Access to Your MS Platform

If you no longer want client computers to access your MS Platform using DCOM, you can disable this access.

To disable DCOM access to your MS Platform:

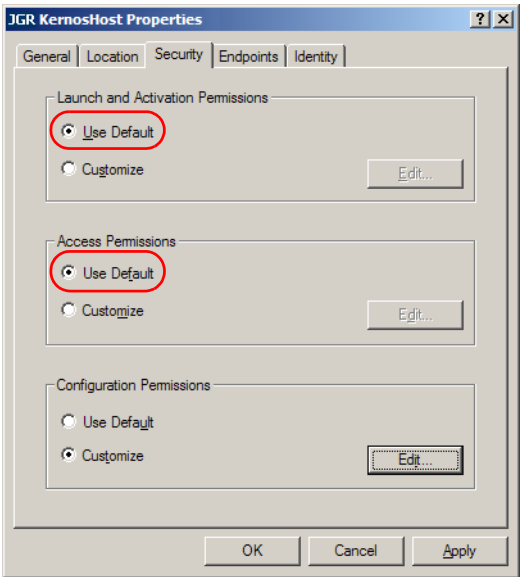
1. Start **Windows**, on the taskbar, click **Start** and select **Run**.
2. In the **Open** box, type "DCOMCNFG.EXE" and click **OK**.
3. In the **Component Services** dialog box, select: **Console Root**> C**omponent Services**> **Computers**> **My Computer**> **DCOM Config** to show the contents of the **DCOM Config** dialog box.



4. In the **DCOM Config** dialog box, right-click **JGR KernosHost**, and select **Properties**.

5. In the **JGR KernosHost Properties** dialog box, click the **Security** tab.
6. Under **Launch and Activation Permissions** and **Access Permissions**, select **Use Default**.

   This ensures the JGR IcSCPIAccess Class component uses the default lists instead of the customized lists.



7. Click **OK**.
8. Restart your MS controller unit.

   The JGR IcSCPIAccess Class component, located on your MS Platform, *cannot* be accessed with DCOM.

7 Using MS Products in an Automated Test Environment

JGR supplies commands that follow the guidelines determined by the SCPI consortium. JGR also supplies COM properties and events allowing you to build your own application.

Your application can be developed using Visual C++, .NET, Visual Basic or any other language that runs under Windows XP.

The present chapter gives you information to help you use the provided commands, drivers, and COM properties and events to remotely control your instruments.

If you need information on how to prepare your MS Platform for remote control, see the corresponding section in this documentation.

Managing the States of Your MS Platform

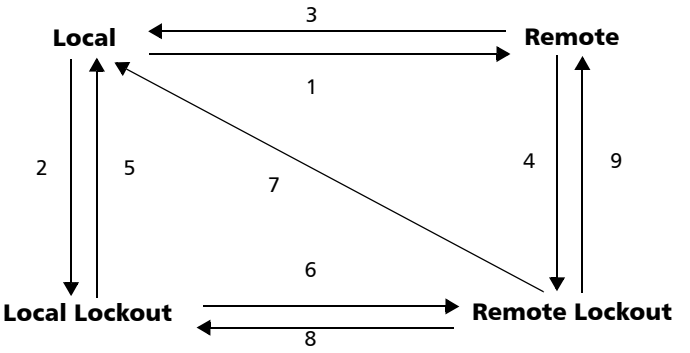
Your MS Platform can be in one of the following states:

State	Characteristics
Local	➤ Front-panel keys of the MS controller unit and peripherals are all functional ➤ Default state at unit turn on
Remote	➤ Front-panel keys of the MS controller unit and peripherals are *not* functional except for the LOCAL key and the power switch ➤ The REMOTE LED of the MS controller unit is lit
Local with lockout	➤ Return to Local mode only possible via commands sent from the system controller ➤ Pressing the front-panel LOCAL key causes an URQ message (see *Standard Status Data Structure* on page 151) ➤ In GPIB, this state is not available when the device has the RL2 capability (see *Preparing Hardware for GPIB Control* on page 98) ➤ The LOCK LED of the MS controller unit is lit

State	Characteristics
Remote with lockout	➤ Front-panel keys of the MS controller unit and peripherals are *not* functional except for the power switch ➤ Return to Local mode only possible via commands sent from the system controller ➤ Pressing the front-panel LOCAL key causes an URQ message (see *Standard Status Data Structure* on page 151) ➤ In GPIB, this state is not available when the device has the RL2 capability (see *Preparing Hardware for GPIB Control* on page 98) ➤ Both the REMOTE and LOCK LEDs of the MS controller unit are lit

Note: *The current state is displayed in the MS Manager status bar.*

The following diagram shows the possible transitions between the different states.



No.	Possible in GPIB if:	Possible in ActiveX by setting:
1	➤ The remote interface message (REM) is set to True. ➤ The device is addressed to listen (MLA). ➤ The return to local interface message (rtl) is set to False.	The *RemoteState* property to True.
2	➤ The remote interface message (REM bus line) is set to True. ➤ The local lockout interface message (LLO) is set to True.	The *LockoutState* property to True.
3	One of the following conditions is true: ➤ The go to local interface message (GTL) is received and the device is addressed to listen (MLA). ➤ The Clear Lockout/Set Local message is received (the REM bus line is set to False). ➤ The return to local interface message (rtl) is set to True and the local lockout interface message (LLO) is set to False.	The *RemoteState* property to False.

No.	Possible in GPIB if:	Possible in ActiveX by setting:
4	The local lockout interface message (LLO) is set to True.	The *LockoutState* property to True.
5	The Clear Lockout/Set Local message is received from the controller (the REM bus line is set to False).	The *LockoutState* property to False.
6	The device is addressed to listen (MLA).	The *RemoteState* property to True.
7	The Clear Lockout/Set Local message is received from the controller (the REM bus line is set to False).	-----
8	The go to local interface message (GTL) is received.	The *RemoteState* property to False.
9	-----	The *LockoutState* property to False.

Note: *See the appendix on COM properties and events for more information.*

Note: *You can also return to local state by turning the device off and on again. However, this operation will cause the controller to lose control of the system (no more local lockout state). All settings made on the device by the controller will be lost (reset of device configuration when turning on).*

Message Management

Each device that is physically connected to the GPIB link has its own input buffer, output queue and error/event queue. These data structures allow storage of incoming messages (single or compound commands that are sent to an instrument), responses from queries, errors and events that may occur.

Data structure	Characteristics	Clearing
Input buffer	➤ Consists of a First-In, First-Out (FIFO) data structure. ➤ Stores Data Bytes (DABs) and END messages. ➤ Delivers messages to the parser in the order that they were received from the I/O control. ➤ Maximum message length: unlimited in DABs (the input buffer size is only limited to the total size of the device memory). ➤ An incoming byte empties the output queue. An error will be raised if the output queue contained data. Consequently, it clears the Message AVailable bit (bit number 4 –MAV from the Status Byte register).	The buffer will be cleared by: ➤ Turning off the power. ➤ Sending a Device Clear (DCL) message to the instrument. ➤ Sending a Selected Device Clear (SDC) message to the instrument.

Data structure	Characteristics	Clearing
	➤ Except for the string and binary block contents, the following transformations are made on the incoming data: character conversion from lower case to upper case. conversion of "<wsp>" characters to spaces. conversion of multiple blanks to a single blank. ➤ Parser begins to process messages when the <PROGRAM MESSAGE TERMINATOR> is received or if the input buffer is full.	
Output queue	➤ Consists of a First-In, First-Out (FIFO) data structure. ➤ When the instrument acts as a talker, it sends response messages (from the output queue) to the controller. Response messages all end with a *<RESPONSE MESSAGE TERMINATOR>*, see the appendix on data types. ➤ Total storage capacity: only limited to the device's memory. ➤ As soon as there is data in the output queue, the Message AVailable bit (bit number 4 –MAV from the Status Byte register) is set to 1. ➤ Remains empty if no query is received or if the query contains an error.	The Output queue will be cleared by: ➤ Reading all the items it contains. ➤ Turning off the power. ➤ Sending a Device Clear (DCL) message to the instrument. ➤ Sending a Selected Device Clear (SDC) message to the instrument. ➤ Attempting to send a command before reading the responses to previous queries (an error will also be raised).

Data structure	Characteristics	Clearing
Error/Event queue	➤ Consists of a First-In, First-Out (FIFO) data structure. ➤ Total storage capacity: 50 errors or events. ➤ Errors or events can be retrieved, one at a time, with *:SYSTem:ERRor[:NEXT]?*. ➤ When an error or event occurs and the Error/Event queue is full, the last item in the queue (the most recent) is removed and the Queue overflow error (error –350) is added. No new items will be stored into the queue until there will be room available. ➤ As soon as there is data in the output queue, the Error AVailable bit (bit number 2 –EAV from the Status Byte register) is set to 1.	The Error/Event queue will be cleared when: ➤ Reading all the items it contains. ➤ Turning off the power. ➤ Sending a Device Clear (DCL) message to the instrument. ➤ Sending a Selected Device Clear (SDC) message to the instrument. ➤ Using the *CLS command.

Standard Status Data Structure

Each device that is physically connected to the GPIB bus has four status registers with a structure complying with the IEEE 488.2 standard. These registers allow the controller to monitor events and get useful information on the status of the devices it controls.

- Standard Event Status Register (ESR)
- Standard Event Status Enable Register (ESE)
- Status Byte Register (STB)
- Service Request Enable Register (SRE)

ESR and ESE

The standard event status register and status enable register information is presented in the following table.

Bits	Mnemonics	Bit Value
7	Power On (PON)	128
6	User Request (URQ)	64
5	Command Error (CME)	32
4	Execution Error (EXE)	16
3	Device-Dependent Error (DDE)	8
2	Query Error (QYE)	4
1	Not Used (N.U.)	0
0	Operation Complete (OPC)	1

The following table presents a summary of the possible operations on ESR and ESE registers.

Register	Read	Write	Clear
ESR	Use *ESR?.	Impossible to write.	➤ Use *CLS. ➤ Read the register.
ESE	Use *ESE?.	Use *ESE.	➤ Use *ESE with a value equal to 0. ➤ At MS Manager startup, the register is set to 0. Use *ESE with a value equal to 0.

STB and SRE

The status byte register and service request enable register information is presented in the following table.

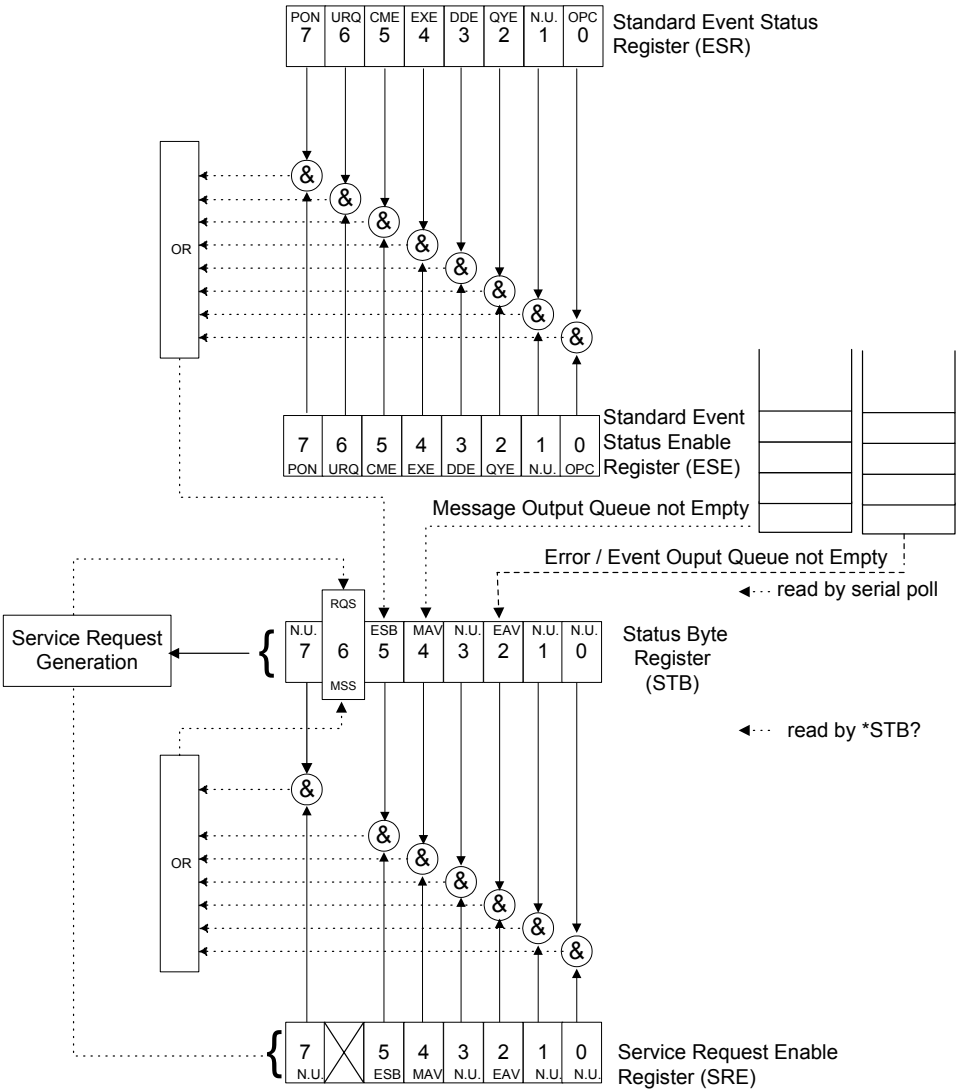
Bits	Mnemonics	Bit Value
7	Not Used (N.U.)	0
6	Master Summary Status (MSS)/ Service Request (RQS)	64
5	Event Summary Bit (ESB)	32
4	Message Available (MAV)	16
3	Not Used (N.U.)	0
2	Error Available (EAV)	4
1	Not Used (N.U.)	0
0	Not Used (N.U.)	0

The following table presents a summary of the possible operations on STB and SRE registers.

Register	Read	Write	Clear
STB	➤ Use *STB?. ➤ Use serial poll (GPIB bus sequence that allows retrieval of the value without interrupting the current process).	Impossible to write; the register content is only modified when the Event registers or Queues are modified.	Use *CLS before sending a query (to clear the Event registers and Queues and by the same token clear the STB register).
SRE	Use *SRE?.	Use *SRE with a value equal to 0 to disable the register or with a value equal to 1 to enable it.	➤ Use *SRE with a value equal to 0. ➤ At MS Manager startup, the register is set to 0.

The diagram displayed on the next page is a useful aid in understanding the general commands and how a service request (SRQ) is generated.

Using a service request, a device notifies the controller that an event requiring special attention occurred. The controller will then find which device generated a SRQ (its RQS bit is set) and the causes of it.


PON 7
URQ 6
CME 5
EXE 4
DDE 3
QYE 2
N.U. 1
OPC 0
Standard Event Status Register (ESR)
OR
7 PON
6 URQ
5 CME
4 EXE
3 DDE
2 QYE
1 N.U.
0 OPC
Standard Event Status Enable Register (ESE)
Message Output Queue not Empty
Error / Event Ouput Queue not Empty
read by serial poll
RQS
MSS
Service Request Generation
N.U. 7
6
ESB 5
MAV 4
N.U. 3
EAV 2
N.U. 1
N.U. 0
Status Byte Register (STB)
read by *STB?
OR
7 N.U.
5 ESB
4 MAV
3 N.U.
2 EAV
1 N.U.
0 N.U.
Service Request Enable Register (SRE)

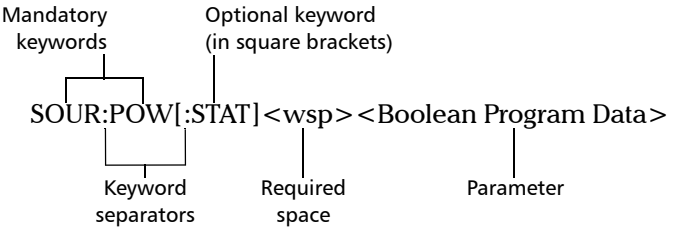
SCPI Command Structure

The information presented in this section provides an overview of GPIB programming. If you need detailed information, refer to:

- The International Institute of Electrical and Electronics Engineers. *IEEE Standard 488.1-1987, IEEE Standard Digital Interface for Programmable Instrumentation*. New York, 1987.
- The International Institute of Electrical and Electronics Engineers. *IEEE Standard 488.2-1992, IEEE Standard Codes, Formats, Protocols and Common Commands For Use with ANSI/IEEE Std. 488.1-1987*. New York, 1992.
- *Standard Commands for Programmable Instruments (SCPI). Volume 1: Syntax and Style*. Vers. 1999.0 May, U.S.A, 1999.

The provided commands follow the guidelines determined by the Standard Commands for Programmable Instruments (SCPI) consortium. A *program message* consists of one or more commands (and/or queries) with their appropriate parameters.

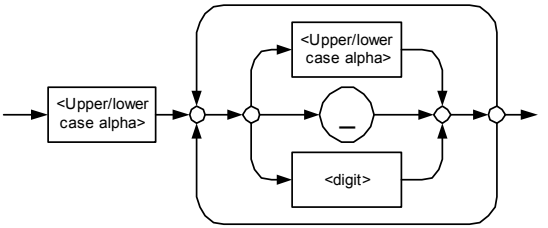
For example, a program message could contain a command used to activate or deactivate a source. The corresponding command syntax would be:



When sending a message containing the previous command, you would actually type: SOUR:POW ON.

The following table shows elements that are commonly used in the commands or queries syntax.

Item	Meaning
[ ]	Enclose optional keywords or parameters. *Do not include square brackets in your program message.*
[1..n]	Indicates that the instrument provides multiple capabilities and that you have to specify which one you want to use. If you omit the value, the command will take effect on the first capability. Multiple capabilities can be found at any branch of the command tree (root, intermediate node or terminal node). Example: If the command is :SENSe*[1..n]*:CORRection:COLLect:ZERO and you want it to take effect on the second SENSe (sensor) capability of the instrument, you may send this: :SENSe*2*:CORRection:COLLect:ZERO. *Do not include square brackets in your program message; simply enter the number.*
<wsp>	Indicates that a space is required ("wsp" stands for "white space"). Corresponds to ASCII character codes (0 to 9 and 11 to 32, in decimal). *Do not include* "<wsp>" *in your program message; simply type a space.*
<digit>	Element used in the construction of various numeric data types. Can take any value between 0 and 9 inclusively (corresponds to ASCII character codes 48 to 57, in decimal).

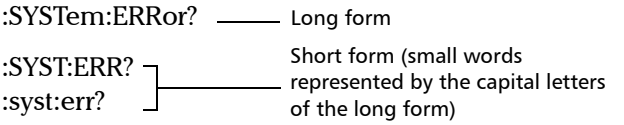
Item	Meaning
<mnemonic>	Element used in the construction of certain data types and program messages.  In the diagram above, ➤ "<Upper/lower case alpha>" corresponds to ASCII character codes (65 to 90 and 97 to 122, in decimal). ➤ "_" corresponds to an underscore character (code 95, in decimal).
< >	Text appearing between angled brackets specifies the command parameter to be sent or the response you will receive from an instrument. *Do not include angled brackets in your program message.*
|	Indicates that one, and only one, value must be selected from the available choices. Example: If the list is 0|1, you can only select 0 *or* 1. *Do not include the pipe character in your program message.*
{ }	Indicate that the enclosed parameters can appear 0 to n times when the command is used. *Do not include braces in your program message.*
:	Mandatory to separate keywords. Can be omitted at the beginning of a program message. For example, you can use either :SYST:ERR or SYST:ERR.

Item	Meaning
;	➤ Mandatory to separate the different commands of a program message when more than one command is sent at a time. In this case, it is called *<PROGRAM MESSAGE UNIT SEPARATOR>*. ➤ Also used to separate responses when multiple queries were sent in a single program message. In this case, it is called *<RESPONSE MESSAGE UNIT SEPARATOR>*.
,	➤ Mandatory to separate parameters in a command or a query. In this case, it is called *<PROGRAM DATA SEPARATOR>*. ➤ Also used to separate the various responses from a query. In this case, it is called *<RESPONSE DATA SEPARATOR>*.

There are also several conventions regarding command syntax:

- Spelling errors will cancel the command or query.
- Commands and queries are not case-sensitive. You can type your program messages using either lower-case or upper-case letters.
- The command or query can be written using only the three- or four-letter shortcuts, only full words, or a combination of both.

  The example below shows the long and the short forms of a same query.



- For readability reasons, you can use extra spaces in your program messages but they will not be taken into account. For more information, see *Message Management* on page 148.

Note: *Omitting the leading* <mnemonic> *path is only possible when you have more than one command or query in the program message that you send.*

Consulting Data Types

If you need information about data types used in JGR's documentation, see the appendix on data types.

Writing Remote Control Code

Complex measurement programs may be written using any programming environment that supports GPIB communication. GPIB development kits are available for most of the popular commercial programming languages.

The MS Platform offers many commands permitting complete remote control of all the MS components. These commands adhere to the SCPI standard.

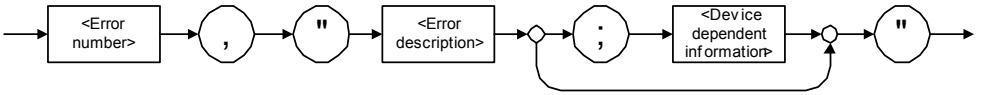
You can find all the commands and queries supported by the MS controller unit in the *IEEE 488.2 and Specific Commands* appendix. For information on commands specific to particular instruments, refer to the instrument's user guide.

When you write code, you must follow these rules on message reception and transmission:

- The controller must have sent a complete message to the instrument (including the message terminator) before retrieving a response.
- The controller must retrieve all the responses from previous queries (including the response terminator) before sending a new message to an instrument.
- The controller must not try to retrieve a response from an instrument if the corresponding query has not been previously sent to the instrument.
- You must pay special attention to queries that return an indefinite ASCII response. To avoid any confusion, the IEEE 488.2 standard requires that this data type be immediately followed by a response termination character. For this reason, when working with compound queries, you must ensure that a query sending an indefinite ASCII response is the last query of the series.
- Be careful when sending program messages containing multiple queries that return large amounts of data. Since the controller can only retrieve data when the instrument has finished processing the queries, it could result in problems ranging from a saturation of the output queue to the complete blocking of the whole system.

Error Message Format

System and device-specific errors are managed by the MS Platform. The generic format for error messages is illustrated in the following figure.



As shown in the above figure, the message contains three parts:

- error number
- error description
- device-dependent information

Error messages ending in a negative number are SCPI-based errors.

For more information on errors, see *Message Management* on page 148. For a complete list of possible errors, see the appendix on SCPI-based errors.

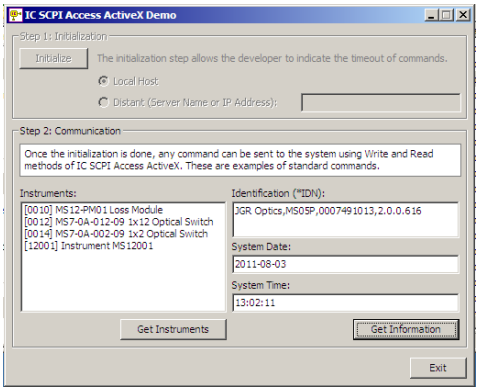
Working with JGR COM Objects

The *MS Manager* CD offers you a demo application to show how to communicate with your MS Platform using COM technology via the provided IcSCPIAccess interface.

The source files can be found on the CD under:
F:\examples\ScpiActiveX

where *F* corresponds to the CD-ROM drive (the CD-ROM drive letter may differ from one computer to another).

This application has been designed with Visual Basic and, therefore, must be compiled using Visual Basic 6.0, Service Pack 5.



With this demo application, you can control your MS Platform either in local (COM technology) or remote (DCOM technology) mode.

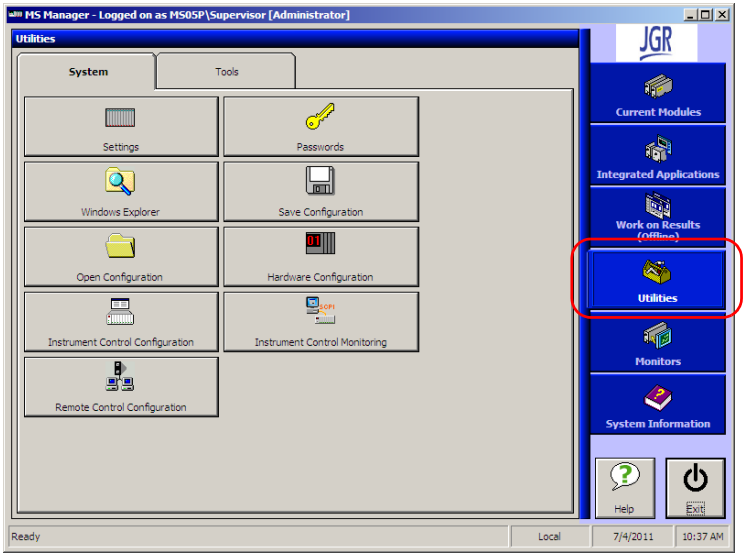
If you want to work in remote mode, see *Preparing for Remote Control with DCOM* on page 108.

Monitoring Remote Commands

MS Manager allows you to monitor remote commands sent to your units, if desired.

To monitor remote commands:

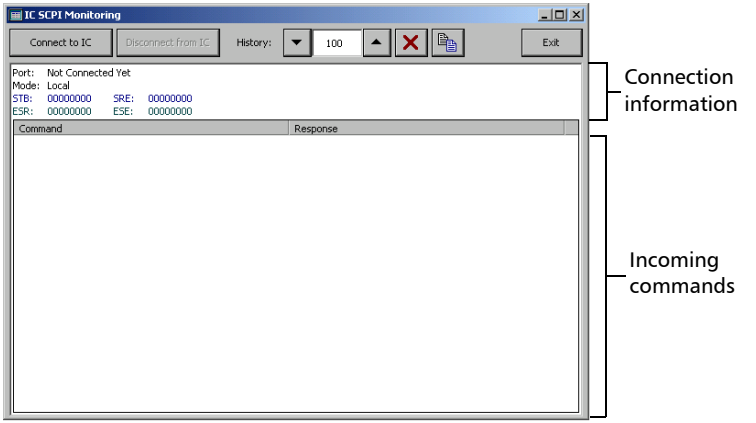
1. Click the **Utilities** function tab.



2. Click **Instrument Control Monitoring** to access the corresponding utility.

3. Do one of the following:

➤ If monitoring using GPIB, ActiveX or RS-232, click **Connect to IC**.



You are automatically connected to the monitoring system. The **Disconnect from IC** button becomes available for you when you are ready to disconnect.

Once connected, your current connection information will appear in the upper part of the window, and the commands will appear as a list in the lower part of the window.

With the **History** parameter, you determine how many commands you want to keep in the list. You can change the number by using the arrow buttons on each side of the list.

To clear the history, click .

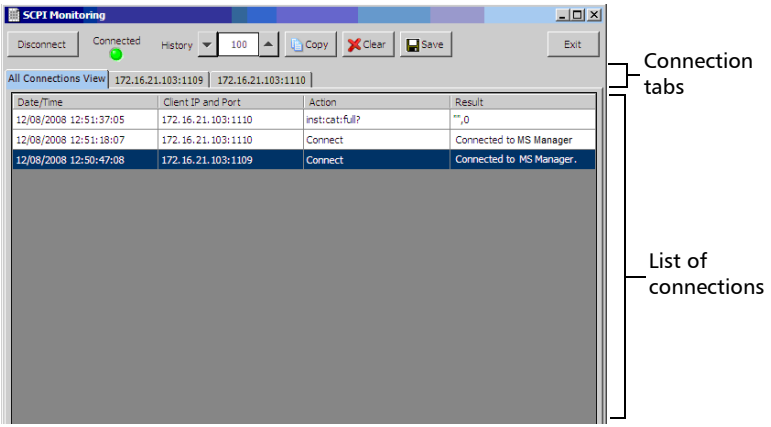
To view the list in any word processor, click copy it to the clipboard, and then paste it in your document. You can use any program, as the list is copied in text format.

To exit the monitoring utility, click **Exit**.

For more information, see the section on using your unit in an automated test environment.

➤ If monitoring using TCP/IP, which provides sending SCPI commands over TCP/IP through Telnet from the JGR Instrument Control, you are automatically connected to the monitoring system.

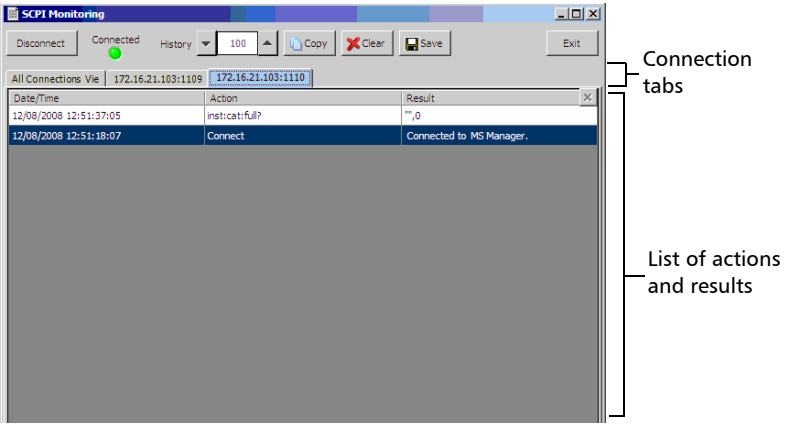
Note: *Port 5024 is designated for sending SCPI commands in the Telnet protocol.*



Once connected, your current connection information will appear in the **All Connections View** tab, and the commands will appear as a list in the lower part of the window.

The **Disconnect** button becomes available for you to click when you are ready to disconnect.

Connection information is also displayed in a separate tab, identified by its IP address, from where you can monitor the commands and other actions sent through TCP/IP over Telnet, as well as the results.



With the **History** parameter, you determine how many commands you want to keep in the list. You can increase or decrease the number by using the arrow buttons on each side of the list.

To clear the history, click **Clear**.

To view the list in any word processor, click **Copy** to copy it to the clipboard, and then paste it in your document. You can use any program, as the list is copied in text format.

To save the list as a file, click **Save**.

To exit the monitoring utility, click **Exit**.

For more information, see the section pertaining to communication through TCP/IP over Telnet.

8 Controlling Modules Remotely

Some modules come with a dedicated application designed to control them remotely from a computer. The main characteristic of this application is that it lets another user control the module exactly as if he had it close at hand.

To control the module remotely, you must configure it on the unit containing the module, including a description that can help you identify it. The user controlling the module from a computer must install the dedicated application (for more information on the installation, refer to the application documentation).

- The module can be controlled both remotely and locally at the same time.
- You will have to configure your remote control again in the following cases:
  - you inserted the module in another slot
  - you applied changes on applications while the module is not inserted in its slot.

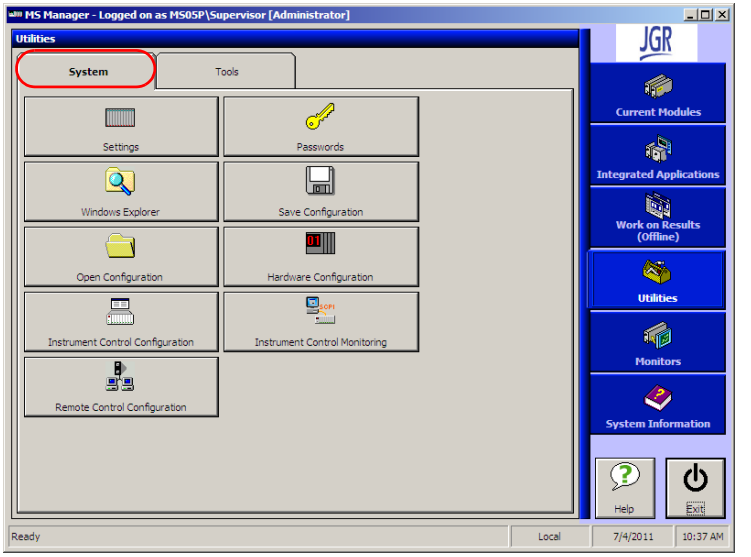
Note: *Some modules do not support the remote control.*

IMPORTANT

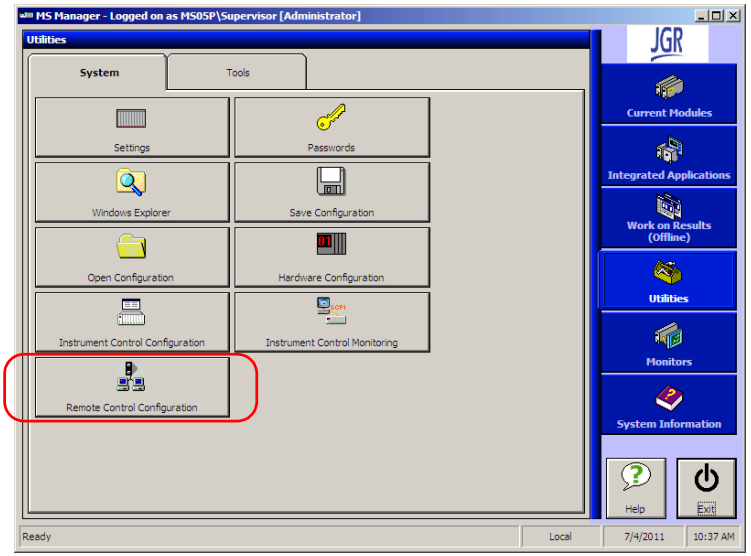
If there is a network failure or a problem with your remotely-controlled module, you can shut down the latter without closing all the other applications. The local and remote module session will then be closed.

To activate or deactivate remote control:

1. From the **Utilities function tab, select the System** tab.

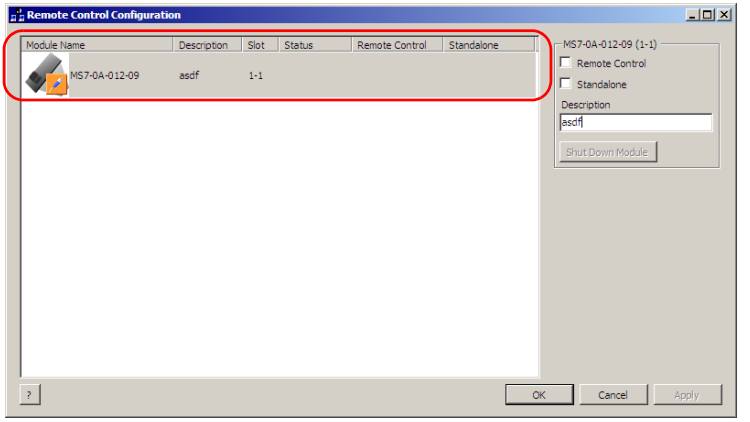


2. Select **Remote Control Configuration.**



3. From the **Remote Control Configuration** window you will see all the inserted modules. Select the module you want to have a remote access on.

Note: *Modules for which remote control is not supported are identified as unrecognized.*



4. Set the parameters:

   ➤ Select **Remote Control** to let another user control the module from a computer.

   ➤ Select **Standalone** to leave the module active even if all users close their dedicated applications.

5. Under **Description**, type a relevant description (test interface ID, for example).

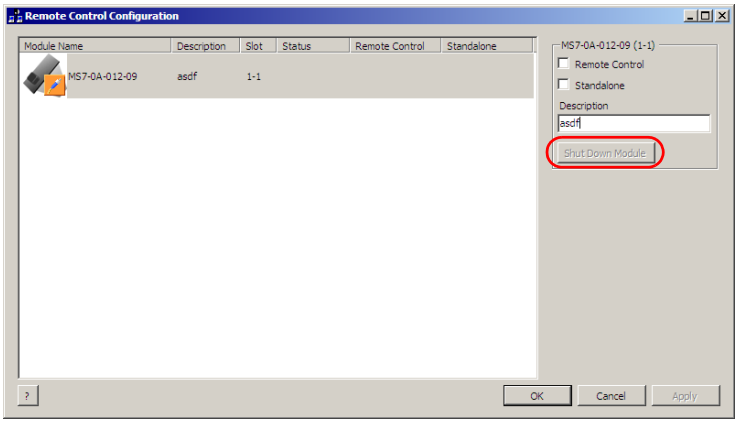
Note: *You can enter up to 10 characters. The description corresponds to the test interface ID or any other relevant description.*

6. Press **Apply** to enter your changes or **OK** to apply your changes and close the window.

Note: *This information will be updated the next time you start the module application and appear in the title bar if the module application allows it. Refer to the corresponding module documentation for more details.*

To shut down the remotely-controlled module:

Press **Shut Down Module**.



9 Maintenance

To help ensure long, trouble-free operation:

- Always inspect fiber-optic connectors before using them and clean them if necessary.
- Keep the unit free of dust.
- Clean the unit casing and front panel with a cloth slightly dampened with water.
- Store unit at room temperature in a clean and dry area. Keep the unit out of direct sunlight.
- Avoid high humidity or significant temperature fluctuations.
- Avoid unnecessary shocks and vibrations.
- If any liquids are spilled on or into the unit, turn off the power immediately, disconnect from any external power source, remove the batteries and let the unit dry completely.

WARNING

Use of controls, adjustments, and procedures for operation and maintenance other than those specified herein may result in hazardous radiation exposure.

Cleaning the Touchscreen (MS05P)

Clean the touchscreen with a soft, non-abrasive cloth dampened with glass cleaner.

Replacing the Fuse

The MS Platform contains a 10 A, L250 V, fast blow, low breaking capacity fuse. However, this fuse is not user-serviceable. If you ever encounter problems with the fuse, contact JGR for assistance.

Recycling and Disposal (Applies to European Union Only)

Recycle or dispose of your product (including electric and electronic accessories) properly, in accordance with local regulations. Do not dispose of it in ordinary garbage receptacles.

This equipment was sold after August 13, 2005 (as identified by the black rectangle).

- Unless otherwise noted in a separate agreement between JGR and a customer, distributor, or commercial partner, JGR will cover costs related to the collection, treatment, recovery, and disposal of end-of-lifecycle waste generated by electronic equipment introduced after August 13, 2005 to an European Union member state with legislation regarding Directive 2002/96/EC.
- Except for reasons of safety or environmental benefit, equipment manufactured by JGR, under its brand name, is generally designed to facilitate dismantling and reclamation.

10 Troubleshooting

Solving Common Problems

Problem	Probable Cause	Recommended Action
MS module does not turn on.	➤ AC power not applied.	➤ Verify AC power cord connection, then power unit back on.
	➤ Main power switch not at “On” position.	➤ Turn main power switch on.
	➤ AC current has been interrupted.	➤ Press on the front On/Off button twice to turn the unit on.
	➤ Fuse blown.	➤ Contact JGR.
Unit is on, but screen remains black.	External monitor not properly connected.	Verify external monitor connection.
External keyboard, printer and/or mouse do not seem to function.	No communication between peripheral and unit or peripheral in wrong connector.	Ensure that the peripheral is connected properly and that the corresponding drivers are installed.
Controller unit does not detect MS modules in an expansion unit.	➤ The expansion unit is not turned on.	➤ Turn on the expansion unit.
	➤ The modules are not properly inserted into the expansion unit.	➤ Ensure that the modules are properly inserted into the unit (remember to turn off your unit first).
	➤ The MS interface cable is not properly connected.	➤ Ensure that the cable linking your expansion unit with the main unit is properly connected.

Problem	Probable Cause	Recommended Action
A USB device is not functioning.	The USB device was not recognized by the system.	Disconnect, then reconnect the USB device.
SCPI commands are not displayed in the **IC SCPI Monitoring** window.	Communication has not been established from this window.	Establish connection by clicking the **Connect to IC** button.
	Communication has not been established on the GPIB or Ethernet port.	➤ Ensure that all cables are properly connected. ➤ Ensure that the **Remote** option is selected and that all parameters are properly set.
This message is displayed on Windows desktop: RAID Volume Degraded.	Malfunctioning hard drive.	➤ Contact JGR.

Removing MS Manager Manually

It is possible that a software remains on the list of installed programs after attempting to remove it with the Add/Remove Programs utility (from the Windows Control Panel).

Before installing a new version of that software, you will have to remove it from the list with a special tool and to delete files from the hard disk.

IMPORTANT

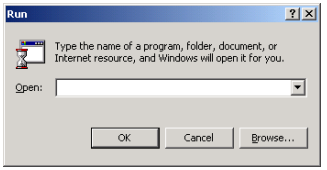
Before performing this procedure, you should always try to remove MS Manager with the Windows Add/Remove Programs utility.

Note: *You must log onto your MS Platform as a local administrator.*

To manually remove MS Manager:

1. Insert the installation CD into the CD-ROM drive.

2. On the Windows taskbar, click **Start** and select **Run**.



3. Use the **Browse** button to find the *Utilities|Windows Installer Clean-Up* folder on the CD-ROM.

4. Select *msicuu.exe and* click **OK**.

5. Select the software to be removed from the list and click **Remove**.

Note: *The msicuu.exe tool is only intended for removing a software name from the list. Therefore, it does not remove the software itself (no deletion of the associated files and registry keys).*

6. Once the operation is complete, close the msicuu.exe application.

7. Manually delete the installed files from the hard disk as folows:

7a. Go to *C:|Program Files|JGR|*.

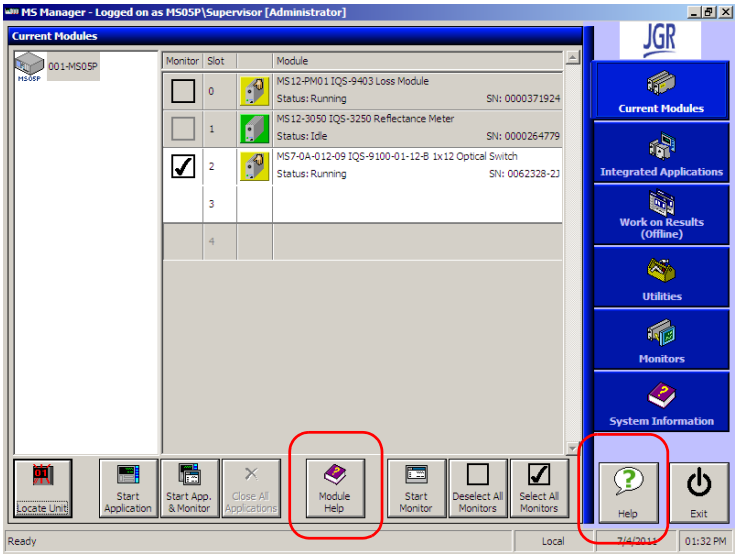
7b. Find the *MS Manager* folder and delete it (as well as all its content).

Viewing Online Documentation

An online version of the MS Platform user guide is available at all times from the application.

To access the online user guide:

Click **Help** in the function bar.



You can also open the online help of the current module by selecting the module and clicking **Module Help**.

Customer Support and Technical Assistance

All inquiries regarding service, calibration and technical assistance should be directed to the Customer Service department:

JGR Optics
160 Michael Cowpland Drive
Ottawa (Ontario) K2M 1P6
Tel: 613-599-1000
Fax: 613-599-1099

Tel.: 1 613 599-1000
Fax: 1 613 599-1099
info@jgroptics.com

Please be prepared to provide the part number, serial number, purchase order number, nature of problem, and ship-to address.

Viewing System Information

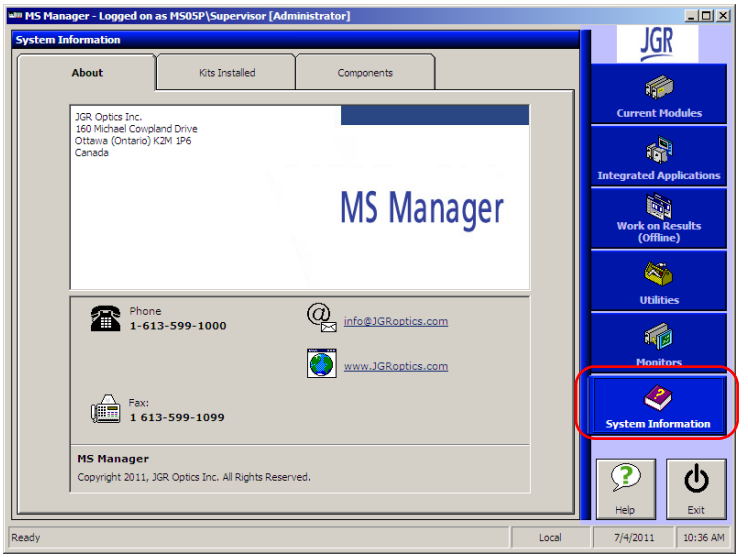
The **System Information** function tab contains important information about your system such as installed kits, components, and part number. To access the **System Information** function tab, click the corresponding tab on the right side of the screen.

Consulting the About Tab

The **About** tab contains information on how to contact JGR should you need to do so.

To access the About tab:

1. From the main window, select the **System Information** function tab.



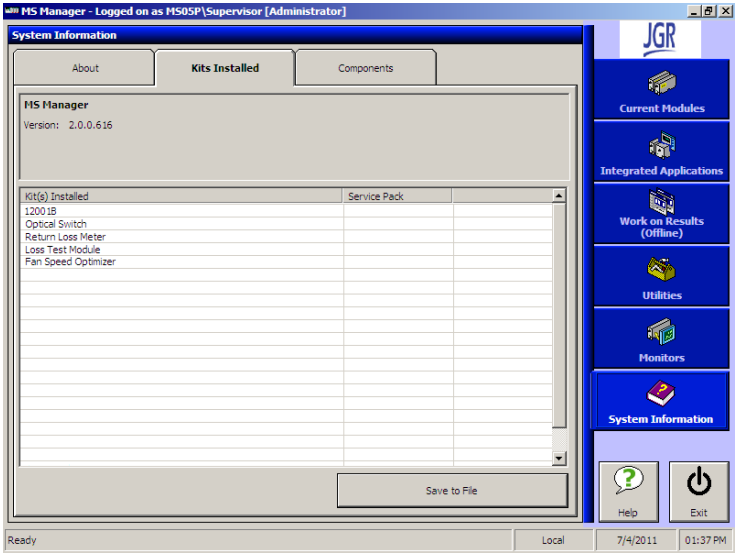
2. Select the **About** tab to view contact details.

Consulting the Kits Installed Tab

The **Kits Installed** tab allows you to view which kits are installed on your unit. Examples of kits are MS Manager itself or any integrated application installed on the unit.

To access the Kits Installed tab:

1. From the main window, select the **System Information** function tab.



2. Select **Kits Installed** to view which kits are installed on your unit.

The first column represents a description of every installed item, the second gives you the item version, and the third column gives you the serial number.

The last two columns are particularly important if you are contacting the Technical Support department, as the version and serial numbers helps to pinpoint and solve your problem more rapidly.

To save a text format (.TXT) of the installed kit list, click **Save to File**. The list is automatically saved as *JGR Install Kit List.TXT* in *D:\MS Manager*. Click **OK** to acknowledge and close the pop-up window.

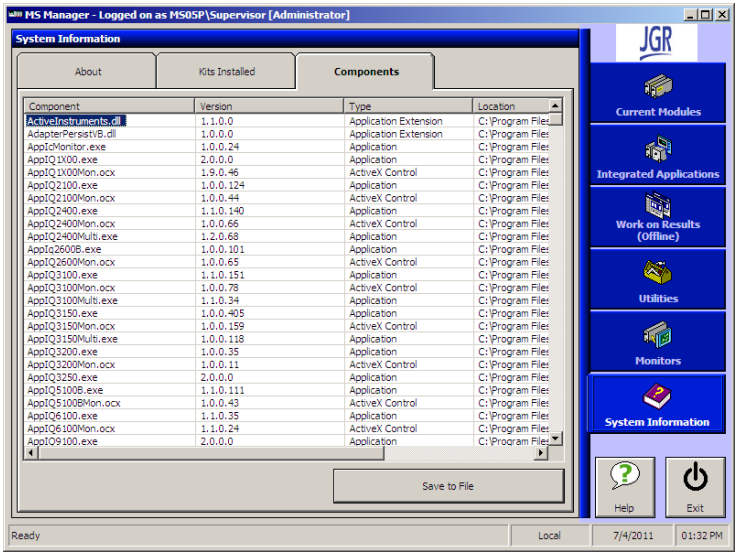
Note: *You cannot change the filename for this list. If you save the list a second time, the first list will be overwritten. To avoid this, you must go to* D:\MS Manager *and rename the first list.*

Consulting the Components Tab

The **Components** tab provides a list of all the applications installed on your unit.

To access the Components tab:

In the **System Information** window, select the **Components** function tab to see a list of the applications on your system.



You can also see which version of the applications you have, their type, and their location on your drive.

To save a text format (.TXT) of the component list, click **Save to File**. The list is automatically saved as *JGR Component List.TXT* on *D:\MS Manager*. Click **OK** to acknowledge the message and close the pop-up window.

Note: *You cannot change the filename for this list. If you save the list a second time, the first list will be overwritten. To avoid this, you must go to* D:\MS Manager *and rename the first list.*

Transportation

Maintain a temperature range within specifications when transporting the unit. Transportation damage can occur from improper handling. The following steps are recommended to minimize the possibility of damage:

- Pack the unit in its original packing material when shipping.
- Avoid high humidity or large temperature fluctuations.
- Keep the unit out of direct sunlight.
- Avoid unnecessary shocks and vibrations.

A SCPI-Based Errors

Error Number	Description	Probable Cause
–100	“Command error”	This is the generic syntax error for devices that cannot detect more specific errors. This code indicates only that a Command Error as defined in IEEE 488.2, 11.5.1.1.4 has occurred.
–101	“Invalid character”	A syntactic element contains a character which is invalid for that type; for example, a header containing an ampersand, SETUP&. This error might be used in place of errors –114, –121, –141, and perhaps some others.
–102	“Syntax error”	An unrecognized command or data type was encountered; for example, a string was received when the device does not accept strings.
–103	“Invalid separator”	The parser was expecting a separator and encountered an illegal character; for example, the semicolon was omitted after a program message unit, *EMC 1:CH1:VOLTS 5.
–104	“Data type error”	The parser recognized a data element different than one allowed; for example, numeric or string data was expected but block data was encountered.
–105	“GET not allowed”	A Group Execute Trigger was received within a program message (see IEEE 488.2, 7.7).
–108	“Parameter not allowed”	More parameters were received than expected for the header; for example, the *EMC common command only accepts one parameter, so receiving *EMC 0,1 is not allowed.
–109	“Missing parameter”	Fewer parameters were received than required for the header; for example, the *EMC common command requires one parameter, so receiving *EMC is not allowed.

Error Number	Description	Probable Cause
–110	“Command header error”	An error was detected in the header. This error message should be used when the device cannot detect the more specific errors described for errors –111 through –119.
–111	“Header separator error”	A character which is not a legal header separator was encountered while parsing the header; for example, no white space followed the header, thus *GMC"MACRO" is an error.
–112	“Program mnemonic too long”	The header contains more that twelve characters (see IEEE 488.2, 7.6.1.4.1).
–113	“Undefined header”	The header is syntactically correct, but it is undefined for this specific device; for example, *XYZ is not defined for any device.
–114	“Header suffix out of range”	The value of a numeric suffix attached to a program mnemonic, see Syntax and Style section 6.2.5.2, makes the header invalid.
–115	“Unexpected number of parameters”	The number of parameters received does not correspond to the number of parameters expected. This is typically due an inconsistency with the number of instruments in the selected group (see section on INSTrument:DEFine:GROup).
–120	“Numeric data error”	This error, as well as errors –121 through –129, are generated when parsing a data element which appears to be numeric, including the non-decimal numeric types.This particular error message should be used if the device cannot detect a more specific error.
–121	“Invalid character in number”	An invalid character for the data type being parsed was encountered; for example, an alpha in a decimal numeric or a “9” in octal data.

Error Number	Description	Probable Cause
–123	“Exponent too large”	The magnitude of the exponent was larger than 32000 (see IEEE 488.2, 7.7.2.4.1).
–124	“Too many digits”	The mantissa of a decimal numeric data element contained more than 255 digits excluding leading zeros (see IEEE 488.2, 7.7.2.4.1).
–128	“Numeric data not allowed”	A legal numeric data element was received, but the device does not accept one in this position for the header.
–130	“Suffix error”	This error, as well as errors –131 through –139, are generated when parsing a suffix. This particular error message should be used if the device cannot detect a more specific error.
–131	“Invalid suffix”	The suffix does not follow the syntax described in IEEE 488.2, 7.7.3.2, or the suffix is inappropriate for this device.
–134	“Suffix too long”	The suffix contained more than 12 characters (see IEEE 488.2, 7.7.3.4).
–138	“Suffix not allowed”	A suffix was encountered after a numeric element which does not allow suffixes.
–140	“Character data error”	This error, as well as errors –141 through –149, are generated when parsing a character data element. This particular error message should be used if the device cannot detect a more specific error.
–141	“Invalid character data”	Either the character data element contains an invalid character or the particular element received is not valid for the header.
–144	“Character data tool long”	The character data element contains more than twelve characters (see IEEE 488.2, 7.7.1.4).
–148	“Character data not allowed”	A legal character data element was encountered where prohibited by the device.

Error Number	Description	Probable Cause
–150	“String data error”	This error, as well as errors –151 through –159, are generated when parsing a string data element. This particular error message should be used if the device cannot detect a more specific error.
–151	“Invalid string data”	A string data element was expected, but was invalid for some reason (see IEEE 488.2, 7.7.5.2); for example, an END message was received before the terminal quote character.
–158	“String data not allowed”	A string data element was encountered but was not allowed by the device at this point in parsing.
–160	“Block data error	This error, as well as errors –161 through –169, are generated when parsing a block data element. This particular error message should be used if the device cannot detect a more specific error.
–161	“Invalid block data”	A block data element was expected, but was invalid for some reason (see IEEE 488.2, 7.7.6.2); for example, an END message was received before the length was satisfied.
–168	“Block data not allowed”	A legal block data element was encountered but was not allowed by the device at this point in parsing.
–170	“Expression error”	This error, as well as errors –171 through –179, are generated when parsing an expression data element. This particular error message should be used if the device cannot detect a more specific error.
–171	“Invalid expression”	The expression data element was invalid (see IEEE 488.2, 7.7.7.2); for example, unmatched parentheses or an illegal character.
–178	“Expression data not allowed”	A legal expression data was encountered but was not allowed by the device at this point in parsing.

Error Number	Description	Probable Cause
–180	“Macro error”	This error, as well as errors –181 through –189, are generated when defining a macro or executing a macro. This particular error message should be used if the device cannot detect a more specific error.
–181	“Invalid outside macro definition”	Indicates that a macro parameter placeholder ($<number) was encountered outside of a macro definition.
–183	“Invalid inside macro definition”	Indicates that the program message unit sequence, sent with a *DDT or *DMC command, is syntactically invalid (see IEEE 488.2, 10.7.6.3).
–184	“Macro parameter error”	Indicates that a command inside the macro definition had the wrong number or type of parameters.
–200	“Execution error”	This is the generic syntax error for devices that cannot detect more specific errors. This code indicates only that an Execution Error as defined in IEEE 488.2, 11.5.1.1.5 has occurred.
–201	“Invalid while in local”	Indicates that a command is not executable while the device is in local due to a hard local control (see IEEE 488.2, 5.6.1.5); for example, a device with a rotary switch receives a message which would change the switches state, but the device is in local so the message can not be executed.
–202	“Settings lost due to rtl”	Indicates that a setting associated with a hard local control (see IEEE 488.2, 5.6.1.5) was lost when the device changed to LOCS from REMS or to LWLS from RWLS.
–203	“Command protected”	Indicates that a legal password-protected program command or query could not be executed because the command was disabled.

Error Number	Description	Probable Cause
–210	“Trigger error”	-----
–211	“Trigger ignored”	Indicates that a GET, *TRG, or triggering signal was received and recognized by the device but was ignored because of device timing considerations; for example, the device was not ready to respond. Note: a DT0 device always ignores GET and treats *TRG as a Command Error.
–212	“Arm ignored”	Indicates that an arming signal was received and recognized by the device but was ignored.
–213	“Init ignored”	Indicates that a request for a measurement initiation was ignored as another measurement was already in progress.
–214	“Trigger deadlock”	Indicates that the trigger source for the initiation of a measurement is set to GET and subsequent measurement query is received. The measurement cannot be started until a GET is received, but the GET would cause an INTERRUPTED error.
–215	“Arm deadlock”	Indicates that the arm source for the initiation of a measurement is set to GET and subsequent measurement query is received. The measurement cannot be started until a GET is received, but the GET would cause an INTERRUPTED error.
–220	“Parameter error”	Indicates that a program data element related error occurred. This error message should be used when the device cannot detect the more specific errors described for errors –221 through –229.
–221	“Settings conflict”	Indicates that a legal program data element was parsed but could not be executed due to the current device state (see IEEE 488.2, 6.4.5.3 and 11.5.1.1.5.)

Error Number	Description	Probable Cause
–222	“Data out of range”	Indicates that a legal program data element was parsed but could not be executed because the interpreted value was outside the legal range as defined by the device (see IEEE 488.2, 11.5.1.1.5.)
–223	“Too much data”	Indicates that a legal program data element of block, expression, or string type was received that contained more data than the device could handle due to memory or related device-specific requirements.
–224	“Illegal parameter value”	Used where exact value, from a list of possible, was expected.
–225	“Out of memory”	The device has insufficient memory to perform the requested operation.
–226	“Lists not same length”	Attempted to use LIST structure having individual LIST's of unequal lengths.
–230	“Data corrupt or stale”	Possibly invalid data; new reading started but not completed since last access.
–231	“Data questionable”	Indicates that measurement accuracy is suspect.
–232	“Invalid format”	Indicates that a legal program data element was parsed but could not be executed because the data format or structure is inappropriate. For example when loading memory tables or when sending a SYSTem:SET parameter from an unknown instrument.

Error Number	Description	Probable Cause
–233	“Invalid version”	Indicates that a legal program data element was parsed but could not be executed because the version of the data is incorrect to the device. This particular error should be used when file or block data formats are recognized by the instrument but cannot be executed for reasons of version incompatibility. For example, a not supported file version, a not supported instrument version
–240	“Hardware error”	Indicates that a legal program command or query could not be executed because of a hardware problem in the device. Definition of what constitutes a hardware problem is completely device-specific. This error message should be used when the device cannot detect the more specific errors described for errors –241 through –249.
–241	“Hardware missing”	Indicates that a legal program command or query could not be executed because of missing device hardware; for example, an option was not installed. Definition of what constitutes missing hardware is completely device-specific.
–250	“Mass storage error”	Indicates that a mass storage error occurred. This error message should be used when the device cannot detect the more specific errors described for errors –251 through –259.
–251	“Missing mass storage”	Indicates that a legal program command or query could not be executed because of missing mass storage; for example, an option that was not installed. Definition of what constitutes missing mass storage is device-specific.
–252	“Missing media”	Indicates that a legal program command or query could not be executed because of a missing media; for example, no disk. The definition of what constitutes missing media is device-specific.

Error Number	Description	Probable Cause
–253	“Corrupt media”	Indicates that a legal program command or query could not be executed because of corrupt media; for example, bad disk or wrong format. The definition of what constitutes corrupt media is device-specific.
–254	“Media full”	Indicates that a legal program command or query could not be executed because the media was full; for example, there is no room on the disk. The definition of what constitutes a full media is device-specific.
–255	“Directory full”	Indicates that a legal program command or query could not be executed because the media directory was full. The definition of what constitutes a full media directory is device-specific.
–256	“File name not found”	Indicates that a legal program command or query could not be executed because the file name on the device media was not found; for example, an attempt was made to read or copy a nonexistent file. The definition of what constitutes a file not being found is device-specific.
–257	“File name error”	Indicates that a legal program command or query could not be executed because the file name on the device media was in error; for example, an attempt was made to copy to a duplicate file name. The definition of what constitutes a file name error is device-specific.
–258	“Media protected”	Indicates that a legal program command or query could not be executed because the media was protected; for example, the write-protect tab on a disk was present.The definition of what constitutes protected media is device-specific.

Error Number	Description	Probable Cause
–260	“Expression error”	[Indicates that a expression program data element related error occurred. This error message should be used when the device cannot detect the more specific errors described for errors –261 through –269.]
–261	“Math error in expression”	[Indicates that a syntactically legal expression program data element could not be executed due to a math error; for example, a divide-by-zero was attempted. The definition of math error is device-specific.]
–270	“Macro error”	[Indicates that a macro-related execution error occurred. This error message should be used when the device cannot detect the more specific errors described for errors –271 through –-279.]
–271	“Macro syntax error”	[Indicates that a syntactically legal macro program data sequence, according to IEEE 488.2, 10.7.2, could not be executed due to a syntax error within the macro definition (see IEEE 488.2, 10.7.6.3.)]
–272	“Macro execution error”	[Indicates that a syntactically legal macro program data sequence could not be executed due to some error in the macro definition (see IEEE 488.2, 10.7.6.3.)]
–273	“Illegal macro label”	[Indicates that the macro label defined in the *DMC command was a legal string syntax, but could not be accepted by the device (see IEEE 488.2, 10.7.3 and 10.7.6.2); for example, the label was too long, the same as a common command header, or contained invalid header syntax.]
–274	“Macro parameter error”	[Indicates that the macro definition improperly used a macro parameter placeholder (see IEEE 488.2, 10.7.3).]

Error Number	Description	Probable Cause
–275	“Macro definition too long”	[Indicates that a syntactically legal macro program data sequence could not be executed because the string or block contents were too long for the device to handle (see IEEE 488.2, 10.7.6.1).]
–276	“Macro recursion error”	[Indicates that a syntactically legal macro program data sequence could not be executed because the device found it to be recursive (see IEEE 488.2, 10.7.6.6).]
–277	“Macro redefinition not allowed”	[Indicates that a syntactically legal macro label in the *DMC command could not be executed because the macro label was already defined (see IEEE 488.2, 10.7.6.4).]
–278	“Macro header not found”	[Indicates that a syntactically legal macro label in the *GMC? query could not be executed because the header was not previously defined.]
–280	“Program error”	[Indicates that a downloaded program-related execution error occurred. This error message should be used when the device cannot detect the more specific errors described for errors –281 through –289. A downloaded program is used to add algorithmic capability to a device. The syntax used in the program and the mechanism for downloading a program is device-specific.]
–281	“Cannot create program”	[Indicates that an attempt to create a program was unsuccessful. A reason for the failure might include not enough memory.]
–282	“”Illegal program name	[The name used to reference a program was invalid; for example, redefining an existing program, deleting a nonexistent program, or in general, referencing a nonexistent program.]
–283	“Illegal variable name”	[An attempt was made to reference a nonexistent variable in a program.]

Error Number	Description	Probable Cause
–284	“Program currently running”	[Certain operations dealing with programs may be illegal while the program is running; for example, deleting a running program might not be possible.]
–285	“Program syntax error”	[Indicates that a syntax error appears in a downloaded program. The syntax used when parsing the downloaded program is device-specific.]
–286	“Program runtime error”	-----
–290	“Memory use error”	[Indicates that a user request has directly or indirectly caused an error related to memory or <data_handle>, this is not the same as “bad” memory.]
–291	“Out of memory”	-----
–292	“Referenced name does not exist”	-----
–293	“Referenced name already exist”	-----
–294	“Incompatible type”	[Indicates that the type or structure of a memory item is inadequate]
–300	“Device-specific error”	[This is the generic device-dependent error for devices that cannot detect more specific errors. This code indicates only that a Device-Dependent Error as defined in IEEE 488.2, 11.5.1.1.6 has occurred.]
–310	“System error”	[Indicates that some error, termed “system error” by the device, has occurred. This code is device-dependent.]
–311	“Memory error”	[Indicates some physical fault in the device's memory, such as parity error.]

Error Number	Description	Probable Cause
–312	“PUD memory lost”	[Indicates that the protected user data saved by the *PUD command has been lost.]
–313	“Calibration memory lost”	[Indicates that nonvolatile calibration data used by the *CAL? command has been lost.]
–314	“Save/Recall memory lost”	[Indicates that the nonvolatile data saved by the *SAV? command has been lost.]
–315	“Configuration memory lost”	[Indicates that nonvolatile configuration data saved by the device has been lost. The meaning of this error is device-specific.]
–320	“Storage fault”	[Indicates that the firmware detected a fault when using data storage. This error is not an indication of physical damage or failure of any mass storage element.]
–321	“Out of memory”	[An internal operation needed more memory than was available.]
–330	“Self-test failed”	-----
–340	“Calibration failed”	-----
–350	“Queue overflow”	[A specific code entered into the queue in lieu of the code that caused the error. This code indicates that there is no room in the queue and an error occurred but was not recorded.]
–360	“Communication error”	[This is the generic communication error for devices that cannot detect the more specific errors described for errors –361 through –363.]
–361	“Parity error in program message”	[Parity bit not correct when data received for example, on a serial port.]
–362	“Framing error in program message”	[A stop bit was not detected when data was received for example, on a serial port (for example, a baud rate mismatch).]

Error Number	Description	Probable Cause
–363	“Input buffer overrun”	[Software or hardware input buffer on serial port overflows with data caused by improper or nonexistent pacing.]
–365	“Time out error”	[This is a generic device-dependent error.]
–400	“Query error”	[This is the generic query error for devices that cannot detect more specific errors. This code indicates only that a Query Error as defined in IEEE 488.2, 11.5.1.1.7 and 6.3 has occurred.]
–410	“Query INTERRUPTED”	[Indicates that a condition causing an INTERRUPTED Query error occurred (see IEEE 488.2, 6.3.2.3); for example, a query followed by DAB or GET before a response was completely sent.]
–420	“Query UNTERMINATED”	[Indicates that a condition causing an UNTERMINATED Query error occurred (see IEEE 488.2, 6.3.2.2); for example, the device was addressed to talk and an incomplete program message was received.]
–430	“Query DEADLOCKED”	[Indicates that a condition causing an DEADLOCKED Query error occurred (see IEEE 488.2, 6.3.1.7); for example, both input buffer and output buffer are full and the device cannot continue.]
–440	“Query UNTERMINATED after indefinite response”	[Indicates that a query was received in the same program message after an query requesting an indefinite response was executed (see IEEE 488.2, 6.5.7.5).]
–500	“Power on”	[The instrument has detected an off to on transition in its power supply.]
–600	“User request”	[The instrument has detected the activation of a user request local control]

Error Number	Description	Probable Cause
–700	“Request control”	[The instrument requested to become the active IEEE 488.1 controller-in-charge]
–800	“Operation complete”	[The instrument has completed all selected pending operations in accordance with the IEEE 488.2, 12.5.2 synchronization protocol]

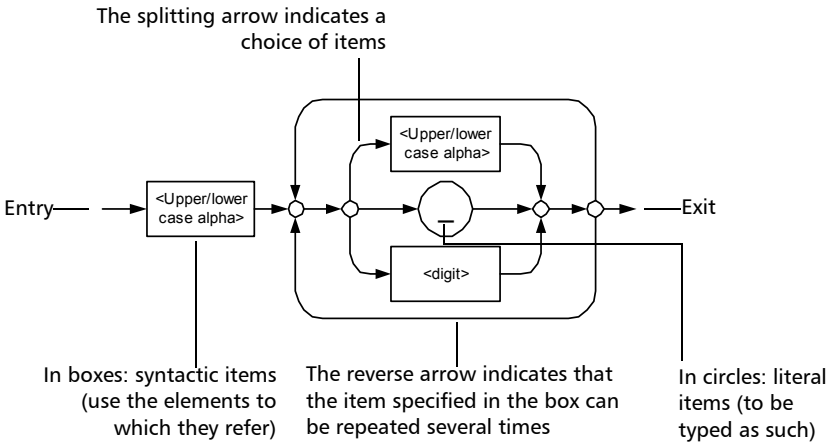
B Data Types

The following section provides an overview of the most common data types that may appear in JGR's documentation on commands and queries. The information is supplied for guidance only.

For more detailed information, please refer to IEEE 488.2 and SCPI standards. Additional reference sources are listed in *SCPI Command Structure* on page 155.

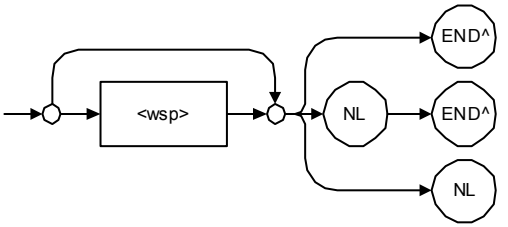
Data types are divided into two groups: <PROGRAM DATA> for the types that are used when you want to send messages to a device and <RESPONSE DATA> for the types that are used when a device sends responses to the controller unit.

The data types are presented in graphics often referred to as "railroad diagrams". The following example illustrates how to interpret such diagrams.



Applicable Data Types for Input—IEEE 488.2

➤ <PROGRAM MESSAGE TERMINATOR>



In the diagram above,

➤ "NL" corresponds to ASCII character code 10, in decimal (0A in binary)

➤ "END ^ " corresponds to the last data byte of the message sent with EOI = True and ATN = False

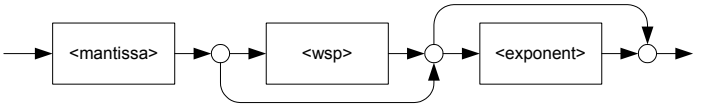
➤ <CHARACTER PROGRAM DATA>

This data type is used to send short mnemonics when a *<DECIMAL NUMERIC PROGRAM DATA>* cannot be used.

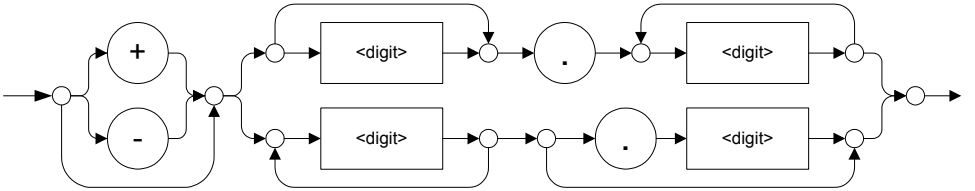
Examples: TRIANGLEWAVE, NCONTINUOUS

➤ <DECIMAL NUMERIC PROGRAM DATA> (or <NRf>)

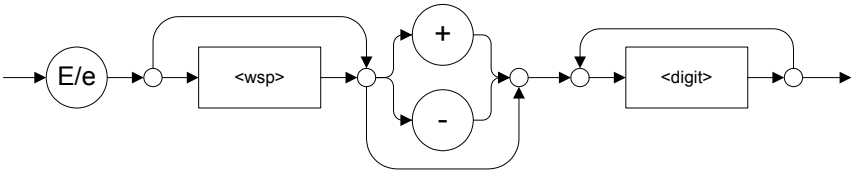
This data type includes <NR1>, <NR2> and <NR3> data types. It is used for decimal fractions with or without an exponent. Instruments adapts the values they receive to fit their degree of precision. For example, if an instrument has a precision of two digits after the decimal point and the incoming value is 12.048, this value is rounded off to 12.05.



The second diagram below illustrates the <mantissa> syntax.



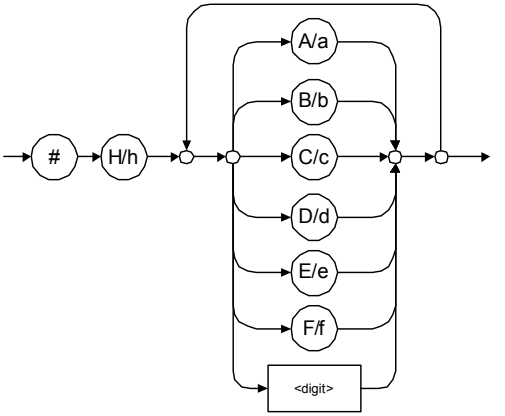
The third diagram illustrates the <exponent> syntax.



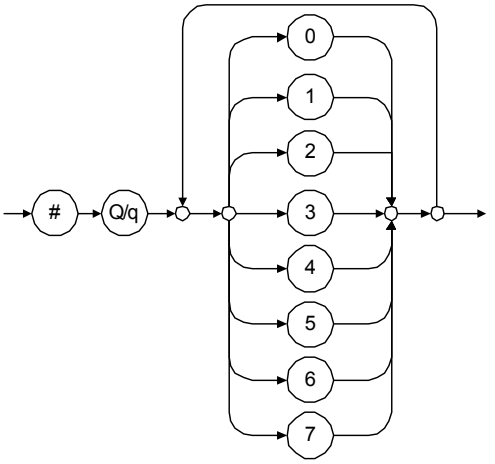
Examples: +2.0 e5, –.56E+4, 6.5e–10

➤ <NON-DECIMAL NUMERIC PROGRAM DATA>

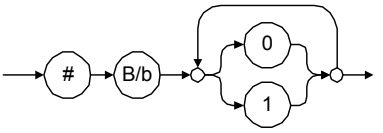
This data type is used for integer representation in hexadecimal (base 16), octal (base 8) or binary (base 2). The numeric representations begins with “#H” for hexadecimal, “#Q” for octal and “#B” for binary.



Examples: #Hf3bc015d, #h01a4, #hfe



Examples: #Q1234567, #q1275, #q07

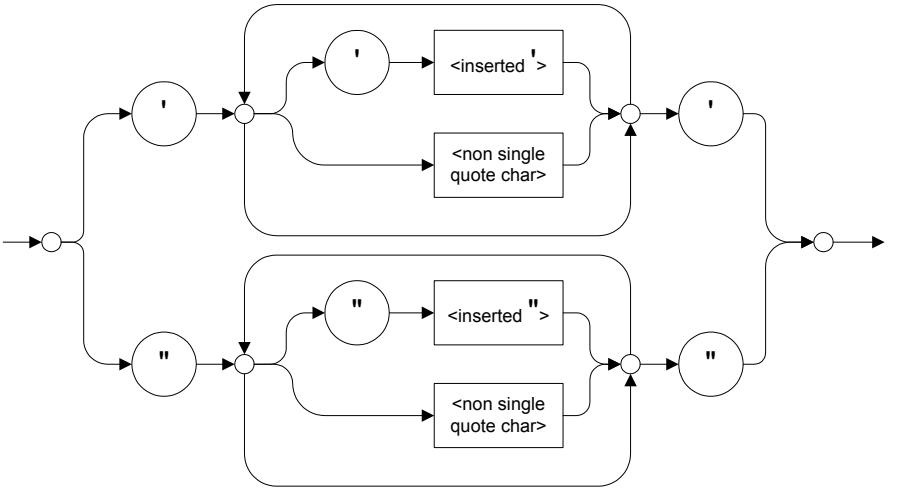


Examples: #B10010111, #b10110, #b1100

➤ <STRING PROGRAM DATA>

This data type is used for strings containing 7-bit ASCII characters that have to be enclosed in either single- or double-quotes delimiters.

If a string needs to contain a character that is exactly the same as the delimiter, make sure to double the character to avoid syntax errors.



Examples: "SCPI Commands", 'SCPI Commands', "SCPI 'Commands'", 'SCPI "Commands"', "SCPI ""Commands""", 'SCPI ''Commands'''

➤ <ARBITRARY BLOCK PROGRAM DATA>

This data type is used to send blocks of arbitrary 8-bit information when you need to work with large amounts of data.

The actual length of the data that you send has the following structure:

➤ The first byte contains the # character.

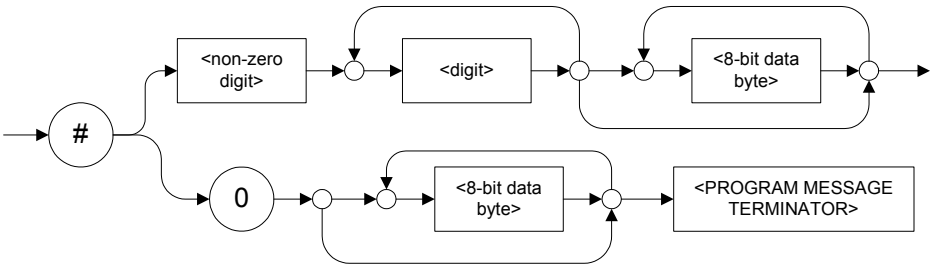
➤ The byte that immediately follows contains the number of subsequent bytes that you have to check to obtain the total length.

Note: *If you use a zero as the first digit (#0), it has to be followed by a <PROGRAM MESSAGE TERMINATOR> to allow the device to detect the end of the <ARBITRARY BLOCK PROGRAM DATA>. This also triggers immediate termination of the message.*

For example, if you send the following data (here, values are expressed in decimal instead of binary for easier readability):

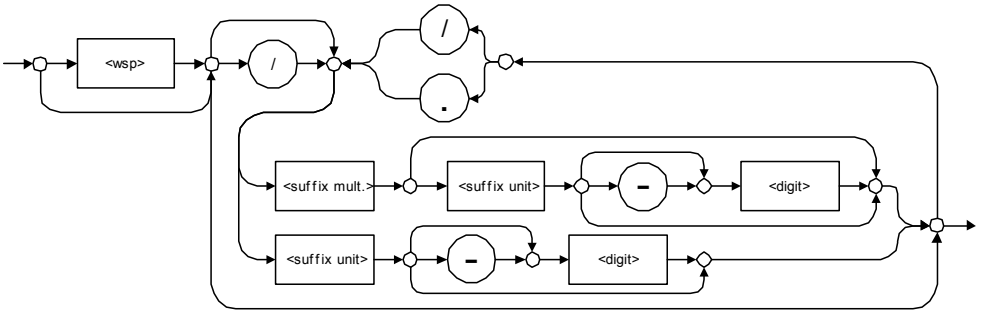
2 1 3 7 6 8 9 2 ...

The byte that immediately follows the # contains 2, which means that you would have to read the two following bytes to know the length (in bytes) of the retrieved data. The bytes indicate 1 and 3. The length is then 13 bytes. The actual response begins at byte number 5, in this case.



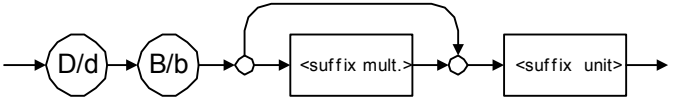
➤ <SUFFIX PROGRAM DATA>

This data type is used when units and multipliers have to be sent.



Examples: nm, kHz, km/s2, uW

A relative unit (dB) can be referenced to an absolute level, as shown on the following diagram.



Examples: db, dbm, dBW

The table below illustrates the possible forms for <suffix mult.>:

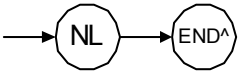
Name	Value	Mnemonic
Exa	1E18	EX
Peta	1E15	PE
Tera	1E12	T
Giga	1E9	G
Mega	1E6	MA
Kilo	1E3	K
Milli	1E-3	M
Micro	1E-6	U
Nano	1E-9	N
Pico	1E-12	P
Femto	1E-15	F
Atto	1E-18	A

The table below gives the possible forms for <suffix unit>:

Reference Unit	Suffix Unit
Degrees	DEG
Radians	RAD
Amperes	A
Volts	V
Hertz	HZ
Meters	M
Watts	W
DBs ref to 1mW	DBM
Decibels	DB
Degrees Celsius	CEL
Degrees Fahrenheit	FAR
Kelvins	K
Seconds	S
Hours	HR
Minutes	MIN

Applicable Data Types for Output —IEEE 488.2

- <RESPONSE MESSAGE TERMINATOR>



In the diagram above,

- "NL" corresponds to ASCII character code 10, in decimal (0A in binary)
- "END ^ " corresponds to the last data byte of the message sent with EOI = True and ATN = False

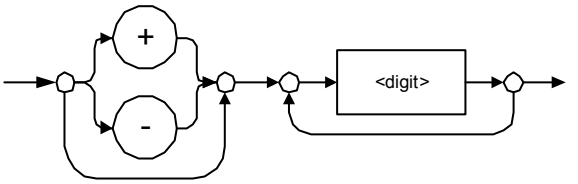
- <CHARACTER RESPONSE DATA>

This data type is used by a device to return short mnemonics when a *<DECIMAL NUMERIC PROGRAM DATA>* cannot be used. The returned information is sent in the long form and in upper case.

Examples: TRIANGLEWAVE, NCONTINUOUS

➤ <NR1 NUMERIC RESPONSE DATA> (or <NR1>)

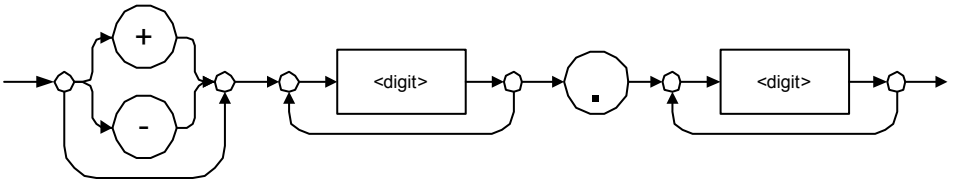
This data type is used by a device to return positive or negative integers.



Examples: 4, –23, 90

➤ <NR2 NUMERIC RESPONSE DATA> (or <NR2>)

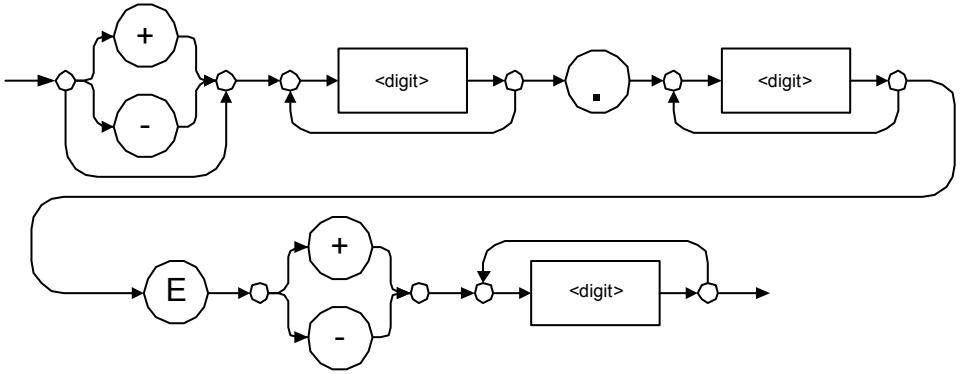
This data type is used by a device to return positive or negative real numbers (fixed-point numbers).



Examples: 23.45, 1.22, –4.55

➤ <NR3 NUMERIC RESPONSE DATA> (or <NR3>)

This data type is used by a device to return positive or negative exponential numbers (floating-point numbers).



Examples: 4.3E–3, –8.9456E8, 123E–5

➤ Special Numeric Values Received on Output

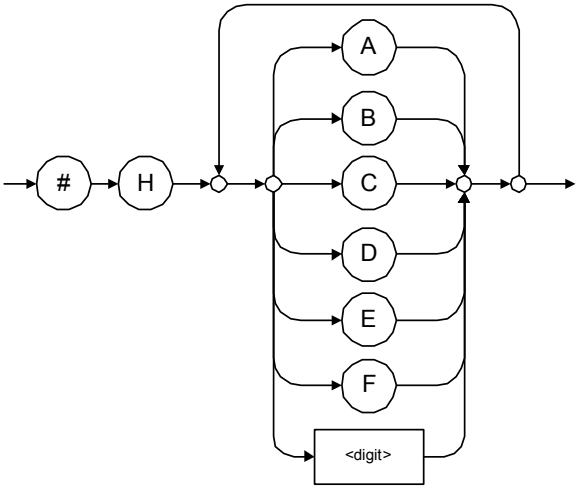
In some cases, an instrument may send values indicating that an unusual event has occurred. The following tables present the possible values.

Value is	ASCII 4 bytes	PACKED 4 bytes
Under range	2143289345.000000	7FC00001
Over range	2143289346.000000	7FC00002
Invalid	2143289347.000000	7FC00003
Inactive	2143289348.000000	7FC00004

Value is	ASCII 8 bytes	PACKED 8 bytes
Under range	9221120237577961472	7FF8000020000000
Over range	9221120238114832384	7FF8000040000000
Invalid	9221120238651703296	7FF8000060000000
Inactive	9221120239188574208	7FF8000080000000

➤ <HEXADECIMAL NUMERIC RESPONSE DATA>

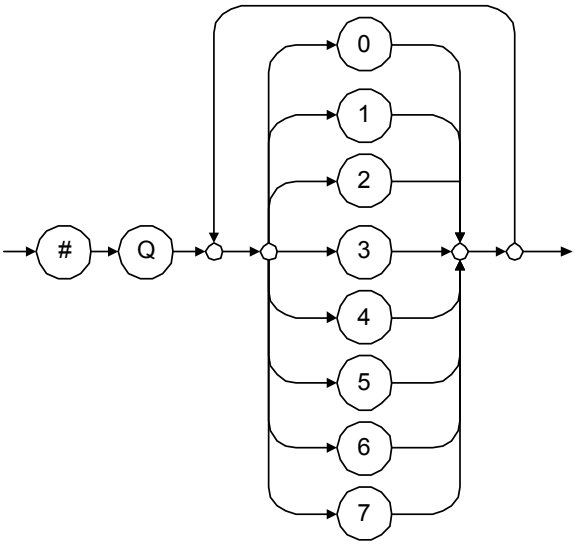
This data type is used by a device to return integer representations in hexadecimal (base 16).



Examples: #HA3C5, #H0123C, #H010F

➤ <OCTAL NUMERIC RESPONSE DATA>

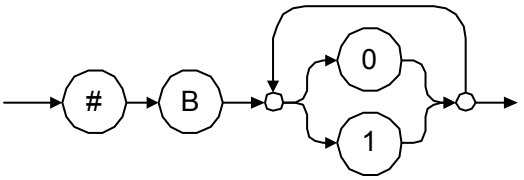
This data type is used by a device to return integer representations in octal (base 8).



Examples: #Q753214, #Q0124, #Q0725

➤ <BINARY NUMERIC RESPONSE DATA>

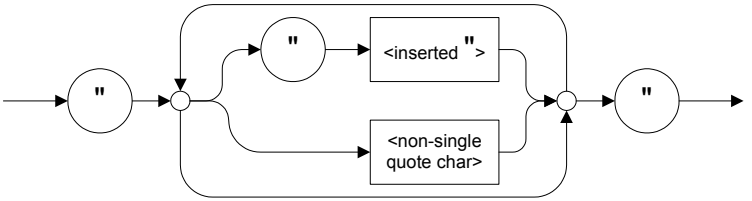
This data type is used by a device to return integer representations in binary (base 2).



Examples: #B11011110101, #B110100, #B0100

➤ <STRING RESPONSE DATA>

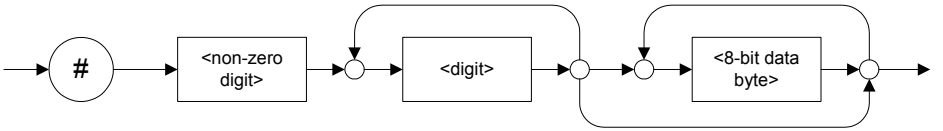
This data type is used by a device to return strings containing 7-bit ASCII characters and especially when text has to be displayed since even the non-printable characters are also returned.



Examples: "SCPI Commands", "SCPI ""Commands"""

➤ <DEFINITE LENGTH ARBITRARY BLOCK RESPONSE DATA>

This data type is used by a device to return blocks of 8-bit binary information with a fixed and predetermined length.



The actual length of the retrieved data has the following structure:

➤ The first byte contains the # character.

➤ The byte that immediately follows contains the number of subsequent bytes that you have to check to know the total length.

For example, if you receive this response (here, values are expressed in decimal instead of binary for easier readability):

2 1 3 7 6 8 9 2 ...

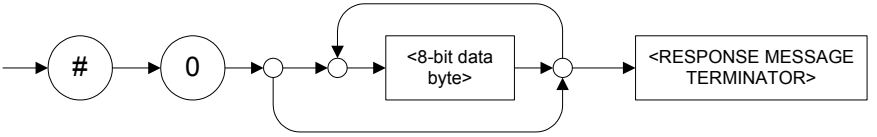
The byte that immediately follows the # contains 2, which means that you have to read the two following bytes to know the length (in bytes) of the retrieved data. The bytes indicate 1 and 3. The length is then be 13 bytes. The actual response begins at byte number 5, in this case.

Examples: #14<DAB><DAB><DAB><DAB>,
#3004<DAB><DAB><DAB><DAB>

where "<DAB>" stands for data byte

➤ <INDEFINITE LENGTH ARBITRARY BLOCK RESPONSE DATA>

This data type is used by a device to return blocks of 8-bit binary information when the block length was not predefined or when data has to be computed later.

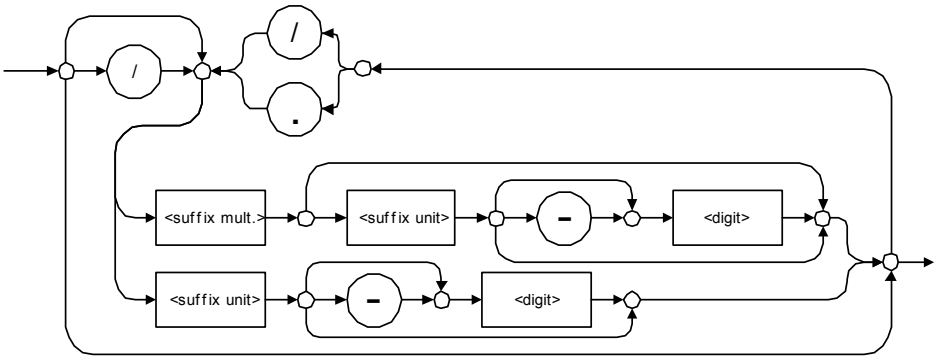


Note: *If you receive a zero as the first digit (#0), it is necessarily followed by a <RESPONSE PROGRAM MESSAGE TERMINATOR> to allow you to detect the end of the <INDEFINITE LENGTH ARBITRARY BLOCK RESPONSE DATA>.*

Example: #0<DAB><DAB><DAB><DAB><terminator> where "<DAB>" stands for data byte.

➤ <SUFFIX RESPONSE DATA>

This data type is used by a device to return units and multipliers.

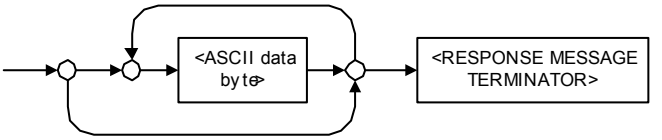


Examples: DBW, W, KHZ

➤ <ARBITRARY ASCII RESPONSE DATA>

This data type is used by a device to return information when it is impossible to use any other data type.

Example: To the *IDN? query, the device returns this response in an arbitrary ASCII bytes format:
JGR Optics,MS10P,0123456789,2.1.0.15



Applicable Data Types for Input—SCPI

SCPI data types include the IEEE 488.2 data types (see *Applicable Data Types for Input—IEEE 488.2* on page 202) with certain additional restrictions.

- <numeric_value>: abbreviated form of the decimal numeric element. It differs from the <DECIMAL NUMERIC PROGRAM DATA> "<NRf>" described in IEEE 488.2.

  Several forms of <CHARACTER PROGRAM DATA> are defined as special forms of numbers. These are: MINimum, MAXimum, DEFault, UP, DOWN, Not A Number (NAN), INFinity and Negative INFinity (NINF). The following special forms are likely to be used by JGR's instruments in certain commands or queries:

  - DEFault: This special <numeric_value> parameter forces the instrument to select a value, which is deemed to be convenient to the user.
  - MINimum|MAXimum: These special <numeric_value> parameters refer to the instrument's limit values. MINimum corresponds to the value closest to negative infinity that the function can accept. MAXimum corresponds to the largest value that the function can accept.
  - <Boolean Program Data>: This form is often used as a shorthand of the <DECIMAL NUMERIC PROGRAM DATA>ON|OFF form.

    <Boolean Program Data> parameters have a value of 0 or 1 and are not followed by any unit.

    On input, an <NRf> is rounded to an integer.

    A non-zero result is interpreted as 1.

    ON and OFF are accepted on input for readability purposes. They correspond respectively to 1 and 0. However, on output, they appear as 1 or 0, never ON or OFF.

Special Numeric Values Received on Output

It is possible that an instrument returns unusual values in certain cases. For information on these values, see Applicable Data Types for Output —IEEE 488.2 *on page 211*.

C IEEE 488.2 and Specific Commands

This chapter presents detailed information about the commands and queries supplied with your MS Platform.

IEEE 488.2 Commands–Quick Reference

The MS Platform recognizes the required commands identified in IEEE 488.2. The table below summarizes these commands. These commands are fully explained on the following pages.

Command	Function
*CLS	Clear status command (See **CLS* on page 224)
*ESE	Standard event status enable command (See **ESE* on page 225)
*ESE?	Standard event status enable query (See **ESE?* on page 227)
*ESR?	Standard event status register query (See **ESR?* on page 229)
*IDN?	Identification query (See **IDN?* on page 231)
*OPC	Operation complete command (See **OPC* on page 233)
*OPC?	Operation complete query (See **OPC?* on page 234)
*RST	Reset command (See **RST* on page 235)

Command	Function
*SRE	Service request enable command (See **SRE* on page 236)
*SRE?	Service request enable query (See **SRE?* on page 238)
*STB?	Read status byte query (See **STB?* on page 240)
*TST?	Self-test query (See **TST?* on page 242)
*WAI	Wait for pending operations to be completed (See **WAI* on page 243)

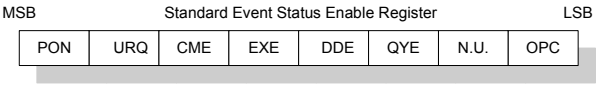
IEEE 488.2 Required Commands

*CLS	
Description	The *CLS command clears the Standard Event Status Register and the Error/Event Queue.
Syntax	*CLS
Parameter(s)	None

*ESE

Description

The *ESE command sets the Standard Event Status Enable Register bits, as defined in the table below. This register contains a mask value for the bits to be enabled in the Standard Event Status Register.



Syntax

*ESE<wsp><RegisterValue>

Parameter(s)

RegisterValue:

The program data syntax for <RegisterValue> is defined as a <DECIMAL NUMERIC PROGRAM DATA> element.

The <RegisterValue>, expressed in base 2, represents the bit values of the Standard Event Status Enable Register.

*ESE

The table below shows the contents of this register.

Bit	Weight	Meaning
PON	128	Turn On Enable
URQ	64	User ReQuest Enable
CMD	32	CoMmanD Error Enable
EXE	16	Execution Error Enable
DDE	8	Device Dependent Error Enable
QRY	4	QueRry Error Enable
N.U.	2	Not used
OPC	1	Operation Complete Enable

A value of 1 in the Enable Register enables the corresponding bit in the Status Register, a value of 0 disables the bit. The value of the <RegisterValue> shall be in the range of 0 through 255.

Example(s)

*ESE 25
where 25 = (bit EXE, bit DDE and bit OPC)

*ESE 0
clears the content of the Standard Event Status Enable register

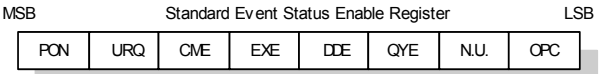
See Also

*ESE?
*ESR?

*ESE?

Description

The *ESE? query allows the programmer to determine the current contents of the Standard Event Status Enable Register. See the contents of this register below.



Syntax

*ESE?

Parameter(s)

None

Response Syntax

<RegisterValue>

*ESE?

Response(s)

RegisterValue:

The response data syntax for <RegisterValue> is defined as a <NR1 NUMERIC RESPONSE DATA> element.

The <RegisterValue> ranges from 0 through 255.

The <RegisterValue> value expressed in base 2 (binary) represents the bit values of the Standard Event Status Enable register. See below.

Bit	Weight	Meaning
PON	128	Turn On Enable
URQ	64	User ReQuest Enable
CMD	32	CoMmanD Error Enable
EXE	16	Execution Error Enable
DDE	8	Device Dependent Error Enable
QRY	4	QueRry Error Enable
N.U.	2	Not used
OPC	1	Operation Complete Enable

Example(s)

*ESE? returns 133
where 133 = (bit PON, bit QYE and bit OPC)

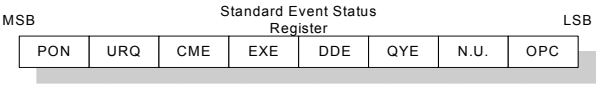
See Also

*ESE
*ESR?

*ESR?

Description

The *ESR? query allows the programmer to determine the current contents of the Standard Event Status Register. Reading the Standard Event Status Register clears it. See the contents of this register below.



Syntax

*ESR?

Parameter(s)

None

Response Syntax

<RegisterValue>

*ESR?

Response(s)

RegisterValue:

The response data syntax for <RegisterValue> is defined as a <NR1 NUMERIC RESPONSE DATA> element.

The <RegisterValue> ranges from 0 through 255.

The <RegisterValue> value expressed in base 2 (binary) represents the bit values of the Standard Event Status register. See below.

Bit	Weight	Meaning
PON	128	Power ON Enable
URQ	64	User ReQuest Enable
CMD	32	CoMmanD Error Enable
EXE	16	Execution Error Enable
DDE	8	Device Dependent Error Enable
QRY	4	QueRry Error Enable
N.U.	2	Not used
OPC	1	Operation Complete Enable

Example(s)

*ESR? returns 33
where 33 = (bit CME and bit OPC)

See Also

*ESE
*ESE?

*IDN?

Description	The intent of the *IDN? query is for the unique identification of devices over the system interface.
Syntax	*IDN?
Parameter(s)	None
Response Syntax	<Identification>
Response(s)	*Identification:* The response data syntax for <Identification> is defined as an <ARBITRARY ASCII RESPONSE DATA> element. The response syntax for the *IDN? query, <Identification> is defined as an <ARBITRARY ASCII RESPONSE DATA> element. This implies that the *IDN? query should be the last <QUERY MESSAGE UNIT> in a <TERMINATED PROGRAM MESSAGE>. The response is organized into four fields separated by commas. The field definitions are as follows: Field 1 (Manufacturer): JGR Optics Field 2 (Model): Instrument Model

*IDN?	
	Field 3 (Serial number): ASCII character (0 if not available) Field 4 (Firmware level): ASCII character (0 if not available) ASCII character 0 represents a single ASCII-encoded byte with a value of 30 (48 decimal). The presence of data in all fields is mandatory. If either field 3 or 4 is not available, the ASCII character 0 shall be returned for that field. A field may contain any 7-bit ASCII-encoded bytes in the range of 20 through 7E (32 through 126 decimal) except commas (2C, 44 decimal) and semicolons (3B, 59 decimal).
Example(s)	*IDN? returns JGR Optics,MS10P,0123456789,2.1.0.15
Notes	The overall length of the *IDN? response is less than or equal to 72 characters.

*OPC

Description	The *OPC command allows synchronization between the instrument and an external controller unit. The *OPC command causes the instrument to set bit 0 (Operation Complete) in the Standard Event Status Register to the TRUE (logic 1) state when the instrument completes all pending operations. Detection of the Operation Complete message can be accomplished by continuous polling of the Standard Event Status Register using the *ESR? common query command. However, using a service request eliminates the need to poll the Standard Event Status Register thereby freeing the controller unit to do other useful work.
Syntax	*OPC
Parameter(s)	None
See Also	*OPC? *WAI

*OPC?

Description	The *OPC? query allows synchronization between the instrument and an external controller unit by reading the Output Queue or by waiting for a service request on the Message Available (MAV) bit in the Status Byte Register. The *OPC? query causes the instrument to place an ASCII character, 1, into its Output Queue when the device completes all pending operations. A consequence of this action is that the MAV bit in the Status Byte Register is set to state 1.
Syntax	*OPC?
Parameter(s)	None
Response Syntax	<Acknowledge>
Response(s)	*Acknowledge:* The response data syntax for <Acknowledge> is defined as a <NR1 NUMERIC RESPONSE DATA> element. The <Acknowledge> response is a single ASCII-encoded byte corresponding to 1. The receipt of an <Acknowledge> response indicates that all pending selected device operations have been completed.
Example(s)	*OPC? Return 1
See Also	*OPC *WAI

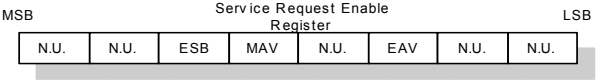
*RST

Description	The *RST command performs a device reset. This command is the third reset level in a three-level reset strategy. The Reset command shall do the following: a) Sets the device-specific functions to a known state that is independent of the past-use history of the device. b) Forces the device into OCIS state (Operation complete Command Idle State). c) Forces the device into OQIS state (Operation complete Query Idle State). The Reset command explicitly DOES NOT affect the following: a) The state of the Communication interface. b) The Output Queue. c) Any Event Enable Register setting, including the Standard Event Status Enable Register setting. d) Any Event Register setting, including the Standard Event Status Register settings. e) Calibration data that affects device specifications. f) The Service Request Enable Register setting.
Syntax	*RST
Parameter(s)	None

*SRE

Description

The *SRE command sets the Service Request Enable Register bits. See the contents of this register below. This register contains a mask value to enable the bits in the Status Byte Register.



Syntax

*SRE<wsp><RegisterValue>

Parameter(s)

RegisterValue:

The program data syntax for <RegisterValue> is defined as a <DECIMAL NUMERIC PROGRAM DATA> element.

The <RegisterValue> value ranges from 0 through 255.

The <RegisterValue>, expressed in base 2 (binary), represents the bit values of the Service Request Enable Register.

*SRE

See the contents of this register below.

Bit	Weight	Meaning
N.U.	128	Not used
N.U.	64	Not used
ESB	32	Event Summary Bit Enable
MAV	16	Message AVailable Enable
N.U.	8	Not used
EAV	4	Error / Event AVailable Enable
N.U.	2	Not used
N.U.	1	Not used

A bit value of zero shall indicate a disabled condition.

Example(s)

*SRE 52
where 52 = (bit ESB, bit MAV and bit EAV)

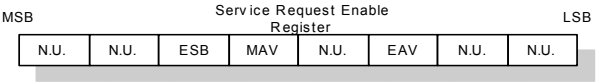
See Also

*SRE?
*STB?

*SRE?

Description

The *SRE? query allows the programmer to determine the current contents of the Service Request Enable Register. See the contents of this register below.



Bit	Weight	Meaning
N.U.	128	Not used
N.U.	64	Not used
ESB	32	Event Summary Bit Enable
MAV	16	Message AVailable Enable
N.U.	8	Not used
EAV	4	Error / Event AVailable Enable
N.U.	2	Not used
N.U.	1	Not used

Syntax *SRE?

Parameter(s) None

Response Syntax <RegisterValue>

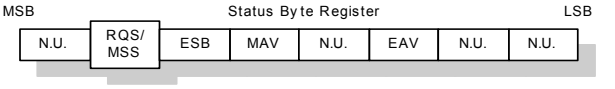
*SRE?

Response(s)	*RegisterValue:* The response data syntax for <RegisterValue> is defined as a <NR1 NUMERIC RESPONSE DATA> element. The <RegisterValue> ranges from 0 through 255. When converted to binary (base 2), the <RegisterValue> represents the current bit values of the Service Request Enable Register.
Example(s)	*SRE Return 32 (bit ESB)
See Also	*SRE *STB?

*STB?

Description

The *STB? query allows the programmer to read the status byte and Master Summary Status bit. See the content of this register below.



Syntax

*STB?

Parameter(s)

None

Response Syntax

<RegisterValue>

*STB?

Response(s)

RegisterValue:

The response data syntax for <RegisterValue> is defined as a <NR1 NUMERIC RESPONSE DATA> element.

The <RegisterValue> ranges from 0 through 255.

The <RegisterValue> value, expressed in base 2 (binary) represents the bit values of the Status Byte Register. See the contents of this register below.

Bit	Weight	Meaning
N.U.	128	Not used
RQS/ MSS	64	ReQuest Service (read by serial polling)/MaSter Summary bit (read by *STB?)
ESB	32	Event Summary Bit Enable
MAV	16	Message AVailable Enable
N.U.	8	Not used
EAV	4	Error / Event AVailable Enable
N.U.	2	Not used
N.U.	1	Not used

Example(s)

*STB? Return 68
where 68 = (bit MSS and bit EAV)

See Also

*SRE
*SRE?

	*TST?
Description	The *TST? query causes an internal self-test and places a response into the Output Queue indicating whether or not the device completed the self-test without any detected errors. Upon successful completion of *TST?, the device settings is restored to their values prior to the *TST?.
Syntax	*TST?
Parameter(s)	None
Response Syntax	<Result>
Response(s)	*Result:* The response data syntax for <Result> is defined as a <NR1 NUMERIC RESPONSE DATA> element. The <Result> value ranges from -32767 through +32767. A <Result> with a value of zero indicates that the self-test has been completed without errors detected. A <Result> with a value not equal to zero indicates that the self-test was not completed or was completed with errors detected.
Example(s)	*TST? Return 0 (self-test was completed with success)

*WAI

Description	The *WAI command shall prevent the device from executing any further commands or queries until the no-operation-pending flag becomes TRUE.
Syntax	*WAI
Parameter(s)	None
Example(s)	*WAI
See Also	*OPC *OPC?

Specific Commands—Quick Reference

The table below contains a summary of the MS Platform specific commands. These commands are fully explained on the following pages.

Command						Parameter(s)
FORMat	[DATA]					ASCii|PACKed[,<Length>]
	[DATA]?					
INSTrument	CATalog?					
	CATalog	FULL?				
SYSTem	DATE					<Year>,<Month>,<Day>
	DATE?					
	ERRor	[NEXT]?				
	TIME					<Hour>,<Minute>,<Seconds>
	TIME?					
	VERSion?					

Specific Commands

:FORMat[:DATA]

Description

The FORMat[:DATA] command selects the data format and <Length>. The <Length> parameter is optional for all data format, its meaning is dependent on the data format selected.

If PACKed type is selected, the data is transferred in a <DEFINITE BLOCK RESPONSE DATA>. The ASCii-type data is automatically identified by its syntax. Therefore, in these cases, the FORMat subsystem is only necessary to determine the output format.

At *RST, ASCii is selected as the default data format and the <Length> is set to 0.

Syntax

:FORMat[:DATA]<wsp>ASCii|PACKed[,<Length>]

Parameter(s)

➤ *Type:*

The program data syntax for the first parameter is defined as a <CHARACTER PROGRAM DATA> element. The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: ASCii|PACKed.

:FORMat[:DATA]

In ASCii format, the numeric data is transferred to ASCii bytes in <NR1 NUMERIC RESPONSE DATA>, <NR2 NUMERIC RESPONSE DATA> or <NR3 NUMERIC RESPONSE DATA> representation, as appropriate.

In PACKed format, data is transferred to a <DEFINITE BLOCK RESPONSE DATA>, in a manner specified in the device documentation.

➤ *Length:*

The program data syntax for <Length> is defined as a <DECIMAL NUMERIC PROGRAM DATA> element.

When ASCii data format is selected, the optional <Length> parameter specifies the number of significant digits to be returned. A <Length> value of zero indicates that the device selects the number of significant digits to be returned. When a <Length> of zero has been specified, the FORMat[:DATA]? query returns zero as its second parameter.

When the PACKed data format is selected, the optional parameter <Length> it not used.

Example(s)

FORM ASC
FORM ASC,6
FORM:DATA PACKED

See Also

FORMat[:DATA]?

:FORMat[:DATA]?

Description

The FORMat[:DATA]? query returns the data format and the <Length>.

At *RST, ASCii is selected as the default data format and the <Length> is set to 0.

Syntax

:FORMat[:DATA]?

Parameter(s)

None

Response Syntax

<Type>,<Length>

Response(s)

➤ *Type:*

The response data syntax for <Type> is defined as a <CHARACTER RESPONSE DATA> element.

The ASCII <Type> is returned when numeric data is transferred to ASCII bytes in <NR1 NUMERIC RESPONSE DATA>, <NR2 NUMERIC RESPONSE DATA> or <NR3 NUMERIC RESPONSE DATA> representation, as appropriate.

The PACKED <Type> is returned when data is transferred to a <DEFINITE BLOCK RESPONSE DATA>, as specified in the device documentation.

➤ *Length:*

The response data syntax for <Length> is defined as a <NR1 NUMERIC RESPONSE DATA> element.

:FORMat[:DATA]?

	When the data is returned in ASCII, the <Length> is the number of significant digits to be returned. A <Length> value of zero indicates that the device selects the number of significant digits to be returned. When the data is returned in PACKED <Type>, the <Length> is not used and always returns 0.
Example(s)	FORM? Return ASCII,6 FORM? Return PACKED,0
See Also	FORMat[:DATA]?

:INSTrument:CATalog?

Description	The INSTrument:CATalog? query returns a comma-separated list of <STRING RESPONSE DATA>, which contains the names of all logical instruments and groups. If no logical instruments are defined, a single null <STRING RESPONSE DATA> is returned. This is not affected by a *RST command.
Syntax	:INSTrument:CATalog?
Parameter(s)	None
Response Syntax	<Catalog>
Response(s)	*Catalog:* The response data syntax for <Catalog> is defined as a <STRING RESPONSE DATA> element. The list of <STRING PROGRAM DATA> contains the names of all logical instruments and groups.
Example(s)	INST:CAT? returns “MS12-PM01 Loss Module”,”MS7-0A-012-09 1x12 Optical Switch”
See Also	INSTrument:CATalog:FULL?

:INSTrument:CATalog:FULL?

Description	The INSTrument:CATalog:FULL? returns a list of <STRING RESPONSE DATA> - <NR1 NUMERIC RESPONSE DATA> pairs. The <STRING RESPONSE DATA> contains the name of the logical instrument. The immediately following <NR1 NUMERIC RESPONSE DATA> formatted number is its associated logical instrument number. All response data elements are separated by commas. If no logical instrument is defined, a null <STRING RESPONSE DATA> value followed by a zero is returned. This is not affected by a *RST command.
Syntax	:INSTrument:CATalog:FULL?
Parameter(s)	None
Response Syntax	<Catalog>
Response(s)	*Catalog:* The response data syntax for <Catalog> is defined as a <STRING RESPONSE DATA> element. The list of <STRING RESPONSE DATA> contains the names of all logical instruments and groups. The immediately following <NR1 NUMERIC RESPONSE DATA> formatted number is its associated logical instrument number.
Example(s)	INST:CAT:FULL? returns “MS12-PM01 Loss Module”,10,”MS7-0A-012-09 1x12 Optical Switch”,12
See Also	INSTrument:CATalog?.

:SYSTem:DATE

Description

The SYSTem:DATE command is used to set the device's internal calendar.

This is not affected by a *RST command.

Syntax

:SYSTem:DATE<wsp><Year>,<Month>,<Day>

Parameter(s)

➤ *Year:*

The program data syntax for <Year> is defined as a <DECIMAL NUMERIC PROGRAM DATA> element.

The <Year> is rounded to the nearest integer. Its range is limited by the capability of the device. The year shall be entered as a four-digit number, including century and millennium information.

➤ *Month:*

The program data syntax for <Month> is defined as a <DECIMAL NUMERIC PROGRAM DATA> element.

The <Month> is rounded to the nearest integer. Its range is 1 to 12 inclusive. The number 1 corresponds to January, 2 to February, and so on.

:SYSTem:DATE

	➤ *Day:* The program data syntax for <Day> is defined as a <DECIMAL NUMERIC PROGRAM DATA> element. The <Day> is rounded to the nearest integer. It ranges from 1 to the number of days in the month from the previous parameter. This command keeps track of the number of days in each month, accounting for leap years through the range of years that it accepts.
Example(s)	SYST:DATE 2001,11,29
See Also	SYSTem:DATE?

:SYSTem:DATE?

Description	The SYSTem:DATE query returns the instrument's internal calendar. This is not affected by a *RST command.
Syntax	:SYSTem:DATE?
Parameter(s)	None
Response Syntax	<Year>,<Month>,<Day>

:SYSTem:DATE?

Response(s)

➤ *Year:*

The response data syntax for <Year> is defined as a <NR1 NUMERIC RESPONSE DATA> element.

The <Year> is a four-digit number, including century and millennium information.

➤ *Month:*

The response data syntax for <Month> is defined as a <NR1 NUMERIC RESPONSE DATA> element.

The <Month> ranges from 1 to 12, inclusively. The number 1 corresponds to January, 2 to February, and so on.

➤ *Day:*

The response data syntax for <Day> is defined as a <NR1 NUMERIC RESPONSE DATA> element.

The <Day> ranges from 1 to the number of days in the month from the previous field. This command keeps track of the number of days in each month, accounting for leap years through the range of years that it accepts.

Example(s) SYST:DATE? returns 2001,11,29

See Also SYSTem:DATE

:SYSTem:ERRor[:NEXT]?

Description

The SYSTem:ERRor[:NEXT]? queries the error/event queue for the next item and removes it from the queue. The response message consists of two fields separated by commas <Code>,<Description[,Info]>.

SYSTem:ERRor[:NEXT]? is a query only and, therefore, does not have an associated *RST state.

Syntax

:SYSTem:ERRor[:NEXT]?

Parameter(s)

None

Response Syntax

<Code>,<Description[,Info]>

Response(s)

➤ *Code:*

The response data syntax for <Code> is defined as a <NR1 NUMERIC RESPONSE DATA> element.

The <Code> is a unique integer in the range [-32768, 32767]. All positive numbers are instrument-dependent. All negative numbers are reserved by the SCPI standard with certain standard error/event codes described in an appendix of this document. The zero value is also used to indicate that no error or event has occurred.

➤ *Description[,Info]:*

The response data syntax for <Description[,Info]> is defined as a <STRING RESPONSE DATA> element.

:SYSTem:ERRor[:NEXT]?

The <Description[,Info]> parameter of the full response is a quoted string containing a description followed by information text [,Info]. Each <Code> has a unique and fixed <Description> associated with it. The <Date> and <Time> are appended to the [,info] separated by a semi-colon using the following format:
<Date><wsp><Time> where
<Date> = Year/Month/Day
<Time> = Hour,Minute,Second (24 hour time)

The maximum length of <Description[,Info]> is 255 characters. For standard defined error/event <Codes>, the <Description> is sent exactly as indicated in the appendix of this document.

Example(s)

SYST:ERR:NEXT? returns -222,"Data out of range"
SYST:ERR:NEXT? returns -222,"Data out of range,instrument monomodule 2100, 2001/11/29 14:56:16.259"

:SYSTem:TIME

Description	This device has an internal clock and implements the SYSTem:TIME command to set the clock time over the interface. This is not affected by a *RST command.
Syntax	:SYSTem:TIME<wsp><Hour>,<Minute>,<Seconds>
Parameter(s)	➤ *Hour:* The program data syntax for <Hour> is defined as a <DECIMAL NUMERIC PROGRAM DATA> element. The <Hour> is always rounded to the nearest integer. It ranges from 0 to 23 inclusively. The device accepts hour information in 24-hour format. ➤ *Minute:* The program data syntax for <Minute> is defined as a <DECIMAL NUMERIC PROGRAM DATA> element. The <Minute> is always rounded to the nearest integer. It ranges from 0 to 59 inclusively.

:SYSTem:TIME

➤ *Seconds:*

The program data syntax for <Seconds> is defined as a <DECIMAL NUMERIC PROGRAM DATA> element.

The <Second> is rounded to the resolution of the clock. It ranges from 0 to 60. A value of 60 is allowed since rounding may cause a number greater than 59.5 to be rounded to 60. When this element is rounded to 60 it shall be set to 0 and the minute value incremented. Any other carries shall be rippled through the date.

Example(s) SYST:TIME 12,47,29

See Also SYSTem:TIME?

:SYSTem:TIME?

Description	This device has an internal clock and implements the SYSTem:DATE? query to get the clock time over the interface. This is not affected by a *RST command.
Syntax	:SYSTem:TIME?
Parameter(s)	None
Response Syntax	<Hour>,<Minute>,<Second>

:SYSTem:TIME?

Response(s)

➤ *Hour:*

The response data syntax for <Hour> is defined as a <NR1 NUMERIC RESPONSE DATA> element.

The <Hour> ranges from 0 to 23. The instruments returns hour information in 24-hour format.

➤ *Minute:*

The response data syntax for <Minute> is defined as a <NR1 NUMERIC RESPONSE DATA> element.

The <Minute> ranges from 0 to 59.

➤ *Second:*

The response data syntax for <Second> is defined as a <NR1 NUMERIC RESPONSE DATA> element.

The <Second> ranges from 0 to 59. The resolution of the clock is the second.

Example(s) SYST:TIME? returns 16,55,38

See Also SYSTem:TIME

:SYSTem:VERSion?

Description	The SYSTem:VERSion? query returns a value corresponding to the SCPI version number to which the device complies. The SYSTem:VERSion? is a query only and, therefore, does not have an associated *RST state.
Syntax	:SYSTem:VERSion?
Parameter(s)	None
Response Syntax	<Version>
Response(s)	*Version:* The response data syntax for <Version> is defined as a <NR2 NUMERIC RESPONSE DATA> element. The <Version> is shown in the form Year.Revision, where Year represents the year-version (for example 1990) and Revision represents an approved revision number for that year. If no approved revisions are claimed, then this extension is 0.
Example(s)	SYSTem:VERSion? returns 1999.0 (no approved revisions are claimed)

D COM Properties and Events

The MS Platform also provides objects based on Microsoft Component Object Model (COM). COM defines a common way to access and create software components and services.

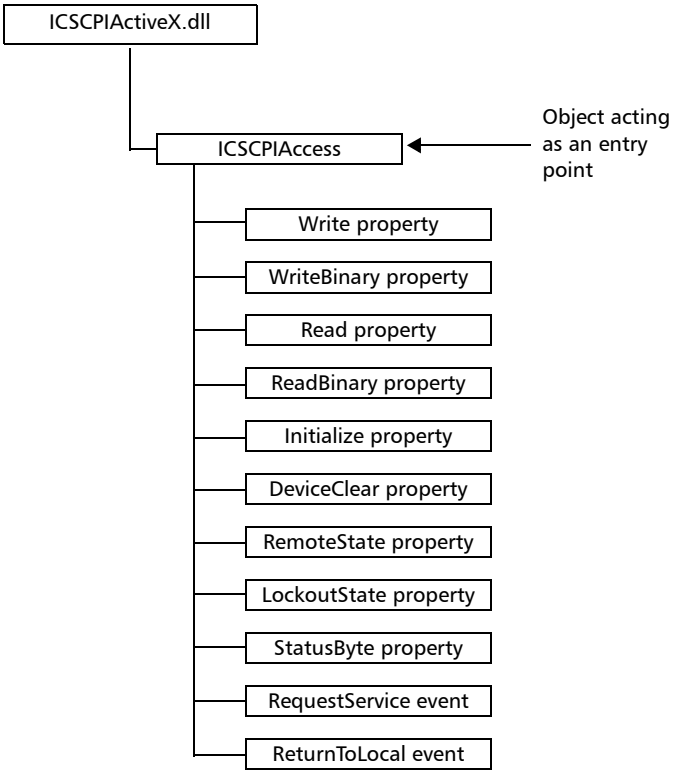
COM promotes the integration and the reuse of software components, as well as interoperability. In order to interoperate, components developed in different languages must adhere to a binary structure specified by Microsoft.

OLE and ActiveX are based on COM. Also, programming languages such as C, C++, Smalltalk, Pascal, Ada, and Java can create and use COM components.

You can build your own programs using the provided properties and events via the IcSCPIAccess interface. For information on how to configure your MS Platform for DCOM control, see *Preparing for Remote Control with DCOM* on page 108.

ActiveX (COM/DCOM)—Quick Reference

The following diagram illustrates the different properties and events available.



These properties and events are fully explained in the following pages.

Properties

	Write
Description	This method allows you to send a program message (single command or compound of commands) to the device input buffer.
Syntax	object.*Write* (*Message*)
Parameter(s)	*Message*: Required. A string value corresponding to the program message to be sent.
Possible error(s)	*Timeout*: This error occurs if the operation could not be completed within the allowed delay. For information on timeout setting, see *Initialize* on page 269.

WriteBinary	
Description	This method allows you to send a program message (single command or compound of commands) as an array of bytes into the device input buffer.
Syntax	object.*Write* (*BinaryArray*)
Parameter(s)	*BinaryArray*: Required. An array of bytes corresponding to the program message to be sent.
Notes	Use this method instead of the *Write* method if you need to send commands in binary (COM is UNICODE).
Possible error(s)	*Timeout*: This error occurs if the operation could not be completed within the allowed delay. For information on timeout setting, see *Initialize* on page 269.

IMPORTANT

Before you retrieve data with the Read or ReadBinary methods, JGR recommends that you specify the format in which the information must be returned. Details on how to correctly set the format can be found hereafter.

Read

Description	This method allows you to retrieve all the data from the device output queue in a UNICODE format.
Syntax	object.*Read*
Parameter(s)	None.
Response(s)	A string value (in UNICODE format).
Notes	This method must be used in conjunction with the *Write* method. Always ensure that a query has been previously sent before attempting to read a response from the output queue. To properly set the data format, send the following command (using the *Write* method): FORM:DATA<wsp>ASCII <number_of_digits> where <*number_of_digits*> corresponds to the number of digits after the decimal point that you require. Remember that the retrieved data must be converted to a numeric format before you can use it in calculations, for example.
Possible error(s)	*Timeout*: This error occurs if the allowed delay has expired before the preceding *Write* operation could send a response to the output queue. For information on timeout setting, see *Initialize* on page 269. *QueryUnterminated*: This error occurs if the output queue is empty (e.g. no query has been made previously).

ReadBinary

Description	This method allows you to retrieve data from the device output queue in a binary format.
Syntax	object.*ReadBinary*
Parameter(s)	None.
Response(s)	An array of bytes.
Notes	This method must be used in conjunction with the *Write* method. Always ensure that a query has been previously sent before attempting to read a response from the output queue. To properly set the data format, send the following command (using the *Write* method): FORM:DATA<wsp>PACKED The retrieved data *does not* need to be converted to a numeric format before you can use it in calculations, for example. To help you know the actual length of the retrieved data, it has the following structure: ➤ The first byte contains the # character. ➤ The byte that immediately follows contains the number of subsequent bytes that you have to check to know the total length.

ReadBinary

	For example, if you receive this response (here, values are expressed in decimal instead of binary for easier readability): # 2 1 3 7 5 8 9 2 ... The byte that immediately follows the # contains 2, which means that you have to read the two following bytes to know the length (in bytes) of the retrieved data. The bytes indicate 1 and 3. The length is then 13 bytes. The actual response begins at byte number 5, in this case.
Possible error(s)	*Timeout*: This error occurs if the allowed delay has expired before the preceding *Write* operation could send a response to the output queue. For information on timeout setting, see *Initialize* on page 269. *QueryUnterminated*: This error occurs if the output queue is empty (that is no query has been made previously).

Initialize

Description	This method allows you to configure the timeout value i.e., the allowed delay for Read and Write operations, in milliseconds.
Syntax	object.*Initialize*(*Timeout*)
Parameter(s)	*Timeout*: Required. A numeric value corresponding to the delay in milliseconds.
Notes	If the *Initialize* method is not invoked, the default value is 10 000 milliseconds.

DeviceClear	
Description	This method performs a *Device Clear* operation as specified in the IEEE 488.1 standard.
Syntax	object.*DeviceClear*
Parameter(s)	None.

RemoteState	
Description	This property returns or sets the device's remote state.
Syntax	object.*RemoteState* (to retrieve the state) object.*RemoteState*=*State* (to set the state) *State*: a Boolean value corresponding to: True: Remote False: Local
Parameter(s)	None.
Response(s)	If the property is used to get the device's remote state, the property returns a Boolean value.
Access	Get/Set

LockoutState

Description	This property returns or sets the device's lockout state.
Syntax	object.*LockoutState* (to retrieve the state) object.*LockoutState*=*State* (to set the state) *State*: a Boolean value corresponding to: True: Lockout False: No lockout
Parameter(s)	None.
Response(s)	If the property is used to get the device's lockout state, the property returns a Boolean value.
Access	Get/Set

StatusByte

Description	This read-only property returns the device's status byte. Refer to IEEE 488.2 standard for status byte description.
Syntax	object.*StatusByte*
Parameter(s)	None.
Response(s)	A value corresponding to the device's status byte.
Notes	This property can be used in conjunction with *RequestService* event (see *RequestService* on page 272) to find out why the device caused a Service Request (SRQ).
Access	Get

Events

RequestService	
Description	This event is triggered whenever the device causes a Service Request (SRQ).
Parameter(s)	None.
Notes	It is the user's responsibility to configure the different registers (*SRE, *ESE) as stated in the IEEE 488.2 standard to receive SRQ. When used in conjunction with *StatusByte* property (see *StatusByte* on page 271), this event allows you to determine the cause of the SRQ.

ReturnToLocal	
Description	This event is triggered when the user presses the Local button from the front panel of the controller unit when the device is in Remote state.
Parameter(s)	None.

E Communicating Through TCP/IP over Telnet

Introducing TCP/IP over Telnet

The JGR Instrument Control provides SCPI automation or remote control over Telnet through TCP/IP as a Windows Service that continuously listens to a port from a Telnet Server (MS) on which modules to be tested are connected.

TCP/IP protocols are used for communication.

Note: *Port 5024 is designated for sending SCPI commands in the Telnet protocol.*

All Windows versions include the Telnet Client and the Telnet Server components. With these components, you can create a remote command console session on a remote computer.

Commands can be executed simply by logging on the Server using the Telnet interface.

Features

- A client can use the Telnet components, which are freely available on Windows, Linux, and Unix operating systems, as well as connect to the service
- A client from any network domain can connect to the service
- A client/session can connect to one module at a time
- A user can connect to multiple modules through multiple sessions
- A client can execute single commands or batch of commands
- A user can terminate any client already connected

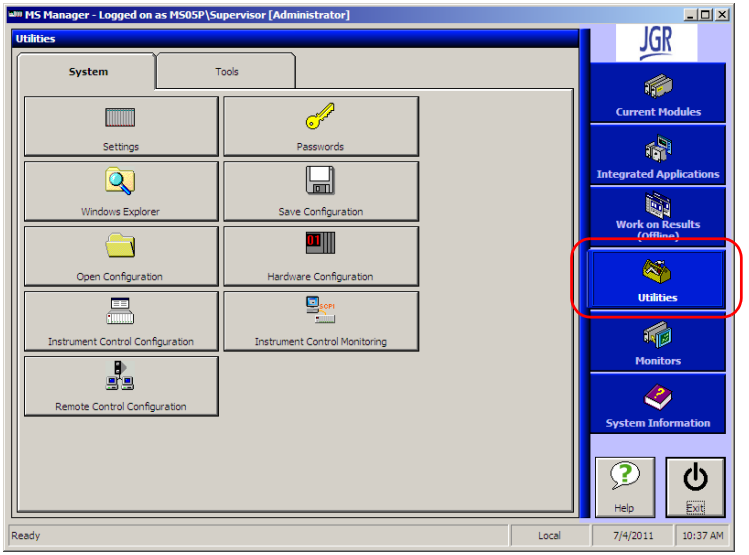
Getting Started with TCP/IP over Telnet

The TCP/IP over Telnet Service, which is part of the JGR Instrument Control, is a mediator between the Telnet client and the testing instrument.

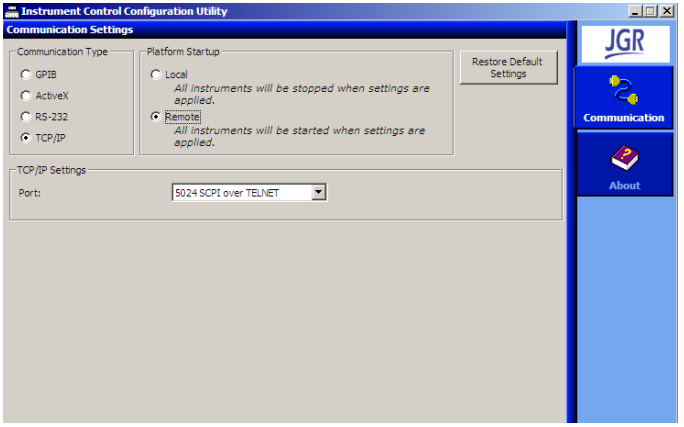
Any request from the Telnet client will be transferred to IcExecutive. IcExecutive then passes the request to the appropriate instrument. The instrument executes the request and returns the response to IcExecutive and to TCP/IP over Telnet accordingly.

To activate the TCP/IP over Telnet:

1. From MS Manager, click the **Utilities** function tab.



2. Click **Instrument Control Monitoring** to access the corresponding utility.



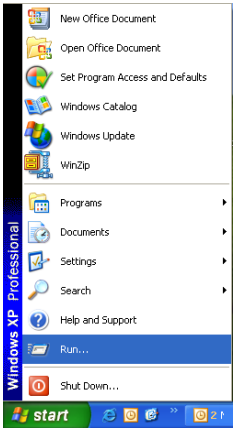
3. Select the **TCP/IP** and **Remote** options.
4. Click **Apply**, and then **Exit**.
5. Restart MS Manager.

Executing SCPI Commands Over Telnet

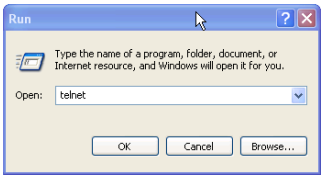
You can remote control the MS05P/MS10P/MS10E system modules by executing SCPI commands through TCP/IP over Telnet. The Telnet Client and the Telnet Server components are being used to create a remote command console session on a remote client computer.

To connect through TCP/IP over Telnet from a remote Windows client:

1. Start Windows.
2. On the taskbar, click **Start**, and then **Run**.

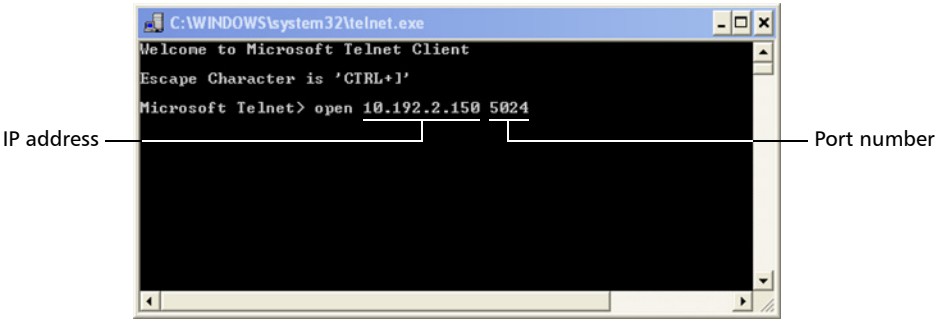


3. In the **Open** field, type *telnet*.



4. Still in the **Open** field, connect to the TCP/IP Telnet Service by typing the *OPEN <IPADDRESS><PORT>* command:

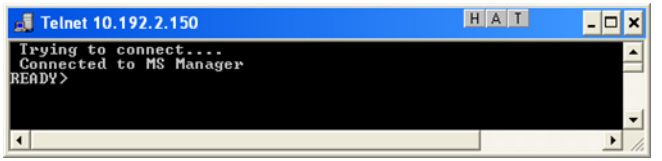
   Example: *open 10.192.2.45 5024*



Note: *Port 5024 is designated for sending SCPI commands in the Telnet protocol.*

5. Press ENTER to establish connection with the Service.

   If the connection is established, the READY> prompt is displayed in the Telnet Editor window.



6. Type a SCPI command to execute in the Editor window.

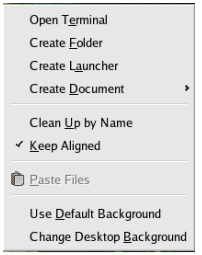
   For more information, see the following sections:

   - *Examples for Releasing Modules* on page 280
   - *Internal Commands of the TCP/IP over Telnet Protocol* on page 283

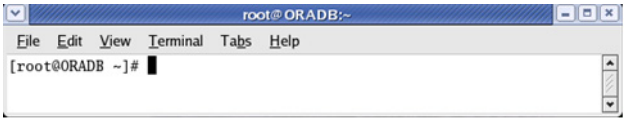
7. Click to close the session.

To connect through TCP/IP over Telnet from a remote Linux client:

1. Right-click on the desktop, and then click **Open Terminal**.



2. The command prompt is displayed in the Telnet Editor window.

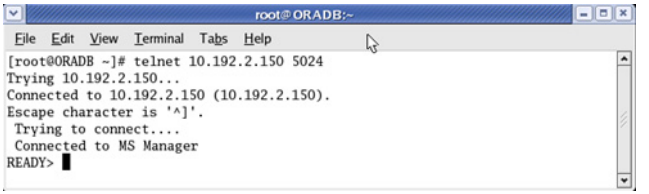


3. Connect to the TCP/IP Telnet Service by typing the *OPEN <IPADDRESS><PORT>* command:

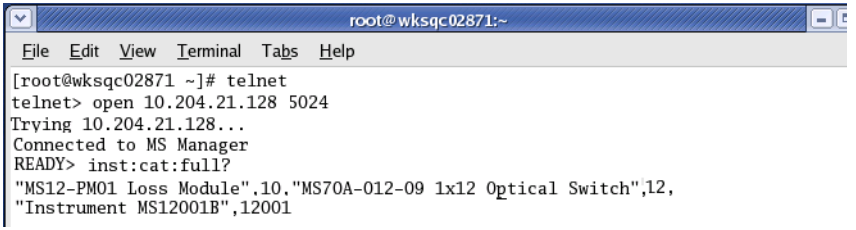
Example: *open 10.192.2.45 5024*

Note: *Port 5024 is designated for sending SCPI commands in the Telnet protocol.*

The connection is established when the message **Connected to MS Manager** is displayed in the Editor window.



4. Type a SCPI command to execute in the Editor window.



For more information, see the following sections:

- *Examples for Releasing Modules* on page 280
- *Internal Commands of the TCP/IP over Telnet Protocol* on page 283

5. Click ✖ to close the session.

Examples for Releasing Modules

A module is blocked as soon as a connection is established from any user session executing a valid instrument command.

For example, when the LINS10:SOURce:DATA:TELecom:CLEar command is executed for the first time by client session 10.192.2.155:1364, the module is blocked for any other client/session until the user releases it.

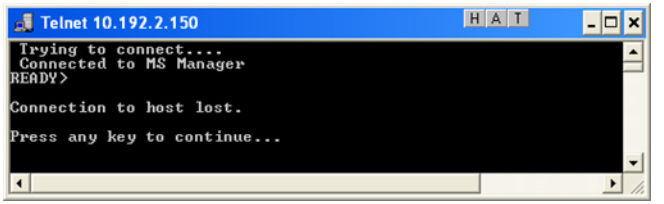
A module is released by one of the following actions:

- Once all the desired commands ave been completed, executing the CLOSE command to end the current session. For more information, see *CLOSE* on page 284;
- Closing the current session by clicking the **Close** button on the Telnet Editor windows' title bar;
- Shutting down and restarting the client computer;
- A network interruption.

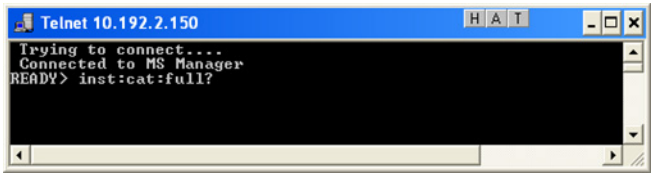
A module can also be released by terminating the communication by executing one of the two following commands:

- KILL LINSxx (For more information, see *KILL LINSxx* on page 285.)
- CLOSE (For more information, see *CLOSE* on page 284.)

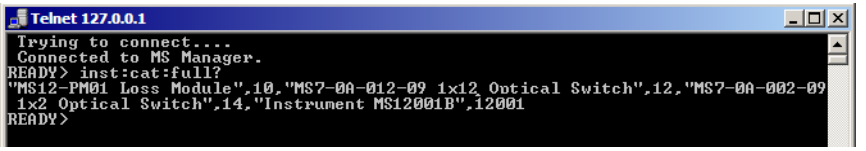
If the connection is not established, the **Connection to host lost** message is displayed in the Telnet Editor window.



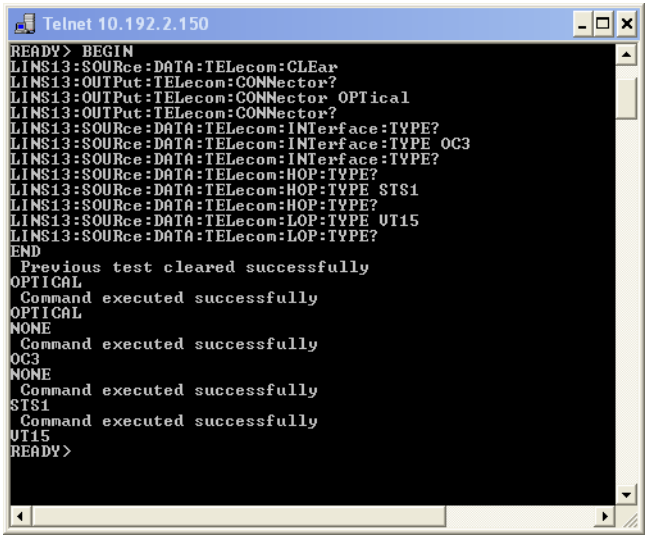
Once the connection is established, type or copy the desired SCPI command in the Telnet Editor window. Then, press ENTER to execute the command.



The response for the above command is displayed in the Telnet Editor window.



To execute multiple commands, copy the desired commands from any Script file and paste them between BEGIN and END blocks in the Telnet Editor window, and then press ENTER.



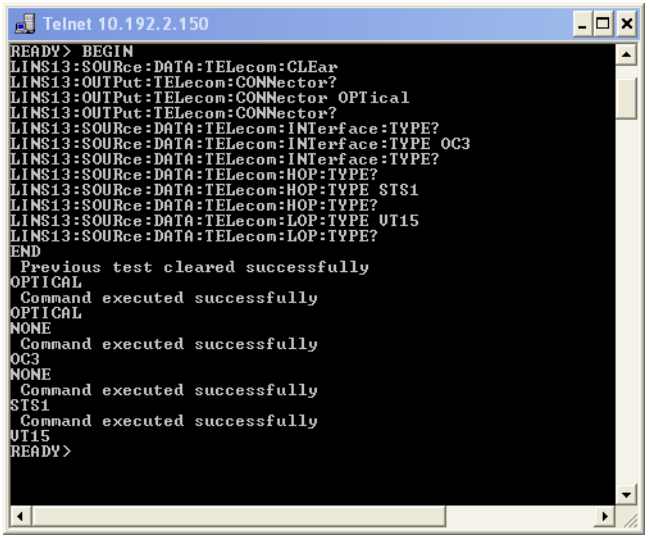
For more information, see *Internal Commands of the TCP/IP over Telnet Protocol* on page 283.

Internal Commands of the TCP/IP over Telnet Protocol

BEGIN and END

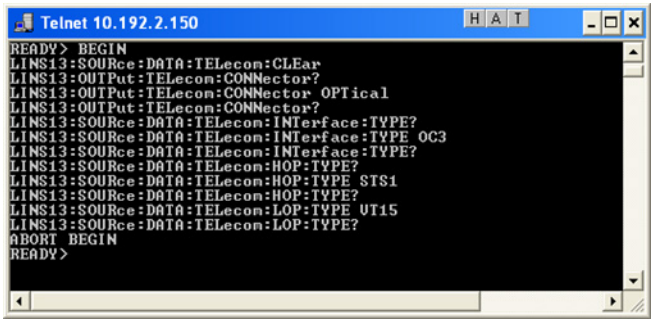
To execute multiple commands, all commands should be inserted within the BEGIN and END blocks in a Telnet Editor window.

For executing a single command, no protocol is required. Simply type or paste the command in the Editor window.



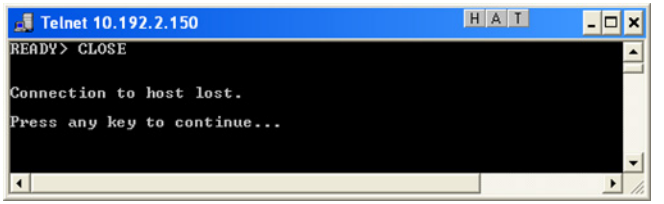
ABORT BEGIN

The ABORT BEGIN command terminates the commands inserted between the BEGIN and END blocks and returns to the READY> prompt in the Editor window.



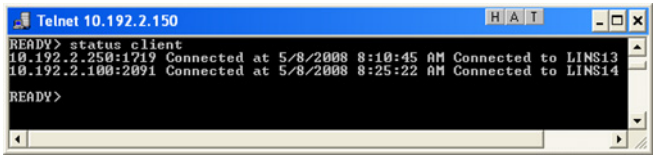
CLOSE

The CLOSE command terminates the current Telnet session.



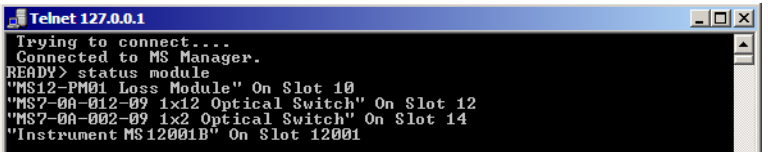
STATUS CLIENT

the STATUS CLIENT command lists out the statuses of clients, along with the connection time and the connected modules.



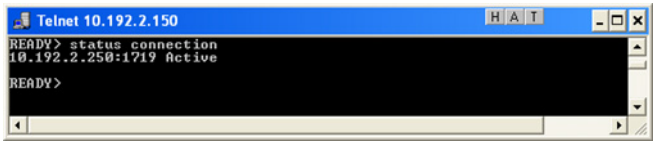
STATUS MODULE

The STATUS MODULE command lists out the statuses of modules; along with the slot numbers where they are connected.



STATUS CONNECTION

The STATUS CONNECTION command displays the connection along with status whether it is **Active** or **Idle**.

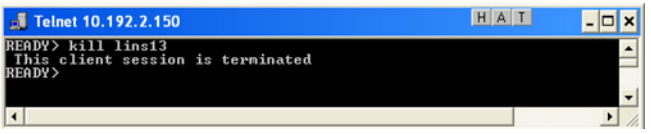


Note: *If any connection is idle for a certain period, the service automatically changes the status to* ***Idle****.*

KILL LINSxx

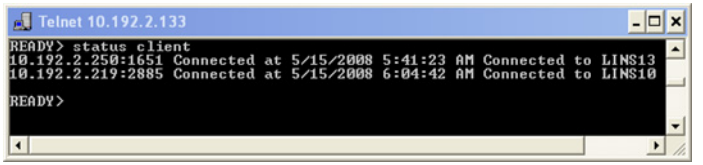
The KILL LINSxx command allows any user to terminate an ongoing session.

In place of xx, type the unit number and the slot number of the module to be remotely controlled.

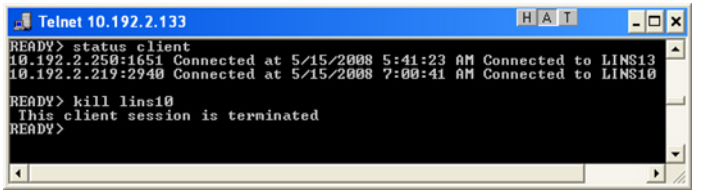


To know the status of the modules in order to terminate connections using the KILL LINSxx command, you can first enter the STATUS CLIENT command. For more information, see *STATUS CLIENT* on page 284.

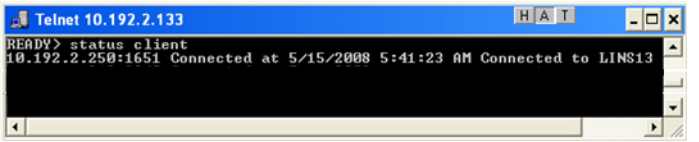
In the example below, two modules are connected: LINS13 and LINS10.



To disconnect the LINS10 module used by another session, enter the *kill lins10* command. The **This client session is terminated** message is displayed once the module is disconnected.



Enter again the STATUS CLIENT command to confirm the termination of the module (LINS10 in our example). Only the information of the remaining connected client is displayed.



Index

A

About tab 180
AC requirements 16
accesses in MS Manager 53
accessing
  MS Manager 34
  utilities 86
ActiveX
  control 89
  linking units 100
  output units and formats 102
  selecting 105
adjusting
  controller unit angle 17
  display 40
  sound 40
after-sales service 179
alarm indicator LED 9
angled brackets 157
ASCii format 102
automatic fan speed 57
automation 87, 88

B

back panel 3, 5, 6
backlight
  button 9
  level 43
benchtop, putting units on 17
books, SCPI and IEEE 488.2 155
braces 157
brackets
  angled 157
  square 156
bumpers, platform 19
bus timing 102
Busy, module status 81
button, on/off 4, 6

C

Canadian Standards Association (CSA) ix
capacitors 15
caution
  of personal hazard 12
  of product hazard 12
certification information ix
changing
  backlight level 43
  communication settings 104
  status 146
  theme color 40
cleaning
  front panel 173
  touchscreen 173
closing MS Manager 35
code writing, GPIB 159
codes, error 161
colon 157
columns, setting monitor windows in 83
COM, local control 89
COM/DCOM
  demo application 162
  events 272
  properties 265
comma 158
commands
  IEEE 488.2 224
  SCPI 155
  specific 245
  TCP/IP over Telnet 283–286
communication
  ActiveX 89
  changing settings 104
  default settings 107
  Ethernet TCP/IP 88, 89
  GPIB 88, 89
  RS-232 ..... 88, 89
  TCP/IP over Telnet 90, 273–275

components tab 183
configuration
  GPIB 98
  loading 60
  opening 59
  saving 59
  saving on exit 61
connecting units together 22
connectors
  for interlock 30
  peripheral 3, 5
contact information 180
contents of hard drive 11
control
  ActiveX 89
  Ethernet TCP/IP 88, 89
  GPIB 88, 89
  remote 87
  RS-232 89
  TCP/IP over Telnet 90
controller unit stand 17
controlling modules 88
conventions, programming 155, 158, 160
conventions, safety 12
cooling fans clearance space 18
copying commands into text document ... 164, 166
covers, unit 15
creating integrated data display 83
CSA ix
current modules 79
current, electrical 16
customizing
  hardware 62
  monitors 83

D

data 51
  peripherals related to 4
  types 201
data input
  IEEE 488.2 202
  SCPI 221
data output IEEE 488.2
  special 214, 222
  standard 211
DCOM
  configuring computers 108
  demo application 162
  technology 89
deactivating LED display 41
default identification of module 62
dial, selection 9
digit 156
disconnecting unit 14
door for interlock 30
drive, DVD-RW 4
DVD-RW 4

E

Enter button 9
error messages in remote control 161
error/event queue 150
Ethernet TCP/IP
  control 88, 89
  port 100
examples
  COM/DCOM 162
  TCP/IP over Telnet release 280
exiting
  and saving configuration 61
  MS Manager 35
expansion unit connectors 3, 5

Explorer, Windows 86
external
door for interlock..... 30
keyboard, installing 43
monitor, installing 38

F

factory settings, reverting to 64
fan speed 57
finding units..... 85
front panel description..... 4, 6
front panel, cleaning 173
function tabs..... 79

G

GPIB
address 89
capability 99
chain..... 89
configuring parameters 106
control..... 88, 89
hardware configuration..... 98
optimizing performance 101
output units and formats 102
port 100
restrictions..... 98
selecting 105

H

handshake rate..... 102
hard drive
activity LED 9
structure 11
hardware, customizing..... 62
help. *see* online user guide
high speed (HS488) 98, 101, 106

I

identification, module..... 62
IEEE 488.1 protocol..... 98, 101, 102
IEEE 488.2 commands 224
indoor use 15
information about system, viewing 180
inlets..... 14
input
buffer..... 148
IEEE 488.2 data types 202
SCPI data types 221
input current 16
inserting a module 25
installed kits 181
installing
external keyboard 43
external monitor 38
mouse..... 43, 46, 50
software..... 33
unit 15
units in rackmount..... 18
instrument control settings 104
integrated
data display, creating..... 83
interface at startup..... 52
interlock
function 30
LED 9

J

JGR address 180

K

keyboard
on-screen 44
PS/2..... 43
keywords, SCPI 157
Kits Installed tab..... 181
knob, selection 9

L

LAN network 89, 90
launching MS Manager 34

layout of on-screen keyboard 44
learning COM/DCOM 162
LED
  deactivating display 41
  description 9
  indicators 4, 6
legs, platform 19
lights, turning off 41
linking units
  Ethernet port 100
  GPIB port 100
  serial port 101
  together 22
list of components 183
list separator 157
loading configuration 60
local control 89, 105
local to remote 146
locating units 85
lock LED 9
lockout status 144, 145
logging off Windows 35

M

main power switch 3, 5, 6
maintenance
  front panel 173
  general information 173
mandatory commands 224
maximum input current 16
messages, remote control 148
mnemonic, definition 157
module
  controlling 88
  current 79
  default identification 62
  detail 79
  insertion 25
  locating 85
  name 62
  number 62
  removal 25
  status 81
monitor
  customizing windows 83
  port 38
monitoring remote commands 163
mounting rack
  cabinet 21
  installing units in 18
  kit 18
mouse
  installing 43, 46, 50
  PS/2 ..... 43, 46, 50
moving, monitor windows 83
MS Platform
  controller 87
  turning off 32, 35
  turning on 32
multiple capabilities 156

N

name, module 62
network, LAN/WAN 89, 90
number, module 62

O

on/off
  button 4, 6
  LED 9
online user guide 178
on-screen keyboard, selecting type 44
opening
  configuration 59
  MS Manager 34
orientation of units 17
Out port 23
outlets, power 3, 5
output IEEE 488.2
  special 214, 222
  standard 211
output queue 149

P

PACked format 102
panel, front 4, 6
passwords in MS Manager 53
PDF. *see* online user guide
peripheral
  connectors 3, 5
  data related 4
peripherals related to 51
pipe character 157
platform
  bumpers 19
  legs 19
platform commands 245
port
  Ethernet 100
  GPIB 100
  MS10P In 22, 23
  MS10P Out 23
  serial 101
position
  display 4, 6, 85
  of unit 22
positioning units on benchtop 17
power
  button 4, 6
  cable 14
  main switch 3, 5, 6
  outlet 3, 5, 6
  plug 14
power source, AC 16
Program Switcher button 9
programmable instruments, standards 88, 155
programming, GPIB 159
protocol
  HS488 98
  IEEE 488.1 98
PS/2
  keyboard 43
  mouse 43, 46, 50

Q

queue
  error/event 150
  input 148
  output 149

R

rackmount
  cabinet 21
  installing units in 18
  kit for unit 18
Ready, module status 81
receiving data
  with Ethernet port 100
  with GPIB port 100
  with serial port 101
register
  diagram 153
  ESE 151
  ESR 151
  SRE 151
  STB 151
remote control
  ActiveX (DCOM) 89, 100
  configuring 105, 108
  description of commands 155
  error messages 161
  GPIB 89, 100
  limitations 98
  messages 148
  methods 1, 87
  monitoring 163
  RS-232 ..... 88, 89, 101
  status 144
  TCP/IP over Telnet 90, 273–286
remote LED 9
remote to local 146
removing a module 25
repairing unit 15
reset settings 64
returned values, setting 102

reverting to default settings 64, 107
rights in MS Manager 53
RS-232
  communication 88
  configuring parameters 106
  control 89
  port 101
  selecting 105
rules
  programming 160
  syntax 158
  syntax conventions 155

S

safeguarding units 30
safety
  caution 12
  conventions 12
  power cable 14
  warning 12
saving
  any configuration 59
  configuration on exit 61
SCPI
  commands 155
  data types 221
  guidelines 88, 155
  references 155
screen brightness 43
security levels in MS Manager 53
selecting
  on-screen keyboard type 44
  startup interface 52
selection dial 9
semicolon 158
sending data
  with Ethernet port 100
  with GPIB port 100
  with serial port 101
separator 157, 158
service request enable register (SRE) 151
setting
  instrument control utility 104
  monitor window columns 83
software
  accessing 34
  automatic launch 52
  exiting 35
  installing 33
  keyboard 44
  upgrading 33
  using passwords 53
  utilities 86
sound, adjusting 40
space 156
specific commands 245
speed, fan 57
square brackets 156
stand for controller unit 17
standard event status
  enable register (ESE) 151
  register (ESR) 151
standard status data structure
  diagram 153
  general 151
startup interface, selecting 52
status
  local 144
  local with lockout 144
  remote 144
  remote with lockout 145
  switching 146
status bar 81
status byte register (STB) 151
status, remote control 144
storage requirements 173
structure of hard drive 11
symbols, safety 12

syntax
rules 155, 158
SCPI 155
symbols 156
system
ghost 11
tree 79
viewing information about 180

T

TCP/IP
configuring parameters 106
selecting 105
TCP/IP control 88, 89
TCP/IP over Telnet
activating 274
commands 283–286
connecting to, Linux 278
connecting to, Windows 276
control 90
features 273
introduction 273
port number 5024 273
release examples 280
technical support 179
temperature for storage 173
theme color, changing 40
touchscreen 37, 173
transportation requirements 173, 184
turning lights on/off 41
turning MS Platform
off 32, 35
on 32
type of on-screen keyboard 44
types and data 51
types, data 201

U

unit
covers 15
disconnecting 14
installing 15
locating 85
position 22
repairing 15
ventilation 15
unit status 144
upgrading software 33
user guide. *see* online user guide
using configuration in different units 59
utilities in MS Manager 86

V

ventilation 15, 57
vertical bar 157
viewing system information 180
virtual keyboard 44
volume, adjusting 40

W

WAN network 89, 90
white space 156
Windows
Explorer 86
XP security 53
wsp 156

© 2011 JGR Optics Inc. All rights reserved.
Printed in Canada (2011-08)

CE