Nucleus4

Communications Guide

PN 141-402, Revision D, Revised 07/26/11

CascadeMicrotech®

Contents

Welcome

Welcome to the *Nucleus™ 4 Communications Guide*. The information provided in this guide explains how to control your probe station from a remote host or from another Microsoft Windows® application running on the probe station computer.

Notational Conventions

This manual uses the following conventions:

- When directing the user to a command found on the menus, a vertical hash mark is used to indicate that a command is nested. So, to direct the user to the Wafer Map command from the *Tools* item on the menu, we use *Tools > Wafer Map*.
- `Syntax strings (commands) appear in this font.`
- All numbers are decimal unless otherwise stated.
- Bit 0 is the low-order bit. If a bit is set to 1, the associated description is true unless otherwise stated.

NOTE

Note is used to indicate important information about the product that is not hazard.

CAUTION

Caution is used to indicate the presence of a hazard which will or can cause minor personal injury or property damage if the warning is ignored.

WARNING

Warning is used to indicate the presence of a hazard which can cause severe personal injury, death or substantial property damage if the warning is ignored.

DANGER

Danger is used to indicate the presence of a hazard which will cause severe personal injury, death or substantial property damage if the warning is ignored.

Where to Get More Information

More information is available from the.se sources:

- Other information or instructions arriving with your system.
- *Nucleus User Guide* or online Help.
- **Release Notes**: Lists features and issues that arose too late to include in other documentation. Release Notes are available from the Help menu in Nucleus or from the Windows Start menu (Start > All Programs > Cascade Microtech>Nucleus 4.0)).
- **World Wide Web**: Cascade Microtech maintains an active site on the World Wide Web at *http://www.cascademicrotech.com*. The site contains current information

about the company and locations of sales offices, new and existing products, contacts for sales, service and technical support information. You can also send e-mail to Cascade Microtech using the web site. Requests for sales, service and technical support information will receive a prompt response.

NOTE

When sending e-mail for technical support, please include information about both the hardware and software, with a detailed description of the problem, including how to reproduce it.

- **Other**: If you purchased your Cascade Microtech product from a third-party vendor, you can contact that vendor for service and support.

Technical Support

If you need technical support, please read everything relevant to the problem in this guide and in online Help before calling customer service.

For immediate sales support or customer service assistance, please contact the Sales Administration Department at 1-800-550-3279 or sales@cmicro.com. Requests for sales, service, and technical support information receive prompt response.

To receive a faster solution, try to recreate the problem to provide us with an exact sequence of events. Please have the following information available, if possible:

- Name, version number and file date of the application in use
- Drive information, including sizes, hard drive controller card brand, partition sizes and partitioning software
- Additional hardware such as specialty video cards, EMS boards, or turbo cards
- Memory-resident programs in use when the problem occurred. Problems can occur when memory-resident programs are not loaded in the correct order
- Exact wording of any error messages
- Notes about any steps you took in trying to solve the problem
- Windows version number and manufacturer

Also see *Nucleus Resources* in the *Nucleus User Guide* for information on generating a diagnostic file for help in troubleshooting.

Chapter

1 Communications Protocols

Cascade Microtech provides four software interfaces for controlling semiautomatic probe stations:

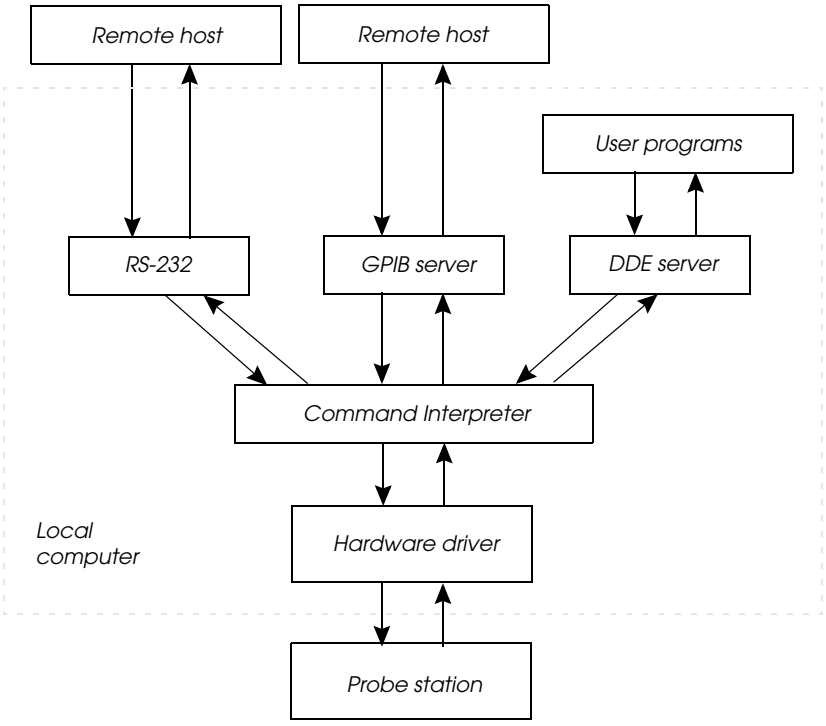
- GPIB Communications — to remotely control the probe station with a separate computer.
- DDE Server Communications — to control hardware with Windows-based applications running on the probe station's computer.
- RS-232 Communications — to support remote communication across the RS-232 serial interface.
- *Nucleus Software* — a menu-based, interactive software running under Microsoft Windows XP. See the *Nucleus User Guide for more information.*

The Command Interpreter (CI) software module accepts commands from other applications, sends them to the probe-station drivers and returns responses.

The GPIB Server receives commands from a remote host over the GPIB and passes them to the CI, which in turn passes them on to the probe station hardware driver. The probe station returns responses to the CI, which passes them to the GPIB Server for sending out over the GPIB to the remote host.

A DDE application program needs to use DDE-specific commands to open communication with the DDE Server. Once that is accomplished, the program sends probe station control commands to the DDE Server, which passes them to the CI. The CI sends the commands to the probe station's hardware driver software, which executes the commands on the probe station. Responses from the hardware are sent back through the CI and the DDE Server to the DDE application program.

Probe station communications model



From an application programmer's point of view, the DDE and GPIB commands are virtually identical. All command sets use the same syntax. The only difference is that there are some commands that are useful only in a GPIB session.

In addition to Cascade's library of prober control commands, the GPIB Server recognizes commands that control the bus. The DDE Server also recognizes the commands in the Windows DDE Management Library.

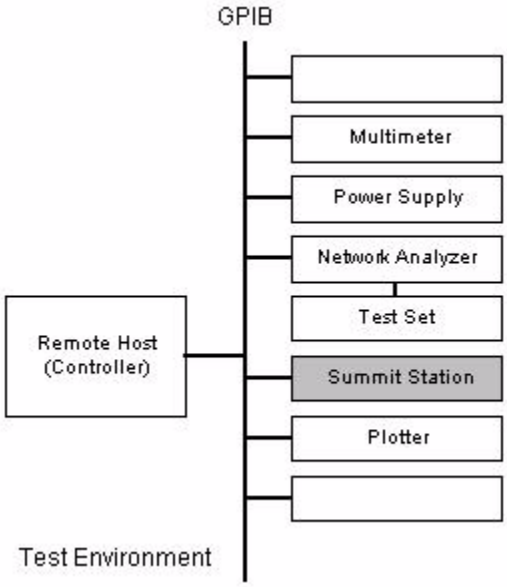
GPIB Communications

Cascade semiautomatic probe stations support remote communication across the General Purpose Interface Bus (GPIB or IEEE-488). This feature allows you to send commands and information between the probe station and a remote host. The probe station remote communication features help you integrate the probe station into an automated test environment.

This section contains the following information:

Using the GPIB Server

For communication across the GPIB, Cascade semiautomatic probe stations have both talker and listener GPIB capabilities. If you have a remote host acting as a GPIB controller, you can send and receive instrument settings across the GPIB. The following diagram shows a sample test environment with several instruments connected to a host through a GPIB.



GPIB Terms

This section describes GPIB and SCPI standards, talker/listener and controller capabilities, and command-complete and query-response handshaking standards.

GPIB and SCPI Standards

The GPIB standard describes a method for allowing instruments to communicate across a bus. This includes a standard method for specifying which instrument is the talker, listener and/or controller of the communication. This also includes a standard method for defining and describing status-reporting structures. The GPIB standard is fully described in the IEEE 488.1-1987 and 488.2-1987 documentation published by IEEE.

Cascade probe stations support commonly used IEEE 488.1-1987 and 488.2-1987 commands such as status-reporting commands.

The Standard Commands for Programmable Instruments (SCPI) standard describes a set of commands and a syntax for instrument communication. The SCPI commands and syntax are built upon the GPIB communication protocol. The SCPI standard is issued and controlled by the SCPI consortium.

Cascade probe-station commands comply with SCPI syntax and protocols but are not defined in the SCPI standard, because the standard does not include commands for probing equipment.

Instrument vs. Controller

Cascade probe stations have both GPIB *talker/listener* and *controller* capabilities. By default, a probe station is a GPIB instrument. It becomes a GPIB controller when directed to make a VNA calibration, or to set the temperature on a thermal controller. The probe station gives up control of the GPIB and reconnects to the host when it returns to the local operation mode.

After a probe station has been in controller mode, the host must issue an IFC (Interface Clear) command to regain control of the probe station. This is because the GPIB switch box breaks off communication with the host to allow the probe station to control instruments. When the connection with the host is re-established, the host must consider the GPIB to be in an unknown state and initialize the bus.

NOTE

If you plan to use your probe station both as an instrument and as a controller, you will need Cascade's optional GPIB switch box. For example, if you plan to use thermal equipment or VNA calibration software and control the probe station from a remote host.

Command and Query Responses

Command-complete and query-response handshaking conforms to the IEEE 488.2 standard. However, using the Meta command **`$:set:resp on`**, you can enable the sending of return strings so that the probe station sends a response string to the host after each command execution.

When return responses are enabled, the probe station sends the host one of the following after each command or query:

- The ASCII string "COMPLETE"
- An error message
- A query response

Unlike commands, queries always return a response, regardless of the Meta-command settings. This is because the host expects an ASCII string in response to a query. If the host receives no response, it can hang. If the probe station cannot return the expected response, it returns an error. Examples of query responses include a set of coordinate values, a version string and a probe plan file name.

The **@**(at sign) precedes all error messages sent by the probe station.

Types of Commands

This section describes types of commands that allow you to control the probe station from your host computer.

Command Groups

GPIB commands	Reads and sets registers in the GPIB status reporting structure.
Meta commands	Specifies system-level functions such as error reporting.
Probe-station control commands (SCPI, EG)	Translates movement along the x-, y-, and z-axes, manages files, and sets up the probe station.

The symbol preceding a command identifies its instruction set as follows:

$:	Precedes a Meta command
:	Precedes a probe station control command
*	Precedes a GPIB command

Probe Station Commands (SCPI, EG)

Probe station commands control the probe station. For example, they move the device, turn vacuum on and off, load probe plans and so on. For backwards compatibility with previous versions of Nucleus, the device ID parameter is always accepted, but ignored for these commands. For new implementations, we recommend avoiding device ID for these commands. and EG Commands for descriptions of the available commands.

Meta Commands

Meta commands control system level-functions such as whether the computer displays commands and responses. Meta commands are preceded by a $ (dollar sign), but have SCPI-compatible syntax (similar to probe station commands). Meta commands comply with these general rules:

- The commands are not case-sensitive.
- Meta commands always start with a $ (dollar sign).
- Parameters to a command follow the command and are separated by spaces.
- There is also a space between the command and its first parameter.
- Commands are separated by a ; (semicolon).
- The keyword separator for multiple keyword commands is : (colon).
- Meta commands do not return a response.
- All commands and responses are ASCII strings.

For example, you can use the Meta command sequence **$:set:resp** on. This command specifies that the probe station return to the host the message "COMPLETE" after each successfully completed remote operation

Table 1. Remote Mode Meta Commands.

Command	Description

`$:set:mode`
`[ SUMMIT ]`
`[ EG ]`
`[ TEL ]`
	Select which of the command sets is to be used when interpreting an incoming command. SUMMIT is the standard SCPI command set.

`$:set:resp`
`[ ON ]`
`[ OFF]`
	Sends an ASCII string `"COMPLETE"` to the host after each operation is completed unless an error occurs. If an error occurs, an error string is sent to the host.

Choose one of the parameter values in brackets ( ).

GPIB Protocol Commands

GPIB protocol commands read, set and clear GPIB status-reporting registers.

GPIB commands are preceded by an * (asterisk). For example, if you wanted to set the service-request enable register (SRE) to a particular value, you would use the ***sre** command.

Also see Status Reporting and Time-Outs.

Queries, which return a response to the remote controller, are followed immediately by a ? (question mark). For example, if you wanted to know the status of the SRE register, use the **`*sre?`** query command.

Table 2. GPIB protocol commands.

Command	Description	Function
`*cls`	*clear-status*	Clear-status command. Clears the ESR, and STB registers in the status-reporting structure. This command does not clear the ESE and SRE registers.
`*ese`	*event-status enable*	Sets the value of the ESE register. See the text for a description of the bits that are set.
`*ese?`	*event-status register*	Returns the value of the ESE register.
`*esr?`	*event-status register*	Returns the value of and clears the standard event-status register (ESR).
`*idn?`	*identification query*	Returns the probe station identification string. The Return string from an *idn? command is: Cascade Microtech, 'Station Type', 'Serial Number', 'Version Major', 'Version Minor' 'Station Type' will be one of the following depending on your station: • Virtual Summit 12K • Virtual S300 • Virtual Alessi 6100 • Virtual Summit 12K Theta • 12K • S300 • Alessi 6100 • Elite 300 • Elite 300 Edge • 12K Theta • S300 Theta S300 Theta - refers to a station type where the auxiliary chucks rotate when the theta axis is moved. 12K Theta - refers to a Summit 12000 series stage that has a motorized theta stage. Example String: Cascade Microtech, S300, 2323, 2, 5
`*opc`	*operation complete*	Sets the OPC bit in the ESR register and bit 1 (the DNE bit) in the STB register.
`*opc?`	*operation complete*	Places a 1 in the output queue when all query selected pending operations are completed.
`*rst`	*reset*	Resets the probe station. This command is the same as the `*cls` command.
`*sre`	*service-request enable*	Sets the value of the SRE register (the enable register for SRQ generation).
`*sre?`	*service-request enable query*	Returns the value of the SRE register.
`*stb?`	*status byte query*	Returns the value of the STB register.

Command	Description	Function
`*tst?`	*self-test query*	Always returns a 0.

For more information about GPIB commands, refer to the *IEEE 488.1-1987* and *488.2-1987* documents. The use of GPIB commands in the Cascade probe station status-reporting structure is described earlier in this guide.

Setting Up for Remote Communication

This section explains how to set up and verify GPIB communication between the probe station and a remote host.

Setting GPIB Communication Parameters

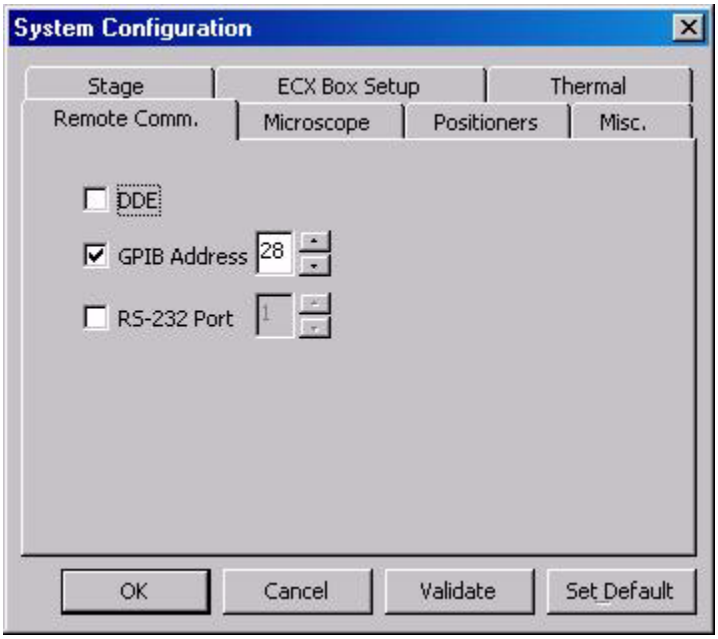
When operating as an instrument, the probe station requires only one address assignment in the host GPIB program.

If the remote host and the probe station specify different GPIB addresses, communication across the GPIB will not occur. Incorrect or inconsistent addressing is one of the most common causes of communication problems.

Use the following procedure to set up GPIB communication parameters on the probe station, then on your remote host. Instructions for setting up addressing and time-out values on your host are general only, and will differ depending on the GPIB software installed on your host.

To Set GPIB Parameters on the Probe Station PC

The following graphic shows the dialog from the Hardware Configurator for setting the GPIB device address.



To access this dialog, follow these steps:

1. Open the Hardware Configurator.
2. Highlight the Communications:GPIB component on the component list. If this component is not in the list, then it must be added.

3. Click 

if the component is running.
4. Click 

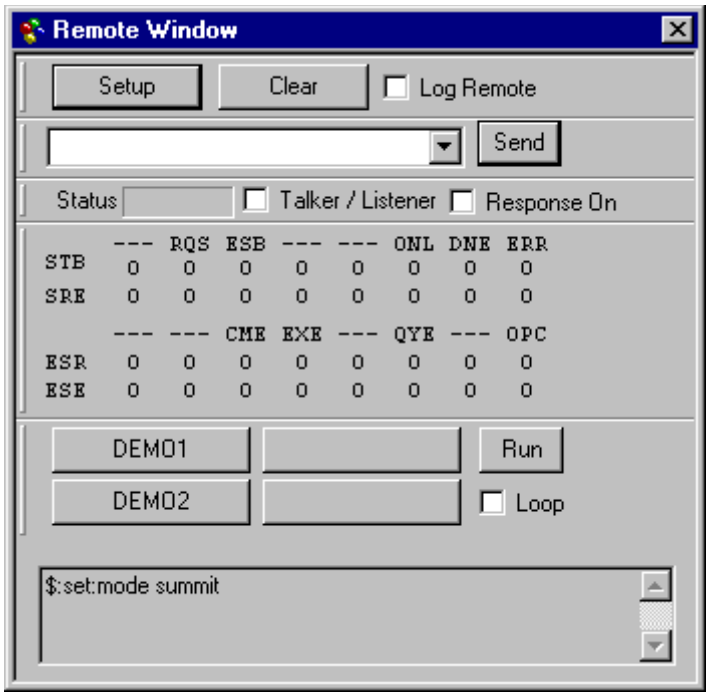
to open the dialog shown above.
5. Change the address as desired. The default value is 28. The address must be identical to the address assigned to the probe station by the host.
6. Save the config file by clicking Save 

.
7. Start the component by clicking Go 

.
8. Once the component is running, the station is ready to receive remote commands over the GPIB bus.

Nucleus User Interface Setup

See the Remote Window in the *Nucleus User Guide* for a complete description of this dialog.



Once you have set the probe station parameters, you are ready to set the GPIB parameters on your host.

To Set GPIB Parameters on the Host PC

1. Assign to the probe station the same address in the host GPIB map as is defined on the station. This address is usually 28.

2. Set the host time-out value for the probe station to an appropriate length of time (for example, 30.0 seconds).
3. Set other communication fields as defined by your host software. For example, the settings for detecting the end of a transition and the end of line termination character. Once communication parameters are defined, you are ready to verify communication between the probe station and remote host.

Example GPIB Parameters – Verifying GPIB Communication

This example shows how to use the National Instruments Win32 Interactive Control program to perform simple communications with Nucleus. This example may be used as a model when other control libraries are used instead of National Instruments.

Commands that are typed in by the operator are shown in **bold** text.

Table 3. Nucleus commands for National Instruments Win32 Interactive Control program

Command	Description
`: ibfind gpib0`	This selects the first interface card as GPIB0.
`gpib0: ibdev 0 28 0 13 1 0x0C0A`	Select and configure a device. See the text for a description of the parameters. See Table 4 for details.

`ud0: ibwrt "*idn?"`
`[0100]  ( cmpl )`
`count:  5`
	Request the device identification string.

`ud0: ibrd 100`
`[2100]  ( end cmpl )`
`count:  11`
`53 75 6d 6d 69 74 20 31  Summit 1`
`32 4b 0a`
	Read the device identification string.
	(Summit 12000-Series)

`ud0: ibwrt "$:set:mode summit"`
`[0100]  ( cmpl )`
`count:  17`
	Set the command interpreter mode to Summit.

`ud0: ibwrt "$:set:resp on"`
`[0100]  ( cmpl )`
`count:  13`
	Turn string responses on.

`ud0: ibwrt ":set:unit metric"`
`[0100]  ( cmpl )`
`count:  18`
	Set the units to metric.

`ud0: ibrd 100`
`[2100]  ( end cmpl )`
`count:  9`
`43 4f 4d 50 4c 45 54 45  COMPLETE`
`0a  .`
	Read the response to setting the units.

`ud0: ibwrt ":move:rel 2 0 100 none"`
`[0100]  ( cmpl )`
`count:  22`
	Move relative in the Y axis.

`ud0: ibrd 100`
`[2100]  ( end cmpl )`
`count:  9`
`43 4f 4d 50 4c 45 54 45  COMPLETE`
`0a`
	Read the response to the relative move.

`ud0: ibwrt ":move:abs? 2"`
`[0100]  ( cmpl )`
`count:  12`
	Request the position of the stage.

`ud0: ibrd 100`
`[2100]  ( end cmpl )`
`count:  24`
`2b 30 30 30 30 30 30 20  +000000`
`2b 30 30 30 35 30 30 20  +000500`
`2b 30 30 30 31 30 30 0a  +000100`
	Read the position of the stage.

The parameters for the National Instruments *ibdev* routine are as follows:

Value	Parameter Name	Description
0	Board Index	Selects GPIB0 as the current interface board to communicate through.
28	Device Primary Address	Selects the address of Nucleus.
0	Secondary Device Address	Nucleus does not use a secondary address.
13	Timeout value	The value is symbolic. The value 13 means 10 seconds.
1	EOT Mode	Sets the EOT line on the last character of a send to Nucleus.
0x0C0A	EOS Mode	Line feed (0x0A) character is used as a terminator. Terminate reads when the line feed is encounter. Assert the EOI line at the end of a read.

Typical GPIB Communication Problems

When lab equipment is shared between users, communication addresses and connections can be unexpectedly changed. Check for the following common problems:

- The bus cable is not plugged into both components
- The bus cable is plugged into the wrong component
- The address assigned to the probe station is incorrect
- The address assigned to the probe station is already assigned to another instrument
- Power to the probe station is not on
- The host has not taken remote control of the probe station
- The probe station is not specified on your host as being online for GPIB communication
- The host time-out values assigned to the probe station are too short for the host to communicate with the station

If you still cannot communicate with your instrument, call the manufacturer of your GPIB card for service.

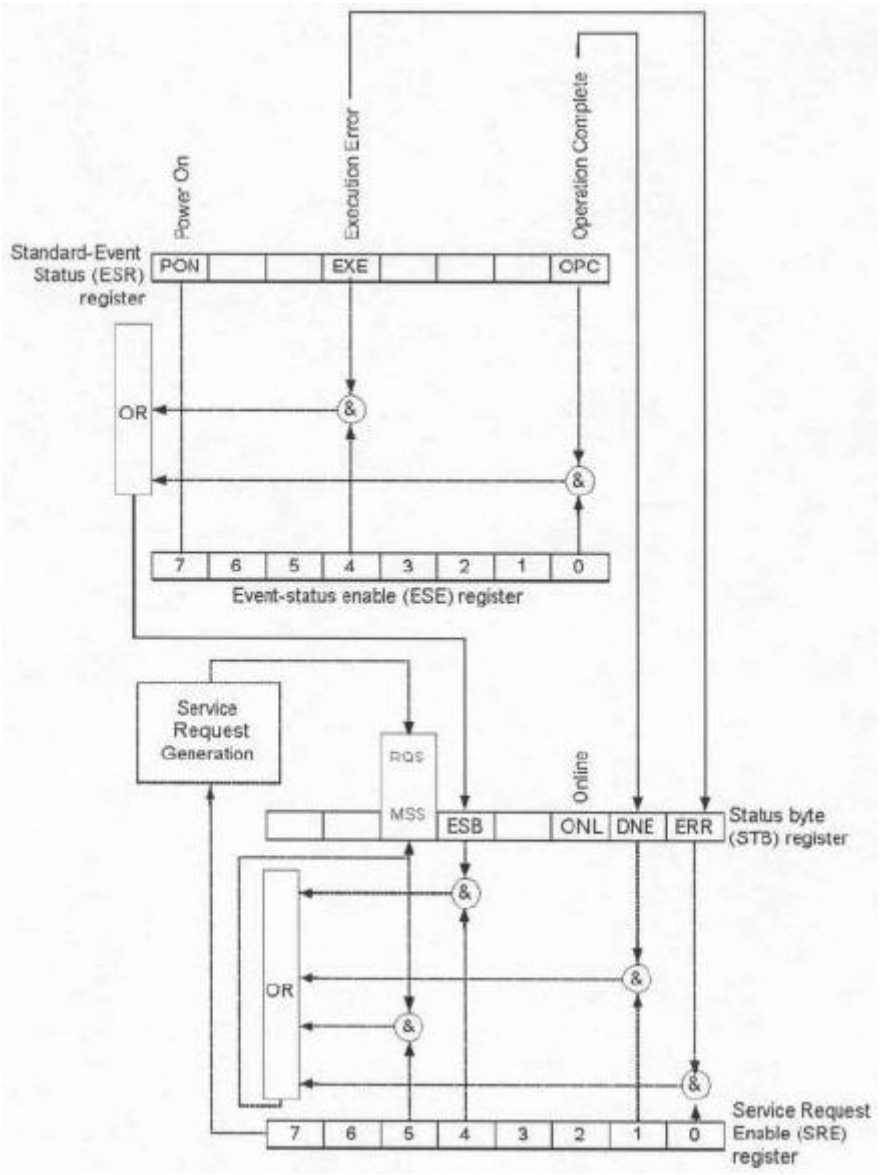
Status Reporting and Time-Outs

Cascade's probe-station status-reporting structure is shown below. Refer to this figure throughout this discussion to see how bit settings are enabled and passed from one register to another.

IEEE 488.1 and 488.2 provide several instrument-defined bits in the status-byte (STB) register. Cascade probe stations define bit 0 of the STB register to be an error (ERR) flag. If a command, query, or execution error occurs, the probe station sets the execution error (EXE) bit in the event-status (ESR) register. The probe station then copies a true value (1) into the ERR bit of the STB register.

Bit 2 of the STB register in the online (ONL) bit. This bit indicates the state of the Host Mode setting. Clicking the Host Mode check box on the Remote Window on the Nucleus UI can toggle the host mode.

Status-reporting structure



This figure also shows that the probe station defines bit 1 of the STB register to be a done-bit (DNE). This bit corresponds to the OPC bit in the ESR register. When the probe station sets the OPC bit to 1, the DNE bit is also set to 1. This allows the host to use a serial poll to quickly determine if operations have completed successfully.

For example, you might not be able to begin a measurement until the device is positioned properly. In your host program, you can use the GPIB ***opc** command immediately after the **:move:absolute command**. Your host program can wait until the OPC bit is true before continuing to the next program instruction.

Enabling Events

The Event-Status Enable (ESE) register (see Status-reporting structure) allows you to enable specific events that occur in the ESR register. In this discussion, for simplicity, the value of the ESE register is 1. This means that all bits except bit 0 (which corresponds to the OPC bit) are disabled.

For example, the host program sends a command string with the last command being the **`*opc`** command. When the probe station completes the operations specified by the command string, it sets the OPC bit in the ESR register. When the OPC bit is set, that value is ANDed with the value of bit 0 in the ESE register. Because at least one event is true, the event-status bit (ESB) in the STB register is then set.

There are two registers you can use to enable events to cause service requests. The ESE register enables events that occur in the event-status register. This stops events from setting the event-status bit in the STB register. However, this does not prevent errors or operation-complete flags from being set in other bits of the STB register. For example, the ESR register OPC bit setting is copied into the DNE bit. Therefore, the STB register bit 1 can still be set true, regardless of the ESE register value.

The SRE register allows you to enable events in the status-byte register. For example, if a power-on state is reported and filters down to set the ESB bit true, you can set bit 5 of the SRE register to 0. This disables the power-on state, preventing it from causing a service request.

Reading the STB Register

You can read the STB register using two methods. You can send a GPIB *stb? query, or you serial-poll the STB register. Serial polling is described later.

Bit 6 of the STB register is the master-status/request-for-service (MSS/RQS) bit. The meaning of this bit depends on the command used to read the register. If you read the STB register with **`*stb?`**, a true value in the MSS bit indicates that a service request is pending or has occurred.

Clearing the Registers

Use the GPIB ***cls** command to clear the ESR and STB registers (reset them to 0). The ESE and SRE register are not cleared by ***cls**. Refer to the *GPIB Protocol Commands* for specific information about the registers affected by the ***cls** command.

NOTE

*Before you send a Meta or probe-station command, you must send a GPIB *cls command to clear old bit settings from the status registers. If you are using a serial poll without SRQs, you must send *cls as a separate string.*

GPIB Time-Out Values

One of the most common communication problems is a time-out value that is too short to allow the instrument to complete the task. You need to choose a time-out value that is long enough for any task your application might perform. For example, if the longest task is to slowly move the device from one location to another, a time-out value of 30 seconds may be sufficient. However, if your program requires the operator to perform a task such as aligning theta or loading a wafer, you may want to increase the time to 5 minutes or more.

The time-out value becomes particularly important if your program sends messages without checking the status to see if the operation completed or without reading the response string for verification. In many applications, relying on time-out values is not as effective as using SRQs or serial polling to synchronize the host and the probe station (see *Alternative Time-Out Methods*).

The time-out value on the probe station indicates how long the probe station will wait for the host to read a response sent by the probe station.

The time-out value used in the host program controls how long the host waits for the probe station to respond after the host sends a command. Once the host has issued a command, the host should be ready to read the ASCII string returned by the probe station. If the remote task completes within the time allowed, the host reads the returned string, and is ready to execute another task.

If there was no time-out value for the probe station, the host would wait indefinitely for a response if the probe station could not finish the host-specified task. This causes the host to hang. The finite time-out value puts a limit on how long the host will wait for a response to come back from the probe station. If the time limit is reached without a response from the probe station, the host program will report an error, or be ready to perform some other task.

Choosing and Setting an Appropriate Time-Out Value

On the host, the probe-station time-out value is set in your host GPIB program. The time-out value you set must be long enough to allow you to complete the longest task you expect to begin. For example, if you expect only to remotely move the device from one location to another at a slow speed, a time-out value of 1 minute might be appropriate. But if you will be performing a manual operation such as **:probeplan:align**, you might want a time-out value of 10 minutes or more, or consider using a serial poll or SRQ handshaking to determine when the command execution is complete without any risk of a time-out.

For a National Instruments GPIB, you can increase the time-out value to approximately 10 minutes. You can disable the time-out by setting the value to 0. However, if you disable the time-out value and the probe station cannot complete a remote command, the host might hang, waiting indefinitely for the probe station to respond.

Alternative Time-Out Methods

Probe stations let you perform many tasks that can take a relatively long time to complete. For these tasks, consider using one of the following methods instead of relying on a time-out value:

- You can use a serial poll and/or SRQs with your own time-out handling or keyboard-check handling. See Using Service Requests (SRQs).
- You can disable or set a very long time-out value before sending the command to the probe station. After the command is completely executed and the host has

received the probe station "COMPLETE" message, you can reset the time-out value.

- You can cancel remote mode, perform the task and then invoke remote mode again.
- You can write your own loop on the host so that a time-out is not considered an error.

Handshake Methods

The method of handshaking used for remote communication must be appropriate for the instruments connected to the bus. For example, the handshake method used with a probe station allows the host to correctly determine when the probe station has completely finished executing a remote command.

This section uses excerpts from a program to illustrate remote-communication concepts. The following list summarizes the conventions used:

- Bits in a byte are numbered 0 through 7, with 0 being the least significant bit.
- The probe station uses positive logic: true = 1; false = 0.
- The variable **`prober`** represents the GPIB address of the probe station.

Using Serial Polls

Serial Polling is a means of handshaking with the probe station that allows the host program to perform other tasks while it is waiting for the station to finish an operation. The example code below shows how to send a **`*cls`** command and a move command followed by the **`*opc`** command to set the OPC bit when the move is finished.

NOTE

*The *cls command must be sent in a separate string, before the string of operation command is sent. However, the *opc command can be the last command in the operation-command string. This avoids timing problems in case the host starts a serial poll before the STB register has been cleared.*

```
// PollUntilDone
// This routine waits until the OPC bit is set in
// the prober station. In order to avoid an infinite
// loop, this should be modified to timeout after
// some period of time.
void PollUntilDone ( int device );
{
   CONST short DONE_BIT = 2;
   short status;

   while(1)
   {
      ReadStatusByte ( board_0, device, &status ) ;
      if ((status & DONE_BIT ) != 0)
         break;
   }
}
// This shows how PollUntilDone() might be used to
// detect when a move is completed. First step is
// to send the CLS command to clear out any previous
```

```
// done bit.
Send_Cmd ( device , "*CLS" );

         // Send the move command followed by the OPC command.
         // The commands are executed in order so the OPC will
         // not execute until the move command is complete.
Send_Cmd ( device , ":MOVE:CONTACT 2 ; *OPC ;" );

         // Wait for the done bit that means the move
         // is complete.
Poll_Until_Done ( device );
```

Using Service Requests (SRQs)

Service requests (SRQs) are a method of handshaking that allows the host to interpret an interrupt as a request for attention from a particular device. To control service requests, you use the event-status (ESR) and service-request enable (ESE) registers.

The ESE register allows you to enable specific events that occur in the ESR register to generate an SRQ. Assume that the value of the ESE register is 1. This means that all bits except bit 0 (the OPC bit) are disabled. So, only an operation-complete flag can filter through the status-reporting structure and generate an SRQ.

```
         // Clear out any previous done bit.
Send_Cmd( prober, "*cls");

         // Enable an SRQ for the DNE bit. When the Operation
         // complete bit is set, we will get an SRQ.
Send_Cmd( prober, "*sre 2");

         // Send a move and OPC commands. These are executed
         // in order so when the move command finishes, the
         // OPC bit will be set which generates the SRQ.
Send_Cmd( prober, ":MOVE_CONTACT 2 ; *OPC ;" );

         // Wait for the SRQ to occur.
WaitSRQ( board_0, &result );
```

Using String-Return Handshaking

The string-return method of handshaking is the simplest form of remote communication. When using this form of handshaking, you send a command to the specified instrument and wait for a response. After you receive an appropriate response, you send the next command.

The response strings to commands must be enabled with the following meta command:

```
$:set:resp on
```

The **`$:set:resp`** command instructs the probe station to return an ASCII string after each command is successfully completed (or an error occurs). After each command is successfully executed, the probe station returns the ASCII string "`COMPLETE`". If an error occurs, the probe station returns the ASCII error message instead. Also, if the command was a query, the probe station returns the response only (or the error message only). The first character in an error message is always the @ ("at") sign.

For GPIB communication, the string-return method of handshaking relies on a timeout value. The timeout value on the host indicates to the host how long the host should wait for a response from the probe station.

```
// Send a move command.
char Str[100];
strcpy( Str, ":MOVE:CONTACT 2" );
ibwrt( prober, Str, strlen(Str) );

         // Read back the response. The response
         // will not be ready to read until the
         // probe station is done moving.
ibrd( prober, Str, sizeof(Str) );
```

Understanding Command Execution

Cascade probe stations execute commands in sequence as they arrive on the bus. The probe station does not read a new command from the bus until it has finished executing the last command.

The GPIB commands (described in detail later) allow you to control the response the probe station sends the host after a remote command has executed. You can specify that the ASCII string "`COMPLETE`" or an error is returned, or that no response is returned.

Query commands always return a response, regardless of the settings of the Meta commands. This is because the host expects an ASCII string in response. If the probe station cannot return the expected query response (such as a set of coordinate values or a version string), the probe station returns an error.

DDE Server Communications

The Nucleus Software Dynamic Data Exchange Server (NuDdeSrv) application allows you to write automated test programs to control your probe station. The DDE Server accepts commands from other Windows-based applications, sends them to the probe station by way of the Command Server, and returns responses.

This section contains information about basic concepts that are used with Windows-based DDE applications as well as DDE Programming examples in C and Visual Basic.

See the *Nucleus User Guide* for information about the operation of the Nucleus DDE Server.

About DDE — Basic Concepts and Terminology

The DDE application is based on the DDE messaging system built into Windows. The concepts and terms presented here are common to all DDE applications.

NOTE

Refer to your Windows programming documentation for DDE Management Library (DDEML) information.

Client/Server Conversation

Communication with the DDE Server follows the client/server model. Essentially, two Windows-based programs carry on a "conversation" by posting messages to each other. These two programs are known as the "server" and the "client". A server, such as Nucleus DDE Server, is the program that has access to data (or system hardware) that the other program needs. A client is the program that requests and receives data from the server.

The client program initiates DDE conversations. The client broadcasts a message to all running Windows-based programs that indicates the type of information it needs. The server that has the information, in this case Nucleus DDE Server, responds and the conversation begins.

Application, Topic and Item

Conversations between a DDE Client and Server are opened with a pair of names that identify the application and topic that the client program needs to converse about. For Nucleus, the application and topic names are always the same (shown below).

NOTE

To see these strings in use, refer to DDE Programming Examples.

The item name is the specific data item or command that the client program is to control.

Application Name:	EDMAIN
Topic Name:	CMI Commands
Item Name:	Any of the remote commands that are supported by Nucleus. For example: :move:load 2

Type of Conversation

NOTE

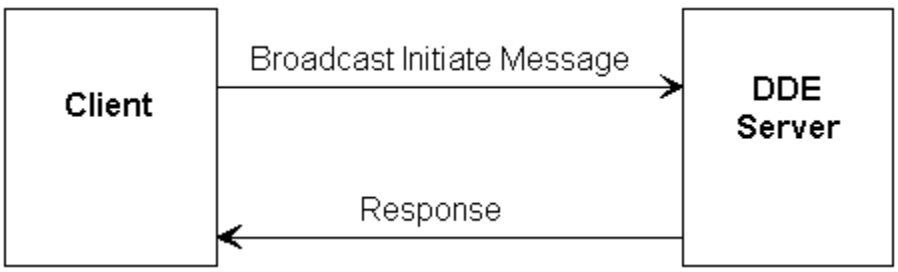
Refer to your Windows programming documentation for more information.

Windows supports three basic types of DDE interprocess conversations: cold link, hot link and warm link. Nucleus DDE Server uses only the cold link (or execution).

A DDE conversation begins when the client calls `DdeConnect()` which broadcasts a message to initiate the conversation.

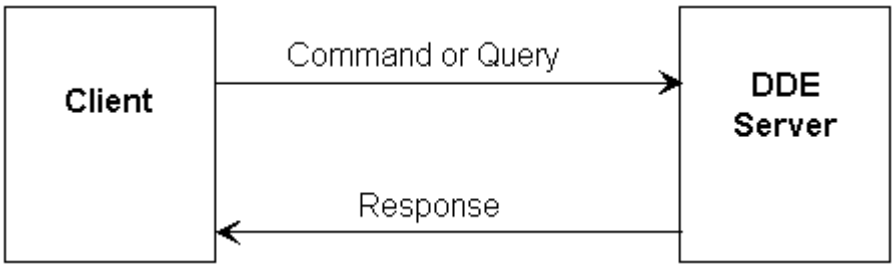
The server responds to the client with an acknowledge message

Client initiates conversation



The client sends commands to the DDE Server using the `DdeClientTransaction()` `Application Program Interface` (API).

DDE Server always returns responses

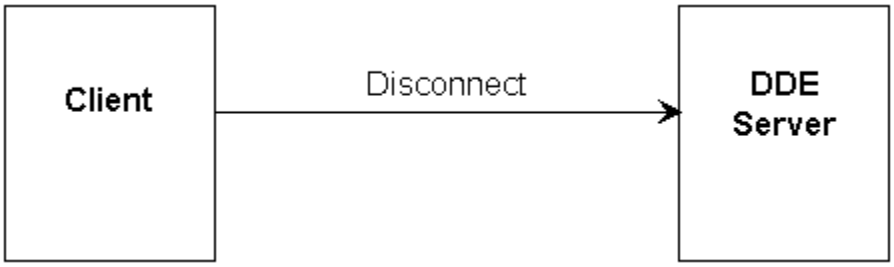


For each client call, the Nucleus DDE Server uses the `DdeClientTransaction()` return value as a pointer to a status string which contains one of the following values:

- A string containing `"COMPLETE"`, indicating that the transaction executed without error.
- A string containing the answer to a query made by the client. The format of the string depends on the nature of the requested data.
- A string containing an error message. The first character of any error-message string is the symbol @. The error message, a new line character combination, and the command string that generated the error condition follow this.

The conversation ends when the client posts a disconnect messages to the server. The DDE Server disconnects without returning an acknowledge message:

Client terminates DDE Server conversations



DDE Programming Examples

Two code fragments that demonstrate the use of the Nucleus DDE Server are provided.

C Example

The following code fragment shows how to send a command to the Nucleus DDE Server. It uses the Windows API DdeClientTransaction() to send a **:move:relative** command.

```
// The following code fragment uses Windows DDEML APIs
// to send a move command to the Nucleus DDE Server
// and receives a status string.
//
// Variables and initial values.
HCONV hConv = NULL;
```

```
DWORD idInst = 0;
HSZ hszService = NULL;
HSZ hszTopic = NULL;
HSZ hszCmd = NULL;
BOOL err = FALSE;
HDDEDATA transResult = NULL;
DWORD dwResult = 0;
char result[300];
// Need long timeout value to allow the station to move and
// respond.
const long TIMEOUT = 60000;
// Initialize the DDEML environment. Create an Instance
// that is used in many other calls to DDE.
if(DMLERR_NO_ERROR != DdeInitialize(&idInst,MyDDECallBack,
      APPCMD_CLIENTONLY, 0))
   err = true;

// Create String handles for the server name, topic and item.
if(!err)
{
   hszService = DdeCreateStringHandle(idInst, "EDMAIN",
      CP_WINANSI);
   hszTopic = DdeCreateStringHandle(idInst, "CMI Commands",
      CP_WINANSI);
   hszCmd = DdeCreateStringHandle(idInst,
      ":MOVE:REL 2 100 100 NONE",
      CP_WINANSI);
   err = (hszService == NULL || hszTopic == NULL ||
      hszCmd == NULL);
}

// Connect to the Nucleus DDE Server. (Open a conversation).
// Captain Picard would say, "Open a channel Mr. Wharf".
if(!err)
{
   hConv = DdeConnect(idInst, hszService, hszTopic, NULL);
   err = hConv == NULL;
   if(err)
      MessageBox(NULL, "Unable to make DDE connection.\n"
         "Make sure Nucleus is running.",
         "DDE CLIENT ERROR", MB_ICONSTOP);
}

// Send the command string to the server.
if (!err)
{
   transResult = DdeClientTransaction(NULL, 0, hConv, hszCmd,
      CF_TEXT, XTYP_REQUEST, TIMEOUT, NULL);

      // Read the result string. TransResult will be a
// valid data handle if the client transaction above
   // was successful, and NULL if it was not. This must be
   // checked since calls to DdeGetData with a NULL handle
   // cause GPF's.
```

```
   if(transResult)
      DdeGetData(transResult, (LPBYTE)result,
         sizeof(result), 0);
   // Display the result string.
   MessageBox(NULL, result,
      "RESULT", MB_ICONINFORMATION);
}
// Close the conversation.
if (hConv != NULL)
   DdeDisconnect( hConv );
// Delete the string handles.
if ((hszService != NULL) & (idInst != NULL))
   DdeFreeStringHandle( idInst, hszService );
if ((hszTopic != NULL) & (idInst != NULL))
   DdeFreeStringHandle( idInst, hszTopic );
if ((hszCmd != NULL) & (idInst != NULL))
   DdeFreeStringHandle( idInst, hszCmd );
// Clear out the DDEML environment.
if (idInst != NULL)
   DdeUninitialize(idInst);
..

// Since we are only doing requests from Nucleus, we don't
// expect to get callbacks to this routine. Nevertheless,
// it is necessary to create a routine with no action.
HDDEDATA CALLBACK MyDDECallBack( UINT wType,
   UINT wFmt, HCONV HConv, HSZ dataHandle1, HSZ dataHandle2,
   HDDEDATA data, DWORD myword1, DWORD myword2)
{
    return NULL;
}
```

Visual Basic Example

The following subroutine uses Visual Basic 6.0 to create a message box that you can use to send commands directly to the Nucleus DDE Server.

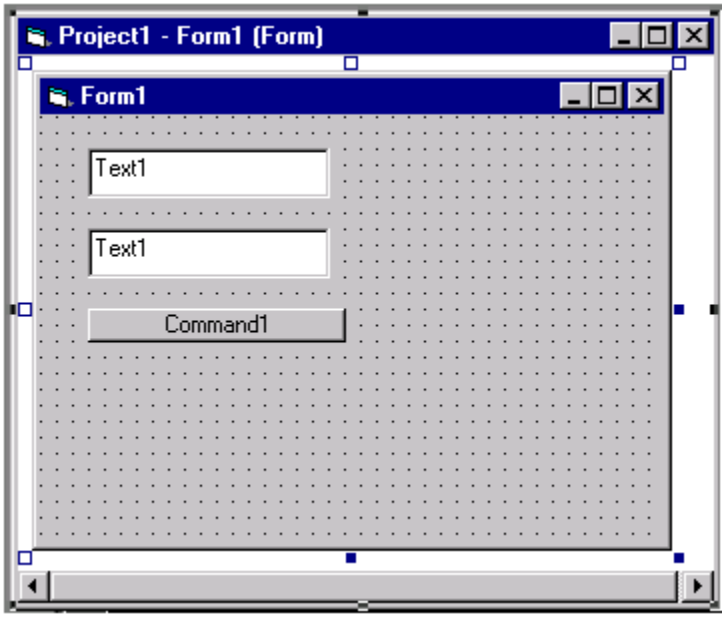
```
' Simple Visual Basic program that communicates
' with Nucleus via DDE.
'
' Steps:
' 1. Start Visual Basic 6.0.
' 2. Create a new Standard EXE program. VB will
'    make a blank form.
' 3. Add a text box. In the properties window
'    change the name of this text box to CmdText.
' 4. Add a second text box. Change its name
'    to RespText.
' 5. Add a button. VB will create the button
'    with the default name Command1.
' 6. Double click on the button (in design mode).
'    VB will open the code for the button. Enter
'    the routine as shown below.
' 7. Press F5 (start run).
```

```
' 8. Enter a command string in the first edit
'    box on the form (CmdText). For example, enter
'    "MOVE:REL 2 100 100 NONE" without the quote marks.
' 9. Make sure that the Nucleus DDE server is running.
' 10. Click the Command1 button. Nucleus will execute
'     the command and return a response string
'     which will be shown in the RespText box.

Option Explicit
Private Sub Command1_Click()
    RespText.LinkItem = CmdText.Text
    RespText.LinkTopic = "EDMAIN|CMI COMMANDS"
    RespText.LinkMode = vbLinkManual
    RespText.LinkRequest
    RespText.LinkMode = vbLinkNone

EndSub
```

Visual Basic form with edit boxes and the Command button added



RS-232 Communications

Cascade semiautomatic probe stations support remote communication across the RS-232 serial interface. This feature allows you to send commands and information between the probe station and a remote host. The probe station remote communication features help you integrate the probe station into an automated test environment.

Requirements

- Null modem cable
- Communications: RS-232 component configured with an Engine: Serial Port component

Setting up the Hardware Configurator

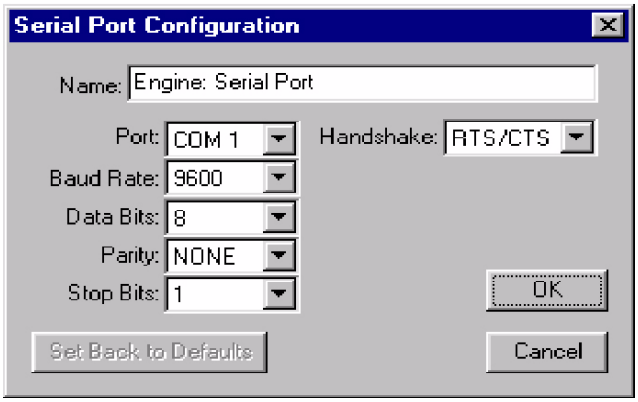
Two components are needed in the Hardware Configurator:

- Communications: RS-232
- Engine: Serial Port

To set up the Hardware Configurator:

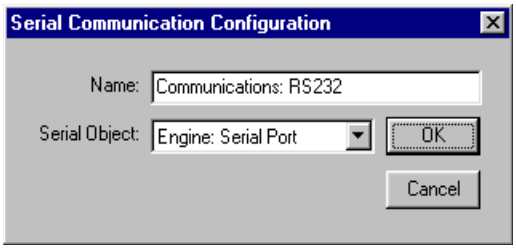
1. From the main toolbar in the Hardware Configurator select the "Add" button. Find the component "Engine: Serial Port" from the component, select the component, and click "OK".
2. Highlight the "Engine: Serial Port" component and press the "Setup" button on the main toolbar.

Serial Port Configuration dialog



3. The settings of this dialog must be consistent with the communication settings of the remote host. The settings in the example dialog may need to be changed, depending on the settings of the remote host.
4. Click on the "Add" button in the main toolbar of the Hardware Configurator, and find **Communications: RS232** in the component list. Select this component and press "OK".
5. Highlight the "Communications: RS232" component and press the "Setup" button on the main toolbar.

Serial Communication Configuration dialog



6. In the "Serial Object" combo box, select the Hardware Configurator Engine component that was setup in Step 2, and press OK.

The Hardware Configurator is now prepared to receive remote commands via RS-232.

Sending Commands

Each remote command must be terminated with an "\r" or an "\n" when sent from the remote computer.

Chapter

2 Command Dictionary

The command dictionary focuses primarily on SCPI commands (Standard Commands for Programmable Instruments) and lists the EG commands (Electroglas probing technology).

Summary of SCPI Commands

The commands listed are applicable to the following platforms:

- Elite Series
- Summit 12000-Series
- S300-Series
- Alessi 6100-Series

These syntax rules must be observed:

- You must use only the clean short or long form of a command. Anything in between will generate an error.
- Parameters must be separated by exactly one (1) character space.

Command	Description
`:align:wafer:busy?`	Returns the current state of the wafer alignment.
`:alig:waf:canc`	Cancels the alignment process before it completes.
`:alig:waf:star`	Starts an auto wafer alignment process.
`:aut:cle:act:data`	Resets all of the automation data to default values.
`:aut:mov:can`	Halts an automation operation.
`:aut:open`	Opens a previously saved automation file.
`:aut:run:temp:canc`	Stops the currently running temperature transition operation.
`:aut:run:temp:star`	Starts an automated temperature transition to the desired temperature.
`:aut:run:temp:stat?`	Returns the status of the currently running temperature transition operation.
`:aut:run:waf:canc`	Stops the currently running wafer automation setup operation.
`:aut:run:waf:star`	Starts an automated wafer operation.
`:aut:run:waf:stat?`	Returns the current status of a running automation setup wafer operation.
`:aut:set:soak:die:tim`	Sets the amount of time that the system waits after a move is complete before attempting to search and touch down on the DUT.

Command	Description
:aut:set:soak:die:tim?	Returns the amount of time the system waits after a move is complete before attempting to search and touch down on the DUT.
:aut:set:soak:temp:tim	Sets the amount of time that the system waits after the target temperature has been reached before attempting to search and touch down on the DUT.
:aut:set:soak:temp:tim?	Gets the amount of time that the system waits after the target temperature has been reached before attempting to search and touch down on the DUT.
:aut:save	Saves the current automation data setup using the "name" parameter.
:aut:set:act	Enables or disables the usage of automation features such as Auto XY, Auto Z, and Probe tip tracking.
:aut:set:act**?**	Returns "ON" or "OFF" to indicate whether automation is currently active.
:aut:set:act:name	Sets the automation name used to process wafer and needle information.
:aut:set:act:name?	Returns the current automation name used to process wafer and needle information.
:aut:dist	Sets the XY total radial distance that must be traveled before an Auto-Z operation is performed.
:aut:dist?	Returns the current distance setting that is used for Auto-Z.
:aut:guar	Defines the guard band that is used during Auto-Z operations.
:aut:guar?	Returns the current Auto-Z guard band distance.
:aut:sear:band	Sets the size of the search band that Auto Z uses when performing an auto focus operation (eVue only).
:aut:sear:band?	Returns the total size of the search band used by Auto Z during an auto focus operation.
:capt:vid	A fully qualified file name that defines where the captured video will be saved.
find:wafer:align:busy?	Returns TRUE if the alignment routine is currently running and FALSE if the alignment routine is no longer busy.
:find:wafer:align:cancel	Halts an alignment operation that is already running.
:find:wafer:align:start	Performs alignment only using live video and does not rely on previously trained targets.
:find:wafer:align:status?	Returns a string containing the current status of the theta alignment operation.
:find:wafer:cent:acc	Defines the accuracy with which to find the edges.
:find:wafer:cent:busy?	Returns TRUE, COMPLETE, or an error string.
:find:wafer:cent:canc	Cancels the find wafer center operation.
:find:wafer:cent:size	Defines the size of the wafer on chuck.
:find:wafer:cent:star	Starts a find wafer center operation. This command returns immediately and should be monitored with the :find:wafer:center:busy? command.
:find:wafer:cent:thic	Defines the thickness of the wafer parameter used during find center command.
:find:wafer:edge:busy?	Returns TRUE if the system is busy finding the edge. Returns FALSE when it is complete.
:find:wafer:edge:cancel	Halts the operation if a find edge operation is currently running.

Command	Description
:find:wafer:edge:start	Starts an edge detection operation and returns immediately.
:find:wafer:edge:status?	Returns a string containing the current status of the edge mapping operation.
:find:wafer:first:die:busy?	Returns TRUE if the find first die operation running. Returns FALSE when the operation is complete.
find:wafer:first:die:cancel	If the find wafer first die operation is currently busy, this command stops the process.
:find:wafer:first:die:start	Starts a search for the first die using the currently selected trained target.
:find:wafer:first:die:status?	Returns a string containing the current status of the find first die operation.
:find:wafer:set:active:name	Takes one parameter <name>, which defines which product name to make active for the :find:wafer commands.
:find:wafer:set:active:name?	Returns the name of the currently active wafer product.
:mov:abs	Moves the device z-axis to a safe position; then moves the x- & y-axis to specified coordinates, and then moves z-axis to a specified position.
:mov:abs?	Returns the device's current position in user coordinates.
:mov:abs:chuc	This command is the same as the :mov:abs command.
:mov:abs:opt	Allows the user to specify the type of coordinates being used in specifying the x-, y-, and z-axis positions.
:mov:align	Moves the device to the alignment position.
:mov:cent	Moves the device to the center (0,0) of the device-movement area.
:mov:cont	Moves the device to contact the wafer.
:mov:cont?	The query returns TRUE if the device is at contact position and FALSE if not.
:mov:down	Lowers the device to a predefined down position. Use :mov:sep instead of this command.
:mov:down?	The query returns TRUE if the device is down and FALSE if the device is up.
:mov:home	Initializes the hardware at the center (machine 0,0) by moving the device to the end of the limit switches, then locating the center (home) position.
:move:kill	Used to stop motion on the specified device.
:mov:load	Moves the device to the load-device position. The load position is either the front-center or front-left corner, depending on your model of probe station.
:mov:micr:abs	Moves the fine focus stage of the eVue microscope to an absolute position.
:mov:micr:abs?	Returns the current stage location of the eVue fine focus stage.
:mov:micr:foc	Performs an Auto-Focus operation.
:mov:prob:abs:die	Moves the device to the x, y die index position, as defined in the active probe plan file.
:mov:prob:abs:die?	The query returns the current probe plan x, y indexes.
:mov:prob:abs:ind	Moves the device to the nth die site. Only sites designated for testing in the active probe plan file are counted.

Command	Description
:mov:prob:abs:ind?	Returns the die-site number corresponding to the current device position.
:mov:prob:abs:ind:label	Moves the device to a die/subsite location.
:mov:prob:abs:ind:subs	Moves the device to a specified die and subsite, using the testable die index and the numeric index for the subsite.
:mov:prob:abs:loc	Moves the device to an x, y die index and subsite location in one move.
:mov:prob:abs:subs	Moves the device to the specified subsite index position, as defined in the active probe plan file.
:mov:prob:abs:subs?	Returns the current probe plan subsite indexes.
:mov:prob:abs:subs:lab	Moves the device to the subsite with the given label on the current die.
:mov:prob:firs:die	Moves the device to the first die site designated for testing in the active probe plan file.
:mov:prob:firs:subs	Moves the device to the first subsite on the current die designated for testing in the active probe plan file.
:mov:prob:last:die	Moves the device to the last die site designated for testing in the active probe plan file.
:mov:prob:last:subs	Moves the device to the last subsite designated for testing in the active probe plan file.
:mov:prob:next:die	Moves the device to the next die site designated for testing in the active probe plan file.
:mov:prob:next:site	Moves the device to the next site designated for testing in the active probe plan file. The site can be either a die site or a subsite, whichever comes next.
:mov:prob:next:subs	Moves the device to the next subsite on the current die designated for testing in the active probe plan file.
:mov:prob:pri:die	Moves the device to the prior die site designated for testing in the active probe plan file.
:mov:prob:pri:site	Moves the device to the prior site designated for testing in the active probe plan file. The site will be either a die site or a subsite, whichever came last.
:mov:prob:pri:subs	Moves the device to the prior subsite on the current die designated for testing in the active probe plan file.
:mov:prob:rel:die	Specifies relative moves within the active probe plan wafer map.
:mov:prob:rel:ind	Specifies relative moves within the active probe plan wafer map.
:mov:rel	Moves the device z-axis to a safe position; moves to the specified x- and y-axis positions, then moves the z-axis to a specified position. Moves are made relative to the current position.
:mov:rel:chuc	This command is the same as the :move:rel command.
:mov:scan:aax	Starts the device moving in the specified directions. Movement continues until the station receives a :move:stop command. The device moves at a specified velocity.
:mov:scan:xax	Starts the device moving in the specified x-axis direction. Movement continues until the station receives a :move:stop command. The device moves at a specified velocity.

Command	Description
:mov:scan:yax	Starts the device moving in the specified y-axis direction. Movement continues until the station receives a :move:stop command. The device moves at the speed a specified velocity.
:mov:scan:zax	Starts the device moving in the specified z-axis direction. Movement continues until the station receives a :move:stop command. The device moves at the specified velocity.
:mov:sep	Moves the device to break the contact between the probes and DUT.
:mov:sep?	The query returns TRUE if the device is at contact position and FALSE if the device is retracted.
:mov:stop:aax	Stops movement along all axes.
:mov:stop:xax	Stops device movement along the x-axis, but allows the device to continue moving along the z- and y-axes
:mov:stop:yax	Stops device movement along the y-axis, but allows the device to continue moving along the z- and x-axes
:mov:stop:zax	Stops movement along the z-axis; that is, all vertical chuck motion.
:mov:up	Raises the device.
:mov:up?	The query returns TRUE if the device is up and FALSE if it is down.
:prob:abs:ref	Moves the reference die to the specific X, Y location without moving the origin at the same time. It is different from :prob:ref which does move the origin when a new location is specified.
:prob:align	Sets the alignment position for a probe plan.
:prob:blink:die	Turns blink die on or off.
:prob:die:size	Sets the die size of the currently loaded wafer map file.
:prob:die:xsiz?	Returns the horizontal size of the die in the currently loaded wafer map.
:prob:die:ysiz?	Returns the vertical size of the die in the currently loaded wafer map.
:prob:flat	Sets the FLAT or NOTCH on a wafer map.
:prob:flat?	Returns a string that contains two values: Type and Size. Type is either FLAT or NOTCH and the Size depends on the Type.
:prob:grid:shif	Shifts the die pattern on the wafer map by X, Y microns.
:prob:grid:shif?	Returns the current shift of the die pattern on the wafer map.
:prob:load	Loads and activates a probe plan file.
:prob:load?	The query returns the current probe plan filename, if any.
:prob:mark:die	Specifies the color used to identify die sites when you display a wafer map.
:prob:mark:die:val	Stores a floating-point value at the given die, subsite and parameter combination.
:prob:mark:subs:val	Stores a floating-point value at the given die, subsite and parameter combination.
:prob:mark:xyd	Similar to the :prob:mark:die command, but uses x, y coordinates to specify die location.

Command	Description
:prob:mark:xyd:val	Stores a floating-point value at the given die and parameter combination.
:prob:mark:xys:val	Stores a floating-point value at the given die, subsite and parameter combination.
:prob:nsub?	Returns the number of subsites marked for testing in the active probe plan file.
:prob:ntes?	Returns the number of die marked for testing in the active probe plan file.
:prob:orig	Used to set the origin for the currently loaded wafer map.
:prob:orig?	Queries the currently loaded wafer map for the origin die.
:prob:par:bin	Sets up the bins for the selected parameter.
:prob:par:bin?	Returns a string value for the parameter that was selected. The string value returned consists of three fields separated by spaces.
:prob:par:col	Sets the color values for the maximum and minimum bin of a particular parameter.
:prob:par:col?	Queries the color settings of a particular parameter.
:prob:par:desc	Sets a descriptive text field used to identify the parameter.
:prob:par:desc?	Returns the descriptive text field used to identify a parameter.
:prob:par:lab	Sets the label for the parameter.
:prob:par:lab?	Returns the label of the specified parameter.
:prob:par:lab:ord?	Returns the parameter labels in order.
:prob:par:val:cle	Clears all of the parameter data values for the current wafer map.
:prob:par:view	Changes the current view parameter in the wafer map window.
:prob:qual:size	Sets the quality area of a wafer.
:prob:qual:size?	Returns the quality area size of the current wafer map.
:prob:ref	Sets the reference die for the currently loaded wafer map.
:prob:ref?	Queries the reference die for the currently loaded wafer map.
:prob:save	Saves the currently loaded wafer map file from memory onto disk.
:prob:set:pal	Defines the colors in the palette.
:prob:set:subs	Defines a subsite location (device position relative to the die site).
:prob:set:subsite:lab:active?	Returns the test status of a subsite at the given index.
:prob:show	Controls the probe plan wafer map display.
:prob:str:size	This command sets the street size of the current wafer map in X and Y.
:prob:str:xsiz?	Returns the X street size of the current wafer map.
:prob:str:ysiz?	Returns the Y street size of the current wafer map.
:prob:subs:del	Deletes the subsite at the specified index.
:prob:subs:del:all	Deletes all of the subsites in the subsite list.
:prob:subs:lab?	Returns the descriptive text field used to identify the subsite.
:prob:subs:tot?	Returns the total number of subsites whether they are marked as active or not.

Command	Description
:prob:subs:xoff?	Returns the X offset associated with a given subsite.
:prob:subs:yoff?	Returns the Y offset associated with a given subsite.
:prob:test:die	Marks die to test for the currently loaded wafer map.
:prob:test:die?	Queries whether die was marked to test for the currently loaded wafer map.
:prob:test:res	Returns the result from a test analysis given the testable die index.
:prob:test:xy:res	Returns the result from a test analysis given the Column and row in the wafer map.
:prob:test:seq	Defines the stepping sequence of the currently load wafer map.
:prob:test:seq?	Returns a string describing the current stepping pattern.
:prob:waf	Sets the diameter of the currently loaded wafer map.
:prob:waf?	Returns the diameter of the currently loaded wafer map.
:prob:waf:ori	Sets the orientation of the current wafer map.
:prob:waf:ori?	Returns the orientation of the current wafer map.
:prob:xy:ori	Sets the orientation of the X and Y axes in the wafer map.
:prob:xy:ori?	Queries the positive direction of the x- and y-axis of the currently loaded wafer map.
:prof:use:waf	Defines whether or not a wafer map is used to define the profile locations used during a Z-Profile.
:prof:use:waf?	Returns whether or not a wafer map file is being used to define auto focus locations for a Z-Profile.
:prof:waf:busy?	Returns TRUE, COMPLETE, or error string, depending on if it is in the middle of doing a Z-Profile.
:prof:waf:canc	Cancels a Z-Profile operation and stops the stage.
:prof:waf:dens	Defines the density of profile points used during a Z-Profile.
:prof:waf:dens?	Returns the currently set ring density.
:prof:waf:fil	Defines the Z-Profile file name that is used when a Z-Profile operation starts.
:prof:waf:fil?	Returns that the current Z-Profile is "300mm Z-Profile Test Wafer.pro".
:prof:waf:foc:opt	Enables the focus optimizer if error recovery is enabled for Z-Profile.
:prof:waf:foc:opt?	Returns the current state of the Z profile focus optimize setting.
:prof:waf:guar	Defines the guard band between the Maximum and Minimum profile points that causes a profile to become invalid.
:prof:waf:mode	Sets the mode that a Z-Profile uses when the operation is started.
:prof:waf:mode?	Returns what mode Z-Profile is currently using.
:prof:waf:qual:scor	Defines the amount of difference that an Auto Focus score can have from the original Z-Profile point. Doesn't fail if score is higher than the original only on the low end.
:prof:waf:retr	Used in failure recovery for Z-Profile. Defined as the number of times the system retries the surrounding area of a profile location.
:prof:waf:sear:band	Defines the amount of total distance that the fine focus stage travels when it is searching for optimal focus during Z-Profile.

Command	Description
:prof:waf:sear:band?	Returns the distance, in microns, used by the fine focus stage during an auto focus operation using Z-Profile
:prof:waf:sear:dist	Used in failure recovery for Z-Profile. Defined as the distance away from the original profile location that the system will move when searching for a better target.
:prof:waf:size	Diameter that is used during profile.
:prof:waf:size?	Returns the current diameter that is used during a Z-Profile.
:prof:waf:star	Starts a Z-Profile operation.
:prof:waf:succ:perc	Indicates the amount of successful auto-focuses as a percent. 95% would mean that 95 out of 100 targets must pass for the profile to be successful.
:prof:waf:test:perc	Defines the success percent that a Z-Profile must have in order to be considered valid.
:prof:waf:test:perc?	Returns the percent of focus points that must be valid before a Z-Profile is considered valid.
:prof:waf:use:rec	Defines whether or not error recovery is used for Z-Profile.
:prof:waf:use:rec?	Returns the current setting for Z-Profile error recovery.
:rev:ori	This command reverses the polarity of the user coordinate system. The reversal only applies to the x- and y-axes.
:set:aut:ligh:aut	Enables or disables the automatic light switching mode of Auto XY correction.
:set:aut:ligh:aut?	Returns the current state of the automatic light switching mode.
:set:aut:ligh:on:del	Defines the amount of time that automatic light switching waits before doing the vision pattern search.
:set:aut:ligh:on:del?	Returns the amount of delay time that the light switching mode is set to use.
:set:aut:ligh:off:del	Defines the amount of time to wait after the light has been turned off.
:set:aut:ligh:off:del?	Returns the amount of time that the system waits after turning the light off during automatic light switching.
:set:aux	Turns power to the probe station auxiliary power outlet on or off.
:set:busy?	Checks to see if the probe station is currently performing any operations.
:set:cham:purg	Controls the amount of MicroChamber air purge (OFF, MANUAL, QUICK, AUTO).
:set:cham:purg?	Returns the current MicroChamber air purge setting.
:set:comp	Turns compensation mode on or off.
:set:comp:fact	Numerically sets the compensation factors for movement along the x- and y-axes.
:set:comp:mode	Numerically sets the compensation factors for movement along the x- and y-axes
:set:cont	Sets the contact location for the Z stage and positioners. It defines the location that the DUT comes in contact with the probes
:set:cont?	Returns the contact position of the z stage.
:set:cont:acti	Turns the contact mode on and off. Optional parameters can be used to enable or disable dialog prompts. A value of "ON" enables prompts to be displayed. This is the default if no parameter is provided. "OFF" disables all prompting for setup of Z-Profile.

Command	Description
:set:cont:acti?	Returns ON or OFF depending on contact status.
:set:cont:ban	Sets the contact search band for contact mode.
:set:cont:ban?	Returns the currently set contact/search band used in contact mode.
:set:cont:mode	Turns on different contact modes.
:set:cont:mode?	Returns values for the different contact modes.
:set:cont:spee	Sets the search speed used for the Z-axis in contact mode.
:set:cont:spee?	Returns the current search speed for Z-axis that is used in contact mode.
:set:cont:use:aut	Enables Auto-Z functionality when Auto XYZ Correction mode is activated. Parameters must be ON or OFF.
:set:cont:use:aut?	Returns ON if Auto-Z is enabled in Auto XYZ Correction mode, otherwise returns OFF.
:set:cont:use:zpr	Enables Z-Profile when programmable contact mode is activated.
:set:cont:use:zpr?	Returns ON if Z-Profile is enabled in programmable contact mode, otherwise returns OFF.
:set:delay	Specifies a time in milliseconds for the probe station to delay after each operation.
:set:delay?	Returns the current delay time.
:set:edg	Turns edge sense mode on or off.
:set:edg?	Returns the status of contact mode: enabled or not enabled.
:set:edg:over	Sets the distance the chuck rises after contact is made with the edge sense probe.
:set:edg:over?	Returns the current overdrive amount that is set in contact mode.
:set:edg:swit	Identifies the non-contacted position of the switch in the edge sense probe.
:set:edg:swit?	Returns the current switch type that is defined for use with edge sense.
:set:host	GPIB only. If a GPIB switch box is connected, this command closes (on) or opens (off) the switch connecting the Summit and Host connectors.
:set:host?	Returns ON or OFF depending on whether the host is connected or not.
:set:ink	Used to fire the inker.
:set:joys:enab	Enables or disables the joystick.
:set:joys:enab?	Queries whether or not the joystick is enabled.
:set:joys:mode	Changes the motion control mode used by the joystick (SCAN, SCANAUTOZ, SCANZ, INDEX, DIE, SUBINDEX, JOG, THETA).
:set:joys:mode?	Returns the currently-defined motion mode for joystick.
:set:ligh	Turns the microscope light on or off.
:set:man	Toggles Nucleus from semiautomatic to manual mode.
:set:man?	Returns ON if the probe station is operating in manual mode and OFF if operating in semiautomatic mode.
:set:micr:AGC	Enables/disables the use of Automatic Gain Control for the microscope.

Command	Description
:set:micr:AGC?	Queries the state of Automatic Gain Control for the microscope.
:set:micr:auto:ill	Runs an auto illumination operation at the current location.
:set:micr:brig	Sets the microscope's brightness level. On eVue systems, it changes the software brightness level. Possible values 0.0 to 1.0.
:set:micr:brig?	Returns the microscope's brightness level value. On eVue returns the brightness level from 0.0 to 1.
:set:micr:cont	Defines the amount of software contrast. Possible values are 0.0 to 1.0
:set:micr:cont?	Returns the amount of software contrast being used.
:set:micr:exp	eVue only. Exposure level in ms from 0.5 to 100.
:set:micr:exp?	eVue only. Returns the exposure level.
:set:micr:zoom	Sets the zoom value used by the A-Zoom microscope. This command also works in the QUAD screen mode.
:set:micr:zoom?	Returns the zoom value used by the A-Zoom microscope. This command has an optional parameter that can be used to query the different zoom levels of each optical path
:set:pres	Presets the coordinate system coordinates for :move:absolute device movements by assigning the specified user coordinates to the current device location.
:set:sal	Turns on and off software alignment mode.
:set:sal?	Returns ON if the probe station is operating in software alignment mode and OFF if it is not.
:set:sep	Sets the separate distance away from contact used in auto z-stepping.
:set:sep?	Returns the separate distance of the z-stage.
:set:station?	Returns the probe station's device identification number.
:set:unit	Specifies the measurement units to be used for all input/output fields.
:set:unit?	Returns the unit of measurement that is being used by distances and coordinates.
:set:used	Controls the use of the device ID parameter for many SCPI commands.
:set:used?	Queries the current setting of the device ID in use.
:set:vac	Turns the central chuck vacuum on or off. Does not affect vacuum on corners or auxiliary devices.
:set:vac?	Returns ON or OFF depending on the state of the vacuum output line (not a sense line).
:set:vac:aux	Turns vacuum on or off to the auxiliary chucks. Elite 300 only.
:set:vac:aux?	Returns ON or OFF depending on the vacuum control line for the requested auxiliary chuck. Elite 300 only.
:set:vac:rin	Enables vacuum to the rings of the chuck. Elite 300 only.
:set:vac:rin?	Returns ON or OFF depending on the enable state of the specified vacuum ring. Elite 300 only.
:set:vel	Specifies the velocity for device movement along each axis. Does not apply to moves made in scan mode.
:set:vis	Enables/disables Auto XYZ Correction mode.

Command	Description
:set:vis:disp	Shows/hides the Auto XYZ Correction window.
:syst:beep	Causes the system to emit a sound.
:syst:conf	Returns a string response containing the system configuration value of the parameter requested.
:syst:del	Makes the probe station wait the specified length of time before executing the next instruction.
:syst:disp	Displays a one-line message on the probe station screen.
:syst:err?	Returns the last error recorded by the probe station.
:syst:iden?	Returns the probe station identification string.
:syst:lim:trav?	Returns the limits of travel for the specified channel.
:syst:oper:mode	Sets Nucleus software into REMOTE (all windows but video are hidden) or LOCAL (normal UI operations) mode.
:syst:oper:mode?	Returns REMOTE or LOCAL depending on what mode is currently active.
:syst:plat?	Returns the status of the platen arm.
:syst:vac:sens?	Returns ON if the system senses vacuum, and returns OFF if it does not sense vacuum.
:syst:vers?	Returns the probe station software release version.
:ther:acti	Sets the thermal chuck to be active.
:ther:acti?	Checks to see that the thermal chuck is active.
:ther:temp:curr?	Returns the current temperature of the thermal chuck in degrees Celsius.
:ther:deac	Deactivates the thermal and sets it to idle.
:ther:iden?	Identifies the type of thermal controller connected to the station.
:ther:temp:sett	Sets the thermal chuck to a temperature.
:ther:stat?	Returns the status of the thermal chuck.
:ther:temp:targ?	Returns the current target temperature.
:ther:temp:wind	Sets the window range of the thermal chuck.
:ther:temp:wind?	Returns the current window range.
:video:record:capture:rate	Set the video capture rate, timestamp and size.
:video:record:close	Closes the current video recording session.
:video:record:countdown:rate	Set the video countdown rate and enable mode.
:video:record:new	Creates a new video recording session.
:video:record:open	Opens an existing video session.
:video:record:save	Saves the current video recording session to a new file.
:video:record:start	Starts video recording.
:video:record:stop	Halts an active video recording.
:vis:need:sear:targ	Searches for a needle tip using a previously trained target.
:vis:need:tra:targ	Trains a needle tip using the current vision train box location.
:vis:read:targ:file	Reads in from file the trained target.
:vis:sear:targ	Performs a pattern req. search using Nucleus Vision.
:vis:set:matc:scor	Sets the threshold score used during vision search operations.
:vis:tra:targ	Trains a target using the region defined by the vision box in the video window.
:vis:writ:targ:file	Writes the trained target to disk.

SCPI Commands and Multiple Channels of Motion

This section provides details on SCPI device IDs and Galaxy Command Channels.

SCPI Command Device IDs in Nucleus Software

DevID	Description
1	Controls a Summit 10000 (not used in Nucleus v3.0 or later)
2	Used to control the chuck on Elite Series, Summit 12000-Series, S300 and Alessi 6100
3	Microscope channel
7	Left #1 Motorized positioner
8	Right #1 Motorized positioner
9	Left #2 Motorized positioner
10	Right #2 Motorized positioner
11	Left #3 Motorized positioner
12	Right #3 Motorized positioner

Galaxy Command Channels

Channel	Description
D	Controls the chuck (not used in Nucleus 3.0 or later)
S	Microscope channel
1	Left #1 Micropositione
2	Right #1 Micropositioner
3	Left #2 Micropositioner
4	Right #2 Micropositioner
5	Left #3 Micropositioner
6	Right #3 Micropositioner

SCPI Commands that Use Device ID

`:MOVe:ABSolute`	`:MOVe:UP`
`:MOVe:ABSolute?`	`:MOVe:UP?`
`:MOVe:ABSolute:OPTions`	`:SET:CONTACT`
`:MOVe:CENTer`	`:SET:CONTACT?`
`:MOVe:CONTact`	`:SET:EDGesense`
`:MOVe:CONTact?`	`:SET:EDGesense?`
`:MOVe:DOWN`	`:SET:EDGesense:OVER`
`:MOVe:DOWN?`	`:SET:EDGesense:OVER?`
`:MOVe:HOME`	`:SET:EDGesense:SWITch`
`:MOVe:RELative`	`:SET:EDGesense:SWITch?`
`:MOVe:SCAN:AAXis`	`:SET:MANual`
`:MOVe:SCAN:XAXis`	`:SET:MANual?`
`:MOVe:SCAN:YAXis`	`:SET:PRESet`
`:MOVe:SCAN:ZAXis`	`:SET:SALign`
`:MOVe:SEParate`	`:SET:SALign?`
`:MOVe:SEParate?`	`:SET:SEPARATE`
`:MOVe:STOP:YAXis`	`:SET:SEPARATE?`

:MOVe:STOP:ZAXis	:SET:VELocity
:MOVe:STOP:AAXis	
:MOVe:STOP:XAXis	

SCPI Commands that Accept but Ignore Device ID

For backwards compatibility with previous versions of Nucleus, the device ID parameter is always accepted, but ignored for these commands. For new implementations, we recommend avoiding device ID for these commands.

:MOVe:ALIGN	:PROBeplan:SAVe
:MOVe:LOAD	:PROBeplan:SHOW
:MOVe:PROBeplan:ABSolute:DIE	:PROBeplan:MARK:DIE
:MOVe:PROBeplan:ABSolute:DIE?	:PROBeplan:MARK:XYDie
:MOVe:PROBeplan:ABSolute:INDex	:PROBeplan:NSUBsites?
:MOVe:PROBeplan:ABSolute:INDex?	:PROBeplan:NTESted?
:MOVe:PROBeplan:ABSolute:INDex:LABel	:PROBeplan:SET:PAL
:MOVe:PROBeplan:ABSolute:INDex:SUBSite	:PROBeplan:SET:SUBSsite
:MOVe:PROBeplan:ABSolute:LOCation	:PROBeplan:SET:SUBSsite:LABel
:MOVe:PROBeplan:ABSolute:SUBSite	:SET:AUXiliary
:MOVe:PROBeplan:ABSolute:SUBSite?	:SET:BUSY?
:MOVe:PROBeplan:ABSolute:LABel	:SET:COMPensation
:MOVe:PROBeplan:FIRSt:DIE	:SET:COMPensation:FACTor
:MOVe:PROBeplan:FIRSt:SUBSite	:SET:COMPensation:MODE
:MOVe:PROBeplan:LAST:DIE	:SET:HOST
:MOVe:PROBeplan:LAST:SUBSite	:SET:HOST?
:MOVe:PROBeplan:NEXT:DIE	:SET:LIGHt
:MOVe:PROBeplan:NEXT:SITE	:SET:LIGHt?
:MOVe:PROBeplan:NEXT:SUBSite	:SET:STATion?
:MOVe:PROBeplan:PRIor:DIE	:SET:UNIT
:MOVe:PROBeplan:PRIor:SITE	:SET:UNIT?
:MOVe:PROBeplan:PRIor:SUBSite	:SET:VACuum
:MOVe:PROBeplan:RELative:DIE	:SYSTem:BEEP
:MOVe:PROBeplan:RELative:INDex	:SYSTem:ERRor?
:PROBeplan:ALIGn	:SYSTem:IDENtification?
:PROBeplan:LOAD	:SYSTem:VERSion?
:PROBeplan:LOAD?	

SCPI Command Descriptions

The command descriptions provide details regarding individual commands. Short commands, descriptions, parameters, examples and related commands are described as appropriate to each command.

:ALIGN:WAFER:BUSY?

Short Command	`:alig:waf:busy?`
Description	Returns the current state of the wafer alignment. An alignment can be started with :align:wafer:start. Possible return values are: SUCCESS, ERROR, BUSY. SUCCESS is returned when the wafer alignment procedure has been completed successfully. ERROR is returned if there was an error during the alignment. BUSY is returned if the alignment process is still working.
Parameters	None

Example	`:align:wafer:start`
	`COMPLETE`
	`:align:wafer:busy?`
	`SUCCESS`

Related Commands	*`:align:wafer:cancel`* *`:align:wafer:start`*

:ALIGN:WAFER:CANCEL

Short Command	`:alig:waf:canc`
Description	Once an alignment is started with the command :align:wafer:start, it can take several minutes to complete. The command :align:wafer:cancel can be used to stop the alignment process before it completes.
Parameters	None

Example	`:align:wafer:start`
	`COMPLETE`
	`:align:wafer:cancel`
	`COMPLETE`

Related Commands	*`:align:wafer:busy?`* *`:align:wafer:start`*

:align:wafer:start

Short Command	*Use Die Size*	`:alig:waf:star`
Description	Starts an auto wafer alignment process. When this command is sent, the vision train box in the video window must be on the repeating target image for the die. An auto alignment process can take a long time, so this command returns as soon as alignment starts. The status can be checked with `:align:wafer:busy?` and it can be canceled with `:align:wafer:cancel`.	
Parameters	*Use Die Size*	If this parameter is TRUE then the die size from the wafer map is used for alignment. If FALSE then it uses small steps to find the repeating pattern. This parameter is optional.

Example	`:align:wafer:start` `COMPLETE`
Related Commands	*`:align:wafer:busy?`* *`:align:wafer:cancel`*

:AUTOMATION:CLEAR:ACTIVE:DATA

Short Command	`:aut:cle:act:data`
Description	This command resets all of the automation data to default values. The active name is set to "Untitled". All needle training and reference data is cleared. This operation is similar to using "new" in the wafer map window.
Parameters	None
Example	`:automation:clear:active:data` `COMPLETE`
Related Commands	*`:automation:set:active:name`* *`:automation:set:active`* *`:automation:set:active?`* *`:automation:clear:active:data`* *`:automation:save`*

:AUTOMATION:MOVE:CANCEL

Short Command	`:aut:mov:can`
Description	Halts an automation operation. Move die commands using automation can take a long time if using die soak times. This command can be used to interrupt an automation step that is in progress. The chuck is left at separate.
Parameters	None
Example	`:automation:move:canel` `COMPLETE`
Related Commands	*See* Automation commands (`:automation`...) and the Automation section in the *Nucleus User Guide*.

:AUTOMATION:OPEN

Short Command	`:aut:open`
Description	This command opens a previously saved automation file. The currently opened automation setup is cleared and the setup information from the "name" parameter is loaded. Only one automation setup is active at a time.

Parameters	*Name*	String parameter that defines the name of the previously saved automation file to open. For names that contain spaces, use quotations as shown in the example.
Example	`:automation:open "Wafer XYZ Celadon"` `COMPELTE`	
Related Commands	*:automation:set:active:name* *:automation:set:active* *:automation:set:active?* *:automation:save*	

:AUTOMATION:RUN:TEMPERATURE:CANCEL

Short Command	`:aut:run:temp:canc`
Description	This command is used to stop the currently running temperature transition operation. It returns "COMPLETE" if the operation was stopped successfully. If the operation encounters an error, an error string will be returned.
Parameters	None
Example	`:automation:run:temperature:cancel` `COMPLETE`
Related Commands	*:automation:run:temperature:start* *:automation:run:temperature:status?* *:automation:clear:active:data* *:automation:save*

:AUTOMATION:RUN:TEMPERATURE:START

Short Command	`:aut:run:temp:star`	
Description	Given the currently loaded automation data setup and a temperature value, this command starts an automated temperature transition to the desired temperature. It returns immediately because this can be a 1+ hour operation. The status of the transition can be monitored with `:automaiton:run:temperature:status?`	
Parameters	*Temperature*	Floating point number that defines the new desired temperature.
Example	`:automation:run:temperature:start 200.0` `COMPLETE`	
Related Commands	*:automation:run:wafer:status?* *:automation:run:temperature:start* *:automation:clear:active:data* *:automation:save*	

:AUTOMATION:RUN:TEMPERATURE:STATUS?

Short Command	`:aut:run:temp:stat?`
Description	This command returns "COMPLETE" if the temperature transition operation has completed successfully. It returns "BUSY" if the operation is still working. If the operation encounters an error, an error string with '@' as the first character will be returned. The error string contains specific information on the type of error.
Parameters	None
Example	`:automation:run:temperature:status?` `BUSY`
Related Commands	*:automation:run:temperature:start* *:automation:run:temperature:start* *:automation:clear:active:data* *:automation:save*

:AUTOMATION:RUN:WAFER:CANCEL

Short Command	`:aut:run:waf:canc`
Description	This command is used to stop the currently running wafer automation setup operation. It returns "COMPLETE" if the operation was stopped successfully. If the operation encounters an error, an error string will be returned.
Parameters	None
Example	`:automation:run:wafer:cancel` `COMPLETE`
Related Commands	*:automation:run:wafer:start* *:automation:run:wafer:status?* *:automation:clear:active:data* *:automation:save*

:AUTOMATION:RUN:WAFER:START

Short Command	`:aut:run:waf:star`
Description	Given the currently loaded automation data setup, this command starts an automated wafer operation. It returns "COMPLETE" immediately once the operation is started or an error string. This can be a long operation so the remote command returns right away and can be monitored using status command. When this operation is complete, the needles are above the requested pad location and the chuck is at the separate distance.
Parameters	None
Example	`:automation:run:wafer:start` `COMPLETE`

Related Commands	*:automation:run:wafer:status?* *:automation:run:temperature:start* *:automation:clear:active:data* *:automation:save*

:AUTOMATION:RUN:WAFER:STATUS?

Short Command	`:aut:run:waf:stat?`
Description	This command returns "COMPLETE" if the automation setup wafer operation has completed successfully. It returns "BUSY" if the operation is still working. If the operation encounters an error, an error string with '@' as the first character will be returned. The error string contains specific information on the type of error.
Parameters	None
Example	`:automation:run:wafer:status?` `BUSY`
Related Commands	*:automation:run:wafer:start* *:automation:run:temperature:start* *:automation:clear:active:data* *:automation:save*

:AUTOMATION:SAVE

Short Command	`:aut:save`	
Description	This command saves the current automation data setup using the "name" parameter.	
Parameters	*Name*	String parameter that defines the file name to save the current automation data setup.
Example	`:automation:save "Wafer XYZ Celadon"` `COMPLETE`	
Related Commands	*:automation:set:active:name* *:automation:set:active* *:automation:set:active?* *:automation:clear:active:data*	

:AUTOMATION:SET:ACTIVE

Short Command	`:aut:set:act`
Description	This command enables or disables the usage of automation features such as Auto XY, Auto Z, and Probe tip tracking when moving from die to die in a wafer map.

Parameters	*Active*	Valid parameters are ON or OFF. • ON enables the previously configured automation features. • OFF disables automation.
Example	`:automation:set:active ON`	
Related Commands	*:automation:set:active?*	

:AUTOMATION:SET:ACTIVE?

Short Command	`:aut:set:act?`
Description	This comand returns "ON" or "OFF" to indicate whether automation is currently active.
Parameters	None
Example	`:automation:set:active?` `ON`
Related Commands	*:automation:set:active*

:AUTOMATION:SET:ACTIVE:NAME

Short Command	`:aut:set:act:name`	
Description	This command sets the automation name used to process wafer and needle information. "Untitled" is the active name after a clear operation.	
Parameters	*Name*	String parameter that defines the name of the automation setup. If using white space characters, the name must be enclosed within quotes.
Example	`:automation:set:active:name "Wafer XYZ Celadon"` `ON`	
Related Commands	*:automation:set:active:name?* *:automation:set:active* *:automation:set:active?*	

:AUTOMATION:SET:ACTIVE:NAME?

Short Command	`:aut:set:act:name?`
Description	This command returns the current automation name used to process wafer and needle information. "Untitled" is the active name after a clear operation.
Parameters	None
Example	`:automation:set:active:name?` `Wafer XYZ Celadon`

Related Commands	*:automation:set:active:name* *:automation:set:active* *:automation:set:active?*

:AUTOMATION:SET:SOAK:DIE:TIME

Short Command	*Time in minutes*	`:aut:set:soak:die:tim`
Description	Sets the amount of time that the system waits after a move is complete before attempting to search and touch down on the DUT.	
Parameters	*Time in minutes*	The amount of time, in minutes, that the system soaks at a die. Decimals are allowed for partial minutes.
Example	This example sets the amount of time that automation waits at a die before attempting to adjust to 1 minute. `:automation:set:soak:die::time 1`	
Related Commands	*:automation:set:soak:die:time?*	

:AUTOMATION:SET:SOAK:DIE:TIME?

Short Command	`:aut:set:soak:die:tim?`
Description	Returns the amount of time, in minutes, that the system waits after a move is complete before attempting to search and touch down on the DUT.

Example	Returns that the amount of time automation waits at a die before attempting to adjust is 1 minute.
	`:automation:set:soak:die:time?`
	`1.0`

Related Commands	*:automation:set:soak:die:time*

:AUTOMATION:SET:SOAK:TEMPERATURE:TIME

Short Command	*Time in minutes*	`minutes:aut:set:soak:temp:tim`
Description	Sets the amount of time that the system waits after the target temperature has been reached before attempting to search and touch down on the DUT.	
Parameters	*Time in minutes*	The amount of time, in minutes, that the system soaks after reaching the target temperature. Decimals are allowed for partial minutes.
Example	Sets the amount of time to wait after temperature is reached to 30 minutes. `:automation:set:soak:temperature:time 30.0`	
Related Commands	*:automation:set:temperature:time?*	

:AUTOMATION:SET:TEMPERATURE:TIME?

Short Command	`:aut:set:soak:temp:tim?`	
Description	Gets the amount of time that the system waits after the target temperature has been reached before attempting to search and touch down on the DUT.	
Example	Returns the amount of time, in minutes, the system waits :automation:get:soak:temperature:time? 30.0	
Related Commands	*`:automation:set:soak:temperature:time`*	

:AUTOZ:DISTANCE

Short Command	*Distance*	`:aut:dist`
Description	(This is an eVue-only command.) Sets the XY total radial distance that must be traveled before an Auto-Z operation is performed. For example: the XY stage location is 0, 0 and Auto-Z distance is set to 20000. If an Auto-Z is performed at this location, the calculated contact is used for all XY within 20000 microns of 0,0. Once the XY stage location goes outside this circular region, another Auto-Z operation is performed.	
Parameters	*Distance*	Defines XY radial distance that must be traveled before another Auto-Z operation is performed.
Example	Sets the guard band to 50 microns. `:autoz:distance 50`	
Related Commands	*`:autoz:distance?`* *`:autoz:guardband`* *`:autoz:guardband?`*	

:AUTOZ:DISTANCE?

Short Command	`:aut:dist?`
Description	Returns the current distance setting that is used for Auto-Z.
Parameters	None

Example	Returns that the current radial distance of Auto-Z is 20000 microns.
	`:autoz:distance?`
	`20000`

Related Commands	*`:autoz:distance?`* *`:autoz:guardband`* *`:autoz:guardband?`*

:AUTOZ:GUARDBAND

Short Command	*Guard Band*	`:aut:guar`

Description	Defines the guard band that is used during Auto-Z operations. If an Auto-Z operation is performed and a contact height is generated that is different by more than the guard band, an error is generated and the Auto-Z is cancelled. For example, the guard band is set to 50 microns and the current contact height of the stage is 10000. If an Auto-Z is performed and newly generated contact height is 10051, an error is generated and the Auto-Z operation is canceled. The same is true if the newly generated contact height is 9949.	
Parameters	*Guard Band*	Defines the distance for the guard band in microns.
Example	Returns that the current radial distance of Auto-Z is 20000 microns. `:autoz:guardband 50`	
Related Commands	*`:autoz:distance`* *`:autoz:distance?`* *`:autoz:guardband`*	

:AUTOZ:GUARDBAND?

Short Command	`:aut:guar?`
Description	Returns the current Auto-Z guard band distance. For details see *:autoz:guardband*.
Parameters	None

Example	Returns that the current guard band distance of Auto-Z is 50 microns.
	`:autoz:guardband?`
	`50`

Related Commands	*`:autoz:distance`* *`:autoz:distance?`* *`:autoz:guardband`*

:AUTOZ:SEARCH:BAND

Short Command	*Search Band*	`:aut:sear:band`
Description	Sets the size of the search band that Auto Z uses when performing an auto focus operation.	
Parameters	*Search Band*	Total distance in microns that Auto Z uses for auto focus

Example	Sets the search band to 200 microns.
	`:autoz:search:band 200`
	`COMPLETE`	

Related Commands	*`:autoz:distance`* *`:autoz:distance?`* *`:autoz:guardband`* *`:autoz:guardband?`* *`:autoz:search:band?`*	

:AUTOZ:SEARCH:BAND?

Short Command	`:aut:sear:band?`
Description	Returns the total size of the search band used by Auto Z during an auto focus operation.
Parameters	None

Example	Returns that Auto Z is using 200 microns for a search band.
	`:autoz:search:band?`
	`200`

Related Commands	*`:autoz:distance`* *`:autoz:distance?`* *`:autoz:guardband`* *`:autoz:guardband?`* *`:autoz:search:band`*

:CAPTURE:VIDEO

Short Command	*Filename*	`:capt:vid`
Description	Captures the current live video frame and saves it to disk as a BMP file. The file name must be a fully qualified path and file name. To include spaces in the file name, enclose it in quotes such as: "C:\Program Files\Cascade\Nucleus\UserData\Video Image.bmp".	
Parameters	*Filename*	A fully qualified file name that defines where the captured video will be saved.
Example	The following example captures the current video frame and saves it to the specified location. `:capt:vid "C:\Program Files\Cascade\Nucleus\UserData\VideoImage.bmp".`	

:FIND:WAFER:ALIGN:BUSY?

Short Command	*Filename*	`:find:waf:alig:busy?`
Description	Returns TRUE if the alignment routine is currently running and FALSE if the alignment routine is no longer busy.	
Example	Returns that the alignment is currently running. `:find:wafer:align:busy?`	

:FIND:WAFER:ALIGN:CANCEL

Short Command	*Filename*	`:find:waf:alig:canc`
Description	This command is used to halt an alignment operation that is already running.	

Example	To Stop a scene alignment during its operation send:
	`:find:wafer:align:cancel`
	`COMPLETE`

:FIND:WAFER:ALIGN:START

Short Command	*Filename*	`:find:waf:alig:star`
Description	Performs an alignment only using live video and does not rely on previously trained targets. The size parameter defines in microns the distance that is used during theta alignment. The larger the number, the more moves the system makes, but it is more accurate. This command returns immediately with the "`COMPLETE`" string if the process was started correctly. Otherwise an error is returned. To use 50mm's of travel for alignment, make the following call "`:find:wafer:align:start 50000`".	
Parameters	*Size*	Defines in microns the distance that is used during theta alignment.
Example	Larger numbers cause more system moves, but result in more accurate results. To use 50mm's of travel for alignment, make the following call "`:find:wafer:align:start 50000`".	

:FIND:WAFER:ALIGN:STATUS?

Short Command	*Filename*	`:find:waf:alig:stat?`
Description	Returns a string containing the current status of the theta alignment operation. Possible values are: • COMPLETE – Scene align for theta has completed successfully and the theta stage is currently located at the aligned angle. • BUSY – the system is still locating aligning the wafer. @ErrorString – The "@" symbol at the start signifies that an error has occurred. ErrorString contains the actual description of the error.	
Example	Indicates that an alignment is actively running. `:find:wafer:align:status?`	

:FIND:WAFER:CENTER:ACCURACY

Short Command	*Accuracy*	`:find:waf:cent:acc`
Description	Defines the distance threshold that the binary search algorithm will use to trigger an edge. Value is expressed in microns. Larger numbers means less accuracy, but a faster completion time.	
Parameters	*Accuracy*	Defines the distance threshold that the binary search algorithm will use to trigger that it has found the edge. 2000 microns is the default. Range can be 0 to 50000.

Example	Sets the accuracy of the wafer center to 2000 microns.
	`:find:wafer:center:accuracy 2000`
	`COMPLETE`

Related Commands	*:find:wafer:center:busy?* *:find:wafer:center:cancel* *:find:wafer:center:size* *:find:wafer:center:start* *:find:wafer:center:thickness*

:FIND:WAFER:CENTER:BUSY?

Short Command	`:find:waf:cent:busy?`
Description	Returns `TRUE`, `COMPLETE`, or an error string depending on the status of a Find Wafer Center operation. "`TRUE`" means the find wafer center is still busy searching. "`COMPLETE`" means the find wafer center has completed successfully. An error string is returned when the find wafer center has not competed successfully. The error string contains the "`@`" symbol at the start just like any other remote command with an error. Finding the wafer center can take a long time to complete, so this command can be used to monitor the status after a :find:wafer:center:start command. When the find wafer center operation is complete, the stage is located at the center of the wafer.
Parameters	None

Example	Find wafer center is currently busy.
	`:find:wafer:center:busy?`
	`TRUE`

Related Commands	*:find:wafer:center:accuracy* *:find:wafer:center:cancel* *:find:wafer:center:size* *:find:wafer:center:start* *:find:wafer:center:thickness*

:FIND:WAFER:CENTER:CANCEL

Short Command	`:find:waf:cent:canc`
Description	This command is used to cancel a Wafer Center operation. Locating the wafer center can take a long time to complete and if for any reason the system needs interrupt the operation this command can be used. :find:wafer:center:start returns immediately and the operation can be monitored with `:find:wafer:center:busy?`.
Parameters	None

Example	Aborts the find wafer center operation.
	`:find:wafer:center:cancel`
	`COMPLETE`

Related Commands	*:find:wafer:center:accuracy* *:find:wafer:center:busy?* *:find:wafer:center:size* *:find:wafer:center:start* *:find:wafer:center:thickness*

:FIND:WAFER:CENTER:SIZE

Short Command	*Size*	`:find:waf:cent:size`
Description	Defines the approximate size of the wafer in mm. This variable used to enhance the ability of the find wafer center to locate an edge. The wafer size must be set before a :find:wafer:center:start operation is performed.	
Parameters	*Size*	Defines the size of the wafer in mm. Valid range is 1 to 300.

Example	Sets the size of the wafer center to 200 mm.
	`:find:wafer:center:size 200`
	`COMPLETE`	

Related Commands	*:find:wafer:center:accuracy* *:find:wafer:center:busy?* *:find:wafer:center:cancel* *:find:wafer:center:start* *:find:wafer:center:thickness*	

:FIND:WAFER:CENTER:START

Short Command	`:find:waf:cent:star`
Description	This command starts a find wafer center operation. The find wafer center operation will use the values that have been setup with the other :find:wafer:center commands. This command returns immediately and can be monitored with the :find:wafer:center:busy? command. When find wafer center operation has completed the stage XY location will be the center of the physical wafer.
Parameters	None

Example	Starts up a find wafer center operation.
	`:find:wafer:center:start`
	`COMPLETE`

Related Commands	*:find:wafer:center:accuracy* *:find:wafer:center:busy?* *:find:wafer:center:cancel* *:find:wafer:center:size* *:find:wafer:center:start* *:find:wafer:center:thickness*

:FIND:WAFER:CENTER:THICKNESS

Short Command	*Thickness*	`:find:waf:cent:thic`

Description	Defines the approximate thickness of the wafer. This value will be used to determine if the Auto-Focus operation is off of the wafer or on it.	
Parameters	*Thickness*	Defines the approximate thickness of the wafer in microns. Default is 300 microns.

Example	Sets the thickness of the wafer center to 300 microns.
	`:find:wafer:center:thickness 300`
	`COMPLETE`	

Related Commands	*:find:wafer:center:accuracy* *:find:wafer:center:busy?* *:find:wafer:center:cancel* *:find:wafer:center:size* *:find:wafer:center:start*	

:FIND:WAFER:EDGE:BUSY?

Short Command	`:find:waf:edge:busy?`
Description	Returns TRUE if the system is busy finding the edge. Returns FALSE when it is complete.
Parameters	None

Example	Indicates that an edge find is currently running.
	`:find:wafer:edge:busy?`
	`TRUE`

Related Commands	*:find:wafer:edge:cancel* *:find:wafer:edge:start* *:find:wafer:edge:status?*

:FIND:WAFER:EDGE:CANCEL

Short Command	`:find:waf:edge:canc`
Description	If a find edge operation is currently running the cancel command halts the operation.
Parameters	None

Example	Stops a currently running edge find.
	`:find:wafer:edge:cancel`
	`COMPLETE`

Related Commands	*:find:wafer:edge:busy?* *:find:wafer:edge:start* *:find:wafer:edge:status?*

:FIND:WAFER:EDGE:START

Short Command	`:find:waf:edge:star`

Description	Starts and edge detection operation and returns immediately. During this operation the stage steps around the outside edge of a wafer or shard and updates the wafer image window on each step. At completion, a polygon file is generated that contains the shape of the shard.	
Parameters	*Filename*	Names the file that contains the polygon information. If a full pathname is required, quotes can be used for spaces.
Example	Starts an edge find running with the given output file. `:find:wafer:edge:start "C:\Test Output\Polygon File.txt"`	
Related Commands	*`:find:wafer:edge:busy?`* *`:find:wafer:edge:cancel`* *`:find:wafer:edge:start`* *`:find:wafer:edge:status?`*	

:FIND:WAFER:EDGE:STATUS?

Short Command	`:find:waf:edge:stat?`
Description	Returns a string containing the current status of the edge mapping operation. Possible values are: • COMPLETE – the edge mapping operation has completed successfully. • BUSY – the operation is still actively mapping the edge of the wafer. • @ErrorString – The "@" symbol at the start signifies that an error has occurred. ErrorString contains the actual description of the error. Example: :find:wafer:edge:status? returns "@Pixel to micron ratio has not been setup." If the pixel to micron ratio has not been defined yet.
Parameters	None

Example	Indicates that the edge find operation has completed successfully.
	`:find:wafer:edge:status?`
	`COMPLETE`

Related Commands	*`:find:wafer:edge:busy?`* *`:find:wafer:edge:cancel`* *`:find:wafer:edge:start`*

:FIND:WAFER:FIRST:DIE:BUSY?

Short Command	*Filename*	`:find:waf:firs:die:busy?`
Description	Returns TRUE if the find first die operation running. Returns FALSE when the operation is complete.	
Parameters	None	

Example	Returns that the find first die operation is currently active.
	`:find:wafer:first:die:busy?`
	`TRUE`	

Related Commands	*:find:wafer:first:die:cancel* *:find:wafer:first:die:start* *:find:wafer:first:die:status?*

:FIND:WAFER:FIRST:DIE:CANCEL

Short Command	`:find:waf:firs:die:canc`
Description	If the find wafer first die operation is currently busy this command stops the process.
Parameters	None

Example	Stops a currently running find first die operation.
	`:find:wafer:first:die:cancel`
	`COMPLETE`

Related Commands	*:find:wafer:first:die:busy?* *:find:wafer:first:die:cancel* *:find:wafer:first:die:start* *:find:wafer:first:die:status?*

:FIND:WAFER:FIRST:DIE:START

Short Command	`:find:waf:firs:die:star`
Description	Starts a search for the first die using the currently selected trained target. The active target can be selected on the "Edge Detect" page. This command returns immediately after starting the first die operation. The return string is either "COMPLETE" or an error string. The SCPI command "`:find:wafer:center:size <size>`" should be called before `:find:wafer:first:die:start` to define the diameter of the currently loaded wafer.
Parameters	None

Example	Begins an operational that finds the wafer edge and then searches for the first trained die.
	`:find:wafer:fist:die:start`
	`COMPLETE`

Related Commands	*:find:wafer:center:size* *:find:wafer:first:die:busy?* *:find:wafer:first:die:cancel* *:find:wafer:first:die:status?*

:FIND:WAFER:FIRST:DIE:STATUS?

Short Command	*Filename*	`:find:waf:firs:die:stat?`

Description	Returns a string containing the current status of the find first die operation. Possible values are: • COMPLETE – Find first die has completed successfully and the stage is currently located at the trained die position. • BUSY – the system is still locating the position of the first die. • @ErrorString – The "@" symbol at the start signifies that an error has occurred. ErrorString contains the actual description of the error.
Parameters	None

Example	Indicates that the currently running find wafer first die operation is still busy.
	`:find:wafer:first:die:status?`
	`BUSY`

Related Commands	*`:find:wafer:first:die:busy?`* *`:find:wafer:first:die:cancel`* *`:find:wafer:first:die:start`*

:FIND:WAFER:SET:ACTIVE:NAME

Short Command	*Filename*	`:find:waf:set:act:nam`
Description	Takes one parameter <name>, which defines which product name to make active for the `:find:wafer` commands.	
Parameters	*Name*	If <name> does not exist then an error is returned. Spaces can be used with <name> if quotes are added to the string.
Example	`:find:wafer:set:active:name "Wafer Type 3"`	
Related Commands	*`:find:wafer:first:die:busy?`* *`:find:wafer:first:die:cancel`* *`:find:wafer:set:active:name?`* *`:find:wafer:first:die:start`* *`:find:wafer:first:die:status?`*	

:FIND:WAFER:SET:ACTIVE:NAME?

Short Command	*Filename*	`:find:waf:set:act:nam?`
Description	Returns the name of the currently active wafer product.	

Example	Returns that the currently selected product is "Wafer Type 3"
	`:find:wafer:set:active:name?`
	`"Wafer Type 3"`	

Related Commands	*`:find:wafer:first:die:busy?`* *`:find:wafer:first:die:cancel`* *`:find:wafer:set:active:name?`* *`:find:wafer:first:die:start`* *`:find:wafer:first:die:status?`*	

:MOVE:ABSOLUTE

<table>
<tr><td rowspan="7">Short Command</td><td>devID</td><td></td><td rowspan="7"><code>:mov:abs</code>

The <code>:move:absolute:chuck</code> and the <code>:move:absolute</code> commands are identical in function.</td></tr>
<tr><td>x</td><td>(none)</td></tr>
<tr><td>y</td><td>(none)</td></tr>
<tr><td>z</td><td>(up)
(down)
(none)</td></tr>
<tr><td>theta</td><td>(none)</td></tr>
<tr><td>coordinate space</td><td>USER
STAGE</td></tr>
<tr><td>Description</td><td colspan="3">This command moves the device to the specified coordinates. The probe station moves the device's z-axis to a safe position before moving the device in x and y. Then it moves the z-axis to the supplied position after the device reaches the new location. The destination is not affected by wafer alignment, the load angle, the wafer map angle, or the wafer map axis settings.
The coordinate space USER is relative to the location currently defined as the user origin. You can use the <code>:set:preset</code> command to change the origin of the coordinate system.</td></tr>
<tr><td rowspan="4">Parameters</td><td>devID</td><td colspan="2">2 for Elite, Summit, S300 and Alessi 6100 emulation. See SCPI Command Device IDs in Nucleus Software.</td></tr>
<tr><td>x</td><td colspan="2">Specifies the x-axis coordinate to which the device will move when this command is executed. This number can be an integer, floating-point, or the character string none. If none, there is no change in the x-axis position.
The actual limits for the value of this parameter depend on the origin of the user coordinate system. The x coordinate value is interpreted as mils or microns. Use the :set:unit command to switch between mils and microns.</td></tr>
<tr><td>y</td><td colspan="2">Specifies the y-axis coordinate to which the device will move when this command is executed. The legal values for this parameter are the same as previously described for the x-axis coordinate.</td></tr>
<tr><td>z</td><td colspan="2">Specifies the device height at the end of the move. This number can be an integer, floating-point, or the character strings: up, down, or none. Entering none for the z position will cause the z-axis to return to its present position after separation is ensured for safe x, y, or theta movement.
For probe stations with pneumatic chucks, use one of the following character strings: up, down, or none. If none, the chuck height remains unchanged.
For probe stations with motorized chucks, this parameter can be an integer, floating-point, or the character strings up, down, and none.</td></tr>
</table>

Optional Parameters	*Theta*	The theta parameter defines the final destination for the theta axis. It is only valid for the chuck channel of motion. Theta is in milli-degrees. This parameter is required to specify the (*coordinate space*) parameter.
	Coordinate Space	Possible values are STAGE and USER. see `:move:absolute?` for a description of coordinates spaces. The (*theta*) parameter is required to specify this parameter.

Example	Lowers the chuck, moves to the specified user coordinates, and raises the chuck again. In this example, coordinates are in microns. The query form of the move command verifies the device position.
	`:mov:abs 2 11980 -34 5000`
	`:mov:abs? 2`
	`+011980 -000034 +5000`	

Related Commands	*`:move:absolute`* *`:move:absolute?`* *`:move:absolute:chuck`* *`:move:absolute:options`* *`:move:relative`* *`:set:unit`*	

:MOVE:ABSOLUTE?

Short Command	*devID*		`:mov:abs?`
	coordinate space	*(user) (theorywiththeta) (theory) (stage)*	
Description	This query returns the device's current position in user coordinates. The values returned are not affected by wafer alignment, the load angle, the wafer map angle, or the wafer map axis settings. If units are in microns, values are returned as six-digit numbers, with preceding 0s to make up the six digits. For example, if the current position is at x, y, and z coordinates 1, -524, and 5000, the query returns the following: `+000001 -000524 +005000` Otherwise, if units are in mils, returns are in floating-point format. Theta, in milli-degrees, is also returned in floating-point format. The coordinates are relative to the location currently defined as the user origin. You can use the `:set:preset` command to change the origin of the coordinate system.		
Parameters	*devID*	2 for Elite, Summit, S300 and Alessi 6100 emulation. See SCPI Command Device IDs in Nucleus Software.	

<table>
<tr><td>Optional Parameters</td><td>coordinate space</td><td>Possible values for coordinate space are: STAGE, USER, THEORYWITHTHETA, and THETA
USER - User coordinates are user defined coordinates for X and Y that take into account user defined zero positions, angle, thermal compensation, and corrected stage values from the ".CAL" file. The Z axis only applies compensation for the ".CAL" file, not user defined zero.
STAGE - coordinates are absolute positions that do NOT take into account user defined zero, angle, or thermal compensation. They do contain corrections from the ".CAL" file.
THEORYWITHTHETA - Same as STAGE, but theta position is returned with response. The theta parameter is padded with blank space characters. If theta is .29 then five spaces will be inserted. If theta is 123.0 then three spaces will be inserted.
THEORY - Same as STAGE.
NOTE
X, Y, Z and Theta are returned in integer formats if units are in metric, or in floating-point if in mils. If the units are in metric_float then floating point accuracy for metric units is returned. In addition if using the chuck device ID theta is returned.</td></tr>
<tr><td>Example</td><td colspan="2">Lowers the chuck, moves to the specified user coordinates and raises the chuck again. You then use the query form of the move command to verify the device position.
<code>:mov:abs 2 +11980 -34 5200</code>
<code>:mov:abs? 2</code>
<code>+011980 -000034 +5200</code>

Note that Stage, Theory, and TheoryWithTheta are all system coordinates.
<code>:mov:abs:opt 2 001234 004567 005000 TheoryWithTheta on on 123</code>
<code>:mov:abs? 2 TheoryWithTheta</code>
<code>+001234 +004567 +005000 123.00</code></td></tr>
<tr><td>Related Commands</td><td colspan="2">:move:absolute
:set:unit
:move:relative
:move:absolute:options
:move:absolute:chuck
:move:relative
:set:preset</td></tr>
</table>

:MOVE:ABSOLUTE:CHUCK

<table>
<tr><td rowspan="7">Short Command</td><td>devID</td><td></td><td rowspan="7"><code>:mov:abs:chuck</code>

The <code>:move:absolute:chuck</code> and the <code>:move:absolute</code> commands are identical in function.</td></tr>
<tr><td>x</td><td>(none)</td></tr>
<tr><td>y</td><td>(none)</td></tr>
<tr><td>z</td><td>(up)
(down)
(none)</td></tr>
<tr><td>theta</td><td>(none)</td></tr>
<tr><td>coordinate space</td><td>USER
STAGE</td></tr>
<tr><td>Description</td><td colspan="3">This command moves the device to the specified coordinates. The probe station moves the device's z-axis to a safe position before moving the device in x and y. Then it moves the z-axis to the supplied position after the device reaches the new location. The destination is not affected by wafer alignment, the load angle, the wafer map angle, or the wafer map axis settings.
The coordinates specified as USER are relative to the location currently defined as the user origin. You can use the <code>:set:preset</code> command to change the origin of the coordinate system.</td></tr>
<tr><td rowspan="4">Parameters</td><td>devID</td><td colspan="2">2 for Elite, Summit, S300 and Alessi 6100 emulation. See SCPI Command Device IDs in Nucleus Software.</td></tr>
<tr><td>x</td><td colspan="2">Specifies the x-axis coordinate to which the device will move when this command is executed. This number can be an integer, floating-point, or the character string none. If none, there is no change in the x-axis position.
The actual limits for the value of this parameter depend on the origin of the user coordinate system. The x coordinate value is interpreted as mils or microns. Use the :set:unit command to switch between mils and microns.</td></tr>
<tr><td>y</td><td colspan="2">Specifies the y-axis coordinate to which the device will move when this command is executed. The legal values for this parameter are the same as previously described for the x-axis coordinate.</td></tr>
<tr><td>z</td><td colspan="2">Specifies the device height at the end of the move. This number can be an integer, floating-point, or the character strings: up, down, or none. Entering none for the z position will cause the z-axis to return to its present position after separation is ensured for safe x, y, or theta movement.
For probe stations with pneumatic chucks, use one of the following character strings: up, down, or none. If none, the chuck height remains unchanged.
For probe stations with motorized chucks, this parameter can be an integer, floating-point, or the character strings up, down, and none.</td></tr>
</table>

Optional Parameters	*Theta*	The theta parameter defines the final destination for the theta axis. It is only valid for the chuck channel of motion. Theta is in milli-degrees. This parameter is required to specify the (*coordinate space*) parameter.
	Coordinate Space	Possible values are STAGE and USER. see `:move:absolute?` for a description of coordinates spaces. The (*theta*) parameter is required to specify this parameter.

Example	Lowers the chuck, moves the device to the specified user coordinates and raises the chuck again. In this example, coordinates are in microns. The query form of the move command verifies the device position.
	`:mov:abs:chuc 2 11980 -34 5000`
	`:mov:abs? 2`
	`+011980 -000034 +5000`	

Related Commands	*`:move:absolute?`* *`:set:unit`* *`:move:relative`* *`:set:preset`* *`:move:absolute:options`*	

:MOVE:ABSOLUTE:OPTIONS

Short Command	*devID*		`:mov:abs:opt`
	x	*(loc), (none*	
	y	*(loc), (none)*	
	z	*(loc), (none)*	
	coordinate space	*(user), (Theorywiththeta), (theory), (stage)*	
	move Z	*(on), (on), (loc)*	
	wait	*(off), (off)*	
	theta	*(loc)*	
Description	This command allows the user to specify the type of coordinates being used in specifying the x-, y- and z-axis and theta positions.		

Parameters	*devID*	2 for Elite, Summit, S300 and Alessi 6100 emulation. See SCPI Command Device IDs in Nucleus Software.
	x	Specifies the x-axis user coordinate to which the device will move when this command is executed. This number can be an integer, or the character string none. If none, there is no change in the x-axis position. The actual limits for the value of this parameter depend on the origin of the user coordinate system. The x coordinate value is interpreted as mils or microns. Use the `:set:unit` command to switch between mils and microns.
	y	The coordinate value (in microns or mils) that the device will move along the y-axis, relative to the current device position. Refer to the x-parameter description for information about the valid range values for this coordinate.
	z	Specifies the z-axis height at the end of the move relative to the current height. This number can be an integer, floating-point, or the character strings: **up, down,** and **none**. Entering **none** for the z position will cause the z-axis to return to its present position after separation is ensured for safe x, y, or theta movement. • For probe stations with pneumatic chucks, use one of the following character strings: **up, down,** or **none**. If **none**, the chuck height remains unchanged. • For probe stations with motorized chucks, this parameter can be an integer or the character strings **up, down,** and **none.**
	coordinate space	x, y, and z are returned in integer formats if units are in microns, or in floating-point if in mils. User coordinates are user defined coordinates that take into account user defined zero positions, angle, thermal compensation and corrected stage values form the ".CAL" file. TheoryWithTheta returns the theta parameter. Theory and Stage coordinates use the system's coordinates.
	moveZ	Either ON or OFF. If set to OFF, the z-axis will not move to separate before a horizontal move. If set to ON, the z-axis will lower to separate.
	wait	Required for backwards compatibility. However, it's implemented as ON, i.e., this command will not return until all moves have completed.
	theta	Specifies theta-axis. This number can be an integer, floating-point, or the character string none. Theta coordinates are in milli-degrees.

Example	The first command tells the probe station to move the chuck to a safe (retracted) position. Next, the device is moved to the position specified by the x- and y-type parameters, and then the z-axis is moved. Because the wait is set ON (wait OFF is not implemented), this command will not return until all moves have completed. The second command queries the station for its position in stage coordinates.
	`:mov:abs:opt 2 001234 004567 005000 stage on on`
	`:mov:abs? 2`
	`001234 004567 005000`

Related Commands	*:move:absolute?* *:move:contact* *:set:unit* *:move:absolute:chuck* *:move:relative* *:set:preset*

:MOVE:ALIGN

Short Command	`:mov:align`
Description	This command moves the device to the alignment position. The device z-axis is moved to its retracted position prior to xy-axis movement. You can use the `:move:contact`, `:move:separate`, and `:set:vacuum` commands with this command to control vacuum and chuck status.
Parameters	None

Example	Centers the device, then uses the `:move:align` command to move the the alignment position.
	`:mov:cent 2`
	`:mov:alig`

Related Commands	*:move:home* *:move:load*

:MOVE:CENTER

Short Command	*devID*	`:mov:cent`
Description	This command moves the device to the center (machine 0,0) of the probe station. The device z-axis is moved to its retracted position prior to xy-axis movement. You can use the `:move:contact`, `:move:separate`, and `:set:vacuum` commands with this command to control vacuum and chuck status.	
Parameters	*devID*	2 for Elite, Summit, S300 and Alessi 6100 emulation.

Example	Centers the device, then uses the `:move:contact` command to raise the chuck.
	`:mov:cent 2`
	`:mov:cont 2`	

Related Commands	*:move:contact* *:move:home* *:move:load*

:MOVE:CONTACT

Short Command	*devID*	`:mov:cont`
Description	This command moves the z-axis to the wafer contact position. It does not turn vacuum on or off. In general, you use this command after a device movement, when the z-axis was away from the wafer — for example, before the device moved. If the chuck is down and the probe station receives a move command, the probe station makes the device movement, but does not raise the chuck after the move is completed. The **`:move:contact`** command allows you to raise the chuck again.	
Parameters	*devID*	2 for Elite, Summit, S300 and Alessi 6100 emulation. See SCPI Command Device IDs in Nucleus Software (all but microscope).

Example	Shows the contact command used with one of the move-probe plan commands. In this example, the probe station lowers the chuck, moves the device, then repositions the device at a subsite. Once the device is correctly located over the subsite, the probe station raises the chuck.
	`:mov:sep 2`
	`:mov:prob:next`
	`:mov:prob:abs:subs 2`
	`:mov:cont 2`
	Shows the contact command used with the query to determine if the chuck was raised completely.
	`:mov:cont 2`
	`:mov:cont? 2`
	`TRUE`	

Related Commands	*:move:contact?* *:move:separate*	

:MOVE:CONTACT?

Short Command	*devID*	`:mov:cont?`
Description	This query returns the character string TRUE if the device is at contact position. Any position 10 microns below or above the set contact position will return TRUE, otherwise it returns FALSE.	
Parameters	*devID*	2 for Elite, Summit, S300 and Alessi 6100 emulation. See SCPI Command Device IDs in Nucleus Software (all but microscope).

Example	Shows the contact command used with the query to determine if the chuck was raised completely.
	`:mov:cont 2`
	`:mov:cont? 2`
	`TRUE`

Related Commands	*`:move:contact`* *`:move:separate`*

:MOVE:DOWN

Short Command	*devID*	`:mov:down`
Description	This command moves the device to the preset down position. The device xy axis does not change. The device remains down after the move to a new location, or until another command is given. Use the command `:move:separate` to move to the separate position. The command is the opposite of the `:move:up` command.	
Parameters	*devID*	2 for Elite, Summit, S300 and Alessi 6100 emulation. See SCPI Command Device IDs in Nucleus Software (all but microscope).

Example	Moves the device to the down position This example assumes that the device was up initially.
	`:mov:down 2`
	The following example shows the down command used with the query to determine if the device is at the down position.
	`:mov:down 2`
	`:mov:down? 2`
	`TRUE`	

Related Commands	*`:move:contact`* *`:move:down?`* *`:move:up?`* *`:move:up`*	

:MOVE:DOWN?

Short Command	*devID*	`:mov:down?`
Description	This query returns the character string TRUE if the device is at the preset down position, otherwise it returns FALSE.	
Parameters	*devID*	2 for Elite, Summit, S300 and Alessi 6100 emulation. See SCPI Command Device IDs in Nucleus Software (all but microscope).

Example	Shows that the device is at the down position.
	`:mov:down 2`
	`:mov:down? 2`
	`TRUE`	

Related Commands	*:move:up* *:move:up?* *:move:down*

:MOVE:HOME

Short Command	*devID*	`:mov:home`
Description	This command uses the hardware to locate the center (machine 0,0) of the station. It moves the device to the end of the limit switches and then moves the device to the center position. Use this command to initialize the probe station hardware. This command lowers the z-axis to move the device, then raises it again when it reaches the home location. This command is different from `:move:center` in that `:move:home` re-initializes the device, instead of merely moving to the currently identified center location. This command does not reset the probe station to the power-up state.	
Parameters	*devID*	2 for Elite, Summit, S300 and Alessi 6100 emulation. Note: applicable to chuck only.
Example	Shows the `:move:home` command: `:mov:home 2`	
Related Commands	*:move:center*	

:MOVE:KILL

Short Command	*devID*	`:mov:kill`
Description	Used to stop motion on the specified device. This command can be used to disable the auto hold correction of micropositioners.	
Parameters	*devID*	Device ID of the channel to kill motion on.
Example	Stops all motion of the first positioner. `:move:kill 7`	

:MOVE:LOAD

Short Command	`:mov:load`
Description	This command moves the device to the load position. The location of the load position depends on its preset value. Summit 12000-Series probe stations load in the front center.
Parameters	None

Example	Use the `:move:load` command to load a wafer on the chuck.
	`:mov:load`
	`When the probe station receives the command, it drops the chuck (if it is not already down) and moves the device to the front of the station to the load position.`

Related Commands	*:move:center* *:move:home*

:MOVE:MICROSCOPE:ABSOLUTE

Short Command	*position*	:mov:micr:abs
Description	Moves the fine focus stage of the eVue microscope to an absolute position. The eVue microscope has a 2mm range. The absolute location is in microns. 0 is the bottom of travel and 2000 is the top. Larger absolute locations moves the focus stage away from the chuck.	
Parameters	*position*	A value in microns between 0 and 2000.
Example	Moves the fine focus stage to the center of travel. `:move:microscope:abs 1000`	

:MOVE:MICROSCOPE:ABSOLUTE?

Short Command	`:mov:micr:abs?`
Description	Returns the current stage location of the eVue fine focus stage. The absolute location is in microns. 0 is the bottom of travel and 2000 is the top.
Parameters	None

Example	Moves the fine focus stage to the center of travel, then query the location, which returns 1000.
	`:move:microscope:abs 1000`
	`:move:microscope:abs?`
	`1000`

:MOVE:MICROSCOPE:FOCUS

Short Command	`:mov:micr:foc`	
Description	Performs an Auto-Focus operation. The full range of travel for the eVue microscope is 2mm. The Up Distance and Down Distance can be used together to form a search region. If the current focus stage position is 500 and the parameters are 1000, 1000 the Auto-Focus operation starts at 0 and moves up to 1500 as the top location.	
Parameters	*Up Distance*	The distance in microns that the fine focus stage moves up during an Auto-Focus.
	Down Distance	The distance in microns that the fine focus stage moves down during an Auto-Focus.

Example	Runs an auto focus with a search band of 1000 microns above and 1000 below the current location. The position of the focus stage at the end of the operation is the location that has the best focus.
	`:move:microscope:focus 1000 1000`
	`:move:microscope:abs?`
	`465`	

:MOVE:PROBEPLAN:ABSOLUTE:DIE

Short Command	*x* *y*	`:mov:prob:abs:die`
Description	Move to an absolute die location on the wafer. The move will be adjusted for the alignment angle (small angle). In addition, the move is affected by the load and wafer map angles, and the wafer map axis settings. There is no check to ensure that the target index position is marked for testing in the probe plan. The move takes place regardless, and it is not considered an error if the position is unmarked. The axis directions and origin relate only to the current probe plan. Using 0,0 for the destination will move the wafer to the reference die. Before using the move commands, you should define test sites with the probe plan Editor menus.	
Parameters	*x*	Specifies the x die index, defined in the probe plan.
	y	Specifies the y die index, defined in the probe plan.

Example	Tells the probe station to move to an index location. You then use the query form of the command to verify the device position.
	`:mov:prob:abs:die 30 40`
	`:mov:prob:abs:die?`
	`30 40`	

Related Commands	*`:move:probeplan:absolute:die?`* *`:move:probeplan:last:die`* *`:move:probeplan:prior:die`* *`:move:probeplan:first:die`* *`:move:probeplan:next:die`* *`:probeplan:load`*	

:MOVE:PROBEPLAN:ABSOLUTE:DIE?

Short Command	`:mov:prob:abs:die?`
Description	Returns the current die location. When the stage is on the reference die, 0,0 will be returned. The values returned will map to the values in the :move:prob:abs:die command. This means the values are affected by the load angle, the wafer map angle and the wafer map axis settings.
Parameters	None

Example	Moves the device to an index location. You then use the query form of the command to verify the device position.
	`:mov:prob:abs:die 30 40`
	`:mov:prob:abs:die?`
	`30 40`

Related Commands	*`:move:probeplan:last:die`* *`:move:probeplan:next:die`* *`:move:probeplan:prior:die`* *`:move:probeplan:absolute:die`* *`:move:probeplan:first:die`* *`:probeplan:load`*

:MOVE:PROBEPLAN:ABSOLUTE:INDEX

Short Command	*n*	`:mov:prob:abs:ind`
Description	This command moves the stage to the *nth* die site listed in the active probe plan. This command returns an error if the specified site has not been defined, such as a test die. Before using the move commands, you should define test sites with the probe plan Editor menus. The probe station lowers the chuck, moves to the specified location, and then raises the chuck again.	
Parameters	*n*	The die-site index number indicates the position of the site in the probe plan die-site list. The index is a positive integer up to five digits long.

Example	Loads a probe plan file and move the device to the 8th die site listed in the probe plan. The probe station is then queried for the current probe position.
	`:prob:load 65ghz_t1.wfd`
	`:mov:prob:abs:index 8`
	`:mov:prob:abs:index?`
	`8`
	`After loading the probe plan file, the probe station prompts you to position the probes on the alignment die of your substrate. When the probes are aligned, you can use the move command to go to the site, as shown above.`	

Related Commands	*`:move:probeplan:absolute:subsite?`* *`:move:probeplan:absolute:die?`* *`:move:probeplan:first:subsite`* *`:move:probeplan:next:site`* *`:move:probeplan:prior:die`* *`:move:probeplan:prior:subsite`* *`:move:probeplan:absolute:die`* *`:move:probeplan:first:die`* *`:move:probeplan:last:die`* *`:move:probeplan:next:subsite`* *`:move:probeplan:prior:site`* *`:move:probeplan:absolute:index?`*	

:MOVE:PROBEPLAN:ABSOLUTE:INDEX?

Short Command	`:mov:prob:abs:ind?`
Description	This query returns the current die site number that is under the probes. If the probes are not over a die site that is selected for testing, then an error condition is returned.
Parameters	None

Example	The following commands load a probe plan file and move the device to the 8th die site listed in the probe plan. The probe station is then queried for the current probe position.
	`:prob:load 65ghz_t1.wfd`
	`:prob:alig`
	`:mov:prob:abs:ind 8`
	`:mov:prob:abs:ind?`
	`8`
	`After loading the probe plan file, the probe station prompts you to position the probes on the alignment die of your substrate. When the probes are aligned, you can use the move command to go to the site, as shown above.`

Related Commands	*:move:probeplan:absolute:index*

:MOVE:PROBEPLAN:ABSOLUTE:INDEX:LABEL

Short Command	*Die index* *Subsite label*	`:mov:prob:abs:ind:lab`
Description	This command will move to a die/subsite location. The die parameter is defined as the Nth testable die and the subsite is defined by the label. The label can include spaces. If spaces are included, the label must be enclosed in quotes.	
Parameters	*Die index*	Testable die index of the currently loaded wafer map.
	Subsite label	Label of the subsite for the move.

Example	Moves to the subsite labeled "voltage" of the 11th testable die.
	`:mov:prob:abs:ind:lab 11 voltage`
	`:mov:prob:abs:ind:lab 11 "voltage out"`	

:MOVE:PROBEPLAN:ABSOLUTE:INDEX:SUBSITE

Short Command	*Die index* *Subsite label*	`:mov:prob:abs:ind:subs`
Description	This command will move to a specified die and subsite, using the testable die index and the numeric index for the subsite.	
Parameters	*Die index*	Testable die index of the currently loaded wafer map.
	Subsite label	Subsite index position of the currently loaded wafer map.
Example	Moves to the 5th subsite of the 11th testable die. `:mov:prob:abs:ind:subs 11 5`	

:MOVE:PROBEPLAN:ABSOLUTE:LOCATION

Short Command	*die x* *die y* *subsite*	`:mov:prob:abs:loc`

Description	This command moves the stage to a subsite location on a die on the wafer. The direction of the move is affected by the load angle, the wafer map angle and the wafer map axis settings (the same as the :move:prob:abs:die command). The probe station lowers the chuck, moves to the specified location, and then raises the chuck again.	
Parameters	*die x*	Specifies the *x* die index, defined in the probe plan.
	die y	Specifies the *y* die index, defined in the probe plan.
	subsite	The *n*th subsite of the specified die.

Example	Load sa probe plan file and then moves the device to the die index 3,2 subsite location 4.
	`:prob:load 65ghz_t1.wfd`
	`:mov:prob:abs:loc 30 40 4`	

Related Commands	*`:move:probeplan:absolute:subsite?`* *`:move:probeplan:absolute:die`* *`:move:probeplan:first:subsite`* *`:move:probeplan:next:site`* *`:move:probeplan:prior:die`* *`:move:probeplan:prior:subsite`* *`:move:probeplan:first:die`* *`:move:probeplan:last:die`* *`:move:probeplan:next:subsite`* *`:move:probeplan:prior:site`* *`:move:probeplan:absolute:index?`*	

:MOVE:PROBEPLAN:ABSOLUTE:SUBSITE

Short Command	*n*	`:mov:prob:abs:subs`
Description	This command moves the stage to the specified subsite on the current die. The subsite itself is an offset of x- and y-axis coordinates from the current die site listed in the active probe plan file. The probe station lowers the chuck, moves to the specified location, and then raises the chuck.	
Error Condition	This command returns an error if the specified subsite has not been defined.	
Parameters	*n*	The die-site index number indicates the position of the site in the probe plan die-site list. The index is a positive integer up to five digits long.

Example	The `:move:probeplan:absolute:subsite` command is usually used with the **`:move:probeplan:next:die`** and other die-movement commands.
	This example assumes that the chuck is up before device movements are made. The following commands load a probe plan, move to a die, move to a subsite on the die.
	`:prob:load 65ghz_t1.wfd`
	`:prob:alig`
	After you load the probe plan file, the probe station asks you to position the probes on the alignment die of your substrate. When the probes are aligned, you can use the move-to-first-die and subsite commands, as shown.
	`:mov:prob:firs:die`
	`:mov:prob:abs:subs 1`
	After performing your testing tasks, you can move to subsequent subsites and make additional tests, as shown with these commands:
	NOTE
	Use :move:next:site to simplify this example.
	`:mov:prob:next:die`
	`:mov:prob:abs:subs 1`
	`:mov:prob:abs:subs 2`
	`:mov:prob:next:die`
	`:mov:prob:abs:subs 2`
	`:moe:prob:abs:subs 1`

Related Commands	*`:move:probeplan:absolute:subsite?`* *`:move:probeplan:absolute:die?`* *`:move:probeplan:first:subsite`* *`:move:probeplan:next:site`* *`:move:probeplan:prior:die`* *`:move:probeplan:prior:subsite`* *`:move:probeplan:absolute:index?`* *`:move:probeplan:absolute:die`* *`:move:probeplan:first:die`* *`:move:probeplan:last:die`* *`:move:probeplan:next:subsite`* *`:move:probeplan:prior:site`* *`:move:probeplan:absolute:index`*

:MOVE:PROBEPLAN:ABSOLUTE:SUBSITE?

Short Command	`:mov:prob:abs:subs?`
Description	This query returns the subsite index that is currently under the probes. If the probes are not over a defined subsite position, an error code will be returned.
Parameters	None

Example	Uses the query to confirm the subsite index number.
	`:mov:prob:firs:die`
	`:mov:prob:abs:subs 1`
	`:mov:prob:abs:subs?`
	`1`

Related Commands	*:move:probeplan:absolute:subsite*

:MOVE:PROBEPLAN:ABSOLUTE:SUBSITE:LABEL

Short Command	*Label*	:mov:prob:abs:subs:lab
Description	This command will move to the subsite with the given label on the current die. The label can include spaces, and does not have to be enclosed in quotes.	
Parameters	*Label*	Text string label of the desired subsite.
Example	Moves to the subsite labeled "voltage" of the current die. `:mov:prob:abs:subs:lab voltage`	

:MOVE:PROBEPLAN:FIRST:DIE

Short Command	`:mov:prob:firs:die`
Description	This command moves the stage to the first die site marked for testing in the active probe plan file. The probe station lowers the chuck, moves to the first die location, then raises the chuck.
Parameters	None
Error Condition	A probe plan must be loaded before the probe station can execute this command. This command also assumes that you have adjusted theta for your DUT. If you load a probe plan, but have not yet adjusted theta, this command moves the device to the location it assumes to be the first die marked for testing. If theta has not been properly adjusted, the new location can be inaccurate.

Example	Enter the commands that load a probe plan into the probe station and move to the first test die.
	`:prob:load 65ghz_t1.wfd`
	`:prob:alig`
	`:mov:prob:firs:die`
	The next command lines show that you can enter other commands, such as **`:move:abs`**, in between `:move:probeplan` commands. In the following example, you use the next-die command to return the device to the next die index listed in the active probe plan file.
	`:mov:abs 2 -181 4500  up`
	`:mov:prob:next:die`
	`:mov:prob:abs:subs 2`
	`:mov:prob:first:die`

Related Commands	*:move:probeplan:absolute:subsite* *:move:probeplan:absolute:subsite?* *:move:probeplan:absolute:die?* *:move:probeplan:first:subsite* *:move:probeplan:next:site* *:move:probeplan:prior:die* *:move:probeplan:prior:subsite* *:move:probeplan:absolute:index?* *:move:probeplan:absolute:die* *:move:probeplan:last:die* *:move:probeplan:next:subsite* *:move:probeplan:prior:site* *:move:probeplan:absolute:index*

:MOVE:PROBEPLAN:FIRST:SUBSITE

Short Command	`:mov:prob:firs:subs`
Description	This command moves the stage to the first subsite of the current die designated for testing in the active probe plan file. The subsite itself is an offset of x- and y-axis coordinates from the current die site. The probe station lowers the chuck, moves to the specified location, and then raises the chuck.
Parameters	None
Error Condition	This command returns an error if the specified subsite has not been defined.

Example	The `:move:probeplan:first:subsite` command is usually used with the `:move:probeplan:next:die` and other die-movement commands.
	This example assumes that the chuck is up before device movements are made. The following commands load a probe plan, move to the first die, move to the first subsite on the die, then move to another subsite.
	`:prob:load 65ghz_t1.wfd`
	`:prob:alig`
	After you load the probe plan file, the probe station asks you to position the probes on the alignment die of your substrate. When the probes are aligned, you can use the move-to-first-die and subsite commands, as shown.
	`:mov:prob:firs:die`
	`:mov:prob:first:subs`
	After performing your testing tasks, you can move to the next subsite and make another test, as shown with these commands:
	`:mov:prob:abs:subs 2`

Related Commands	*:move:probeplan:absolute:subsite* *:move:probeplan:absolute:subsite?* *:move:probeplan:absolute:die?* *:move:probeplan:next:site* *:move:probeplan:prior:die* *:move:probeplan:first:die* *:move:probeplan:last:die* *:move:probeplan:prior:subsite* *:move:probeplan:absolute:index?* *:move:probeplan:absolute:die* *:move:probeplan:last:die* *:move:probeplan:next:subsite* *:move:probeplan:prior:site* *:move:probeplan:absolute:index*

:MOVE:PROBEPLAN:LAST:DIE

Short Command	`:mov:prob:last:die`
Description	This command moves the stage to the last die site designated for testing in the active probe plan file. The probe station lowers the chuck, moves to the last die location, then raises the chuck.
Parameters	None

Example	The following command lines show the device moving from one die to the next. Then to the last die.
	`:mov:prob:next:die`
	`:mov:prob:next:die`
	`:mov:prob:last:die`

Related Commands	*:move:probeplan:absolute:subsite* *:move:probeplan:absolute:die* *:move:probeplan:first:die* *:move:probeplan:next:site* *:move:probeplan:prior:die* *:move:probeplan:prior:subsite* *:move:probeplan:absolute:index?* *:move:probeplan:absolute:subsite?* *:move:probeplan:absolute:die?* *:move:probeplan:first:subsite* *:move:probeplan:next:subsite* *:move:probeplan:prior:site* *:move:probeplan:absolute:index*

:MOVE:PROBEPLAN:LAST:SUBSITE

Short Command	`:mov:prob:last:subs`
Description	This command moves the stage to the last subsite of the current die listed in the active probe plan file. The probe station lowers the chuck, moves to the last subsite location, then raises the chuck.

Parameters	None

Example	The following command lines show the device moving from one subsite to the next, then to the last subsite marked for testing.
	`:mov:prob:next:die`
	`:mov:prob:next:subs`
	`:mov:prob:next:subs`
	`:mov:prob:next:subs`
	`:mov:prob:last:subs`

Related Commands	*`:move:probeplan:absolute:subsite`* *`:move:probeplan:absolute:die`* *`:move:probeplan:first:die`* *`:move:probeplan:last:die`* *`:move:probeplan:next:subsite`* *`:move:probeplan:prior:site`* *`:move:probeplan:absolute:index`* *`:move:probeplan:absolute:subsite?`* *`:move:probeplan:absolute:die?`* *`:move:probeplan:first:subsite`* *`:move:probeplan:next:site`* *`:move:probeplan:prior:site`* *`:move:probeplan:absolute:index?`*

:MOVE:PROBEPLAN:NEXT:DIE

Short Command	`:mov:prob:next:die`
Description	This command moves the stage to the next die site listed in the active probe plan file. The probe station lowers the chuck, moves to the specified location, and then raises the chuck.
Error Conditions	This command returns an error if the device is already positioned at the last die to be tested because there is no next die.
Parameters	None
Example	Positions the device at the next die site listed in the active probe plan file: `:mov:prob:next:die`
Related Commands	*`:move:probeplan:absolute:subsite`* *`:move:probeplan:absolute:die`* *`:move:probeplan:first:die`* *`:move:probeplan:last:die`* *`:move:probeplan:next:site`* *`:move:probeplan:prior:die`* *`:move:probeplan:prior:subsite`* *`:move:probeplan:absolute:index?`* *`:move:probeplan:absolute:subsite?`* *`:move:probeplan:absolute:die?`* *`:move:probeplan:first:subsite`* *`:move:probeplan:last:subsite`* *`:move:probeplan:next:subsite`* *`:move:probeplan:prior:site`* *`:move:probeplan:absolute:index`*

:MOVE:PROBEPLAN:NEXT:SITE

Short Command	`:mov:prob:next:site`
Description	This command moves the stage to the next die site or subsite (whichever is next) listed in the active probe plan file. The probe station lowers the chuck, moves to the specified location, and then raises the chuck.
Error Conditions	This command returns an error if the device is already positioned at the last site to be tested because there is no next site.
Parameters	None
Example	Positions the device at the next test site listed in the active probe plan file: `:mov:prob:next:site`
Related Commands	*`:move:probeplan:absolute:subsite`* *`:move:probeplan:absolute:die`* *`:move:probeplan:first:die`* *`:move:probeplan:last:die`* *`:move:probeplan:next:die`* *`:move:probeplan:prior:die`* *`:move:probeplan:prior:subsite`* *`:move:probeplan:absolute:index?`* *`:move:probeplan:absolute:subsite?`* *`:move:probeplan:absolute:die?`* *`:move:probeplan:first:subsite`* *`:move:probeplan:last:subsite`* *`:move:probeplan:next:subsite`* *`:move:probeplan:prior:site`* *`:move:probeplan:absolute:index`*

:MOVE:PROBEPLAN:NEXT:SUBSITE

Short Command	`:mov:prob:next:subs`
Description	This command moves the stage to the next subsite of the current die listed in the active probe plan file. The probe station lowers the chuck, moves to the specified location, and then raises the chuck.
Error Conditions	This command returns an error if there are no defined subsites or if the device is already positioned at the last subsite to be tested because there is no next subsite.
Parameters	None
Example	Shows how to tell the probe station to position the device at the next subsite listed in the active probe plan file: `:mov:prob:next:subs`

Related Commands	*:move:probeplan:absolute:subsite* *:move:probeplan:absolute:die* *:move:probeplan:first:die* *:move:probeplan:last:die* *:move:probeplan:next:die* *:move:probeplan:prior:die* *:move:probeplan:prior:subsite* *:move:probeplan:absolute:index?* *:move:probeplan:absolute:subsite?* *:move:probeplan:absolute:die?* *:move:probeplan:first:subsite* *:move:probeplan:last:subsite* *:move:probeplan:next:site* *:move:probeplan:prior:site* *:move:probeplan:absolute:index*

:MOVE:PROBEPLAN:PRIOR:DIE

Short Command	`:mov:prob:pri:die`
Description	This command moves the stage to the previous (prior) die site listed in the active probe plan file. The probe station lowers the chuck, moves to the specified location, and then raises the chuck.
Error Conditions	This command returns an error if the probes are already located on the first die because there is no prior die to move to.
Parameters	None
Example	The following command line moves the device to the prior die site listed in the currently active probe plan file: `:mov:prob:pri:die`
Related Commands	*:move:probeplan:absolute:subsite* *:move:probeplan:absolute:die* *:move:probeplan:first:die* *:move:probeplan:last:die* *:move:probeplan:next:die* *:move:probeplan:next:subsite* *:move:probeplan:prior:subsite* *:move:probeplan:absolute:index?* *:move:probeplan:absolute:subsite?* *:move:probeplan:absolute:die?* *:move:probeplan:first:subsite* *:move:probeplan:last:subsite* *:move:probeplan:next:site* *:move:probeplan:prior:site* *:move:probeplan:absolute:index*

:MOVE:PROBEPLAN:PRIOR:SITE

Short Command	`:mov:prob:pri:site`

Description	This command moves the stage to the previous (prior) die site, or subsite, listed in the active probe plan file. If the probes are located at the first subsite in the list, the device moves to the prior die site. Otherwise, the device moves to the prior subsite. The probe station lowers the chuck, moves to the specified location, and then raises the chuck.
Error Conditions	This command returns an error if the probes are already located at the first subsite on the first die because there is no prior site to move to.
Parameters	None
Example	The following command line moves the device to the prior site listed in the currently active probe plan file. `:mov:prob:pri:site`
Related Commands	*`:move:probeplan:absolute:subsite`* *`:move:probeplan:absolute:die`* *`:move:probeplan:first:die`* *`:move:probeplan:last:die`* *`:move:probeplan:next:die`* *`:move:probeplan:next:subsite`* *`:move:probeplan:prior:subsite`* *`:move:probeplan:absolute:index?`* *`:move:probeplan:absolute:subsite?`* *`:move:probeplan:absolute:die?`* *`:move:probeplan:first:subsite`* *`:move:probeplan:last:subsite`* *`:move:probeplan:next:site`* *`:move:probeplan:prior:site`* *`:move:probeplan:absolute:index`*

:MOVE:PROBEPLAN:PRIOR:SUBSITE

Short Command	`:mov:prob:pri:subs`
Description	This command moves the stage to the previous (prior) subsite of the current die listed in the active probe plan file. The probe station lowers the chuck, moves to the specified location, and then raises the chuck.
Error Conditions	This command returns an error if the probes are already located on the first subsite listed in the subsite list of the current die, because there is no prior subsite.
Parameters	None
Example	The following command line moves the device to the prior subsite listed in the currently active probe plan file. `:mov:prob:pri:subs`

Related Commands	*:move:probeplan:absolute:subsite* *:move:probeplan:absolute:die* *:move:probeplan:first:die* *:move:probeplan:last:die* *:move:probeplan:next:die* *:move:probeplan:next:subsite* *:move:probeplan:prior:site* *:move:probeplan:absolute:index?* *:move:probeplan:absolute:subsite?* *:move:probeplan:absolute:die?* *:move:probeplan:first:subsite* *:move:probeplan:last:subsite* *:move:probeplan:next:site* *:move:probeplan:prior:die* *:move:probeplan:absolute:index*

:MOVE:PROBEPLAN:RELATIVE:DIE

Short Command	*xIndex*	*(none)*	`:move:prob:rel:die`
	yIndex	*(none)*	
	z	*(none)* *(up)* *(down)*	
Description	This command allows you to move to a die location relative to the current location. The direction of the move is affected by the load angle, the wafer map angle and the wafer map axis settings. Moves can only be made to a defined die site on the map. Also, you must start the move from a defined die site. Attempts to start from or move to non-die locations will result in an error. Positive x and y values defined when the wafer map was created determine the actual direction of travel. This command is the same as: **`:move:probeplan:relative:index`**.		
Parameters	*xIndex*	Indicates the new die location relative to the current die. Valid entries are integer or **none**.	
	yIndex	Indicates the new die location relative to the current die. Valid entries are integer or **none**.	
	z	Determines the position of the z-axis after travel has completed. Valid entries are **up, down,** or **none**, and the actual position (Summit 12000-Series stations only). Accepts floating-point arguments	

Example	A series of relative moves will return to the original die with the chuck in the down position. Note that before a relative move can be issued, a probe plan must be loaded, aligned and the stage moved to a defined die position.
	`:prob:load "\CMI\300m_fet.wfd"`
	`:prob:alig`
	`:mov:prob:abs:index 80`
	`:mov:prob:rel:index 2 0 UP`
	`:mov:prob:rel:index 0 -3 UP`
	`:mov:prob:rel:index -2 0 UP`
	`:mov:prob:rel:index 0 3 DOWN`		

Related Commands	*:move:probeplan:absolute:subsite* *:move:probeplan:absolute:die* *:move:probeplan:first:die* *:move:probeplan:last:die* *:move:probeplan:next:die* *:move:probeplan:next:subsite* *:move:probeplan:prior:site* *:move:probeplan:absolute:index?* *:move:probeplan:absolute:subsite?* *:move:probeplan:absolute:die?* *:move:probeplan:first:subsite* *:move:probeplan:last:subsite* *:move:probeplan:next:site* *:move:probeplan:prior:die* *:move:probeplan:absolute:index*

:MOVE:PROBEPLAN:RELATIVE:INDEX

Short Command	*xIndex*	*(none)*	`:move:prob:rel:ind`
	yIndex	*(none)*	
	z	*(none)* *(up)* *(down)*	
Description	This command allows you to specify relative moves within the active probe plan wafer map. Moves can only be made to a defined die site on the map. Also, you must start the move from a defined die site. Attempts to start from or move to non-die locations will result in an error. Positive x and y values defined when the wafer map was created determine the actual direction of travel. This command is the same as: `move:probeplan:relative:die`.		
Parameters	*xIndex*	Indicates the new die location relative to the current die. Valid entries are integer or **none**.	
	yIndex	Indicates the new die location relative to the current die. Valid entries are integer or **none**.	
	z	Determines the position of the z-axis after travel has completed. Valid entries are **up, down,** or **none**, and the actual position (Summit 12000-Series stations only). Accepts floating-point arguments	

Example	Shows a series of relative moves that will return to the original die with the chuck in the down position. Note that before a relative move can be issued, a probe plan must be loaded, aligned and the stage moved to a defined die position.
	`:prob:load "\CMI\300m_fet.wfd"`
	`:prob:alig`
	`:mov:prob:abs:index 80`
	`:mov:prob:rel:index 2 0 UP`
	`:mov:prob:rel:index 0 -3 UP`
	`:mov:prob:rel:index -2 0 UP`
	`:mov:prob:rel:index 0 3 DOWN`		

Related Commands	*:move:probeplan:absolute:subsite* *:move:probeplan:absolute:die* *:move:probeplan:first:die* *:move:probeplan:last:die* *:move:probeplan:next:die* *:move:probeplan:next:subsite* *:move:probeplan:prior:site* *:move:probeplan:absolute:index?* *:move:probeplan:absolute:subsite?* *:move:probeplan:absolute:die?* *:move:probeplan:first:subsite* *:move:probeplan:last:subsite* *:move:probeplan:next:site* *:move:probeplan:prior:die* *:move:probeplan:absolute:index*

:MOVE:RELATIVE

Short Command	*devID*		`:mov:rel`
	x	*(none)*	
	y	*(none)*	
	z	*(none)* *(up)* *(down)*	
	theta	*(none)* *(loc)*	
Description	This command moves the device z-axis to a safe position, moves to the specified x-, and y-axis positions (theta is optional), and then moves the z-axis to the specified position. Moves are made relative to the current position. The commanded move is not affected by the wafer alignment, the load angle, the wafer map angle, or the wafer map axis settings. The probe station waits for the device movement to complete before beginning execution of the next command. This command is identical to the `:move:relative:chuck` command.		

Parameters	*devID*	2 for Elite, Summit, S300 and Alessi 6100 emulation. See SCPI Command Device IDs in Nucleus Software.
	x	The coordinate value (in microns or mils) that the device will move along the x-axis, relative to the current device position. This number can be an integer, floating-point, or the character string none. If none is entered there is no change in the x-axis direction. The actual range of valid coordinate values depends on the current device position. If you enter a value outside the probe station limit switches, the probe station returns an error.
	y	The coordinate value (in microns or mils) that the device will move along the y-axis, relative to the current device position. Refer to the x-parameter description for information about the valid range values for this coordinate.
	z	Specifies the z-axis height at the end of the move relative to the current height. This number can be an integer, floating-point, or the character strings: **up, down,** and **none**. Entering **none** for the z position will cause the z-axis to return to its present position after separation is ensured for safe x, y, or theta movement. If the Z parameter is "0" it is an explicit instruction not to change the Z height at the destination. The probe station auto separates if necessary and moves to the new location, but if Edge Sense is enabled it will not invoke a search for the sensor. Use the "UP" to invoke the search for Edge Sense. For probe stations with a pneumatic chuck, use one of the following character strings: **up, down,** and **none**. If **none**, the chuck height remains unchanged. For probe stations with a motorized chuck, this parameter can be an integer or the character strings **up, down,** and **none**.
Optional Parameters	*theta*	Specifies theta-axis. This number can be an integer, floating-point, or the character string **none**. Theta coordinates are in milli-degrees.

Example	The first command returns the current device position. The second command moves the device to a new position, and, when the move completes, returns the new position coordinates.
	`:mov:abs? 2`
	`022659 -008430 000500`
	`:mov:rel 2`
	`604 181 100`
	`:mov:abs? 2`
	`022055 -008249 000600`	

Related Commands	*:move:absolute* *:move:relative:chuck* *:move:absolute* *:move:absolute:options*	

:MOVE:RELATIVE:CHUCK

<table>
<tr><td rowspan="5">Short Command</td><td>devID</td><td></td><td rowspan="5"><code>:mov:rel:chuc</code></td></tr>
<tr><td>x</td><td>(none)</td></tr>
<tr><td>y</td><td>(none)</td></tr>
<tr><td>z</td><td>(none)
(up)
(down)</td></tr>
<tr><td>theta</td><td>(none)
(loc)</td></tr>
<tr><td>Description</td><td colspan="3">This command moves the device z-axis to a safe position, moves to the specified x-, and y-axis positions (theta is optional), and then moves the z-axis to the specified position. Moves are made relative to the current position. The probe station waits for the device movement to complete before beginning execution of the next command.
This command is identical to the <code>:move:relative</code> command.</td></tr>
<tr><td rowspan="4">Parameters</td><td>devID</td><td colspan="2">2 for Elite, Summit, S300 and Alessi 6100 emulation.</td></tr>
<tr><td>x</td><td colspan="2">The coordinate value (in microns or mils) that the device will move along the x-axis, relative to the current device position. This number can be an integer, floating-point, or the character string none. If none is entered there is no change in the x-axis direction.
The actual range of valid coordinate values depends on the current device position. If you enter a value outside the probe station limit switches, the probe station returns an error.</td></tr>
<tr><td>y</td><td colspan="2">The coordinate value (in microns or mils) that the device will move along the y-axis, relative to the current device position. Refer to the x-parameter description for information about the valid range values for this coordinate.</td></tr>
<tr><td>z</td><td colspan="2">Specifies the z-axis height at the end of the move relative to the current height. This number can be an integer, floating-point, or the character strings: up, down, and none. Entering none for the z position will cause the z-axis to return to its present position after separation is ensured for safe x, y, or theta movement.
For probe stations with a pneumatic chuck, use one of the following character strings: up, down, and none. If none, the chuck height remains unchanged.
For probe stations with a motorized chuck, this parameter can be an integer or the character strings up, down, and none.</td></tr>
<tr><td>Optional Parameters</td><td>theta</td><td colspan="2">Specifies theta-axis. This number can be an integer, floating-point, or the character string none. Theta coordinates are in milli-degrees.</td></tr>
</table>

Example	The first command returns the current device position. The second command moves the device to a new position, and, when the move completes, returns the new position coordinates.
	`:mov:abs? 2`
	`022659 -008430 000500`
	`:mov:rel:chuc`
	`2 604 181 100`
	`:mov:abs? 2`
	`022055 -008249 000600`

Related Commands	*`:move:absolute`* *`:move:relative`* *`:move:absolute?`* *`:move:absolute:options`*

:MOVE:SCAN:AAXIS

NOTE

Use this command for making theta adjustments.

Short Command	*devID*		`:mov:scan:aax`
	(left)	*(right)*	
	(in)	*(out)*	
	(up)	*(down)* *(con)* *(sep)*	
	(ufast)	*(vfast)* *(fast)* *(medium)* *(slow)* *(vslow)* *(uslow)*	
Description	This command moves the device continuously in the specified x-, y-, and z-axis directions until a stop command is received or until the limit of travel is reached. This command moves the device diagonally, in three dimensions, toward one of the corners of the station. If you enter a `:move:stop` command, the probe station halts the specified device movement. If you do not use a `:move:stop` command to stop device movement, the device stops scanning when it reaches the end of the limit switches. To move the device along only one axis, refer to the `:move:scan:xax`, `:move:scan:yax`, and `:move:scan:zax` commands. **NOTE** Use this command for making theta adjustments.		

<table>
<tr><td rowspan="10">Parameters</td><td>devID</td><td>2 for Elite, Summit, S300 and Alessi 6100 emulation. See SCPI Command Device IDs in Nucleus Software.</td></tr>
<tr><td>(left)</td><td>Indicates the direction of device movement along the x-axis.</td></tr>
<tr><td>(right)</td><td>This parameter can have the value left or right.
NOTE
When viewed through a microscope, it appears that the probes are moving in the opposite direction.</td></tr>
<tr><td>(in)</td><td>Indicates the direction of device movement along the y-axis.</td></tr>
<tr><td>(out)</td><td>This parameter can have the value in or out. If you select out, this command moves the device toward the front of the probe station. If you select in, this command moves the device toward the rear of the station.</td></tr>
<tr><td>(up)</td><td>Chuck channel only. Up raises the chuck towards the upper limit switch.</td></tr>
<tr><td>(down)</td><td>Chuck channel only. Down lowers the chuck towards the lower limit switch.</td></tr>
<tr><td>(con)</td><td>Indicates a move to contact.</td></tr>
<tr><td>(sep)</td><td>Indicates a move to separate position.</td></tr>
<tr><td>(ufast..)</td><td>Specifies the velocity at which the device will move along the axes. This parameter can have one of the following values: ufast, vfast, fast, medium, vslow or uslow. The default value for this parameter is ultra fast (ufast).</td></tr>
<tr><td>Example</td><td colspan="2">Load a contact substrate on the rear left corner of the chuck. You do not yet know the coordinates of the substrate. Although the load-device command centers the device, you must still scan the device so the appropriate corner of the chuck is under the microscope. This example assumes you have set the appropriate vacuum control knob to the on position.
<code>:mov:load 2</code>
<code>:mov:scan:aax 2 right in down medium</code>
The device moves to the rear right corner of the station. When the device reaches the appropriate position, use the following stop command to stop device movement in all axes:
<code>:mov:stop:aax 2</code>
You can now adjust probe-to-DUT height and perform other tasks.</td></tr>
<tr><td>Related Commands</td><td colspan="2">:move:scan:xaxis
:move:scan:yaxis
:move:scan:zaxis
:move:stop:aaxis
:move:stop:zaxis</td></tr>
</table>

:MOVE:SCAN:XAXIS

<table>
<tr><td rowspan="4">Short Command</td><td>devID</td><td></td><td rowspan="4">:mov:scan:xax</td></tr>
<tr><td>(right)</td><td>(left)</td></tr>
<tr><td rowspan="2">(ufast)</td><td rowspan="2">(vfast)
(fast)
(medium)
(slow)
(vslow)
(uslow)</td></tr>
<tr><td></td></tr>
<tr><td>Description</td><td colspan="3">This command moves the device continuously in the specified x-axis direction until a stop command is received or until the limit of travel is reached. This command moves the device horizontally only.
You do not have to enter a stop command for horizontal device movement. If you do not use a :move:stop command to stop device movement, the device stops when it reaches the right or left end of the limit switches.</td></tr>
<tr><td rowspan="4">Parameters</td><td>devID</td><td colspan="2">2 for Elite, Summit, S300 and Alessi 6100 emulation. See SCPI Command Device IDs in Nucleus Software.v</td></tr>
<tr><td>(left)</td><td colspan="2">Indicates the direction of device movement along the x-axis.</td></tr>
<tr><td>(right)</td><td colspan="2">IThis parameter can have the value left or right.
NOTE
When viewed through a microscope, it appears that the probes are moving in the opposite direction.</td></tr>
<tr><td>(ufast..)</td><td colspan="2">Specifies the velocity at which the device will move along the axes. This parameter can have one of the following values: ufast, vfast, fast, medium, slow, vslow, or uslow. The default value for this parameter is ultra fast (ufast).
NOTE
Use this command for making theta adjustments.</td></tr>
<tr><td>Example</td><td colspan="3">Scan horizontally to the next element on your substrate so you set the velocity to ultra slow:
:mov:scan:xax 2 left uslow
The device moves slowly to the left. When the device reaches the appropriate position, use the following stop command to halt x-axis device movement:
:mov:stop:xax 2
You can also use the :move:stop:aaxis command to stop all device movement.</td></tr>
<tr><td>Related Commands</td><td colspan="3">:move:scan:aaxis
:move:scan:yaxis
:move:scan:zaxis
:move:stop:aaxis
:move:stop:zaxis</td></tr>
</table>

:MOVE:SCAN:YAXIS

<table>
<tr><td rowspan="3">Short Command</td><td>devID</td><td></td><td rowspan="3"><code>:mov:scan:yax</code></td></tr>
<tr><td>(in)</td><td>(out)</td></tr>
<tr><td>(ufast)</td><td>(vfast)
(fast)
(medium)
(slow)
(vslow)
(uslow)</td></tr>
<tr><td>Description</td><td colspan="3">This command moves the device continuously in the specified y-axis direction until a stop command is received or until the limit of travel is reached. This command moves the device vertically only—toward the front and back of the station.
You do not have to enter a stop command to stop vertical device movement. If you do not use a <code>:move:stop</code> command to stop device movement, the device stops moving when it reaches the front or rear limit switch.
NOTE
Use this command for making theta adjustments.</td></tr>
<tr><td rowspan="4">Parameters</td><td>devID</td><td colspan="2">2 for Elite, Summit, S300 and Alessi 6100 emulation. See SCPI Command Device IDs in Nucleus Software.v</td></tr>
<tr><td>(in)</td><td colspan="2">Indicates the direction of device movement along the y-axis. This parameter can have the value in or out..</td></tr>
<tr><td>(out)</td><td colspan="2">If you select out, this command moves the device toward the front of the probe station. If you select in, this command moves the device toward the rear of the station. The positive direction of the y-axis is toward the front of the station, relative to the view through the microscope
NOTE
When viewed through a microscope, it appears that the probes are moving in the opposite direction.</td></tr>
<tr><td>(ufast..)</td><td colspan="2">Specifies the velocity at which the device will move along the axes. This parameter can have one of the following values: ufast, vfast, fast, medium, slow, vslow, or uslow. The default value for this parameter is ultra fast (ufast).</td></tr>
<tr><td>Example</td><td colspan="3">Scan to the top of your DUT until you locate the alignment die. The scan speed is set to medium.
<code>:mov:scan:yax 2 out medium</code>
The device moves to the front of the station. When the device reaches the appropriate position, use the following stop command to stop device movement, then enable the joystick for exact device positioning.
<code>:mov:stop:yax 2</code></td></tr>
<tr><td>Related Commands</td><td colspan="3"><code>:move:scan:aaxis</code>
<code>:move:scan:xaxis</code>
<code>:move:scan:zaxis</code>
<code>:move:stop:aaxis</code>
<code>:move:stop:zaxis</code></td></tr>
</table>

:MOVE:SCAN:ZAXIS

Short Command	*devID*		`:mov:scan:zax`
	(up)	*(down)* *(con)* *(sep)*	
	(ufast)	*(vfast)* *(fast)* *(medium)* *(slow)* *(vslow)* *(uslow)*	
Description	This command moves the device continuously in the specified z-axis direction (up or down) until a stop command is received or until the limit of travel is reached. You do not have to enter a stop command to stop vertical device movement. If you do not use a `:move:stop` command to stop device movement, the device stops moving when it reaches the up or down limit switch.		
Parameters	*devID*	2 for Elite, Summit, S300 and Alessi 6100 emulation. See SCPI Command Device IDs in Nucleus Software.v	
	(up)	Chuck channel only. Up raises the chuck towards the upper limit switch.	
	(down)	Chuck channel only. Down lowers the chuck towards the lower limit switch.	
	(con)	Indicates a move to contact.	
	(sep)	Indicates a move to separate position.	
	(ufast..)	Specifies the velocity at which the device will move along the axes. This parameter can have one of the following values: **ufast, vfast, fast, medium, slow, vslow,** or **uslow**. The default value for this parameter is ultra fast (**ufast**).	

Example	Scan up from the device. The scan speed is set to medium.
	`:mov:scan:zax 2 up medium`
	The chuck moves up in the station. When the chuck reaches the appropriate height, use the following stop command to stop device movement.
	`:mov:stop:zax 2`		

Related Commands	*`:move:scan:aaxis`* *`:move:scan:xaxis`* *`:move:scan:yaxis`* *`:move:stop:aaxis`* *`:move:stop:zaxis`*		

:MOVE:SEPARATE

Short Command	*devID*	`:mov:sep`
Description	This command moves the device to break the contact between the probes and the DUT.	

Parameters	*devID*	2 for Elite, Summit, S300 and Alessi 6100 emulation. See SCPI Command Device IDs in Nucleus Software.

Example	Uses the separate command to lower the chuck so that the probes are no longer in contact with the wafer (DUT). The query is then used to determine if the chuck was lowered.
	`:mov:sep 2`
	`:mov:sep? 2`
	`TRUE`	

Related Commands	*`:move:contact`* *`:move:separate`*	

:MOVE:SEPARATE?

Short Command	*devID*	`:mov:sep?`
Description	This query returns the character string TRUE if the device is 10 microns above or anywhere below the set separate position, otherwise it returns FALSE.	
Parameters	*devID*	2 for Elite, Summit, S300 and Alessi 6100 emulation. See SCPI Command Device IDs in Nucleus Software.

Example	Uses the separate command to lower the chuck so that the probes are no longer in contact with the wafer. The query is then used to determine if the chuck was lowered.
	`:mov:sep 2`
	`:mov:sep? 2`
	`TRUE`	

Related Commands	*`:move:contact`* *`:move:separate`*	

:MOVE:STOP:AAXIS

Short Command	*devID*	`:mov:stop:aax`
Description	This command stops device movement along the x-, y-, and z-axes. Use this command to stop device movement started with `:move:scan` commands.	
Parameters	*devID*	2 for Elite, Summit, S300 and Alessi 6100 emulation. See SCPI Command Device IDs in Nucleus Software

Example	A move-scan command sent to the probe station before the stop command is issued.
	`:mov:scan:aaxis 2 left in up uslow`
	`:syst:del 3000`
	`:move:stop:aaxis 2`
	`The first command line sets the acceleration rate for the scan to be ultra slow. The probe station then scans for 3 seconds. After 3 seconds, the probe station reads the stop-movement command and halts all device movement along the x-, y- and z-axes.`	

Related Commands	*:move:scan:aaxis* *:move:scan:yaxis* *:move:stop:zaxis* *:move:scan:xaxis* *:move:scan:zaxis*

:MOVE:STOP:XAXIS

Short Command	*devID*	`:mov:stop:xax`
Description	This command stops device movement along the x-axis, but allows the device to continue moving along the z- and y-axes. Use this command to stop device movements started with `:move:scan` commands.	
Parameters	*devID*	2 for Elite, Summit, S300 and Alessi 6100 emulation. See SCPI Command Device IDs in Nucleus Software.

Example	A move-scan command sent to the probe station before the stop command is issued.
	`:mov:scan:aaxis 2 right in down uslow`
	`:mov:stop:xaxis 2`	

Related Commands	*:move:scan:aaxis* *:move:scan:yaxis* *:move:scan:xaxis* *:move:stop:aaxis*	

:MOVE:STOP:YAXIS

Short Command	*devID*	`:mov:stop:yax`
Description	This command stops device movement along the y-axis, but allows the device to continue moving along the z- and x-axes. Use this command to stop device movements started with `:move:scan` commands.	
Parameters	*devID*	2 for Elite, Summit, S300 and Alessi 6100 emulation. See SCPI Command Device IDs in Nucleus Software.

Example	A move-scan command sent to the probe station before the stop command is issued.
	`:mov:scan:aaxis 2 right in down uslow`
	`:mov:stop:yaxis 2`	

Related Commands	*:move:scan:aaxis* *:move:scan:yaxis* *:move:scan:xaxis* *:move:stop:aaxis*	

:MOVE:STOP:ZAXIS

Short Command	*devID*	`:mov:stop:zax`

Description	This command stops device movement along the z-axis, but allows the device to continue moving along the x- and y-axes. Use this command to stop device movements started with `:move:scan` commands.	
Parameters	*devID*	2 for Elite, Summit, S300 and Alessi 6100 emulation. See SCPI Command Device IDs in Nucleus Software.

Example	A move-scan command sent to the probe station before the stop command is issued.
	`:mov:scan:aaxis 2 right in down uslow`
	`:mov:stop:zaxis 2`	

Related Commands	*`:move:scan:aaxis`* *`:move:scan:yaxis`* *`:move:scan:xaxis`* *`:move:stop:aaxis`*	

:MOVE:UP

Short Command	*devID*	`:mov:up`
Description	This command is the same as the `:move:contact` command. You can use the `:move:contact`, `:move:separate`, `:move:center`, and `:set:vacuum` commands with this command to control vacuum and chuck status.	
Parameters	*devID*	2 for Elite, Summit, S300 and Alessi 6100 emulation. See SCPI Command Device IDs in Nucleus Software.

Example	Raises the chuck. This example assumes that the chuck was down initially.
	`:mov:up 2`
	The following example shows the query form used to determine if the chuck was raised completely.
	`:mov:up 2`
	`:mov:up? 2`
	`TRUE`	

Related Commands	*`:move:center`* *`:move:down`* *`:move:load`* *`:move:scan:xaxis`* *`:move:scan:zaxis`* *`:move:stop:zaxis`* *`:move:contact`* *`:move:home`* *`:move:scan:aaxis`* *`:move:scan:yaxis`* *`:move:stop:aaxis`* *`:move:up`*	

:MOVE:UP?

Short Command	*devID*	`:mov:up`
Description	This query returns the character string TRUE if the chuck position is equal to or above the set contact position, otherwise it returns FALSE.	
Parameters	*devID*	2 for Elite, Summit, S300 and Alessi 6100 emulation. See SCPI Command Device IDs in Nucleus Software.

Example	Shows that the chuck is completely up.
	`:mov:up 2`
	`:mov:up? 2`
	`TRUE`	

Related Commands	*`:move:down`* *`:move:up`*	

:PROBEPLAN:ABSOLUTE:REFERENCE

Short Command	*X*	`:prob:abs:ref`
	Y	
Description	Moves the reference die to the specific X, Y location without moving the origin at the same time. It is different from `:prob:ref` which does move the origin when a new location is specified.	
Parameters	*X*	Defines the column or X location in user coordinate space.
	Y	Defines the row or Y location in user coordinate space.
Example	Moves the reference location to the die 3, 4 using user coordinate space. `:prob:abs:ref 3 4`	
Related Commands	*`:probeplan:reference`* *`:probeplan:reference?`* *`:probeplan:origin`* *`:probeplan:origin?`*	

:PROBEPLAN:ALIGN

Short Command	*pause*	*(on)* *(off)*	`:prob:alig`
	xPos		
	ypos		

Description	This command sets the alignment position for a probe plan. If only *devID* is specified in the command line, the station reads the alignment position from the active probe plan and moves the station to that position. A message will appear that prompts you to move the stage to the actual alignment position and to press ENTER. In so doing, you allow the station to compensate for differences between the actual DUT and the probe plan wafer map. The parameters, pause, xPos and yPos allow you to remotely control the alignment position and prompting. They must be used together. **CAUTION** If you do not use this command and the wafer has been moved since the probe plan was created, `:move:probeplan` commands may not position the device exactly where you need it to be. **NOTE** You must issue this command immediately after a `:probeplan :load` command.	
Optional Parameters	*pause*	Controls whether or not the station will prompt after it has reached alignment position. On indicates that prompting is enabled. This parameter is required to specify the (*xPos*) and (*yPos*) parameter.
	xPos	Specifies the alignment position in USER coordinates to go to. This new position overrides the values stored in the probe plan file. The (*pause*) parameter is required to specify this parameter.
	ypos	Specifies the alignment position in USER coordinates to go to. This new position overrides the values stored in the probe plan file. The (*pause*) and (*xPos*) parameters are required to specify this parameter.

Example	Shows a probe plan being loaded and the alignment position being set:
	`:prob:load "CMI\300m_fet.ppd"`
	`:prob:alig`
	Shows a probe plan being loaded and the alignment position being set:
	`:prob:load "CMI\300m_fet.ppd"`
	`:prob:alig OFF 2300 -12000`
	Uses the current stage location as the reference die location.
	`:ProbePlan:Align OFF`	

Related Commands	*`:probeplan:load`* *`:probeplan:load?`*	

:PROBEPLAN:BLINK:DIE

Short Command	*active*	*(on)* *(off)*	`:prob:blin:die`
Description	This command will turn blink die mode on or off. Blink die mode will blink the die at the current location in the wafer map window.		

Parameters	*active*	Must be either on or off. This parameter can turn blink die mode either on or off.
Example	Turns on blink die mode. `:prob:blin:die ON`	

:PROBEPLAN:DIE:SIZE

Short Command	*x* *y*	`:prob:die:size`
Description	Use this command to set the die size of the currently loaded wafer map file. The x parameter represents the horizontal size and y represents the vertical size of the die. The units used for the x and y parameters can be set using the `:set:unit` command.	
Parameters	*x*	Indicates the x die size of the wafer map in micron or mils depending on the current units.
	y	Indicates the y die size of the wafer map in micron or mils depending on the current units.
Example	Sets the die size to 30000x20000 in the currently loaded wafer map. The current units are in microns. `:prob:die:size 30000 20000`	

:PROBEPLAN:DIE:XSIZE?

Short Command	`:prob:die:xsiz?`
Description	This query command returns the horizontal size of the die in the currently loaded wafer map. The value is returned using the current units (see `:set:unit` command).
Parameters	None
Example	`Returns the horizontal size of the die in the currently loaded wafer map. The current units are in microns.`

:PROBEPLAN:DIE:YSIZE?

Short Command	`:prob:die:ysiz?`
Description	This query command returns the vertical size of the die in the currently loaded wafer map. The value is returned using the current units (see **:set:unit** command).
Parameters	None

Example	Returns the vertical size of the die in the currently loaded wafer map. The current units are in microns.
	`:prob:die:ysize?`
	`20000`

:PROBEPLAN:FLATNOTCH

Short Command	*Type* *Size*	`:prob:flat`
Description	This command will set the FLAT or NOTCH on a wafer map. It does not set the orientation of the FLAT or NOTCH. To set the orientation, use `:prob:waf:ori`.	
Parameters	*Type*	Possible values are FLAT or NOTCH. If Type is FLAT then size is the length of the flat. If Type is NOTCH then size is the diameter of the notch.
	Size	Uses mm or inches based off of the `:set:unit` command.
Example	Sets a notch on the wafer to 5mm. `:prob:flat NOTCH 5`	

:PROBEPLAN:FLATNOTCH?

Short Command	`:prob:flat?`
Description	This command returns a string that contains two values: *Type and Size.* Type is either FLAT or NOTCH and the *Size* depends on the *Type*. If *Type is* FLAT, size is the length of the flat. If *Type* is NOTCH, size is the diameter of the notch.
Parameters	None

Example	Returns that the wafer has notch 5mm in diameter and another wafer that has flat of 60mm.
	`:prob:flat?`
	`NOTCH 5`
	`:prob:flat?`
	`FLAT 60`

:PROBEPLAN:GRID:SHIFT

Short Command	*x* *y*	`:prob:grid:shif`
Description	This command will shift the die pattern on the wafer map by X, Y microns.	
Parameters	*x*	Specifies in microns the distance to shift the wafer map horizontally.
	y	Specifies in microns the distance to shift the wafer map vertically.
Example	The wafer map will be shifted 1000, 1000 microns. `prob:grid:shif 1000 1000`	

:PROBEPLAN:GRID:SHIFT?

Short Command	`:prob:grid:shif?`

Description	This command will return the current shift of the die pattern on the wafer map.
Parameters	None

Example	Shifts the die pattern on the wafer map by -1000 microns in X and Y.
	`prob:grid:shif?`
	`-1000 -1000`

:PROBEPLAN:LOAD

Short Command	*pathname*	`:prob:load`
Description	This command loads a probe plan file from disk into RAM and makes the file active. Once a probe plan has been loaded, you can access it from the menu system or with commands. Sending this command will clear out any currently loaded probe plan file. If you would like to save the current probe plan before loading another one, use the `:probeplan:save` command. Immediately following a `:probeplan:load` command, you must issue a `:probeplan:align` command.	
Error Condition	If you try to load a file that is not a probe plan, this command returns an error. An error is also returned if you try to load a file that does not exist or that is not located at the specified path.	
Parameters	*pathname*	Specifies the full path and filename of the probe plan file you want to load. File and path names follow the Windows naming conventions. The filename must have a three-character file extension. In general, probe plan files have a .wfd extension. If the filename contains spaces, you must surround the filename with quotes.
Example	Loads the probe plan file *example\new_test.wfd*. `:prob:load "c:\cmi\example\new_test.wfd"`	
Related Commands	*`:probeplan:align`* *`:move:probeplan:absolute:die?`* *`:move:probeplan:absolute:index?`* *`:move:probeplan:absolute:subsite?`* *`:move:probeplan:first:subsite`* *`:move:probeplan:last:subsite`* *`:move:probeplan:next:site`* *`:move:probeplan:prior:die`* *`:move:probeplan:absolute:die`* *`:move:probeplan:absolute:index`* *`:move:probeplan:absolute:subsite`* *`:move:probeplan:first:die`* *`:move:probeplan:last:die`* *`:move:probeplan:next:die`* *`:move:probeplan:prior:site`* *`:probeplan:load?`*	

:PROBEPLAN:LOAD?

Short Command	`:prob:load?`
Description	This query returns the current probe plan filename. If no probe plan is loaded, an error string is returned.
Parameters	None

Example	Queries for the currently active probe plan filename.
	`:probeplan:load?`
	`TEST_01.wfd`

Related Commands	*:probeplan:align* *:move:probeplan:absolute:die?* *:move:probeplan:absolute:subsite?* *:move:probeplan:first:subsite* *:move:probeplan:last:subsite* *:move:probeplan:next:site* *:move:probeplan:prior:die* *:move:probeplan:prior:subsite* *:move:probeplan:first:die* *:move:probeplan:last:die* *:move:probeplan:next:die* *:move:probeplan:next:subsite* *:move:probeplan:prior:site* *:probeplan:load?*

:PROBEPLAN:MARK:DIE

Short Command	*n* *color*	`:prob:mark:die`
Description	This command specifies the color used to identify die sites when you display a wafer map. The command uses indexes that are stored in the palette (see :probeplan:set:palette command).	
Parameters	*n*	Indicates which of the tested die should be marked. The die number is the index number for the die marked as testable in the wafer map.
	color	Specifies which color the wafer map uses to mark tested die. Use the index number corresponding to the color you want in the palette. There are 256 possible colors numbered 0 to 255.
Example	Sets the 10th testable die to the palette color that is stored in index 16. `:prob:mark:subs:val 10 4 3 45.6787`	

:PROBEPLAN:MARK:DIE:VALUE

Short Command	*n* *parameter* *value*	`:prob:mark:die:val`

Description	This command will store a floating-point value at the given die and parameter combination. The color that is represented on the wafer map for this die will depend on the bin color assigned to the value.	
Parameters	*n*	Indicates which of the tested die should be marked. The die number is the index number for the die marked as testable in the wafer map.
	parameter	Indicates which parameter index is being set. This can be any valid parameter in the range of 1 to 255.
	value	Is a floating point number that is stored at the given die and parameter combination.
Example	Sets the 10th testable die using the 3rd parameter to a floating-point value. `:prob:mark:die:value 10 3 45.6787`	
Related Commands	*:probeplan:mark:die* *:probeplan:mark:xydie:value* *:probeplan:parameters:bin* *:probeplan:parameters:description* *:probeplan:parameters:value:clear* *:probeplan:mark:subsite:value* *:probeplan:mark:xysubsite:value* *:probeplan:parameters:color* *:probeplan:parameters:label*	

:PROBEPLAN:MARK:SUBSITE:VALUE

Short Command	*n*	`:prob:mark:subs:val`
	subsite	
	parameter	
	value	
Description	This command will store a floating-point value at the given die, subsite and parameter combination. The color that is represented on the wafer map for this subsite will depend on the bin color assigned to the value.	
Parameters	*n*	Indicates which of the tested die should be marked. The die number is the index number for the die marked as testable in the wafer map.
	subsite	Indicates which of the subsites store the value. Valid subsites can be in the range of 0 to 255. In order to store a value at a subsite it must be a valid subsite.
	parameter	Indicates which parameter index is being set. This can be any valid parameter in the range of 1 to 255.
	value	Is a floating point number that is stored at the given die and parameter combination.
Example	Sets the 10th testable die, 4th subsite using the 3rd parameter to a floating-point value. `:prob:mark:subs:val 10 4 3 45.6787`	

Related Commands	*:probeplan:mark:die* *:probeplan:mark:xysubsite:value* *:probeplan:parameters:color* *:probeplan:parameters:label* *:probeplan:mark:xydie:value* *:probeplan:parameters:bin* *:probeplan:parameters:description* *:probeplan:parameters:value:clear*

:PROBEPLAN:MARK:XYDIE

Short Command	*x* *y* *color*	`:prob:mark:xyd`
Description	This command is very similar to the `:probeplan:mark:die` command, the only difference is that X,Y coordinates are used to specify the die location. Any die can be selected whether it is marked as testable or not. The command uses indices that are stored in the palette.	
Parameters	*x*	X represents the die column for the specified die location defined in user coordinate space.
	y	Y represents the die row for the specified die location defined in user coordinate space.
	color	Specifies which color the wafer map uses to mark tested die. Use the index number corresponding to the color you want in the palette. There are 256 possible colors numbered 0 to 255.
Example	Sets the die at location 4, 5 to the palette color that is stored in index 16. `:prob:mark:xydie 4 5 16`	

:PROBEPLAN:MARK:XYDIE:VALUE

Short Command	*x* *y* *parameter* *value*	`:prob:mark:xyd:val`
Description	This command will store a floating-point value at the given die and parameter combination. This command is very similar to the `:probeplan:mark:die:value` command, the difference is that the die is referenced by an X, Y combination instead of an index. The X,Y coordinates are in user coordinates. The color that is represented on the wafer map for this die will depend on the bin color assigned to the value.	

Parameters	*x*	Indicates in which column the die is located. The X parameter is given in the user coordinate system.
	y	Indicates in which row the die is located. The Y parameter is given in the user coordinate system.
	parameter	Indicates which parameter index is being set. This can be any valid parameter in the range of 1 to 255.
	value	Is a floating point number that is stored at the given die and parameter combination.
Example	Sets the die at 4, 5 using the 3rd parameter to a floating-point value. `:prob:mark:die:value 4 5 3 45.6787`	
Related Commands	*:probeplan:mark:die* *:probeplan:mark:die:value* *:probeplan:parameters:bin* *:probeplan:parameters:description* *:probeplan:parameters:value:clear* *:probeplan:mark:subsite:value* *:probeplan:mark:xysubsite:value* *:probeplan:parameters:color* *:probeplan:parameters:label*	

:PROBEPLAN:MARK:XYSUBSITE:VALUE

Short Command	*x*	`:prob:mark:xys:val`
	y	
	subsite	
	parameter	
	value	
Description	This command will store a floating-point value at the given die, subsite and parameter combination. The die is referenced by the X, Y user coordinate system. The color that is represented on the wafer map for this subsite will depend on the bin color assigned to the value.	
Parameters	*x*	Indicates in which column the die is located. The X parameter is given in the user coordinate system.
	y	Indicates in which row the die is located. The Y parameter is given in the user coordinate system.
	subsite	Indicates which of the subsites to store the value. Valid subsites can be in the range of 0 to 255. In order to store a value at a subsite it must be a valid subsite.
	parameter	Indicates which parameter index is being set. This can be any valid parameter in the range of 1 to 255.
	value	Is a floating point number that is stored at the given die and parameter combination.

Example	Sets the die at 3, 7 using the 4th subsite and the 11th parameter to hold a floating-point value. `:prob:mark:die:value 3 7 4 11 45.6787`
Related Commands	*:probeplan:mark:die* *:probeplan:parameters:bin* *:probeplan:parameters:description* *:probeplan:parameters:value:clear* *:probeplan:mark:xydie:value* *:probeplan:parameters:color* *:probeplan:parameters:label*

:PROBEPLAN:NSUBSITES?

Short Command	`:prob:nsub?`
Description	This query returns the number of active subsites defined in the probe plan subsite list.
Parameters	None

Example	Requests the number of active subsites. In this case, there are eight subsites in the subsite list.
	`:prob:nsub?`
	`8`

Related Commands	*:move:probeplan:absolute:subsite* *:probeplan:ntested?*

:PROBEPLAN:NTESTED?

Short Command	`:prob:ntes?`
Description	This query returns the number of die marked for testing in the currently active probe plan. If you have not yet loaded a probe plan file, this query returns nothing.
Parameters	None

Example	Requests the number of die marked for testing. In this example, you first load a probe plan file. You then enter the die-tested query. In this example, the number of die currently marked for testing is 181 die.
	`:prob:load "example\vband\65ghz_t1.wfd"`
	`:prob:ntes?`
	`18`

Related Commands	*:move:probeplan:absolute:subsite* *:move:probeplan:absolute:die* *:move:probeplan:first:die* *:move:probeplan:last:die* *:move:probeplan:next:die* *:move:probeplan:prior:die* *:move:probeplan:next:site* *:move:probeplan:prior:site* *:move:probeplan:first:subsite* *:move:probeplan:next:subsite* *:move:probeplan:prior:subsite*

:PROBEPLAN:ORIGIN

Short Command	*X*	`:prob:orig`
	Y	
Description	This command can be used to set the origin for the currently loaded wafer map. The X and Y parameters are the column and row from the left bottom corner of the wafer map. Columns start at 0 on the left and are positive to the right. Rows start at 0 on the bottom and are positive to the top. The origin command is used to define the user coordinate space, so the X and Y parameters are in terms of the left bottom corner. Once the origin of a wafer map has been set up, all other commands that use column and row wafer coordinates will use the origin as a base. The origin is not visible on the wafer map. The X and Y die location can be any number. For example: `:prob:orig -1 -1` will set the left, bottom corner to have coordinates 1,1.	
Parameters	*X*	Defines the X column of the origin from the leftmost column.
	Y	Defines the Y row of the origin from the bottommost row.

Example	Sets the origin 3 columns over from the left and 4 rows up from the bottom.
	`:prob:orig 3 4`
	`:prob:orig?`
	`3 4`	

:PROBEPLAN:ORIGIN?

Short Command	`:prob:orig?`
Description	This command queries the currently loaded wafer map for the origin die. The values returned are in relation to the left bottom corner of the wafer map and are in column, row (X, Y) order.
Parameters	None

Example	Shows that the origin is 4 columns over and 5 rows up from the left bottom corner of the wafer map.
	`:prob:orig?`
	`4 5`

:PROBEPLAN:PARAMETERS:BIN

Short Command	*number*	`:prob:par:bin`
	bin total	
	high	
	low	
Description	This command sets up the bins for the selected parameter. The high and low values define the maximum and minimum bin values. The first and last bins go to infinity. All of the values for the other bins are evenly spaced using the bin total for the number of bins. If colors are already assigned for the maximum and minimum then a gradient fill algorithm is used to fill in the bin color values between them. Bin values for this parameter that were already set up prior to this command will be erased.	

Parameters	*number*	Specifies the parameter number. The valid range for a parameter number is 1 to 255.
	bin total	Defines the total number of bins for this parameter.
	high	Defines the maximum threshold value for this parameter.
	low	Defines the minimum threshold value for this parameter.
Example	Sets up parameter 3 with 10 bins ranging in value from 56.78 to 1006.45. `:prob:par:bin 3 10 1006.45 56.78`	

:PROBEPLAN:PARAMETERS:BIN?

Short Command	*number*	`:prob:par:bin?`
Description	This query command returns a string value for the parameter that was selected. The string value returned consists of three fields separated by spaces. The first field is the total number of bins for this parameter. The next field is the maximum threshold value and the last field is the minimum threshold value.	
Parameters	*number*	Specifies the parameter number. The valid range for a parameter number is 1 to 255.

Example	Queries the 3rd parameter, which has 10 bins ranging in value from 56.78 to 1006.45.
	`:prob:par:bin? 3`
	`10 1006.45 56.78`	

:PROBEPLAN:PARAMETERS:COLOR

Short Command	*number*		`:prob:par:col`
	red	*red*	
	green	*green*	
	blue	*blue*	
Description	This command sets the color values for the maximum and minimum bin of a particular parameter. The first set (row) of red, green and blue values are used to set the maximum bin and the second are for the minimum bin. A gradient fill algorithm is used to fill in the bin color values between maximum and minimum.		

Parameters	*number*		Specifies the parameter number. The valid range for a parameter number is 1 to 255.
	red		Sets the intensity of red in the pixels for the maximum bin. The intensity scale runs from 0 to 255, with 0 being no red and 255 being the brightest red.
	green		Sets the intensity of green in the pixels for the maximum bin. The intensity scale runs from 0 to 255, with 0 being no green and 255 being the brightest green.
	blue		Sets the intensity of blue in the pixels for the maximum bin. The intensity scale runs from 0 to 255, with 0 being no blue and 255 being the brightest blue.
Example	Sets the color values for parameter one. The maximum bin is set to Red and the minimum bin is set to Blue. `:prob:par:col 1 255 0 0 0 0 255`		

:PROBEPLAN:PARAMETERS:COLOR?

Short Command	*number*	`:prob:par:col?`
Description	This command queries the color settings of a particular parameter. The command returns a string containing six numbers, which are Red, Green, and Blue intensities of two colors. The first color is for the maximum bin and the second color is the minimum bin.	
Parameters	*number*	Specifies the parameter number. The valid range for a parameter number is 1 to 255.

Example	Queries the color settings for parameter one. The return string has 255 0 0 for the maximum bin which is Red, and 0 0 255 for the minimum bin which is Blue.
	`:prob:par:col? 1`
	`255 0 0 0 0 255`	

:PROBEPLAN:PARAMETERS:DESCRIPTION

Short Command	*number* *description*	`:prob:par:desc`
Description	This command sets a descriptive text field used to identify the parameter. The text may contain spaces, but the text field must be enclosed in quotations. Ex: "Example text". If spaces are used that are not enclosed in quotes, an error will result.	
Parameters	*number*	Specifies the parameter number. The valid range for a parameter number is 1 to 255.
	description	Specifies a text field used to describe the parameter. If spaces are needed in the text then the entire text field needs to be enclosed in quotes. Ex: "this is a test field".
Example	`Sets` the description field of parameter 3 to "Parameter number three". `:prob:par:desc 3 "Parameter number three"`	

:PROBEPLAN:PARAMETERS:DESCRIPTION?

Short Command	*number*	`:prob:par:desc?`
Description	This query command will return the descriptive text field used to identify a parameter. This field can be set with the `:probe:par:desc` command.	
Parameters	*number*	Specifies the parameter number. The valid range for a parameter number is 1 to 255.

Example	`Sets` the description text field of parameter 3 then retrieves the text value.
	`:prob:par:desc 3 "Parameter number three"`
	`:prob:par:desc? 3`
	`Parameter number three`	

:PROBEPLAN:PARAMETERS:LABEL

Short Command	*number* *label*	`:prob:par:lab`
Description	This command sets the label for the parameter. The label is a text field used to identify the parameter. The label is visible in the parameter combo box of the wafer map window.	
Parameters	*number*	Specifies the parameter number. The valid range for a parameter number is 1 to 255.
	label	Specifies text used for the label text field of the parameter. If spaces are needed in the text then the entire text field needs to be enclosed in quotes. Ex: "this is a test field".
Example	Sets the label for parameter 5 to "Parameter Five". `:prob:par:label 5 "Parameter Five"`	

:PROBEPLAN:PARAMETERS:LABEL?

Short Command	*number*	`:prob:par:lab?`
Description	This query command will return the label of the specified parameter. This field can be set with the **`:probe:par:lab`** command.	
Parameters	*number*	Specifies the parameter number. The valid range for a parameter number is 1 to 255.

Example	Sets the label of parameter 5 then retrieves the text value.
	`:prob:par:lab 5 "Number Five"`
	`:prob:par:lab? 5`
	`Number Five`	

:PROBEPLAN:PARAMETERS:LABEL:ORDINAL?

Short Command	*Ordinal*	`:prob:par:lab:ord?`

Description	This command returns the parameter labels in order, but gaps in the index sequence are condensed.
Parameters	*Ordinal* The ordinal index value of the parameter table.

Example	Given the parameter table listed below an ordinal query of 3 will return the 17th parameter.
	`:prob:par:lab:ord? 3`
	`Resistance`

Parameter Index	Parameter Label	Parameter Ordinal Query
1	Volt	1
3	Cap	2
17	Resistance	3

:PROBEPLAN:PARAMETERS:VALUE:CLEAR

Short Command	`:prob:par:val:cle`
Description	Clears all of the parameter data values for the current wafer map. This includes die parameter values and subsite parameter values. The bin setup and interval settings are not affected by this command.
Parameters	None
Example	Clears the data from the currently loaded wafer map. `:prob:par:val:cle`

:PROBEPLAN:PARAMETERS:VIEW

Short Command	*number* *subsite*	`:prob:par:view`
Description	This command will change the current view parameter in the wafer map window. The subsite parameter is optional. If it is not specified, the reference or 0 subsite is used. If the parameter does not exist, an error will be returned.	
Parameters	*number*	Specifies the parameter id number. The valid range for a parameter number is 1 to 255. A number of 0 will set the parameter view to the "none" setting.
	subsite	Optional parameter that gives the subsite location for the parameter number. If this value is not specified, the reference subsite will remain unchanged.

Example	Changes the wafer map display to show parameter 3.
	`prob:par:view 3`
	Changes the wafer map display to show parameter 3, subsite 2.
	`prob:par:view 3 2`
	If parameter 45 doesn't exist, "@parameter does not exist" will be returned.
	`prob:par:view 45`
	If subsite 45 doesn't exist, "@subsite does not exist" will be returned.
	`prob:par:view 45`

:PROBEPLAN:QUALITY:SIZE

Short Command	*Size*	`:prob:qual:size`
Description	This command will set the quality area of a wafer. The quality area can not be larger than wafer size. The `:set:unit` command can be used to change the interpretation of size parameter. It can be in either English or Metric units.	
Parameters	*Size*	Size of the quality area in mm or inches based on `:set:unit`.
Example	Sets the quality area of the wafer to 190mm. `:prob:qual:size 190`	

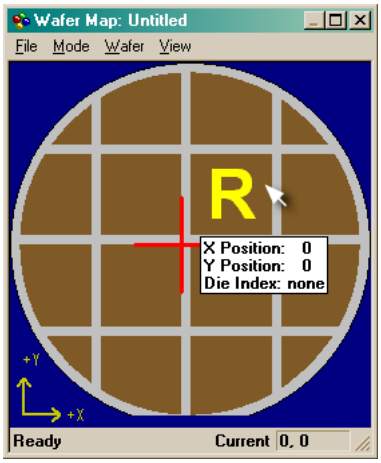
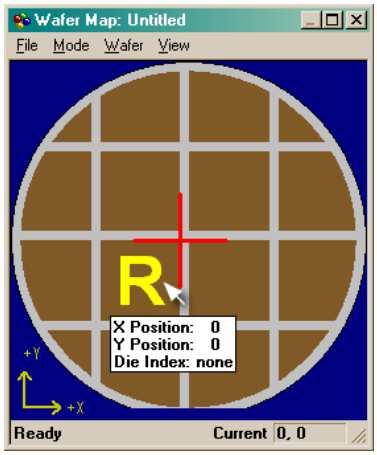
:PROBEPLAN:QUALITY:SIZE?

Short Command	`:prob:qual:size?`
Description	This command returns the quality area size of the current wafer map. It uses mm or inches based on `:set:unit` command.
Parameters	*None*

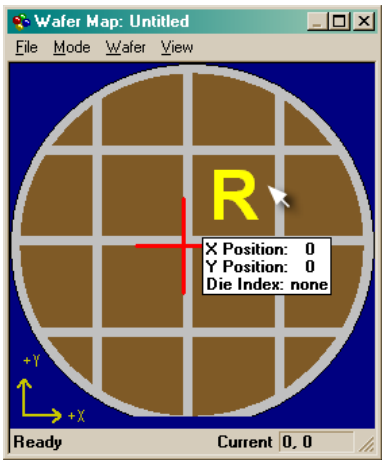
Example	Returns 190mm as the current quality area.
	`:prob:qual:size?`
	`190`

:PROBEPLAN:REFERENCE

Short Command	*X* *Y*	`:prob:ref`
Description	This command can be used to set the reference and origin die for the currently loaded wafer map. The x-and y-parameters are the column and row in wafer map user coordinates. The reference is the die location of the probe to pad alignment. It appears as an "R" on the wafer map. This command will also move the origin so the reference die will maintain the same user coordinates it had before the `:prob:ref` call. If you wish to give the reference die coordinates of x, y, make a "`:probeplan:origin 0 0`" call after `:prob:ref`. This combination will set the origin to the left-bottom with coordinates 0,0 and put the reference die at x, y.	

Parameters	*X*	Defines the X column of the reference in user coordinates.
	Y	Defines the Y row of the reference in user coordinates.
Example 1	 *Default 4x4 wafer map*	This wafer map has 16 die in a 4x4 grid. The reference and origin are located on the same die, which is the default for a new wafer map. As you can see the reference die has coordinates of 0,0
	After sending "`:probeplan:reference -1 -1`", the reference has moved to location -1,-1 and the origin has also moved to location -1,-1. So, the reference die still has the coordinates of 0,0.	 *Wafer map after sending :probeplan:reference -1 -1 command*

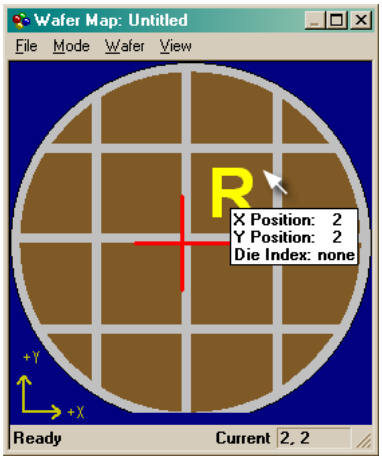
Example 2



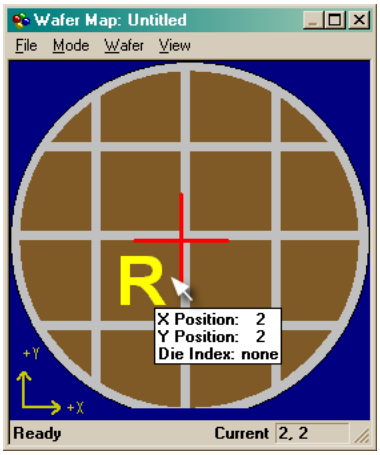
Default 4x4 wafer map

In this example, we have taken the original 4x4 wafer map with the reference and the origin located on the same spot. The reference die has coordinates of 0,0.

Now send the command "`:probeplan:origin 0 0`". After this command, the origin is located at the bottom-left corner of the wafer map. The origin command uses the left-bottom corner as the 0,0 location. For more information, see the "`:probeplan:origin`" command. The reference die now has coordinates of 2,2, because it is two columns over and two columns up from the origin.



Wafer map after sending :probeplan:origin 0 0 command



Wafer map after sending :probeplan:reference 1 1 command

Now send the `:probeplan:reference 1 1` command, and the reference die moves to the location shown above. The origin will also move one column over and one column down. The reference die has changed locations, but the user coordinates for the reference die have not changed. The reference still has user coordinates 2,2 because the origin has moved.

:PROBEPLAN:REFERENCE?

Short Command	`:prob:ref?`

Description	This command can be used to query the reference die for the currently loaded wafer map. The reference is the die location of the probe to pad alignment. It appears as an R on the wafer map. The return values are in wafer map user coordinates.
Parameters	None

Example	Returns the reference die of 3, 4.
	`:prob:ref?`
	`3 4`

:PROBEPLAN:SAVE

Short Command	*pathname*	`:prob:save`
Description	This command saves the currently loaded wafer map file from memory onto disk. The pathname parameter is optional. If it is not supplied then the wafer map is saved using the current name and location.	
Optional Parameters	*pathname*	The fully qualified pathname to which the wafer map file will be saved. If a pathname is not specified, then the wafer map will be saved using the default name that it was opened with. If the filename contains spaces, you must surround the filename with quotes (see example).
Example	Shows how to save a wafer map file. `:prob:save "c:\cmi\example\new test.wfd"`	

:PROBEPLAN:SET:PALETTE

Short Command	*n*	`:prob:set:pal`
	red	
	green	
	blue	

Description

Use this command to define the colors in the palette. The relative intensity of red, green, and blue determines the color. Intensity ranges from 0 to 255, with 0 being off and 255 being the brightest intensity.

There are 256 possible colors in the palette, numbered 0 to 255. The first 16 are predefined, as shown below. You can assign custom colors to any of the palette indexes.

Index	*Color*
0	**Black**
1	**Blue**
2	**Green**
3	**Cyan**
4	**Red**
5	**Magenta**
6	**Brown**
7	**Dull White**
8	**Dark Gray**
9	**Light blue**
10	**Light Green**
11	**Light Cyan**
12	**Light Red**
13	**Light Magenta**
14	**Light Yellow**

Parameters

n	Use this parameter to specify the index number (that is, location in the palette) of the color you are defining. You can use any number between 0 and 255.
red	Sets the intensity of red in the pixels for the maximum bin. The intensity scale runs from 0 to 255, with 0 being no red and 255 being the brightest red.
green	Sets the intensity of green in the pixels for the maximum bin. The intensity scale runs from 0 to 255, with 0 being no green and 255 being the brightest green.
blue	Sets the intensity of blue in the pixels for the maximum bin. The intensity scale runs from 0 to 255, with 0 being no blue and 255 being the brightest blue.

Example

Defines color 100 to be red, then sets 100 as the color to display die #30 in the wafer map.

```
:prob:set:palette 100 255 0 0
:prob:mark:die 30 100
```

Related Commands

:probeplan:mark:die

:PROBEPLAN:SET:SUBSITE

Short Command	*index*		`:prob:set:subs`
	x		
	y		
	{label}		
	{active}	*(on)* *(off)*	
Description	This command defines a subsite location (device position relative to the die site). The subsite is defined as x- and y-axis offsets from the die location. Optional parameters to this command are the label, and whether or not the subsite is active. After defining a subsite, use the `:move:probeplan:absolute:subsite` command to position the device at the subsite location.		
Parameters	*index*	Identifies the specific subsite. You may define up to 255 different subsites. Each subsite must have an index number to differentiate it from the other subsites. For example, if you have three subsites, use 0, 1, and 2 as the three subsite index numbers. The 0 subsite is also the reference point inside the wafer. If an index number is larger than the total number of subsites, the index will be changed to one greater than the last subsite	
	x	The x offset for the subsite. This coordinate value is an offset from the defined die sites in the active wafer map. The values you can assign to this parameter depend on the size of the die.	
	y	The y offset for the subsite. This coordinate value is an offset from the defined die sites in the active wafer map file. The values you can assign to this parameter depend on the size of the die.	
Optional Parameters	*label*	A label for this subsite. If it includes spaces, it must be enclosed in quotation marks. If not given, the default "Label" is assigned. This parameter is required to specify the (*active*) parameter.	
	active	Active defines whether or not this subsite is a part of the active subsite stepping sequence. See the :move:probeplan:next:subsite command. Possible values are "on" and "off". If Active is not given then the subsite is "on" by default. The (*label*) parameter is required to specify this parameter.	
Example	Defines subsite number 1 as being 24 microns in X and –117 microns in Y, away from the reference location. The label is given the value Test one and it is defined as being active. `:prob:set:subs 1 24 –117 "Test One" on`		

:PROBEPLAN:SET:SUBSITE:LABEL:ACTIVE?

Short Command	*subsite index*	`:prob:set:subs:lab:act?`

Description	This command returns the test status of a subsite at the given index. Possible return values are "ON" or "OFF". If a subsite is "ON" then it is in the subsite stepping sequence. A subsite that is "OFF" is not in the stepping sequence.	
Parameters	*subsite index*	Defines the subsite index number. Valid numbers range from 0 to 255.

Example	The subsite located at index 4 is not part of the subsite stepping sequence.
	`:prob:set:subs:lab:act? 4`
	`OFF`	

:PROBEPLAN:SHOW

Short Command	*(on)*	*(off)*	`:prob:show`
	(on)	*(off)*	
	(on)	*(off)*	
Description	This command controls the probe plan wafer map display. You can specify: • Whether or not the wafer map is displayed • Whether or not the wafer-map window is always displayed on top of other windows • Whether or not die site coordinate are displayed		
Parameters	*(on) (off)*	The *(on) (off)* parameter determines whether or not the wafer map is displayed. Use *(on)* to display the map.	
	(on) (off)	The *(on) (off)* parameter determines whether or not the wafer map window is always displayed on top of other windows.	
	(on) (off)	The *(on) (off)* parameter determines whether or not die site coordinate are displayed.	
Example	Causes the wafer map to always be displayed on top of other windows, even when it is not the active window, and to display die site coordinates. `:prob:show on`		
Related Commands	*:probeplan:load* *:probeplan:mark:die*		

:PROBEPLAN:STREET:SIZE

Short Command	*XSize*	`:prob:str:size`
	YSize	
Description	This command sets the street size of the current wafer map in X and Y. The street size can not be larger than the die size. The size parameters may be in microns or mils.	
Parameters	*XSize*	Size of the X Street in microns or mils.
	YSize	Size of the Y Street in microns or mils.
Example	Sets the street size of the wafer map to 1000, 1000. `:prob:str:size 1000 1000`	

:PROBEPLAN:STREET:XSIZE?

Short Command	`:prob:str:xsiz?`

Description	This command returns the X street size of the current wafer map. It uses microns or mils based on `:set:unit.`
Parameters	None

Example	Returns that the size of the street is 1000 microns.
	`:prob:str:xsiz?`
	`1000`

:PROBEPLAN:STREET:YSIZE?

Short Command	`:prob:str:ysiz?`
Description	This command returns the Y street size of the current wafer map. It uses microns or mils based on `:set:unit`.
Parameters	None

Example	Returns that the size of the street is 1000 microns.
	`:prob:str:ysiz?`
	`1000`

:PROBEPLAN:SUBSITES:DELETE

Short Command	*Index*	`:prob:subs:del`
Description	This command will delete the subsite at the specified index.	
Parameters	*index*	Specifies the subsite index to delete.
Example	Deletes the subsite located at index 4. `prob:subs:del 4`	

:PROBEPLAN:SUBSITES:DELETE:ALL

Short Command	`:prob:subs:del:all`
Description	This command will delete all of the subsites in the subsite list.
Parameters	None
Example	Remove all subsites from the subsite list. `prob:subs:del:all`

:PROBEPLAN:SUBSITES:LABEL?

Short Command	*subsite*	`:prob:subs:lab?`
Description	This command will return the descriptive text field used to identify the subsite. This label can be set with the **`:probeplan:set:subsite`** command.	
Parameters	*subsite*	Defines the subsite index number. Valid numbers range from 0 to 255.

Example	Retrieves the descriptive label for the 3rd subsite, which is "Subsite three".
	`:prob:subs:lab? 3`
	`Subsite three`	

:PROBEPLAN:SUBSITES:TOTAL?

Short Command	`:prob:subs:tot?`	
Description	This command will return the total number of subsites whether they are marked as active or not. This command differs from the **`:probplan:nsubsites?`** in that the it returns the total number of subsites, not just the active ones.	
Parameters	None	

Example	Returns 15 as the total number of subsites defined in the wafer map file.
	`:prob:subs:tot?`
	`15`	

:PROBEPLAN:SUBSITES:XOFFSET?

Short Command	`:prob:subs:xoff?`	
Description	This command will return the X offset associated with a given subsite. The X offset is defined as the horizontal distance in microns from the reference point within a die. Offsets are given in the normal Cartesian coordinate system.	
Parameters	*subsite*	Defines the subsite index number. Valid numbers range from 0 to 255.

Example	Retrieves the X offset for the 3rd subsite,
	`:prob:subs:xoff? 3`
	`-4064`	

:PROBEPLAN:SUBSITES:YOFFSET?

Short Command	*subsite*	`:prob:subs:yoff?`
Description	Returns the Y offset associated with a given subsite. The Y offset is defined as the vertical distance in microns from the reference point within a die. Offsets are given in the normal Cartesian coordinate system.	
Parameters	*subsite*	Defines the subsite index number. Valid numbers range from 0 to 255.

Example	Retrieves the Y offset for the 3rd subsite.
	`:prob:subs:xoff? 3`
	`1089`	

:PROBEPLAN:TEST:DIE

Short Command	*X* *Y* *Mark*	`:prob:test:die`

Description	This command can be used to mark die for test of the currently loaded wafer map. The X and Y parameters are the column and row in the wafer map user defined coordinates. An optional parameter can be used to deselect a die for test. The testing sequence is recomputed after this command is issued, which might change the testable index of dies that appear after this one.	
Parameters	*X*	Defines the X position or column of the die to mark in user coordinates.
	Y	Defines the Y position or row of the die to mark in user coordinates.
Optional Parameters	*Mark*	Either ON or OFF depending on whether or not the die is to be marked for test. Use ON to mark the die for test, and OFF to unmark the die for test. If this parameter is not supplied then the default value for mark is.

Example	Sets the die at column 4 and row 5 to be marked for test.
	`:prob:test:die 4 5 ON`
	Deselects the die at column 4 and row 5.
	`:prob:test:die 4 5 OFF`
	`:prob:test:die? 4 5`
	`OFF`	

:PROBEPLAN:TEST:DIE?

Short Command	*X* *Y*	`:prob:test:die?`
Description	This command can be used to query whether or not a has been marked for test in the currently loaded wafer map. The X and Y parameters are the column and row in the wafer map user defined coordinates.	
Parameters	*X*	Defines the X position or column of the die to query in user coordinates.
	Y	Defines the Y position or row of the die to query in user coordinates.

Example	Sets the die at column 4 and row 5 to be marked for test.
	`:prob:test:die 4 5 ON`
	`:prob:test:die? 4 5`
	`ON`	

:PROBEPLAN:TESTANALYSIS:RESULT?

Short Command	*Index*	`:mov:prob:res`
Description	This command will return the result from a test analysis given the testable die index. Returns FAIL, PASS, DISABLED or NOTESTRESULT. PASS and FAIL indicate that the testable die index passed or failed. DISABLED will be returned if the combo pass/fail logic has not been setup.	
Parameters	*Index*	Testable die index.
Example	Checks testable die index number 4 for a result FAIL. `:prob:test:res 4`	

:PROBEPLAN:TESTANALYSIS:XY:RESULT?

Short Command	*X* *Y*	`:prob:test:xy:res`
Description	This command will return the result from a test analysis given the Column and row in the wafer map. Returns FAIL, PASS, DISABLED or NOTESTRESULT. PASS and FAIL indicate that the X, Y die passed or failed. DISABLED will be returned if the combo pass/fail logic has not been set up. NOTESTRESULT will be returned if the X, Y die does not have any stored data at its location.	
Parameters	*X*	Column of the wafer map die in user coordinates.
	Y	Row of the wafer map die in user coordinates.
Example	Checks wafer map die 3 2 for a result FAIL. `:prob:test:xy:res 3 2`	

:PROBEPLAN:TESTING:SEQUENCE

Short Command	*Direction* *Horizontal* *Vertical*	`:prob:test:seq`
Description	This command can be used to define the stepping sequence of the currently load wafer map.	
Parameters	*Direction*	Specifies the major axis and bi-direction for the stepping sequence. Possible values are HORIZONTAL, VERTICAL, BIDIRHORIZ, BIDIRVERT: • HORIZONTAL - single direction along the horizontal. • VERTICAL - single direction along the vertical. • BIDIRHORIZ - Bi-directional along the horizontal axis. Horizontal serpentine motion. • BIDIRVERT - Bi-directional along the horizontal axis. Vertical serpentine motion.
	Horizontal	Specifies the direction of the horizontal stepping sequence. Values are LEFTRIGHT or RIGHTLEFT. • LEFTRIGHT - The horizontal sequence will run from Left to Right. • RIGHTLEFT - The horizontal sequence will run from Right to Left.
	Vertical	Specifies the direction of the vertical stepping sequence. Values are TOPBOTTOM or BOTTOMTOP. • TOPBOTTOM - The vertical sequence will run from Top to Bottom. • BOTTOMTOP - The vertical sequence will run from Bottom to Top.
Example	Sets up a stepping sequence that runs bi-directional with X as the major axis. It runs from Left to Right and Bottom to Top in a horizontal serpentine motion with X as the major axis. `:prob:test:seq BIDIRHORIZ LEFTRIGHT BOTTOMTOP`	

:PROBEPLAN:TESTING:SEQUENCE?

Short Command	`:prob:test:seq?`
Description	This function returns a string describing the current stepping pattern. The string contains three values: Direction, Horizontal, and Vertical. Possible values for *Direction* are: • HORIZONTAL - single direction along the horizontal. • VERTICAL - single direction along the vertical. • BIDIRHORIZ - Bi-directional along the horizontal axis. Horizontal Serpentine motion. • BIDIRVERT - Bi-directional along the horizontal axis. Vertical Serpentine motion. Possible values for *Horizontal* are: • LEFTRIGHT - The horizontal sequence will run from Left to Right. • RIGHTLEFT - The horizontal sequence will run from Right to Left. Possible values for *Vertical* are: • TOPBOTTOM - The vertical sequence will run from Top to Bottom. BOTTOMTOP - The vertical sequence will run from Bottom to Top.
Parameters	None

Example	Returns that the wafer map stepping sequence is bi-directional with X as the major axis and it runs from Left to Right and Top to Bottom.
	`:prob:test:seq?`
	`BIDIRHORIZ LEFTRIGHT TOPBOTTOM`

:PROBEPLAN:WAFER:ORIENTATION

Short Command	*Orientation*	`:prob:waf:ori`
Description	This command sets the orientation of the current wafer map. Possible values are 0, 90, 180, 270. 0 is bottom, 90 equals left, 180 equals top, 270 equals right.	
Parameters	*Orientation*	Possible values are 0, 90, 180, 270.
Example	Sets the current orientation of the flat or notch to the left. `:prob:waf:ori 90`	

:PROBEPLAN:WAFER:ORIENTATION?

Short Command	`:prob:waf:ori?`
Description	This command returns the orientation of the current wafer map. Possible values are 0, 90, 180, 270.
Parameters	None

Example	The wafer map has the flat/notch to the LEFT.
	`:prob:waf:ori?`
	`9`

:PROBEPLAN:WAFERSIZE

Short Command	*size*	`:prob:waf`
Description	This command will set the diameter of the currently loaded wafer map. The size parameter can be in either millimeters or inches, depending on the current units that were set with the **`:set:unit`** command.	
Parameters	*size*	Defines the diameter of the wafer in the currently loaded wafer map. This parameter can be in either Metric (mm.) or English (in.) units depending on the :set:unit command.
Example	Ssets the diameter of the current wafer map to 200 mm. `:prob:waf 200`	

:PROBEPLAN:WAFERSIZE?

Short Command	*size*	`:prob:waf?`
Description	This query command will return the diameter of the currently loaded wafer map. The value that is returned can be either millimeters or inches, depending on the current units that were set with the **`:set:unit`** command.	
Parameters	*size*	Defines the diameter of the wafer in the currently loaded wafer map. This parameter can be in either Metric (mm.) or English (in.) units depending on the :set:unit command.

Example	Queries the diameter of the current wafer map.
	`:prob:waf?`
	`200`	

:PROBEPLAN:XY:ORIENTATION

Short Command	*XAxis*	`:prob:xy:ori`
	YAxis	
Description	This command will set the orientation of the X and Y axes in the wafer map. This affects the Row, Column readings of the die. RIGHT or LEFT are acceptable parameters for the X Axis, and describe the direction of the positive axis. For the Y Axis UP and DOWN are possible values.	
Parameters	*XAxis*	Acceptable values are RIGHT, LEFT.
	YAxis	Acceptable values are UP, DOWN.
Example	Sets the positive X axis RIGHT and Y axis UP. `:prob:xy:ori RIGHT UP`	

:PROBEPLAN:XY:ORIENTATION?

Short Command	`:prob:xy:ori?`

Description	This command queries the positive direction of the X and Y-axis of the currently loaded wafer map. Possible return values for the X-axis are "RIGHT" and "LEFT". Return values for the Y-axis are "UP" and "DOWN".
Parameters	None

Example	Returns the values of RIGHT for X and UP for Y.
	`:prob:xy:ori?`
	`RIGHT UP`

:PROFILE:USE:WAFERMAP

Short Command	*Use Map*	`:prof:use:waf`
Description	Defines whether or not a wafer map is used to define the profile locations used during a Z-Profile. To set the wafer map that is used during a profile use: :probeplan:load. Note: The wafer must be aligned and the PTPA set in order for Z-Profile with wafer map to work properly. The selected dies of the wafer map are used as the profile locations.	
Parameters	*Use Map*	Defines whether or not to use a wafer map during a Z-Profile operation. Valid values are ON or OFF.
Example	Enables the use of a wafer map to define the points used during a profile. `:profile:use:wafermap 0`	
Related Commands	*`:profile:use:wafermap?`* *`:profile:wafer:busy?`* *`:profile:wafer:density`* *`:profile:wafer:density?`* *`:profile:wafer:filename`* *`:profile:wafer:filename?`* *`:profile:wafer:mode?`* *`:profile:wafer:mode`* *`:profile:wafer:size`* *`:profile:wafer:size?`* *`:profile:wafer:start`*	

:PROFILE:USE:WAFERMAP?

Short Command	`:prof:use:waf?`
Description	Returns whether or not a wafer map file is being used to define auto focus locations for a Z-Profile.
Parameters	None

Example	Returns that a wafer map is being used to define XY profile locations.
	`:profile:use:wafermap?`
	`ON`

Related Commands	*:profile:use:wafermap* *:profile:wafer:busy?* *:profile:wafer:density* *:profile:wafer:density?* *:profile:wafer:filename* *:profile:wafer:filename?* *:profile:wafer:mode?* *:profile:wafer:mode* *:profile:wafer:size* *:profile:wafer:size?* *:profile:wafer:start*

:PROFILE:WAFER:BUSY?

Short Command	`:prof:waf:busy?`
Description	Returns TRUE or COMPLETE depending on the state of a Z-Profile operation. TRUE means that the Z-Profile is currently busy and COMPLETE means the Z-Profile has finished. A Z-Profile operation can take a long time to finish, so this command can be used to check the status of the currently running Z-Profile.
Parameters	None

Example	Returns that a Z-Profile is currently in progress.
	`:profile:wafer:busy?`
	`TRUE`

Related Commands	*:profile:use:wafermap* *:profile:use:wafermap?* *:profile:wafer:density* *:profile:wafer:density?* *:profile:wafer:filename* *:profile:wafer:filename?* *:profile:wafer:mode?* *:profile:wafer:mode* *:profile:wafer:size* *:profile:wafer:size?* *:profile:wafer:start*

:PROFILE:WAFER:CANCEL

Short Command	`:prof:waf:canc`
Description	This command is used to cancel a Z-Profile operation. Z-Profiles can take a long time to complete and if for any reason the system needs to interrupt the operation, this command can be used. The command `:profile:wafer:start` sends a return immediately and the Z-Profile process can be monitored with the `:profile:wafer:busy?` query. This allows time for the cancel command to take effect.
Parameters	None

Example	Aborts a Z-Profile operation.
	`:profile:wafer:cancel`
	`COMPLETE`

Related Commands	*`:profile:wafer:start`* *`:profile:wafer:busy?`* *`:profile:wafer:retries`* *`:profile:wafer:search:distance`* *`:profile:wafer:success:percent`* *`:profile:wafer:guardband`*

:PROFILE:WAFER:DENSITY

Short Command	*Ring Density*	`:prof:waf:dens`
Description	Defines the density of profile points used during a Z-Profile. Valid values are: LOW, MEDIUM, and HIGH.	
Parameters	*Ring Density*	Valid values are LOW, MEDIUM, or HIGH
Example	Sets the current ring density to HIGH. `:profile:wafer:density HIGH`	
Related Commands	*`:profile:use:wafermap`* *`:profile:use:wafermap?`* *`:profile:wafer:busy?`* *`:profile:wafer:density?`* *`:profile:wafer:filename`* *`:profile:wafer:filename?`* *`:profile:wafer:mode?`* *`:profile:wafer:mode`* *`:profile:wafer:size`* *`:profile:wafer:size?`* *`:profile:wafer:start`*	

:PROFILE:WAFER:DENSITY?

Short Command	`:prof:waf:dens?`
Description	Returns the currently set ring density. Possible return values are: LOW, MEDIUM, and HIGH.
Parameters	None

Example	Returns the current ring density.
	`:profile:wafer:density?`
	`HIGH`

Related Commands	*:profile:use:wafermap* *:profile:use:wafermap?* *:profile:wafer:busy?* *:profile:wafer:density* *:profile:wafer:filename* *:profile:wafer:filename?* *:profile:wafer:mode?* *:profile:wafer:mode* *:profile:wafer:size* *:profile:wafer:size?* *:profile:wafer:start*

:PROFILE:WAFER:FILENAME

Short Command	*Pathname*	`:prof:waf:fil`
Description	Defines the Z-Profile file name that is used when a Z-Profile operation starts. All Z-Profile files have the ".pro" extension.	
Parameters	*Pathname*	This is a fully qualified pathname to file that is used during a Z-Profile. To include spaces, you must enclose the name in "quotation marks".
Example	Sets the current Z-Profile to "300mm Z-Profile Test Wafer.pro". `:profile:wafer:filename "300mm Z-Profile Test Wafer.pro"`	
Related Commands	*:profile:use:wafermap* *:profile:use:wafermap?* *:profile:wafer:busy?* *:profile:wafer:density* *:profile:wafer:density?* *:profile:wafer:filename?* *:profile:wafer:mode?* *:profile:wafer:mode* *:profile:wafer:size* *:profile:wafer:size?* *:profile:wafer:start*	

:PROFILE:WAFER:FILENAME?

Short Command	`:prof:waf:fil?`
Description	Returns the currently active Z-Profile file name.
Parameters	None

Example	Returns that the current Z-Profile is "300mm Z-Profile Test Wafer.pro".
	`:profile:wafer:filename?`
	`300mm Z-Profile Test Wafer.pro`

Related Commands	*:profile:use:wafermap* *:profile:use:wafermap?* *:profile:wafer:busy?* *:profile:wafer:density* *:profile:wafer:density?* *:profile:wafer:filename* *:profile:wafer:mode?* *:profile:wafer:mode* *:profile:wafer:size* *:profile:wafer:size?* *:profile:wafer:start*

:PROFILE:WAFER:FOCUS:OPTIMIZE

Short Command	*Enable On*	`:prof:waf:foc:opt`
	Enable Off	
Description	If Z profile error recovery is enabled (see :profile:wafer:use:recovery), this command can be used to enable auto focus optimize. If there is an error during auto focus and focus optimized is enabled then the software will attempt to find a better focus target within the field of view by analyzing regions of contrast.	
Parameters	*Enable On*	Turns ON the auto focus optimize routine.
	Enable Off	Turns OFF the auto focus optimize routine.
Example	Enables the focus optimizer if error recovery is enabled for Z-Profile. `:profile:wafer:focus:optimize ON`	
Related Commands	*:profile:use:wafermap* *:profile:use:wafermap?* *:profile:wafer:busy?* *:profile:wafer:density?* *:profile:wafer:filename* *:profile:wafer:filename* *:profile:wafer:filename?* *:profile:wafer:mode?* *:profile:wafer:mode* *:profile:wafer:size* *:profile:wafer:size?* *:profile:wafer:start* *:profile:wafer:focus:optimize?* *:profile:wafer:search:band* *:profile:wafer:search:band?* *:profile:wafer:testsite:percent* *:profile:wafer:testsite:percent?* *:profile:wafer:use:recovery* *:profile:wafer:use:recovery?*	

:PROFILE:WAFER:FOCUS:OPTIMIZE?

Short Command	`:prof:waf:foc:opt?`
Description	Returns the current state of the Z profile focus optimize setting. Returns ON or OFF depending on the state.

Parameters	None

Example	Returns that the focus optimizer has been enabled for Z-Profile.
	`:profile:wafer:focus:optimize?`
	`ON`

Related Commands	*:profile:use:wafermap* *:profile:use:wafermap?* *:profile:wafer:busy?* *:profile:wafer:density?* *:profile:wafer:filename* *:profile:wafer:filename* *:profile:wafer:filename?* *:profile:wafer:mode?* *:profile:wafer:mode* *:profile:wafer:size* *:profile:wafer:size?* *:profile:wafer:start* *:profile:wafer:focus:optimize* *:profile:wafer:search:band* *:profile:wafer:search:band?* *:profile:wafer:testsite:percent* *:profile:wafer:testsite:percent?* *:profile:wafer:use:recovery* *:profile:wafer:use:recovery?*

:PROFILE:WAFER:GUARDBAND

Short Command	`:prof:waf:guar`	
Description	This value defines the amount of difference in Z heights between maximum and minimum that can be tolerated before a Z-Profile is considered invalid.	
Parameters	*Band*	The amount of difference in maximum and minimum Z heights that is allowable.

Example	If the difference between the maximum and minimum Z height is larger than 100, then the profile is considered a failure and will not be used.
	`:profile:wafer:guardband 100`
	`COMPLETE`	

Related Commands	*:profile:wafer:start* *:profile:wafer:busy?* *:profile:wafer:cancel* *:profile:wafer:retries* *:profile:wafer:search:distance* *:profile:wafer:success:percent*	

:PROFILE:WAFER:MODE

Short Command	*Mode*	`:prof:waf:mode`
Description	Sets the mode that a Z-Profile uses when the operation is started. Possible values are "MANUAL" or "SEMIAUTO". In order to perform a "SEMIAUTO" Z-Profile, you must be using an eVue digital imaging system.	

Parameters	*Mode*	Valid mode parameters are "MANUAL" or "SEMIAUTO".
Example	Sets the current Z-Profile to use manual mode. `:profile:wafer:mode MANUAL`	
Related Commands	*:profile:use:wafermap* *:profile:use:wafermap?* *:profile:wafer:busy?* *:profile:wafer:density* *:profile:wafer:density?* *:profile:wafer:filename* *:profile:wafer:filename?* *:profile:wafer:mode?* *:profile:wafer:size* *:profile:wafer:size?* *:profile:wafer:start*	

:PROFILE:WAFER:MODE?

Short Command	`:prof:waf:mode?`
Description	Returns what mode Z-Profile is currently using. Possible values are "MANUAL" or "SEMIAUTO".
Parameters	None

Example	Returns that the current Z-Profile is in manual mode.
	`:profile:wafer:mode?`
	`MANUAL`

Related Commands	*:profile:use:wafermap* *:profile:use:wafermap?* *:profile:wafer:busy?* *:profile:wafer:density* *:profile:wafer:density?* *:profile:wafer:filename* *:profile:wafer:filename?* *:profile:wafer:mode* *:profile:wafer:size* *:profile:wafer:size?* *:profile:wafer:start*

:PROFILE:WAFER:QUALITY:SCORE

Short Command	*Score*	`:prof:waf:qual:scor`
Description	Each Auto Focus has a quality score. The first Z-Profile point is used as the baseline score for all other points. If a subsequent profile point has an Auto Focus score that is different by more than <Score> then that point is considered a failure. When a profile point is a failure, the Z-Profile will attempt to retry the surrounding region based on the :profile:wafer:retries value. Example: First Profile Auto-Focus score = 1254 and <Score> = 500. Any Auto-Focus with a better score is good, any point with a score less than (1254 - 500 =) 754 is considered a failure.	

Parameters	*Score*	Defines the amount of difference in Auto-Focus score that two Z-Profile points can tolerate before the Z-Profile considers that profile point a failure.

Example	If the difference between the first quality score and any subsequent score is more than 100 then that point is considered a failure.
	`:profile:wafer:quality:score 100`
	`COMPLETE`	

Related Commands	*`:profile:wafer:start`* *`:profile:wafer:busy?`* *`:profile:wafer:cancel`* *`:profile:wafer:retries`* *`:profile:wafer:search:distance`* *`:profile:wafer:success:percent`* *`:profile:wafer:guardband`*	

:PROFILE:WAFER:RETRIES

Short Command	*Retries*	`:prof:waf:retr`
Description	Used in failure recovery for Z-Profile. Defined as the number of times the system retries the surrounding area of a failed profile location. If the initial Z-Profile location fails then the recovery systems is activated.	
Parameters	*Retries*	Number of times the system will retry a surrounding area before it considers the location a failure.

Example	Sets the number of retires to 4.
	`:profile:wafer:retries 4`
	`COMPLETE`	

Related Commands	*`:profile:wafer:start`* *`:profile:wafer:busy?`* *`:profile:wafer:cancel`* *`:profile:wafer:search:distance`* *`:profile:wafer:success:percent`* *`:profile:wafer:guardband`*	

:PROFILE:WAFER:SEARCH:BAND

Short Command	*Search Band*	`:prof:waf:sear:band`
Description	When using auto focus with Z-Profile, this command defines the amount of total distance that the fine focus stage travels when it is searching for optimal focus.	
Parameters	*Search Band*	Distance in microns that is used during an auto focus operation during Z-Profile.
Example	Sets the search band for auto focus during Z-Profile to 500. `:profile:wafer:search:band 500`	

Related Commands	*:profile:use:wafermap* *:profile:use:wafermap?* *:profile:wafer:busy?* *:profile:wafer:density* *:profile:wafer:filename* *:profile:wafer:filename?* *:profile:wafer:mode* *:profile:wafer:mode?* *:profile:wafer:size* *:profile:wafer:size?* *:profile:wafer:start* *:profile:wafer:focus:optimize* *:profile:wafer:focus:optimize?* *:profile:wafer:search:band?* *:profile:wafer:testsite:percent* *:profile:wafer:testsite:percent?* *:profile:wafer:use:recovery* *:profile:wafer:use:recovery?*

:PROFILE:WAFER:SEARCH:BAND?

Short Command	`:prof:waf:sear:band?`
Description	Returns the distance that is used by the fine focus stage during an auto focus operation when Z-Profiling a wafer. The distance is in microns.
Parameters	None

Example	Returns that auto focus is using 500 microns as search band during a Z-Profile.
	`:profile:wafer:search:band`
	`500`

Related Commands	*:profile:use:wafermap* *:profile:use:wafermap?* *:profile:wafer:busy?* *:profile:wafer:density* *:profile:wafer:filename* *:profile:wafer:filename?* *:profile:wafer:mode* *:profile:wafer:mode?* *:profile:wafer:size* *:profile:wafer:size?* *:profile:wafer:start* *:profile:wafer:focus:optimize* *:profile:wafer:focus:optimize?* *:profile:wafer:search:band?* *:profile:wafer:testsite:percent* *:profile:wafer:testsite:percent?* *:profile:wafer:use:recovery* *:profile:wafer:use:recovery?*

:PROFILE:WAFER:SEARCH:DISTANCE

Short Command	*Distance*	`:prof:waf:sear:dist`
Description	Used in failure recovery for Z-Profile. Search distance is the distance away from the original failed profile location that the system moves when searching for a better target.	
Parameters	*Distance*	Distance away from the original failed profile location that the system moves when searching for a better target.

Example	If there is a failure, the search distance used is 10000.
	`:profile:wafer:search:distance 10000`
	`COMPLETE`	

Related Commands	*`:profile:wafer:start`* *`:profile:wafer:busy?`* *`:profile:wafer:cancel`* *`:profile:wafer:retries`* *`:profile:wafer:success:percent`* *`:profile:wafer:guardband`*	

:PROFILE:WAFER:SIZE

Short Command	*Size*	`:prof:waf:size`
Description	Sets the diameter that is used during a Z-Profile.	
Parameters	*Size*	Diameter that is used during profile. Valid values are 100, 150, 200, 300. All numbers are in mm. (Size – Diameter that is used during profile. Valid values are 100, 150, 200, 300. All numbers are in mm)

Example	Sets the diameter used for a profile to 300mm and starts a profile running.
	`:profile:wafer:size 300`
	`:profile:wafer:start`	

Related Commands	*`:profile:use:wafermap`* *`:profile:use:wafermap?`* *`:profile:wafer:busy?`* *`:profile:wafer:density`* *`:profile:wafer:density?`* *`:profile:wafer:filename`* *`:profile:wafer:filename?`* *`:profile:wafer:mode?`* *`:profile:wafer:mode`* *`:profile:wafer:size`* *`:profile:wafer:size?`* *`:profile:wafer:start`*	

:PROFILE:WAFER:SIZE?

Short Command	`:prof:waf:size?`

Description	Returns the current diameter that is used during a Z-Profile. Valid return values are 100, 150, 200, 300. All values are in mm.
Parameters	None

Example	Returns the current diameter used for a profile.
	`:profile:wafer:size?`
	`300`

Related Commands	*:profile:use:wafermap* *:profile:use:wafermap?* *:profile:wafer:busy?* *:profile:wafer:density* *:profile:wafer:density?* *:profile:wafer:filename* *:profile:wafer:filename?* *:profile:wafer:mode?* *:profile:wafer:mode* *:profile:wafer:size* *:profile:wafer:start*

:PROFILE:WAFER:START

Short Command	`:prof:waf:star`
Description	Starts a Z-Profile operation. Once all of the values for a Z-Profile have been set up, the :profile:wafer:start command is used to start the profile running. The commands listed in the "Related Commands" section define all the possible setup values that Z-Profile can have.
Parameters	None
Example	Starts a profile running with current values set for a Z-Profile. `:profile:wafer:start`
Related Commands	*:profile:use:wafermap* *:profile:use:wafermap?* *:profile:wafer:busy?* *:profile:wafer:density* *:profile:wafer:density?* *:profile:wafer:filename* *:profile:wafer:filename?* *:profile:wafer:mode?* *:profile:wafer:mode* *:profile:wafer:size* *:profile:wafer:size?*

:PROFILE:WAFER:SUCCESS:PERCENT

Short Command	*Percent*	`:prof:waf:succ:perc`
Description	Indicates the amount of successful auto-focuses that Z-Profile must achieve. 95 would mean that 95% of the total targets must be successful for the Z-Profile to be considered good.	

Parameters	*Percent*	Defines the success percent that the Z-Profile must achieve in order to consider the current Z-Profile valid. Valid ranges are 10 to 100.

Example	When the next Z-Profile is started 100% of the points must be successful or the Z-Profile will fail.
	`:profile:wafer:success:percent 100`
	`COMPLETE`	

Related Commands	*:profile:wafer:start* *:profile:wafer:busy?* *:profile:wafer:cancel* *:profile:wafer:retries* *:profile:wafer:search:distance* *:profile:wafer:guardband*	

:PROFILE:WAFER:TESTSITE:PERCENT

Short Command	*Percent*	`:prof:waf:test:perc`
Description	Defines the success percent that a Z-Profile must have in order to be considered valid. If the percent is 95% and there are 40 total points then at least 38 points must pass auto focus. 100% means all points must pass.	
Parameters	*Percent*	The percent value must be between 10 and 100.
Example	Sets the success percent to 95%. `:profile:wafer:testsite:percent 95`	
Related Commands	*:profile:use:wafermap* *:profile:use:wafermap?* *:profile:wafer:busy?* *:profile:wafer:density* *:profile:wafer:filename* *:profile:wafer:filename?* *:profile:wafer:mode* *:profile:wafer:mode?* *:profile:wafer:size* *:profile:wafer:size?* *:profile:wafer:start* *:profile:wafer:focus:optimize* *:profile:wafer:focus:optimize?* *:profile:wafer:search:band* *:profile:wafer:search:band?* *:profile:wafer:testsite:percent?* *:profile:wafer:use:recovery* *:profile:wafer:use:recovery?*	

:PROFILE:WAFER:TESTSITE:PERCENT?

Short Command	`:prof:waf:test:perc?`
Description	Returns the percent of focus points that must be valid before a Z-Profile is considered valid.

Parameters	None

Example	Returns that the success percent is 95.
	`:profile:wafer:testsite:percent?`
	`95`

Related Commands	*`:profile:use:wafermap`* *`:profile:use:wafermap?`* *`:profile:wafer:busy?`* *`:profile:wafer:density`* *`:profile:wafer:filename`* *`:profile:wafer:filename?`* *`:profile:wafer:mode`* *`:profile:wafer:mode?`* *`:profile:wafer:size`* *`:profile:wafer:size?`* *`:profile:wafer:start`* *`:profile:wafer:focus:optimize`* *`:profile:wafer:focus:optimize?`* *`:profile:wafer:search:band`* *`:profile:wafer:search:band?`* *`:profile:wafer:testsite:percent?`* *`:profile:wafer:use:recovery`* *`:profile:wafer:use:recovery?`*

:PROFILE:WAFER:USE:RECOVERY

Short Command	*Enable*	`:prof:waf:use:rec`
Description	Defines whether or not error recovery is used for Z-Profile. If a profile point is invalid and error recovery is ON then the software makes an attempt to recover from the error otherwise the point is counted as failed and Z-Profile moves on to the next location. Settings for error recovery can be set with :profile:wafer:focus:optimize and :profile:wafer:retries.	
Parameters	*Enable*	Possible values are ON or OFF.
Example	Enables Z-Profile error recovery. `:profile:wafer:use:recovery ON`	
Related Commands	*`:profile:use:wafermap`* *`:profile:use:wafermap?`* *`:profile:wafer:busy?`* *`:profile:wafer:density`* *`:profile:wafer:filename`* *`:profile:wafer:filename?`* *`:profile:wafer:mode`* *`:profile:wafer:mode?`* *`:profile:wafer:size`* *`:profile:wafer:size?`* *`:profile:wafer:start`* *`:profile:wafer:focus:optimize`* *`:profile:wafer:focus:optimize?`* *`:profile:wafer:search:band`* *`:profile:wafer:search:band?`* *`:profile:wafer:testsite:percent?`* *`:profile:wafer:use:recovery?`*	

:PROFILE:WAFER:USE:RECOVERY?

Short Command	`:prof:waf:use:rec?`
Description	Returns the current setting for Z-Profile error recovery. Possible return values are ON or OFF. See :profile:wafer:use:recovery.
Parameters	None
Example	Returns that profile uses error recovery. `:profile:wafer:use:recovery?` ON
Related Commands	*:profile:use:wafermap* *:profile:use:wafermap?* *:profile:wafer:busy?* *:profile:wafer:density* *:profile:wafer:filename* *:profile:wafer:filename?* *:profile:wafer:mode* *:profile:wafer:mode?* *:profile:wafer:size* *:profile:wafer:size?* *:profile:wafer:start* *:profile:wafer:focus:optimize* *:profile:wafer:focus:optimize?* *:profile:wafer:search:band* *:profile:wafer:search:band?* *:profile:wafer:testsite:percent?* *:profile:wafer:use:recovery*

:REVERSE:ORIENTATION

Short Command	*devID*	*(on)* *(off)*	`:rev:ori`
Description	This command reverses the polarity of the user coordinate system. This reversal only applies to the x- and y-axes.		
Parameters	*devID*	2 for Elite, Summit, S300 and Alessi 6100 emulation. This command can only be used with the chuck channel.	
	(on)(off)	The on/off parameter is **off** for the standard coordinate space and **on** for reversed coordinate space. The default setting is **off**.	
Example	`:rev:ori 2 on` `:rev:ori 2 off`		
Related Commands	*:move:absolute* *:move:absolute?* *:move:absolute:chuck* *:move:absolute:options* *:move:relative* *:move:relative:chuck*		

:SET:AUTOXY:LIGHT:AUTO

Short Command	*Mode*	`:set:aut:ligh:aut`
Description	This command can be used to enable or disable the automatic light switching mode of Auto XY correction. When automatic light switch mode is ON, the light control is automatically turned on before a pattern recognition operation is performed and then shuts the light off after the operation is over. The delay time in milliseconds to wait after turning the light on, can be set with :set:autoxy:light:on:delay command. The :set:autoxy:light:off:delay can be used to set the delay used to wait for the light to turn off.	
Parameters	*Mode*	Return values are ON or OFF.

Example	Enable the automatic light switching during Auto XY die stepping.
	`:set:autoxy:light:auto ON`
	`ON`	

Related Commands	*`:set:autoxy:light:auto?`* *`:set:autoxy:light:on:delay`* *`:set:autoxy:light:off:delay`*	

:SET:AUTOXY:LIGHT:AUTO?

Short Command	`:set:aut:ligh:aut?`
Description	Returns the current state of the automatic light switching mode. Return values are ON or OFF.
Parameters	None

Example	Returns OFF, meaning that the automatic light switching mode is OFF.
	`:set:autoxy:light:auto?`
	`OFF`

Related Commands	*`:set:autoxy:light:auto`* *`:set:autoxy:light:on:delay`* *`:set:autoxy:light:off:delay`*

:SET:AUTOXY:LIGHT:ON:DELAY

Short Command	*Time*	`:set:aut:ligh:on:del`
Description	Defines the amount of time to wait after the light has been turned on before a pattern recognition operation is performed. Some illuminators take longer to turn on then others, so this command can be used to adjust the amount of time that automatic light switching delays before doing the vision pattern search.	
Parameters	*Time*	The amount of milliseconds to wait before doing a pattern search on the current die.

Example	Sets the amount of time to 200 ms to wait after the light is turned on.
	`:set:autoxy:light:on:delay 200`
	`COMPLETE`	

Related Commands	*:set:autoxy:light:auto* *:set:autoxy:light:on:delay?* *:set:autoxy:light:off:delay*

:SET:AUTOXY:LIGHT:ON:DELAY?

Short Command	`:set:aut:ligh:on:del?`
Description	Returns the amount of time in milliseconds that the light on delay is set to use. This value can be changed with the :set:autoxy:light:on:delay command.
Parameters	None

Example	Returns a delay time of 200 ms.
	`:set:autoxy:light:on:delay?`
	`200`

Related Commands	*:set:autoxy:light:auto* *:set:autoxy:light:on:delay* *:set:autoxy:light:off:delay*

:SET:AUTOXY:LIGHT:OFF:DELAY

Short Command	*Time*	`:set:aut:ligh:off:del`
Description	This command is used to define the amount of time to wait after the light has been turned off. Each type of microscope illuminator can have a different amount of time before the light has been fully dissipated.	
Parameters	*Time*	The amount of time in milliseconds to wait after the light has been turned off.

Example	Returns sets the off delay time to 200 ms.
	`:set:autoxy:light:off:delay`
	`200`	

Related Commands	*:set:autoxy:light:auto* *:set:autoxy:light:on:delay* *:set:autoxy:light:off:delay?*	

:SET:AUTOXY:LIGHT:OFF:DELAY?

Short Command	`:set:aut:ligh:off:del?`
Description	Returns the amount of time that the system waits after turning the light off during an automatic light switch operation. The time interval can be set with the command :set:autoxy:light:off:delay command
Parameters	None

Example	Returns the value of the light off timer.
	`:set:autoxy:light:off:delay?`
	`200`

Related Commands	*:set:autoxy:light:auto* *:set:autoxy:light:on:delay* *:set:autoxy:light:off:delay*

:SET:AUXILIARY

Short Command	*(on)* *(off)*	`:set:aux`
Description	This command turns the power to the probe station auxiliary outlet on or off.	
Parameters	*(on)* *(off)*	Specifies whether auxiliary power on the station is **on** or **off**. The default setting is **off**.
Example	Turns on auxiliary power on. `:set:aux on`	

:SET:BUSY?

Short Command	`:set:bus?`	
Description	This query checks to see if the probe station is currently performing any operations.	
Parameters	*(true)* *(false)*	Returns TRUE if the probe station is busy and FALSE if it is idle.

Example	The following shows this command sent to an idle probe station.
	`:set:busy?`
	`FALSE`	

:SET:CHAMBER:PURGE

Short Command	*Purge State*	`:set:cham:purg`
Description	Controls the amount of MicroChamber air purge. This command is available only on the Elite 300 probe station.	
Parameters	*Purge State*	Possible values are OFF, MANUAL, QUICK, AUTO. OFF: turns off MicroChamber air purge. MANUAL: the air flow meter on the station is responsible for the amount air flow to the chamber. QUICK: sets the purge state to maximum air flow. AUTO: the software is in charge of controlling the amount of air flow to optimize temperature transitions.
Example	Sets the air chamber purge state to QUICK. `:set:chamber:purge QUICK`	
Related Commands	*`:set:chamber:purge?`*	

:SET:CHAMBER:PURGE?

Short Command	`:set:cham:purg?`

Description	Returns the current MicroChamber air purge setting. Possible values are OFF, MANUAL, QUICK, and AUTO. See :set:chamber:purge. This command is only available on the Elite probe station.
Parameters	None

Example	Returns that the system is using QUICK purge.
	`:set:chamber:purge?`
	`QUICK`

Related Commands	*`:set:chamber:purge`*

:SET:COMPENSATION

Short Command	*(on)* *(off)* *(thermal)*	`:set:comp`
Description	This command turns compensation mode on or off. When turned on, the currently defined compensation factor is used to correct for thermal expansion or other device-position variables. The correction is applied to **ALL** **`:move:absolute`**, **`:move:relative`**, and **`:move:probeplan`** commands. It is also applied to device-position queries. This command functions identically to the **`:set:compensation:mode`** command. Generally, you should use the Nucleus menus to calculate compensation factors. Or, you can numerically set the compensation factors with the **`:set:compensation:factor`** command.	
Parameters	*(on)*	Specifies whether or not to use the compensation mode. This parameter can have the value on, off or thermal.
	(off)	The default value is off.
	(thermal)	The thermal mode will use a coefficient of expansion along with the current temperature to dynamically calculate new temperature compensation factors. The coefficient of expansion can be set in the Nucleus software.
Example	Turns on compensation. `:set:comp on`	

:SET:COMPENSATION:FACTOR

Short Command	*xFact*	`:set:comp:fact`
	yFact	
Description	This command numerically sets the compensation factors for movement along the x- and y-axes. The compensation factor is a value by which device moves are multiplied in order to compensate for thermal expansion and other device position variables. Use the `:set:comp` command to turn on compensation. If you do not know the necessary compensation factors, you can use Nucleus menu operations to measure the difference between the nominal and actual position of a test site and calculate the compensation factors.	

Parameters	*xFact*	The value used to calculate the distance the device should move along the x-axis (that is, right or left movement). The nominal distance is multiplied by this value to get the actual distance.
	yFact	The value used to calculate the distance the device should move along the y-axis (that is, forward or backward movement). The nominal distance is multiplied by this value to get the actual distance.
Error Condition	An error results if the compensation factor causes device movement parameters to fall outside the limits of device travel.	

Example	The following example sets the x- and y-axis compensation factors to increase the move distance by 1 percent along each axis, then turns on compensation mode, moves the device, and queries for the new location. Note that the returned device location is the same as the nominal location, not the actual position with compensation applied. The compensation factor is also applied to the query so that the use of compensation is not reflected in the display of coordinates.
	`:set:comp:fact 1.01 1.01`
	`:set:comp:mode on`
	`:mov:abs 2 +11980 -34 none`
	`:mov:abs? 2`
	`+011980 -000034`	

Related Commands	`ALL` *`:move:absolute`* `commands`
	`ALL` *`:move:probeplan:absolute:die`* `commands`
	`ALL` *`:move:relative`* `commands`
	`:set:compensation:mode`
	`:set:compensation`	

:SET:COMPENSATION:MODE

Short Command	*(on)* *(off)* *(thermal)*	`:set:comp:mode`
Description	This command turns compensation mode on or off. When turned on, the currently defined compensation factor is used to correct for thermal expansion or other device-position variables. The correction is applied to **ALL** **`:move:absolute`**, **`:move:relative`**, and **`:move:probeplan`** commands. It is also applied to device-position queries. This command functions identically to the **`:set:compensation`** command. Generally, you should use the Nucleus menus to calculate compensation factors. Or, you can numerically set the compensation factors with the **`:set:compensation:factor`** command.	
Parameters	*(on)*	Specifies whether or not to use the compensation mode. This parameter can have the value **on**, **off** or **thermal**.
	(off)	The default value is **off**.
	(thermal)	The thermal mode will use a coefficient of expansion along with the current temperature to dynamically calculate new temperature compensation factors. The coefficient of expansion can be set in the Nucleus software.

Example	Turns on compensation mode. `:set:comp:mode on`

Related Commands	`ALL :move:absolute commands`
	`ALL :move:probeplan commands`
	`ALL :move:relative commands`
	`:set:compensation:mode`
	`:set:compensation`

:SET:CONTACT

Short Command	*devID* *contact*	`:set:cont`
Description	This command will set the contact location for the z-stage, and will set the contact location for each positioner used. When used for the z-stage, it defines the location that the DUT comes in contact with the probes. This parameter can be in either microns or mils, depending on the current units that were set with the **`:set:unit`** command. When used for the positioners, a second parameter is not used and the z-contact position defaults to the current position and the z-axis is reset to zero. **NOTE** When contact zones are enabled, this command will set the contact height for the zone that the stage is currently in. For example, if the stage is positioned so that auxilliary chuck 1 is in the probing area, then the contact height for that zone will be affected.	
Parameters	*devID*	2 for the chuck on Elite, Summit, S300 and Alessi 6100 emulation. See SCPI Command Device IDs in Nucleus Software. (not applicable to microscope).
	contact	For the chuck only, defines the z-stage location that the DUT comes in contact with the probes. This parameter can be in either Metric (microns) or English (Mil) units depending on the :set:unit command.

Example	Sets chuck contact at 5000 microns.
	`:set:unit metric`
	`:set:contact 2 5000`	

:SET:CONTACT?

Short Command	*devID*	`:set:cont?`
Description	This command will return the contact position of the Z stage. This parameter can be in either microns or mils, depending on the current units that were set with the **`:set:unit`** command. **NOTE** When contact zones are enabled, this command will return the contact height for the zone that the stage is currently in. For example, if the stage is positioned so that auxilliary chuck 1 is in the probing area, then the contact height for that zone will be returned.	

Parameters	*devID*	2 for Elite, Summit, S300 and Alessi 6100 emulation. See SCPI Command Device IDs in Nucleus Software. (not applicable to microscope).

Example	Shows that the contact location is at 5000 microns.
	`:set:unit?`
	`metric`
	`:set:contact? 2`
	`5000`	

:SET:CONTACT:ACTIVE

Short Command	*Mode*	*On* *Off*	`:set:cont:acti`
	Prompt	*On* *Off*	
Description	This command turns Contact mode on or off. Prompt is an optional parameter that can be used to enable or disable dialog prompts during Z-Profile setups.		
Parameters	*Mode*	ON OFF	
	Prompt	ON enables prompts to be displayed when turning on Z Profile. OFF disables all prompting. This parameter is optional and will default to ON if not provided.	
Example	Enables contact mode and disables the prompt dialogs. `:set:contact:active ON OFF`		

Related Commands	`:set:contact:mode`
	`:set:contact:use:zprofile`
	`:set:contact:use:zprofile?`
	`:set:contact:use:autoz`
	`:set:contact:use:autoz?`		

:SET:CONTACT:ACTIVE?

Short Command	`:set:cont:acti?`
Description	This command returns Contact mode status: On; Off.
Parameters	None

Example	Returns that the Contact mode is currently active.
	`:set:contact:active?`
	`ON`

:SET:CONTACT:BAND

Short Command	*Band*	`:set:cont:ban`
Description	This command sets the Contact/Search band for Contact mode. Use the `:set:unit` command to specify the units being used.	

Parameters	*Band*	Band value in units: defined by the `:set:unit` command.

Example	Sets the units to metric; sets the contact band to 500 microns; and then checks the command state.
	`:set:unit metric`
	`:set:contact:band 500`
	`:set:contact:band?`
	`500`	

:SET:CONTACT:BAND?

Short Command	`:set:cont:ban?`
Description	This command returns the current Contact/Search band for Contact mode. The command can be used to return the units that the command is using.
Parameters	None

Example	This command returns the Contact/Search band value.
	`:set:contact:band?`
	`500`

:SET:CONTACT:MODE

Short Command	*Standard*	`:set:cont:mode`
	Edgesense	
	Programmable	
Description		
Parameters	*Standard*	Sets the mode to Standard.
	Edgesense	Turns on the Hardware Edge Sense mode.
	Programmable	Sets the Contact mode to Programmable.

Example	Sets the contact mode to programmable and then checks the command state.
	`:set:contact:mode PROGRAMMABLE`
	`:set:contact:mode?`
	`PROGRAMMABLE`	

:SET:CONTACT:MODE?

Short Command	`:set:cont:mode?`
Description	This command returns a value: Standard; Edgesense; Programmable, depending on the active mode.
Parameters	None

Example	Returns the contact mode that is currently set.
	`:set:contact:mode?`
	`PROGRAMMABLE`

:SET:CONTACT:SPEED

Short Command	*Speed*	`:set:cont:spee`
Description	This command sets the search speed used for the Z-axis in Contact mode. Use the `:set:unit` command to specify the units to be used.	
Parameters	*Speed*	Speed value in units defined by the `:set:unit` command.

Example	Sets contact speed to 500 microns per second and then checks the speed.
	`:set:unit metric`
	`:set:contact:speed 500`
	`:set:contact:speed?`
	`500`	

:SET:CONTACT:SPEED?

Short Command	`:set:cont:spee?`
Description	This command returns the current search speed for the Z-axis in Contact mode. The `:set:unit` command may be used to set the returned units.
Parameters	None

Example	Returns the contact speed.
	`:set:contact:speed?`
	`500`

:SET:CONTACT:USE:AUTOZ

Short Command	*Mode*	`:prof:cont:use:aut`
Description	Enables Auto-Z functionality when Auto XYZ Correction mode is activated. Parameters must be ON or OFF.	
Parameters	*Mode*	ON enables Auto Z when using Auto XYZ Correction mode. OFF disables Auto Z.
Example	Enables Auto Z when Auto XYZ Correction mode is enabled. `:set:contact:use:autoz ON`	
Related Commands	`:set:contact:mode` `:set:contact:use:autoz?`	

:SET:CONTACT:USE:AUTOZ?

Short Command	`:prof:cont:use:aut?`
Description	Returns ON if Auto-Z is enabled in Auto XYZ Correction mode, otherwise returns OFF.
Parameters	None
Example	`:set:contact:use:autoz?` `ON`

Related Commands	`:set:contact:mode` `:set:contact:use:autoz`

:SET:CONTACT:USE:ZPROFILE

Short Command	*Mode*	`:prof:cont:use:zpr`
Description	Enables Z-Profile functionality when programmable contact mode is activated. Parameters must be ON or OFF.	
Parameters	*Mode*	ON Enables Z-Profile in programmable contact mode. OFF disables Z-Profile.
Example	Turns on Z-Profile. `:set:contact:use:profile ON`	

Related Commands	`:set:contact:mode`
	`:set:contact:active`
	`:set:contact:use:zprofile?`	

:SET:CONTACT:USE:ZPROFILE?

Short Command	`:prof:cont:use:zpr?`
Description	Returns ON if Z-Profile is enabled in programmable contact mode. OFF is returned if Z-Profile is not enabled.
Parameters	None

Example	Returns that Z-Profile has been enabled with programmable contact mode.
	`:set:contact:use:profile?`
	`ON`

Related Commands	`:set:contact:mode`
	`:set:contact:active`
	`:set:contact:use:zprofile`

:SET:DELAY

Short Command	*milliseconds*	`:set:del`
Description	This command specifies a time in milliseconds for the probe station to delay after each operation. During the delay, the probe station does not execute any commands.	
Parameters	*milliseconds*	Specifies the length of time, in milliseconds, which the probe station will delay before executing the next command. You can specify a delay time in the range 0 to 65535 msec (65.5 seconds). By default, the delay is 0.

Example	The first line specifies a delay time of 2 seconds and the second line specifies a delay time of 1 minute.
	`:set:del 2000`
	`:set:del 60000`	

Related Commands	*:set:delay?* *:system:delay*

:SET:DELAY?

Short Command	`:set:del?`
Description	This query returns the current delay time. The delay is the number of milliseconds the probe station waits after each operation before executing the next command. Use the `:set:delay` command to set the delay.
Parameters	None

Example	Shows a delay time of 2 seconds:
	`:set:del?`
	`2000`
	`The following example shows a delay time of 1 minute:`
	`:set:del?`
	`60000`

Related Commands	*:set:delay* *:system:delay*

:SET:EDGESENSE

Short Command	*devID*	*(on)* *(off)*	`:set:edg`
Description	This command turns edge sense mode on or off. When on, the chuck up position is the point at which the DUT makes contact with an edge sense probe plus any specified overdrive distance. Using edge sense makes it easier to raise the chuck to the optimal probing height. To set the parameters required to operate in edge sense mode, use the `:set:edgesense:overdrive` and `:set:edgesense:switch` commands.		
Parameters	*devID*	2 for Elite, Summit, S300 and Alessi 6100 emulation. Applicable to chuck only.	
	(on)(off)	Use **on** to activate edge sense mode and **off** to turn off edge sensing.	

Example	The following example sets the non-contacted position of the edge sense probe, the amount of overdrive, and turns edge sense mode on. In this example, the amount of overdrive is set to 75.
	`:set:edg:swit 2 nopen`
	`:set:edg:over 2 75`
	`:set:edg 2 on`		

Related Commands	*:set:edgesense:overdrive* *:set:edgesense:switch* *:move:contact*		

:SET:EDGESENSE?

Short Command	*devID*	*(on)* *(off)*	`:set:edg?`

Description	Returns whether or not contact mode is enabled.	
Parameters	*devID*	2 for Elite, Summit, S300 and Alessi 6100 emulation. Applicable to chuck only.

Example	The following example returns the status of edge sense: on or off.
	`:set:edg? 2`
	`ON`	

Related Commands	*`:set:edgesense:overdrive`* *`:set:edgesense:switch`* *`:move:contact`*	

:SET:EDGESENSE:OVERDRIVE

Short Command	*devID* *microns*	`:set:edg:over`
Description	This command sets the distance the chuck rises after contact is made with the edge sense probe. This command affects chuck movement only when the probe station is operating in edge sense mode.	
Parameters	*devID*	2 for Elite, Summit, S300 and Alessi 6100 emulation. Applicable to chuck only.
	microns	Specifies the distance the chuck will rise after contacting the edge sense probe.

Example	The following example sets the non-contacted position of the edge sense probe, the amount of overdrive, and turns edge sense mode on.
	`:set:edg:swit 2 nopen`
	`:set:edg:over 2 2`
	`:set:edg 2 on`	

Related Commands	*`:set:edgesense:overdrive`* *`:set:edgesense:switch`* *`:move:contact`*	

:SET:EDGESENSE:OVERDRIVE?

Short Command	`:set:edg:over?`
Description	This command returns the current overdrive amount that is set in Contact mode. The `:set:unit` command may be used to set the returned units.
Parameters	None

Example	Returns that the overdrive amount is 10 microns.
	`:set:edgesense:overdrive?`
	`10`

Related Commands	*`:set:edgesense:overdrive`* *`:set:edgesense:switch`* *`:move:contact`*

:SET:EDGESENSE:SWITCH

Short Command	*devID*		`:set:edg:swit`
	(nopen)	*(nclosed)*	

Description	This command identifies the non-contacted position of the switch in the edge sense probe. For example, if the switch opens when the DUT touches the probe tip, then the non-contacted position is closed.	
Parameters	*devID*	2 for Elite, Summit, S300 and Alessi 6100 emulation. Applicable to chuck only.
	(nopen)	Use **nopen** if the switch is normally open and **nclosed** if the switch is normally closed.
	(nclosed)	Use **nopen** if the switch is normally open and **nclosed** if the switch is normally closed.

Example	The following example sets the noncontacted position of the edge sense probe, the amount of overdrive, and turns edge sense mode on.
	`:set:edg:swit 2 nopen`
	`:set:edg:over 2 2`
	`:set:edg 2 on`	

Related Commands	*:set:edgesense:overdrive?* *:set:edgesense:switch?* *:set:edgesense* *:move:contact* *:set:edgesense:overdrive*	

:SET:EDGESENSE:SWITCH?

Short Command	`:set:edg:swit?`
Description	This command returns the current switch type that is defined for use with edge sense. Possible values are: "NOSWITCH"; "NOPEN"; "NCLOSED".
Parameters	None

Example	Returns that the switch type is normally closed.
	`:set:edgesense:switch?`
	`NCLOS`

Related Commands	*:set:edgesense:overdrive?* *:set:edgesense:switch?* *:set:edgesense* *:set:edgesense:overdrive*

:SET:HOST

Short Command	*(on)* *(off)*	`:set:host`
Description	This command closes the switch that connects the probe station to the host in the optional GPIB switch box. **NOTE** *Use this command for GPIB applications only.*	
Parameters	*(on)(off)*	Use **on** to connect the probe station to the host through the GPIB switch box and **off** to disconnect.
Error Condition	If there is no GPIB switch box installed, this command is ignored.	

Example	The following command closes the connection in the GPIB switch box, then queries for the status.
	`:set:host on`
	`:set:host?`
	`ON`

Related Commands	*`:set:host?`*

:SET:HOST?

Short Command	`:set:host`
Description	This query returns **on** if the host is connected through the optional GPIB switch box and **off** if the host is disconnected.
Error Condition	An error results if there is no GPIB switch box installed.

Example	The following command disconnects the host from the probe station in the GPIB switch box, then queries for the status.
	`:set:host off`
	`:set:host?`
	`OFF`

Related Commands	*`:set:host`*

:SET:INKER

Short Command	*Length*	`:set:ink`
Description	This command can be used to fire the inker. The (*Length*) parameter is the time in milliseconds that the inker is activated.	
Parameters	*Length*	Amount of time inker is kept activated in milliseconds. Valid is 0 to 2001.
Example	Fres the inker for 100 milliseconds. `:set:ink 100`	

:SET:JOYSTICK:ENABLED

Short Command	*enabled*	*(on)* *(off)*	`:set:joys:enab`
Description	This command can be used to enable or disable the joystick. When the joystick is disabled it can not control stage movements.		
Parameters	*enabled*	Possible values are either on or off. On will enable the joystick and off will disable the joystick	
Example	Disables the joystick. `:set:joystick:enabled OFF`		

:SET:JOYSTICK:ENABLED?

Short Command	`:set:joys:enab?`	
Description	This command will query whether or not the joystick is enabled. Possible return values are on or off. If on is returned the joystick is enabled, and if off is returned the joystick is disabled.	
Parameters	None	
Example	Shows that the joystick has been disabled. `:set:joystick:enabled? OFF`	

:SET:JOYSTICK:MODE

Short Command	*Mode*	`:set:joys:mode`
Description	Changes the motion control mode that is used by the joystick. This is the same as selecting a different mode in the combo box drop down on the motion control window. See *Nucleus User Guide* – Motion control window.	
Parameters	*Mode*	All motion modes are valid: SCAN, SCANAUTOZ, SCANZ, INDEX, DIE, SUBINDEX, JOG, THETA.
Example	Sets the joystick mode to INDEX. `:set:joystick:mode INDEX`	
Related Commands	*`:set:joystick:mode?`*	

:SET:JOYSTICK:MODE?

Short Command	`:set:joys:mode?`
Description	Returns the currently defined motion mode. Possible values are SCAN, SCANAUTOZ, SCANZ, INDEX, DIE, SUBINDEX, JOG, THETA. See *Nucleus User Guide* – Motion control window for a complete description of the different joystick modes.
Parameters	None

Example	Returns that the joystick is currently in INDEX mode.
	`:set:joystick:mode?`
	`INDEX`

Related Commands	*`:set:joystick:mode`*

:SET:LIGHT

Short Command	*(on)* *(off)*	`:set:ligh`
Description	This command turns the microscope light on or off.	
Parameters	*(on) (off)*	Specifies whether the microscope light for the probe station optics is **on** or **off**. By default, the light is **off**.

Example	Shows how to turn the microscope light on. `:set:ligh on`

:SET:LIGHT?

Short Command	`:set:ligh?`
Description	This command r eturns the current state of the microscope light.
Parameters	None

Example	Turns the light on and off querying the state after each command.
	`:set:light ON`
	`:set:light?`
	`ON`
	`:set:light OFF`
	`:set:light?`
	`OFF`

Related Commands	*`:set:light`*

:SET:MANUAL

Short Command	*devID*	*(on)* *(off)*	`:set:man`
Description	This command toggles Nucleus from semiautomatic to manual mode. The station must be in manual mode before using the x- and y-axis knobs on the probe station to move the device. The status window continues to track and display the current device location while in manual mode. Manual Mode is only available on the chuck channel of motion. When you exit manual mode, Nucleus lowers the chuck, moves the motors momentarily, and raises the chuck again.		
Parameters	*devID*	2 for Elite, Summit 12000-Series and S300.	
	(on)(off)	Specifies whether manual mode is on or off. The default is off.	
Example	Shows how to turn on manual mode. `:set:man 2 on`		
Related Commands	*`:set:manual?`*		

:SET:MANUAL?

Short Command	*devID*	*(on)* *(off)*	`:set:man?`
Description	This query returns ON if the probe station is operating in manual mode and OFF if operating in semiautomatic mode.		
Parameters	*DevID*	2 for Elite, Summit 12000-Series and S300.	

Example	Shows how to turn on manual mode and query for the status.
	`:set:man 2 on`
	`:set:man? 2`
	`ON`		

Related Commands	*:set:manual*

:SET:MICROSCOPE:AGC

Short Command	`:set:micr:agc`	
Description	This commands enables or disables the use of Automatic Gain Control for the microscope.	
Parameters	*Mode*	ON enables AGC OFF disables AGC
Example	This command turns on AGC: `:set:micr:agc on`	
Related Commands	*:set:micr:agc?*	

:SET:MICR:AGC?

Short Command	`:set:micr:agc?`
Description	This commands queries the current state of Automatic Gain Control for the microscope.
Parameters	None
Example	This command checks AGC: `:set:micr:agc?`
Related Commands	*:set:microscope:agc*

:SET:MICROSCOPE:AUTO:ILLUMINATE

Short Command	*Contrast/Brightness*	*On* *Off*	`:set:micr:auto:ill`
Description	(eVue Only) Runs an auto illuminate routine at the current AF bounding box. Contrast/Brightness is an optional parameter. If it is not provided, then Brightness, Exposure, and Contrast are all adjusted.		
Parameters	*Contrast/Brightness*	ON or OFF. Defines whether or not Contrast and Brightness are adjusted along with the Exposure. • If this parameter is OFF, then only Exposure is adjusted. • If this parameter is ON, then all are adjusted. • If parameter is not provided, then ON is used.	

Example	Runs an auto illumination at the current site.
	`:set:microscope:auto:illuminate`
	`COMPLETE`		

Related Commands	*:set:microscope:brightness* *:set:microscope:brightness?* *::set:microscope:exposure* *:set:microscope:exposure?* *:set:microscope:contrast* *:set:microscope:contrast?*

:SET:MICROSCOPE:BRIGHTNESS

Short Command	*Brightness*	`:set:micr:brig`
Description	Set the brightness level of a zoom microscope. This command has different actions depending on the type of hardware present. On A-Zoom based systems, it is used to set the microscope's light level. The A-Zoom only displays a one digit number on the actual digital control for brightness, i.e., 15 looks like "1" on the digital control. On eVue-based systems, it adjusts the amount of software brightness applied to the video. See	
Parameters	*Brightness*	Amount of brightness that is applied to the live video. A-Zoom Systems:Brightness is an integer value from 0 to 99. eVue Systems: Possible values range from 0.0 to 1.0. 1.0 is higher level of brightness than 0.0.

Example	Sets the brightness level to 0.23 on an eVue-based system.
	`:set:microscope:brightness 0.23`
	`COMPLETE`	

Related Commands	*:set:microscope:brightness?* *::set:microscope:exposure* *:set:microscope:exposure?* *:set:microscope:contrast* *:set:microscope:contrast?*	

:SET:MICROSCOPE:BRIGHTNESS?

Short Command	`:set:micr:brig?`
Description	Returns the brightness level of a Zoom microscope. On an A-Zoom based system, this command returns a brightness integer value from 0 to 99. The A-Zoom only displays a one-digit number on the actual digital control for brightness, i.e.,15 looks like "1" on the digital control. On eVue systems, this command returns the brightness based on a scale from 0.0 to 1.0, with 1.0 being higher in brightness level.
Parameters	None

Example	Returns that the brightness level is set to .23 on a eVue based system.
	`:set:microscope:brightness?`
	`0.230`

Related Commands	*:set:microscope:brightness* *:set:microscope:contrast* *:set:microscope:contrast?* *:set:microscope:exposure* *:set:microscope:exposure?*

:SET:MICROSCOPE:CONTRAST

Short Command	*Contrast*	`:set:micr:cont`
Description	Only supported on an eVue system. This command changes the level of contrast in the Live Video.	
Parameters	*Contrast*	Amount of contrast that is applied to the live video. Possible values range from 0.0 to 1.0. 1.0 is a higher level of contrast than 0.0.

Example	Sets the contrast level to 0.5.
	`:set:microscope:contrast 0.5`
	`COMPLETE`	

Related Commands	*:set:microscope:brightness* *:set:microscope:brightness?* *:set:microscope:contrast?* *:set:microscope:exposure* *:set:microscope:exposure?*	

:SET:MICROSCOPE:CONTRAST?

Short Command	`:set:micr:cont?`
Description	Only supported on an eVue system. This command returns the current contrast level. Possible values are 0.0 to 1.0.
Parameters	None

Example	Returns that the contrast level is set to .5.
	`:set:microscope:contrast?`
	`0.5`

Related Commands	*:set:microscope:brightness* *:set:microscope:brightness?* *:set:microscope:contrast* *:set:microscope:exposure* *:set:microscope:exposure?*

:SET:MICROSCOPE:EXPOSURE

Short Command	*Exposure*	`:set:micr:exp`
Description	Supported only on an eVue system. This command sets the amount of exposure that is used by the CCDs of an eVue microscope. Exposure rate of the CCDs has an effect on the frame rate. Faster exposure rates correspond to less light, but mean faster frames per second.	

Parameters	*Exposure*	Set the level of exposure used for the eVue microscope in milliseconds. Valid range is 0.5 to 100 ms.

Example	Sets the exposure rate to 30 milliseconds.
	`:set:microscope:exposure 30`
	`complete`	

Related Commands	*`:set:microscope:brightness`* *`:set:microscope:brightness?`* *`:set:microscope:contrast`* *`:set:microscope:contrast?`* *`:set:microscope:exposure?`*	

:SET:MICROSCOPE:EXPOSURE?

Short Command	`:set:micr:exp?`
Description	(This command applies only to an eVue system.) This command sets the amount of exposure that is used by the CCDs of an eVue microscope. Exposure rate of the CCDs has an effect on the frame rate. Faster exposure rates correspond to less light, but mean faster frames per second.
Parameters	None

Example	Returns that the exposure rate is set to 30 milliseconds.
	`:set:microscope:exposure?`
	`30`

Related Commands	*`:set:microscope:brightness`* *`:set:microscope:brightness?`* *`:set:microscope:contrast`* *`:set:microscope:contrast?`* *`:set:microscope:exposure`*

:SET:MICROSCOPE:ZOOM

Short Command	*Zoom*	`:set:micr:zoom`
Description	Valid only with a Zoom microscope that is connected to Nucleus or the eVue microscope. On A-Zoom systems, this command sets the zoom value used by the microscope. On eVue systems, this command sets the proper CCD zoom level to the appropriate optical path.	
Parameters	*Zoom*	Sets the current zoom level of the microscope. A-Zoom: valid ranges are 0 to 99. eVue 10x: valid ranges are 0.5 to 5.0. eVue 40x: valid ranges are 0.5 to 20.0.

Example	Sets the zoom value to 35 and checks the command state.
	`:set:microscope:zoom 35`
	`:set:microscope:zoom?`
	`35`	

Related Commands	*`:set:microscope:zoom?`*	

:SET:MICROSCOPE:ZOOM?

Short Command	`:set:micr:zoom?`	
Description	Valid only with a Zoom microscope that is connected to Nucleus or the eVue microscope. On A-Zoom systems, this command returns an integer value from 0 to 99, which is the zoom value set for the microscope. On eVue systems, this command, returns values ranging from 0.5 to 5.0 for a 10x system and 0.5 to 20.0 for a 40x system. The CCD parameter can be used to query a specific optical path in Multi-Camera mode.	
Parameters	*CCD*	The CCD parameter is valid only on eVue systems and is an optional parameter. Valid CCD parameters are 1, 2, or 3. If not provided, then CCD 1 is used.

Example	Returns a zoom value on CCD2 of an eVue system.
	`:set:microscope:zoom? 2`
	`2.5`	

Related Commands	*`:set:microscope:zoom`*	

:SET:PRESET

Short Command	*devID* *x* *y*	`:set:pres`
Description	This command presets the coordinate system coordinates for `:move:absolute` device movements by assigning the specified user coordinates to the current device location. You can assign any valid user coordinates to the current device position. **NOTE** The ***`:set:preset`*** command and the user values of status window in the UI are tied together. For example, pressing Set Zero in the status window will make a call to ***`:set:preset`***, and ***`:set:preset`*** will change the user values in the status window.	
Parameters	*devID*	2 for Elite, Summit, S300 and Alessi 6100 emulation. See SCPI Command Device IDs in Nucleus Software.
	x	Identifies the x coordinate that will be assigned to the current device location. The value of the parameter is interpreted as either mils or microns. Use the `:set:unit` command to switch between mils and microns. By default, the user coordinate system has the same origin as the probe-station machine coordinate system.
	y	Identifies the y coordinate that will be assigned to the current device location. The valid range of values for this command is the same as for the x parameter.

Example	Sets the user 0,0 location. This example assumes that you have already positioned the device at the location to which you want to assign the user 0,0. You can use `:move:abs?` to return the current device location, as shown here:
	`:mov:abs? 2`
	`+001158 -003341 000400`
	`:set:pres 2 0 0`
	`:mov:abs? 2`
	`+000000 +000000 000400`
	The following example shows how to assign coordinates x=150, y=281 to the current device location:
	`:set:pres 2 150 281`
	`:mov:abs? 2`
	`+000150 +000281 000400`

Related Commands	*:move:absolute* *:move:absolute:chuck* *:move:absolute?* *:move:absolute:options*

:SET:SALIGN

Short Command	*devID*		`:set:sal`
	(on)	*(off)*	
	x offset		
	y offset		
Description	This command turns on and off software alignment mode. When software alignment is on, Nucleus adds offsets to device movements to compensate for rotational misalignment between the DUT and the probe station axes. This is an alternative to aligning the hardware by adjusting theta. The x offset and y offset parameters are optional. If they are provided then Nucleus will use the angle created by x offset and y offset as the alignment angle. If these parameters are not given then the angle that was set up in Nucleus is used.		
Parameters	*devID*	2 for Elite, Summit, S300 and Alessi 6100 emulation. Applicable to chuck only.	
	(on) *(off)*	Specifies whether software angle compensation is on or off. Use the character strings **on** and **off** for this parameter.	
	x offset	Used along with the y offset parameter to define the angle of software correction. This parameter is given in microns. This parameter is optional. If the x offset parameter is given then the must also be given.	
	y offset	Used along with the x offset parameter to define the angle of software correction.	
Example	The following command turns on software alignment mode: `:set:sal 2 on`		
Related Commands	*:set:salign?*		

:SET:SALIGN?

Short Command	*devID*	`:set:sal?`
Description	This query returns on if the probe station is operating in software alignment mode and off if it is not.	
Parameters	*devID*	2 for Elite, Summit, S300 and Alessi 6100 emulation. Applicable to chuck only.

Example	The following commands turns on software alignment mode, then queries for the status:
	`:set:sal 2 on`
	`:set:sal? 2`
	`ON`	

Related Commands	*`:set:salign`*	

:SET:SEPARATE

Short Command	*devID* *separate*	`:set:sep`
Description	This command will set the separate distance away from contact used in auto Z stepping. It defines the distance away from contact that the DUT needs to be in order to clear the probes. Since separate is a distance, it must be a positive number. This parameter can be in either microns or mils, depending on the current units that were set with the **`:set:unit`** command.	
Parameters	*devID*	2 for Elite, Summit, S300 and Alessi 6100 emulation. See SCPI Command Device IDs in Nucleus Software (all but microscope).
	separate	Defines the Z distance to travel away from contact so the DUT is clear of the probes. This parameter must be greater than 0 because if defines a distance.

Example	Sets separate distance at 500 microns.
	`:set:unit metric`
	`:set:sep 2 500`	

:SET:SEPARATE?

Short Command	*devID*	`:set:sep?`
Description	This command will return the separate distance of the Z stage. This parameter can be in either microns or mils, depending on the current units that were set with the **`:set:unit`** command.	
Parameters	*devID*	2 for Elite, Summit, S300 and Alessi 6100 emulation.See SCPI Command Device IDs in Nucleus Software (all but microscope).

Example	Shows that the separate distance is 500 microns.
	`:set:unit?`
	`metric`
	`:set:separate? 2`
	`500`

:SET:STATION?

Short Command	`:set:stat?`
Description	This command will return 2, which is the number for Summit 12000 B-Series probe stations.
Parameters	None

Example	The following query line requests the identification number from a device in the system. The returned character (2) indicates that the device is a Summit 12000 B-Series probe station.
	`:set:stat?`
	`2`

:SET:UNIT

Short Command	*(metric)* *(English)* *(metric_float)*	`:set:unit`
Description	This command specifies the units of measurement that will be used for all input and output fields. For example, if you set the units of measurement to metric units (the default setting), x- and y-axis coordinates are in microns, and the wafer diameter is in mm. If you set the units of measurement to English units, the x- and y-axis coordinate values are in mils and the wafer diameter is in inches.	
Parameters	*(metric)*	Specifies that metric units of measurement will be used for distances and other values. By default, the units are metric. Parameters can be abbreviated to **metr**.
	(English)	Specifies that English units of measurement will be used for distances and other values. By default, the units are metric. Parameters can be abbreviated to **engl**.
	(metric_float)	Specifies that metric units of measurement will be used for distances and other values. In addition the METRIC_FLOAT unit type will return microns with 2-digit precision from commands like :move:absolute?
Example	Sets English as the units of measurement. `:set:unit English`	
Related Commands	*`:set:unit?`* *`:move:absolute:options`* *`:move:absolute?`* *`:move:absolute`* *`:set:preset`*	

:SET:UNIT?

Short Command	`:set:unit?`
Description	This query command returns the unit of measurement which is being used by distances and coordinates previously through the `:SET:UNIT` command. It either returns "METRIC", "ENGLISH" or METRIC_FLOAT as a response.
Parameters	None

Example	Shows how to determine the current unit of measurement.
	`:set:unit?`
	`METRIC`

Related Commands	*`:set:unit`*

:SET:USEDEVICEID

Short Command	*active*	*(on)* *(off)*	`:set:used`
Description	This command is obsolete. It is accepted and will not generate an error. All the commands controlled by this command will accept the deviceID parameter, but ignore it. For all new implementations, we recommend not using deviceID unless it is documented as required. This command used to control the use of the device ID parameter (*devID*) for many SCPI commands. When the value was set on, the device ID was used in the affected commands. When the value was off, the ID was not used in the affected commands.		
Parameters	*active*	Must be either on or off, but the value is ignored.	

:SET:USEDEVICEID?

Short Command	`:set:used?`
Description	This command is now obsolete. It will always return OFF. All commands previously affected by this command will always accept, but ignore Device ID.
Parameters	None

:SET:VACUUM

Short Command	*(on)* *(off)*	`:set:vac`
Description	This command turns the vacuum on or off in the center of the chuck. This command does not affect the corner (or auxiliary-device) vacuum. Nor does it affect the size of the DUT hold-down area. The corner vacuum controls and the size control for the center vacuum area are manual. Any vacuum ports not covered by the DUT should be turned off for maximum efficiency.	

Parameters	*(on)*	Specifies that vacuum to the chuck top is on or off (the default). When set to **on**, this command turns on the vacuum to the chuck top.
	(off)	When set to **off**, this command turns off the chuck top vacuum.
Example	Shows how to turn on vacuum to the chuck top: `:set:vac on`	
Related Commands	*`:move:load`* *`:set:vacuum?`* *`:set:vacuum:auxiliary`* *`:set:vacuum:auxiliary?`* *`:set:vacuum:ring`* *`:set:vacuum:ring?`*	

:SET:VACUUM?

Short Command	`:set:vac?`
Description	Returns ON or OFF depending on the state of the vacuum output line (not a sense line). Use the command :system:vacuum:sense to detect if the system actually has vacuum or not.
Parameters	None

Example	Returns that the vacuum control line is turned on.
	`:set:vacuum?`
	`ON`

Related Commands	*`:set:vacuum`* *`:set:vacuum:auxiliary`* *`:set:vacuum:auxiliary?`* *`:set:vacuum:ring`* *`:set:vacuum:ring?`* *`:system:vacuum:sense?`*

:SET:VACUUM:AUXILIARY

Short Command	`:set:vac::aux`	
Description	Used to turn vacuum on or off to the auxiliary chucks. The auxiliary chuck vacuum operates independently from the master chuck vacuum. *Only valid on the Elite probe station.*	
Parameters	*Aux. Chuck*	Possible values are 1 or 2.
	Enable	Possible values are ON or OFF.
Example	Shows how to turn vacuum on to the first auxiliary chuck. `:set:vacuum:auxiliary 1 ON`	
Related Commands	*`:set:vacuum`* *`:set:vacuum?`* *`:set:vacuum:auxiliary?`* *`:set:vacuum:ring`* *`:set:vacuum:ring?`*	

:SET:VACUUM:AUXILIARY?

Short Command	`:set:vac:aux?`	
Description	Returns "ON" or "OFF" depending on the vacuum control line for the requested auxiliary chuck. *Only valid on the Elite probe station.*	
Parameters	*Aux. Chuck*	Defines which Auxiliary chuck have vacuum turned on or off. Possible values are 1 or 2

Example	Returns "ON" if the vacuum control line is active.
	`:set:vacuum:auxiliary? 1`
	`ON`	

Related Commands	*`:set:vacuum`* *`:set:vacuum?`* *`:set:vacuum:auxiliary`* *`:set:vacuum:ring`* *`:set:vacuum:ring?`*	

:SET:VACUUM:RING

Short Command	`:set:vac:rin`	
Description	Used to enable vacuum to the rings of the chuck. The vacuum rings are dependant on master vacuum. If master vacuum is off and a vacuum ring is enabled, then it does not turn on vacuum to the ring. If master vacuum is on, then it does turn vacuum on to the specified ring. Actual vacuum is controlled with :set:vacuum command. *Only valid on the Elite probe station.*	
Parameters	*Ring Size*	Possible values are 75MM, 150MM, 200MM, 300MM.
	Enable	Possible values are ON or OFF.

Example	The first example enables the 75MM vacuum ring. If the master vacuum is ON, then vacuum to this ring is turned on. The second example disables the 300MM vacuum ring.
	`:set:vacuum:ring 75MM ON`
	`:set:vacuum:ring 300MM OFF`	

Related Commands	*`:set:vacuum`* *`:set:vacuum?`* *`:set:vacuum:auxiliary`* *`:set:vacuum:auxiliary?`* *`:set:vacuum:ring?`*	

:SET:VACUUM:RING?

Short Command	*Ring Size*	`:set:vac:rin?`
Description	Returns "ON" or "OFF" depending on the enable state of the specified vacuum ring. *Only valid on the Elite probe station.*	
Parameters	*Ring Size*	Possible values are 75MM, 150MM, 200MM, 300MM.

Example	Returns "ON" if the vacuum to that ring is enabled. `:set:vacuum:ring? 75MM` ON
Related Commands	*:set:vacuum* *:set:vacuum?* *:set:vacuum:auxiliary* *:set:vacuum:auxiliary?* *:set:vacuum:ring*

:SET:VELOCITY

Short Command	*devID*		`:set:vel`
	(xyz)		
	(ufast)	(vfast) (fast) (medium) (slow) (vslow) (uslow)	
Description	This command specifies the velocity setting for chuck and device movements along all axes on semiautomatic probe stations. This command affects the velocity used for the *:move:absolute, :move:relative,* and die stepping commands. This command does not apply to moves made in scan mode.		
Parameters	*devID*	2 for Elite, Summit, S300 and Alessi 6100 emulation. See SCPI Command Device IDs in Nucleus Software.	
	(xyz)	Sets the velocity for horizontal device movements (along the x- and y-axes) or for vertical chuck movements (along the z-axis).	
	(ufast)..	Specifies the velocity as **ufast, vfast, fast, medium, slow, vslow**, or **uslow**. The default value for this parameter is ultra fast (ufast).	

Example	Moves the device at slow velocity and raises the chuck at medium velocity.
	`:set:vel 2 xy slow`
	`:set:vel 2 z medium`
	`:mov:abs 2 7000 7000 750`		

Related Commands	*:move:absolute* *:move:absolute:chuck* *:move:relative*		

:SET:VISION

Short Command	*(on)*	`:set:vis`
	(off)	
Description	Use this command to enable/disable Auto XYZ Correction mode.	
Parameters	*on*	Enables Auto XYZ Correction mode.
	off	Disables Auto XYZ Correction mode.
Example	Enables Auto XYZ Correction mode. `:set:vis on`	

Related Commands	*:move:probeplan:absolute:die* *:move:probeplan:absolute:index* *:move:probeplan:absolute:subsite* *:move:probeplan:first:subsite* *:move:probeplan:next:die* *:move:probeplan:next:subsite* *:move:probeplan:prior:site* *:move:probeplan:relative:die* *:move:probeplan:absolute:location* *:move:probeplan:first:die* *:move:probeplan:last:die* *:move:probeplan:last:subsite* *:move:probeplan:next:site* *:move:probeplan:prior:die* *:move:probeplan:prior:subsite* *:move:probeplan:relative:index*

:SET:VISION:DISP

Short Command	*Vision*	*(on)* *(off)*	`:set:vis:disp`
	Advanced	*(on)* *(off)*	
Description	Use this command to show/hide the Auto XYZ Correction window and its advanced settings.		
Parameters	*Vision*	Show the Auto XYZ Correction window. Possible values are ON/OFF.	
	Advanced	Show the advanced window. Possible values are ON/OFF.	
Example	Displays the Auto XYZ Correction window with the advanced settings present. `:set:disp on on`		
Related Commands	*:move:probeplan:absolute:die* *:move:probeplan:absolute:index* *:move:probeplan:absolute:subsite* *:move:probeplan:first:subsite* *:move:probeplan:next:die* *:move:probeplan:next:subsite* *:move:probeplan:prior:site* *:move:probeplan:relative:die* *:move:probeplan:absolute:location* *:move:probeplan:first:die* *:move:probeplan:last:die* *:move:probeplan:last:subsite* *:move:probeplan:next:site* *:move:probeplan:prior:die* *:move:probeplan:prior:subsite* *:move:probeplan:relative:index*		

:SYSTEM:BEEP

Short Command	`:syst:beep`	
Description	This command causes the system to emit a sound.	
Parameters	None	
Example	The following causes the system to emit a "beep" sound: `:syst:beep`	

:SYSTEM:CONFIGURATION?

Short Command	*Limits* *Channels*	`:syst:conf?`
Description	Returns a string response containing the system configuration value of the parameter requested. Responses will vary depending on the parameter string.	
Parameters	*Limits*	Returns the limit of stage travel as a single number. Assumes a square stage with the center of travel being 0,0.
	Channels	Returns a string containing the currently available channels. The string contains a space between each channel. Ex: "2 3 7" would signify that the Chuck, Scope, and Positioner are available. The IDs returned are the same as the device IDs for the move absolute command: 2 – Chuck, 3 – Scope, 7 – Positioner One, 8 – Positioner Two, up until 12 for Positioner six.

Example	Queries the system for the channels that are available. The return string shows that the Chuck and Scope are the only two channels available.
	`:system:configuration? channels`
	`2 3`	

:SYSTEM:DELAY

Short Command	*msec*	`:syst:del`
Description	This command makes the probe station wait the specified length of time before executing the next instruction. During this delay period, the probe station does not accept commands from application programs or remote hosts. This is a one-time delay. If you want to specify a delay for the probe station to wait after the execution of every instruction, use the **`:set:delay`** command.	
Error Condition	For remote communications applications, avoid setting a delay that is longer than your timeout values. Consider using serial polls or SRQ handshaking instead.	
Parameters	*msec*	Specifies the length of time, in milliseconds, which the probe station will delay. The delay parameter can be a value in the range 0 to 65535 (65.5 seconds).

Example	The delay features are useful with the **`:move:scan`** commands. In this example, you want to scan at slow speeds for 2 sec, then stop device movement.
	`:mov:scan:xax 2 right slow`
	`:syst:del 2000`
	`:mov:stop:xax 2`

Related Commands	*`:set:delay`*

:SYSTEM:DISPLAY

Short Command	*string*	`:syst:disp`
Description	This command displays a one-line message on the probe station screen. The message remains until you press **Enter**, or click OK on the message box, to confirm the message. Use this command to insert prompts, warnings, and other messages into a test program.	
Error Condition	If you have a remote control application using timeout values and the time it takes you to respond to the message exceeds the time limit, you will time out the probe station. Consider using serial polls or SQR handshaking instead.	
Parameters	*string*	The ASCII string representing the message you want to display on the probe station screen. The character string can be up to 50 characters long and must be enclosed in double quotes, for example, "text here is correct." The double quotes that delineate the string are not displayed on the probe station screen as part of the string, and you cannot include quotes in the string. Characters following the second double quote are interpreted as a part of your application program or as a part of another command. If you want to include quotes in your message, use single quotes, for example 'quote here' is correct.

Example	Shows a message that tells the user to perform a particular test. In this example, the second **`:move:relative`** command executes as soon as you confirm the message Measure S11 by pressing a key on the probe station.
	`:mov:rel 2 14 0 400`
	`:syst:disp "Measure S11. Press enter key."`
	`:mov:rel 2 14 0 400`	

:SYSTEM:ERROR?

Short Command	`:syst:err?`

Description	This query returns the last error recorded by the probe station. The returned error is an ASCII string. If no error has occurred, this query returns an empty string. This feature is especially useful when troubleshooting your system. For example, if your application program is not working as expected, you might have created an error situation that your program does not recognize. In this case, you can use the return-error feature to see what error occurred. This can help you trace the error to a particular line in your test program. The error string can be cleared by using an optional parameter. If the error has been cleared and no new error occurs then "COMPLETE" is returned, indicating that no new errors have ocurred since calling this command.	
Optional Parameters	*Clear Error*	Possible values are ON or OFF. If the parameter is not present then the error is not cleared. A value of ON will clear the error after it is returned.

Example	Uses the `:system:error?` command to see the last error message that occurred.
	`:syst:err?`
	`@The number entered is greater than the upper limit of 6024`
	In the previous case, the last error recorded indicates that you entered (or sent) coordinates for a device movement. However, the coordinates were outside the probe station limit switches.
	By using `"ON"` it returns the current error and then clears it.
	`:system:error? ON`
	`@The number entered is greater than the upper limit of 6024`
	So the next time the command is called `"COMPLETE"` is returned.
	`:system:error?`
	`COMPLETE`	

:SYSTEM:IDENTIFICATION?

Short Command	`:syst:iden?`
Description	This query returns the probe station identification string. This is defined in the Station component of the Hardware Configurator.
Parameters	None

Example	Requests the identification string from the probe station. The string returned by your the probe station will differ depending on the probe-station serial number and version.
	`:syst:iden?`
	`Summit 12000`

:SYSTEM:LIMITS:TRAVEL?

Short Command	*DevID*	`:syst:lim:trav?`

Description	This command will return the limits of travel for the specified channel. Currently only the stage is implemented. Values returned for the X and Y axis are in microns, and the theta axis is in milli-degrees. On systems without motorized theta, 0 will be returned for theta limits. On Summit 12000-Series systems the Z axis limits are in motor counts. The return string will be formatted as follows: +X -X +Y -Y +Z -Z +Theta -Theta.	
Parameters	*DevID*	Specifies which channel of motion which to query.

Example	Queries the stage for limits on a Summit 12000-Series system. This system does not have a motorized theta so 0 0 is returned. X and Y each have a limit of +/-103000, the Z axis is 5500 at the top of travel to 0 at the bottom.
	`:syst:lim:trav? 2`
	`+103000 -103000 +103000 -103000 5500 0 0 0`	

:SYSTEM:OPERATION:MODE

Short Command	*Mode*	`:syst:oper:mode`
Description	Sets Nucleus software into REMOTE or LOCAL mode. When in REMOTE mode, all windows are hidden except the video window and point & shoot is disabled. Joystick still functions in REMOTE mode. LOCAL mode restores normal operations to the UI and point & shoot is re-enabled.	
Parameters	*Mode*	Possible values are REMOTE or LOCAL.
Example	Sets the system into LOCAL mode. `:system:operation:mode LOCAL`	
Related Commands	*`:system:operation:mode?`*	

:SYSTEM:OPERATION:MODE?

Short Command	`:syst:oper:mode?`
Description	Returns REMOTE or LOCAL depending on what mode is currently active.
Parameters	None

Example	Returns that the system is in LOCAL mode.
	`:system:operation:mode?`
	`LOCAL`

Related Commands	*`:system:operation:mode`*

:SYSTEM:PLATEN?

Short Command	`:syst:plat?`
Description	This command returns the status of the platen arm. Possible return values are "UP", "DOWN" or "PARTIAL". If "UP" is returned then the platen arm is fully raised. If "DOWN" is returned then the platen is fully lowered. If "PARTIAL" is returned then the platen arm is in-between the up and down positions.
Parameters	None

Example	Shows that the platen arm is fully raised.
	`:system:platen?`
	`UP`

:SYSTEM:VACUUM:SENSE?

Short Command	`:syst:vac:sens?`
Description	Returns ON if the system senses vacuum, OFF if it does not sense vacuum. This command can not detect the presence of a wafer, it can only return whether or not the system senses vacuum.
Parameters	None

Example	Returns that vacuum is sensed on the system.
	`:system:vacuum:sense?`
	`ON`

Related Commands	*`:set:vacuum?`* *`:set:vacuum`*

:SYSTEM:VERSION?

Short Command	`:syst:vers?`
Description	This query returns the probe station software release version.
Parameters	None

Example	Shows a sample response when you send the version command to the probe station. In this example, the probe station version number is 4.0.
	`:syst:vers?`
	`4.0`

Related Commands	*`:system:identification?`*

:SYSTEM:VERSION:EXTENDED?

Short Command	`:syst:vers:ext?`
Description	This query gets version information for more components returned in a comma-separated list. The returned version string may be expanded in the future, so the parsing should not rely on a specific length or number of comma-separated sub-strings returned. The fields returned are: • Nucleus version number in the format: major.minor.revision.build • eVue PCI driver version • eVue internal firmware version • eVue internal daughter-card firmware version • eVue PCI card (in the controller) internal firmware version • eVue Remote firmware version Note that if no eVue is present, all those versions will be zeroes and the remote "Unknown"

Parameters	None

Example	**With an eVue:**
	`:SYSTem:VERSion:EXTended?`
	`4.0.0.23,20070816,20090612,20070816,20090614,V1.05`
	Without an eVue:
	`:SYSTem:VERSion:EXTended`
	`4.0.0.81,00000000,00000000,00000000,00000000,Unknown`

:THERMALCHUCK:ACTIVATE

Short Command	`:ther:acti`
Description	This command activates the attached thermal controller unit using the current settings. **NOTE** The thermal chuck must first be deactivated before you can set a temperature. After the temperature is set it will not take effect until the system is activated. Please see the Example section and :thermalchuck:temperature:settemp, :thermalchuck:deactivate.
Parameters	None

Example	Sets the thermal chuck to become active.
	`:thermalchuck:deactivate`
	`:thermalchuck:temperature:settemp 60.0`
	`:thermalchuck:activate"`

:THERMALCHUCK:ACTIVATE?

Short Command	`:ther:acti?`
Description	This command determines whether the attached thermal controller is in the active or inactive state.
Parameters	None
Example	Checks the thermal chuck for active/inactive status. `:ther:temp:acti?`

:THERMALCHUCK:DEACTIVATE

Short Command	`:ther:deac`
Description	This command deactivates the thermal and sets it to idle. **NOTE** The thermal chuck must first be deactivated before you can set a temperature. After the temperature is set, it will not take effect until the system is activated. Please see the Example section and :thermalchuck:temperature:settemp, :thermalchuck:activate. Some thermal controllers can not be set to idle. Please check your thermal control documentation for details on your model.
Parameters	None

Example	Sets the thermal chuck to idle.
	`:thermalchuck:deactivate`
	`:thermalchuck:temperature:settemp 60.0`
	`:thermalchuck:activate"`

:THERMALCHUCK:IDENTIFICATION

Short Command	`:ther:iden?`

Description	Returns a string that identifies the type of thermal controller connected to the station.
	Possible responses are:
	`"Temptronics 3020"`
	`"Temptronics 3000"`
	`"Temptronics 300"`
	`"Temptronics 3200"`
	`"Temptronics 3500"`
	`"ATT"`
	`"Cascade"`
	`"ERS"`
	`"ESPEC"`
	`"Virtual"`
	`"None"`
	`"Unknown"`

Parameters	None
Example	`:ther:iden?` `ERS`
Related Commands	*`:thermalchuck:activate`* *`:thermalchuck:activate?`* *`:thermalchuck:deactivate`* *`:thermalchuck:identification`* *`:thermalchuck:temperature:current?`* *`:thermalchuck:temperature:settemp`* *`:thermalchuck:status?`* *`:thermalchuck:temperature:target?`* *`:thermalchuck:temperature:window`* *`:thermalchuck:temperature:window?`*

:THERMALCHUCK:TEMPERATURE:CURRENT?

Short Command	`:ther:temp:curr?`
Description	This command returns the current temperature of the thermal chuck in degrees Celsius.
Parameters	None

Example	Returns the thermal chuck temperature.
	`:ther:temp:curr?`
	`-34.5`

:THERMALCHUCK:TEMPERATURE:SETTEMP

Short Command	*value*	`:ther:temp:sett`
Description	This command can be used to set the thermal chuck to a temperature. The value is a floating point number in degrees Celsius. **NOTE** The thermal chuck must first be deactivated before you can set a temperature. After the temperature is set, it will not take effect until the system is activated. Please see the example section and :thermalchuck:activate, :thermalchuck:deactivate.	
Parameters	*value*	The desired temperature for the thermal chuck.

Example	`:thermalchuck:deactivate`
	`:thermalchuck:temperature:settemp 60.0`
	`:thermalchuck:activate`	

:THERMALCHUCK:STATUS?

Short Command	`:ther:stat?`
Description	This command returns the status of the thermal chuck. Possible return values are: `AT TEMP`, `RUNNING`, `IDLE`, or `ERROR`, depending on the specific thermal controller used.
Parameters	None

Example	Returns the status of the thermal chuck.
	`:ther:sta?`
	`AT TEMP`

:THERMALCHUCK:TEMPERATURE:TARGET?

Short Command	`:ther:temp:targ?`
Description	This command returns the current target temperature last set with the `:ther:temp:sett` command. If no "settemp" command has been issued, this value shall be 0.0 degrees Celsius.
Parameters	None

Example	Returns the current target temperature of the thermal chuck.
	`:ther:temp:targ?`
	`-25.4`

:THERMALCHUCK:TEMPERATURE:WINDOW

Short Command	*window*	`:ther:temp:wind`
Description	This command sets the window range of the thermal chuck when the unit is in the inactive state. The `:thermalchuck:temperature:window` is not supported on Titan thermal chucks.	

Parameters	*window*	Entire range of window (+0.1 <= WINDOW <= 9.9)
Example	The following command sets the window range to 5.0 degrees Celsius. `:ther:temp:wind 5.0`	

:THERMALCHUCK:TEMPERATURE:WINDOW?

Short Command	`:ther:temp:wind?`	
Description	This command returns the current window range. If the **`:ther:temp:wind`** command has not been issued, this value shall be 0.2 degrees Celsius. This command is not supported on Titan thermal controllers.	
Parameters	*window*	Entire range of window (+0.1 <= WINDOW <= 9.9)

Example	Returns the current window range of the thermal chuck.
	`:ther:temp:wind?`
	`2.5`	

:VIDEO:RECORD:CAPTURE:RATE

Short Command	`:vid:rec:capt:rate`	
Description	This command sets up the capture parameters used for subsequent recording sessions.	
Parameters	*Capture rate*	A numeric value for cature rate, dependent on the following argument.
	SEC | MIN | HOUR	An indication of the type of unit to use with the capture rate.
	ON | OFF	When enabled, places a timestamp text message at the lower right corner of captured images.
	FULL | HALF | QUAR	Captured image size of full, half or quarter size of displayed image.
Example	Sets up for 12 frames per second, with timestamp and full images. `:vid:rec:capt:rate 12 sec on full`	
Related Commands	`:video:record:countdown:rate`	

:VIDEO:RECORD:CLOSE

Short Command	`:vid:rec:close`
Description	This command closes the current video session.
Parameters	None
Example	`:vid:rec:clos`

Related Commands	`:video:record:new`
	`:video:record:open`
	`:video:record:save`

:VIDEO:RECORD:COUNTDOWN:RATE

Short Command	`:vid:rec:coun:rate`	
Description	This command sets up the condition for automatic stopping of recording.	
Parameters	*Countdown time*	Length of time to wait before stopping the recording. Units are based on the following argument.
	SEC | MIN | HOUR | FRAME	Unit of measure for countdown timer. Use of time does not depend on number of frames captured. Use of frames does not depend on the time it takes to capture frames.
	ON | OFF	If enabled, the countdown timer will stop recording when the conditions are met.
Example	Stop recording when ten frames have been captured. `vid:rec:coun:rate 10 fram on`	
Related Commands	`:video:record:capture:rate` `:video:record:start`	

:VIDEO:RECORD:NEW

Short Command	`:vid:rec:new`	
Description	This command creates a new video session.	
Parameters	*<filename>*	Name of new session to create, in quotes. Can be relative to UserData directory, or fully qualified name.

Example	`:vid:rec:new "TestWafer1"`
	or
	`:vid:rec:new "C:\RootDir\TestWafer1"`	

Related Commands	`:video:record:close`
	`:video:record:open`
	`:video:record:save`	

:VIDEO:RECORD:OPEN

Short Command	`:vid:rec:open`	
Description	This command opens an existing video session. The file name can be the name of the session as found in the system UserData directory, or a full path qualified name.	
Parameters	*<filename>*	Name of existing video session, in quotes.

Example	`:vid:rec:open "VideoCapture"`
	or
	`:vid:rec:open`	

Related Commands	`:video:record:close`
	`:video:record:new`
	`:video:record:save`	

:VIDEO:RECORD:SAVE

Short Command	`:vid:rec:save`	
Description	This command saves the current session by its active name, or saves the session under a new name. The most recent recorded video will not be available for viewing until the session has been closed, saved, or saved under a new name. If no name is provided, it will be saved under the current name. If a name is provided, it will be saved under that name.	
Optional Parameters	*<filename>*	Name of session to save, in quotes.

Example	Save current session
	`:vid:rec:save`
	or
	Save current session to a new name
	`:vid:rec:save "CopyTest`	

Related Commands	`:video:record:close`
	`:video:record:new`
	`:video:record:open`	

:VIDEO:RECORD:START

Short Command	`:vid:rec:star`
Description	This command begins video recording.
Parameters	None
Example	`:vid:rec:star`
Related Commands	`:video:record:stop`

:VIDEO:RECORD:STOP

Short Command	`:vid:rec:stop`
Description	This command halts a recording in progress (started with the :vid:rec:star command).
Parameters	None
Example	`vid:rec:stop`
Related Commands	`:video:record:start`

:VISION:NEEDLE:SEARCH:TARGET

Short Command	`:vis:need:sear:targ`

Description	This command searches for a needle tip using a previously trained target. The "name" parameter defines the id used to search. It returns four values separated by white space character: X shift, Y shift, Z shift and match score. The shift values are in microns and the match score can be minimum match score to 1000. 1000 is a perfect match. If "OFF" is used for Search Z , 0.0 is returned for Z. If an error occurs, '@' is returned with the error string. If the target can't be found, "0.0 0.0 0.0 0" is returned.	
Parameters	*Name*	String parameter that defines the name of the trained needle.
	Search Z	Possible values are "ON" or "OFF". • If "ON" then the search includes finding the Z height difference. • If "OFF" then only horizontal differences are returned.
Example	`:vision:needle:search:target "Celadon 1" ON` `32.3 -10.2 5.3 850`	
Related Commands	*:vision:needle:train:target*	

:VISION:NEEDLE:TRAIN:TARGET

Short Command	`:vis:need:tra:targ`	
Description	This command trains a needle tip using the current vision train box location. The "name" parameter defines the ID used to access the needle training from search or file commands. It returns "COMPLETE" if the needle was trained successful ly. Otherwise, "@description of error" is returned.	
Parameters	*Name*	String parameter that defines the name of the trained needle.
Example	`:vision:train:needle "Celadon 1"` `COMPLETE`	
Related Commands	*:vision:needle:search:target*	

:VISION:READ:TARGET:FILE

Short Command	*Vision ID*	`:vis:read:targ:file`
Description	Reads a previously trained target from disk. Once the target is read in from disk it is considered trained and ready for use. The bounding region of the vision target is also restored when it is read from disk.	
Parameters	*Vision ID*	Used to identify which target to read from disk. Possible values range from 1 to 50.

Example	Reads the vision target id 12 from disk.
	`:vision:read:target:file 12`
	`COMPLETE`	

Related Commands	*:vision:train:target* *:vision:search:target* *:vision:write:target:file*	

:VISION:SEARCH:TARGET

Short Command	`:vis:sear:targ`
Description	Performs a pattern req. search using Nucleus Vision. Nucleus Vision must be purchased and installed on the system before this command will work. In order to use this command the target must first be trained using the Auto XY Correction window. After a search is preformed the return values are the X, Y shift of the target and a match score. If the match score is 0, the target was not found in the field of view. The X and Y values are oriented such that a call to ":move:rel X Y" will move the target directly into the target train box of the live video window. Please refer to Auto XY Correction in the *Nucleus User Guide* for more information.
Parameters	None

Example	The following example performs a pattern req. search and returns that the target has moved 30 microns in X and -20 microns in Y with a match score of 830.
	`:vis:sear:targ`
	`30 -20 830`

Related Commands	*`:move:relative`*

:VISION:SET:MATCH:SCORE

Short Command	*Match Score*	`:vis:set:matc:scor`
Description	This command can be used to set the threshold score used during vision search operations. If the current search target yields a score that is below this threshold, the vision system returns that no matches could be found. In some cases, it can be effective to lower the match score to yield passing targets. The possible range for match score is 0 to 1000. When the match score is set at 1000, only targets that are a perfect match will succeed. The default setting is 700.	
Parameters	*Match Score*	Defines the threshold score that is used as the criteria for deciding whether or not a target is a good match.

Example	Sets the match score threshold to 700.
	`:vision:set:match:score 700`
	`COMPLETE`	

Related Commands	*`:vision:train:target`* *`:vision:search:target`* *`:vision:write:target:file`* *`:vision:read:target:file`*	

:VISION:TRAIN:TARGET

Short Command	*Vision ID*	`:vis:tra:targ`

Description	This remote command trains a target using the region defined by the vision box in the video window. Once a target has been trained, you can search for a match using the `:vision:search:target` command. The location of the training region is recorded for each target, which ensures that the offset of multiple trained targets can be managed.	
Parameters	*Vision ID*	Used to identify which target to train. The current location of the vision train box in the video window defines the location where the image is trained. Possible values range from 1 to 50. If the parameter is not provided, then the Auto XY Correction target is trained.

Example	Trains the vision target ID to 12.
	`:vis:trai:target 12`
	`COMPLETE`	

Related Commands	*`:vision:search:target`* *`:vision:write:target:file`* *`:vision:read:target:file`*	

:VISION:WRITE:TARGET:FILE

Short Command	*Vision ID*	`:vis:writ:targ:file`
Description	Writes a trained vision target to disk. This can be retrieved later by using the :vision:read:target:file command. Before a target can be stored to disk it must first be trained.	
Parameters	*Vision ID*	Used to identify which target to write to disk. Possible values range from 1 to 50.

Example	Writes the vision target ID 12 to disk.
	`:vision:write:target:file 12`
	`COMPLETE`	

Related Commands	*`:vision:train:target`* *`:vision:search:target`* *`:vision:read:target:file`*	

EG Commands

NOTE

The commands where "no effect" is listed in the Comment column are harmless, unless you depend on the actions that are stubbed out.

Commands & Parameters	Description	Comment
?An	Return depends on n: n=0 temperature (**AT**123.4) n=1setpoint (**AS**111) n=2 delay (**AD**0) n=3 model ID (**AM**2)	In Nucleus software n=1 is the only valid setting. n=3 ID is always "AM2" for a non EG hot chuck.

Commands & Parameters	Description	Comment
?E	Returns last errors number n (returned as **E**n). If n=0 there was no recorded error.	
?H	Return the current location of the stage in a string formatted as HXnnnnnnYmmmmmm where nnnnnn and mmmmmm are the values for X and Y respectively. Returns values in 0.1 mil steps. All values are rounded to nearest whole number. X-value returned gets more positive as the stage moves to the left. Y-value returned gets more positive as the stage moves to the rear (away from the operator).	The home position (0,0) is in the center in Nucleus, but lower right corner on an EG prober. Location 0,0 is the reference die after the wafer has been aligned. The values returned account for the street angle from alignment (small angle). The values are not affected by the load angle, wafer map angle or wafer map axis.
?H*	Same as ?H but returns values in microns.	Nucleus extension to EG command set.
?I	Format returned: **IX**x1**Y**y1**X**x2**Y**y2**D**d. -Reference die position is returned in (x1, y1). These values are in 0.1 mils. -Chuck center coordinates are (x2, y2). These values are in 0.1 mils -Wafer diameter in mm is d **Example: IX**10000**Y**10000**X**0**Y**0**D**200. In this example the reference die is at 10000, 10000 the center of the chuck is 0,0 and the wafer diameter is 200mm.	In Nucleus, the center of the chuck will always be 0,0, so the x2, y2 will always be 0,0. Chuck center is the Home position.
MD	Move the stage by multiples of the die size relative to the current position. Format of the command is MDXnnYmm where nn and mm are the X and Y values respectively.	Moves are adjusted for street angle from the alignment (small angle). The wafer load angle will not affect the direction of the move.
MM,*MM,MR	Move the stage relative to the current position. The format of the commands are: MMXnnnYmmmm *MMXnnnYmmm MRXnnnYmmm Where nnn and mmm are the X and Y values respectively.	The moves are adjusted for the street angle from the alignment (small angle). The moves are not affected by the load angle, wafer map angle, or the wafer map axis.
Command	*Metric Mode (see SM1U1)*	*English Mode (see SM1U0)*
MM	2.5 Microns per count	2.54 Microns per count
*MM	1.0 Microns per count	2.54 Microns per count
MR	1.0 Microns per count	2.54 Microns per count
The direction of the move is as follows:		
Command	*X+ Stage Direction*	*Y+ Stage Direction*
MM	Left	Rear
*MM	Left	Rear
MR	Right	Front

Commands & Parameters	Description	Comment
MOXnYm	Instructs the prober to move to a specific X, Y location on the wafer	n-is the X column in the current wafer map. m-is the Y row in current wafer map.
?O	Return current option settings. If a component is started and not disabled in any way, n is returned as 1, otherwise 0. Format returned: **Om**n**A**n**P**n**B**n**S**n**W**n**H**n**T**n**D**n**K**n Mn Material handling An Auto alignment Pn Profiler Bn OCR Sn SECS Wn Wafer mapping Hn Hot chuck Tn Temperature compensation Dn Ink dot inspection Kn Probe mark inspection	Nucleus will always return 0 for Material handling, Auto alignment, Profiler, OCR, SECS, Ink dot inspection, Probe mark inspection, and Temperature compensation. Hn/Hot chuck and Wn/Wafer Mapping are the only items available for Nucleus.
?P	Return the current location of the stage in terms of die rows and columns. The return string is formatted XnnYmm where nn and mm are the row and column position respectively. The location is based on the settings for the reference die, which include wafer axis and reference coordinate offset. Values returned are not affected by the load angle	Relates only to Nucleus wafer map indexes. Does not change with quadrant settings.
?S	Returns the prober status chuck position (up, down), Wafer On Chuck (off, on), Edge Sensor (no contact, contact). Format returned: **SZ**(u,d)**W**(0,1)**C**(0,1)	Nucleus does not have a wafer on the chuck sense, so this will always return that there is a wafer on the chuck or W1. Nucleus does not support Z Profiling so C(0,1) only applies to Edge Sense.

Commands & Parameters	Description	Comment
?Wn	Request. n-0 full wafer ID n=1-device ID, n=2- lot number, n=3 wafer number. No n at all equals ?W0. Example scenario: Operator entered ABCD for device ID and 123 for lot ID and current wafer is number 23 in the cassette. OCR (when present in the future) read "38318CB-19 HJKL". ?W0 or ?W Return full Wafer ID (device, lot and slot). OCR present: The exact read string "38318CB-19 HJKL" will be returned. OCR not present: Concatenated operator entered strings will be returned for device- lot-slot#. A dash character will separate the items. ABCD-123-23 ?W1 Return operator entered device ID ABCD will be returned even if OCR is present. 23 will be returned even if OCR is present.	Nucleus will always return " " for any n.
	For all the above: ?W1, ?W2, ?W3 will always return operator entered strings even if OCR is present. If the operator entered nothing, an empty string will be returned.	
?Z	Returns current Z height **Z**n n=position in units of 0.1 mil.	S300 will return physical distances in .1 mil and Summit 12000-Series will return motor counts.
*?Z	Returns current Z height **Z**n. n=position in units of one micron.	Summit 12000-Series will return motor counts.
AA	Perform wafer auto align.	Nucleus does not support auto alignment. The command will have no effect.
AP	Abort probing.	Has no effect.
CE	Clear buzzer, alarm light and error condition.	Has no effect.
CP	Clean probes.	Has no effect.
HO	Move to the prober home position.	Nucleus probe stations home is in the center of travel.

Commands & Parameters	Description	Comment
HW	Handle a wafer	In Nucleus, this command will bring up a message box instructing the user to manual load/unload and align a wafer. When the user is finished they must close this dialog to continue.
ID	Returns an ID string.	Prober Model, Software Version, Part Number, and serial number are returned. Ex: PPP.02.5P.N.sssss Where sssss is the serial number, and PPP is the type of prober.
LAn	Set all lights OFF n=0 / ON n=1	Will turn the lamp on or off.
LCAmGnYnRn	Light Control (alarm light and buzzer) m: 0=OFF 1= ON n: 0=OFF 1= ON 2=ON (blink on EG) A = alarm => buzzer ON/OFF G = green light Y = yellow light R = red light	Has no effect.
LO	Load next wafer.	Has no effect.
MDXnYn	Move to die index.	Directions hardwired. Positive x moves the probes x die steps to the right. Positive y moves y die steps up. Wafer load angle has no effect on the die indexes
MF	Move to preset die.	On Nucleus this is the reference die.
MMXnYn	Move relative in machine steps. Metric mode => n in 2.5 micron steps. English mode => n is in 0.1 mil steps.	Positive x moves chuck to the left, and positive y to the rear of the prober.
*MMXnYn	Move relative in machine steps. Metric mode => n in 1 micron steps. English mode => n is in 0.1 mil steps.	Positive x moves chuck to the left, and positive y to the rear of the prober.
MRXn.nYn.n	Move Relative with floating-point parameters. Metric mode => n.n in micron steps. English mode => n.n is in 0.1 mil steps.	Positive x moves chuck to the right, and positive y to the front of the prober (opposite to MM).
PA	Pause probing.	Has no effect.
SM1Un	Set metric n=1 or English n=0 units for input.	
SM2Qn	Set probing quadrant.	Has no effect.
SM3Fn	Set flat orientation n = 0…359 degrees. Only 0,90,180,270 applies.	Has no effect.
SM4Pn	Select probe mode.	Has no effect.
SM11Qn	Select coordinate quadrant	Has no effect.

Commands & Parameters	Description	Comment
SM15Mnnnnn nnnnnnnn	Enable response messages. Each n is 1 for enabling or 0 for disabling i.e. no MC/MF is returned after command is executed. Only three "n" settings has any meaning in Nucleus. 1 -- MF/MC on x/y motion 2 -- MF/MC on Z motion 4 -- MF/MC on rest of commands	Only for MC/MF response settings has an effect. Nucleus does not support any other unsolicited messages.
SM33Hn	Set hot chuck temperature, where "n" is degrees Celsius.	
SP1	Set actual die size.	
SP2	Set reference die location in user coordinates. Format of the command is SP2XnYm, where n and m are the new column and row coordinates.	Not needed for MM, *MM and MR moves.
SP4D	Set wafer diameter. Format of command is SP4Dn, where n is the wafer diameter in mm.	
SP5Zn	Set Z over travel. n is in 0.1 mils. Set the distance Z stage will continue to travel beyond contact with the wafer as determined by the edge sensor.	Nucleus does not support a Z profiler.
*SP5Zn	As above, but n is in units of 1 micron.	Nucleus does not support a Z profiler.
SP6Zn	Set Z clearance. n is in 0.1 mils. Sets the downward travel of the Z chuck after contact is lost.	Has no effect in Nucleus. This command relates to a Z Profiler, which is not supported in Nucleus
*SP6Zn	As above, but n is in units of 1 micron.	Has no effect in Nucleus. This command relates to a Z Profiler, which is not supported in Nucleus
UL	Unload current wafer back to its source cassette slot.	Has no effect
WM	Send x/y with test start on/off.	Has no effect. Can only be OFF.
ZD	Move chuck down to separate position.	Probe contact must be set for wafer.
ZRn	Move Z relative to the current position. n is in units of 0.1 mils.	Warning: Displayed coordinates on a Summit 12000-Series are in motor counts only.
*ZRn	As above but n is in units of 1 micron.	
ZMn	Move Z to any height within the valid up/down range. n is in units of 0.1 mils	Warning: Displayed coordinates on Summit 12000-Series are in motor counts only.
*ZMn	As above but n is in units of 1 micron.	

Commands & Parameters	Description	Comment
ZU	Move chuck up to contact position.	Probe contact must be set for wafer.

Index

Cascade Microtech, Inc.
Corporate Headquarters
toll free: +1-800-550-3279
phone: +1-503-601-1000
email: cmi_sales@cmicro.com

Germany
phone: +49-811-60005-0
email: cmg_sales@cmicro.com

Japan
phone: +81-3-5615-5150
email: cmj_sales@cmicro.com

China
phone: +86-21-3330-3188
email: cmc_sales@cmicro.com

Singapore
phone: +65-6873-7482
email: cms_sales@cmicro.com

Taiwan
phone: +886-3-5722810
email: cmt_sales@cmicro.com

www.cascademicrotech.com