

In-Sight[®] Wafer ID Native Mode Communications Reference Manual



2026 September 21
Revision: 26.2.0.62

Legal Notices

The software described in this documentation is provided "as is" and "as available," without warranty of any kind, express or implied, including but not limited to the warranties of merchantability, fitness for a particular purpose, and non-infringement. In no event shall Cognex Corporation and its global affiliates (collectively, "Cognex") be liable for any claims, damages, or other liability arising from, out of, or in connection with the software or your use of the software.

Your use of this software is governed by the full terms and conditions of the Cognex Software License Agreement. For complete licensing details and legal terms, please visit <https://www.cognex.com/terms-and-conditions>.

The information in this document is subject to change without notice and should not be construed as a commitment by Cognex. Cognex is not responsible for any errors that may be present in either this document or the associated software.

Companies, names, and data used in examples herein are fictitious unless otherwise noted. No part of this document may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, nor transferred to any other media or language without the written permission of Cognex Corporation.

Portions of the hardware and software provided by Cognex may be covered by one or more U.S. and foreign patents, as well as pending U.S. and foreign patents listed on the Cognex web site at: cognex.com/patents.

In the event of any inconsistency between the English language version of this documentation and any other language version, the English language version shall take precedence.

Copyright © 2026. Cognex Corporation. All Rights Reserved.

The following are registered U.S. trademarks of Cognex Corporation:

2DMAX, 3D LIGHTBURST TECHNOLOGY, ADVANTAGE, ALIGNPLUS, ASSEMBLYPLUS, COGNEX, COGNEX VISION FOR INDUSTRY, CVL, DATAMAN, DVT, EASYBUILDER, EDGE INTELLIGENCE, HOTBARS, IDMAX, IN-SIGHT, LIGHTBURST, MX, OMNIVIEW, PATFIND, PATFLEX, PATINSPECT, PATMAX, PATMAX REDLINE, PATQUICK, POWERGRID, QUICKSORT, ULTRALIGHT, VISION SOLUTIONS, VISIONPRO, VISIONVIEW

The following are trademarks of the Cognex Corporation:

1DMAX, 3D SYMBOLIC LIGHT, 3D-LOCATE, 3DMAX, ONEVISION, PINPOINT, RANGEMAX, SLX, VIDI, OCRMAX, INSPECTEDGE, VISION SOLUTIONS

Portions copyright © Microsoft Corporation. All rights reserved.

Portions copyright © MadCap Software, Inc. All rights reserved.

Other product and company trademarks identified herein are the trademarks of their respective owners.

Precautions

To reduce the risk of injury or equipment damage, observe the following precautions when you install the Cognex product:

- Connectivity is possible through the following options:
 - 24VDC (+/- 10%) output connection using a UL or NTRL listed power supply
- Any other voltage creates a risk of fire or shock and can damage the components. Applicable national and local wiring standards and rules must be followed.
- This product is intended for industrial use in automated manufacturing or similar applications.
 - The safety of any system incorporating this product is the responsibility of the assembler of the system.
 - Do not install Cognex products where they are exposed to environmental hazards such as excessive heat, dust, moisture, humidity, impact, vibration, corrosive substances, flammable substances, or static electricity.
 - Route cables and wires away from high-current wiring or high-voltage power sources to reduce the risk of damage or malfunction from the following causes: over-voltage, line noise, electrostatic discharge (ESD), power surges, or other irregularities in the power supply.
 - Do not expose the image sensor to laser light. Image sensors can be damaged by direct, or reflected, laser light. If your application requires laser light that might strike the image sensor, use a lens filter at the corresponding laser wavelength. For suggestions, contact your local integrator or application engineer.
 - This product does not contain user-serviceable parts. Do not make electrical or mechanical modifications to product components. Unauthorized modifications can void your warranty.
 - Changes or modifications not expressly approved by the party responsible for regulatory compliance could void the user's authority to operate the equipment.
 - Include service loops with cable connections.
 - Ensure that the cable bend radius begins at least six inches from the connector. Cable shielding can be degraded or cables can be damaged or wear out faster if a service loop or bend radius is tighter than 10X the cable diameter.
 - This device should be used in accordance with the instructions in this manual.
 - All specifications are for reference purposes only and can change without notice.

Symbols

The following symbols indicate safety precautions and supplemental information:

 **WARNING:** This symbol indicates a hazard that could cause death, serious personal injury or electrical shock.

 **CAUTION:** This symbol indicates a hazard that could result in property damage.

 **Note:** This symbol indicates additional information about a subject.

 **Tip:** This symbol indicates suggestions and shortcuts that might not otherwise be apparent.

Table of Contents

Legal Notices	2
Precautions	3
Symbols	4
Table of Contents	5
Native Mode Communications	9
About the Native Mode Protocol	9
Basic Native Mode Syntax	9
Extended Native Mode Syntax	10
Communicating with In-Sight Sensors via Serial Port Using Native Mode commands	10
Communicating with In-Sight Sensors via Telnet Using Native Mode Commands	11
Basic Native Mode Commands	12
File & Job Commands	12
Load File	12
Store File	14
Read File	16
Write File	18
Delete File	20
Get File	22
Set Job	24
Store Job	26
Read Job	28
Write Job	30
Delete Job	33
Get Job	35
Image Commands	37
Read BMP	37
Read Image	39
Write BMP	41
Write Image	43
Settings & Cell Value Commands	44
Get Value	44
Set Integer	47
Set Float	51
Set String	53
Get Info	55
Execution & Online Commands	56
Set Online	56
Get Online	58
Set Event	59
Set Event and Wait	61
Reset System	63
Abort Execution	64
Extended Native Mode Commands	65

Get	65
Get Filelist	65
Evaluate	67
Evaluate Inputs	67
Evaluate Outputs	67
WaferID Inputs & Outputs	67
AcquireConfig	76
AcquireConfigAndRunSheet	77
CopyConfig	78
GetConfig	79
GetConfigAppendSEMICHkToT7	83
GetConfigEx	84
GetConfigRemoveSEMIChecksum	88
GetConfigResult	89
GetConfigTune	91
GetFilterList	94
GetWaferAll	96
GetWaferAllEx	99
GetWaferCompatibilityMode	102
GetWaferFailCharsOutRegion	103
GetWaferRegionTolerance	104
GetWaferSummary	105
ReadConfig	107
ResetStats	109
SetConfig	110
SetConfigAccept	113
SetConfigAppendSEMICHkToT7	114
SetConfigBC3Digits Inputs	115
SetConfigBC3Digits Outputs	115
SetConfigBC3Digits Example	115
SetConfigBCHyphen Inputs	116
SetConfigBCHyphen Outputs	116
SetConfigBCHyphen Example	116
SetConfigBCType	117
SetConfigCharSize	118
SetConfigChecksum	119
SetConfigCharSpacing	120
SetConfigColor	121
SetConfigEnable	122
SetConfigEx	124
SetConfigFieldDef	127
SetConfigFieldString	129
SetConfigLEDOVERRIDE	130
SetConfigLightMode	132
SetConfigLightPower	133
SetConfigMark	134
SetConfigMatchString	135
SetConfigName	137

SetConfigOptimization	139
SetConfigOrientation	140
SetConfigRegion	141
SetConfigRemoveSEMChecksum	143
SetConfigRetry	144
SetConfigStrokeThickness	145
SetConfigThresh	146
SetConfigTune	147
SetWaferAcqEnable	154
SetWaferCompatibilityMode	155
SetWaferFailCharsOutRegion	156
SetWaferOrder	157
SetWaferRegionTolerance	158
SetWaferScoreAs100	159
SetWaferTimeout	160
TuneConfig	161
TuneConfigEx	164
Protocol Commands	167
About Protocol Commands	167
In-Sight Protocol	167
TEL Protocol	168
TSK Protocol	168

Native Mode Communications

[About the Native Mode Protocol](#)

[Basic Native Mode Syntax](#)

[Extended Native Mode Syntax](#)

[Communicating with In-Sight Sensors via Serial Port Using Native Mode Commands](#)

[Communicating with In-Sight Sensors via Telnet Using Native Mode Commands](#)

Tips:

- The In-Sight wafer reader has significant built-in intelligence. For example, each Config can be tried in order, based on the score (from highest to lowest) of the most recent read, allowing the reader to automatically adjust light modes based on the read results. Therefore, it is recommended that the Wafer ID user interface be used to configure the wafer reader (e.g., optimizing Configs, enabling the Retry option, etc.) and the programmatic interface only be used to load/save jobs and to issue the In-Sight Protocol [READ](#) command to trigger the wafer reader. The READ command trigger provides better control over the returned data. [Native Mode commands](#) should only be used to trigger a reader when absolutely necessary; using Native Mode commands to manage the wafer reader can easily override the built-in intelligence of the reader.
- If using Native Mode commands to manage the wafer reader, the **EV ReadConfig -1** Native Mode command provides the same functionality as the In-Sight Protocol [READ](#) command.

About the Native Mode Protocol

The In-Sight Native Mode protocol is an ASCII protocol that allows an In-Sight sensor to be controlled from any of the following:

- Custom application programs running on a PC
- Remote hosts that support standard [serial communications](#)
- [Telnet](#) over an Ethernet network

The Native Mode protocol is divided into two sets of commands: [Basic](#) and [Extended](#). Basic Native Mode commands are two characters long, plus parameters (if any) and a terminator character; extended Native Mode commands include additional functions or commands. The commands are not case sensitive. The terminator is CR + LF (ASCII characters 13 + 10) when sending Native Mode commands using a telnet connection. When using Native Mode commands over a serial port, an alternate terminator character may be specified in the Native Mode Details; the default terminator is CR (ASCII character 13).

When a Native Mode command is remotely issued to an In-Sight sensor, the In-Sight sensor processes the command and then returns a response, consisting of an ASCII string followed by the terminator character. Commands that *set* values return **1** for "success", **0** for "unrecognized command", or a negative number for "failure". Commands that *get* values return various values, depending on the command.

Note: A subset of Native Mode commands can be sent from any PLC (that allows a generic CIP¹ message) over an EtherNet/IP connection using Explicit Messaging.

Basic Native Mode Syntax

The general syntax of a Basic Native Mode command is as follows:

¹ Control and Information Protocol

[command][parameter][terminator]

where *[command]* consists of two characters. Some Native Mode commands take an optional *[parameter]*.

The Basic Native Mode command is completed with a *[terminator]* character. The default terminator is CR + LF (ASCII characters 13 + 10). When using Native Mode commands over a serial port, an alternate terminator character may be specified in the Native Mode Details.

Basic Native Mode commands are not case-sensitive. No spaces exist between parameters.

Notes:

- Native Mode commands that are executed on a Job (such as [Write Job](#) and [Delete Job](#)) are included primarily for backwards compatibility.
- The checksum that is used by Native Mode file transfer commands is a standard 16-bit Cyclical Redundancy Check.

Extended Native Mode Syntax

The general syntax of an Extended Native Mode command is as follows:

[command1] [command2] [input parameters][terminator]

where *[command1]* is any one of the Extended Native Mode commands ([Get](#), [EV](#)). *[command2]* represents the secondary command, indicating which data to send, receive, or execute, based upon *[command1]*. *[input parameters]* supply optional supplementary information to the command.

The Extended Native Mode command is completed with a *[terminator]* character. The default terminator is CR + LF (ASCII characters 13 + 10). When using Native Mode commands over a serial port, an alternate terminator character may be specified in the Native Mode Details.

Extended Native Mode commands are not case-sensitive. Note that spaces exist between certain input parameters.

Communicating with In-Sight Sensors via Serial Port Using Native Mode commands

1. On the Sensor menu in Spreadsheet, click **Serial Port Settings**.
2. In the **Serial Port Settings** dialog, choose **Native** from the **Mode** drop-down list.
3. Configure the remote serial device to use the same port settings as the In-Sight sensor.

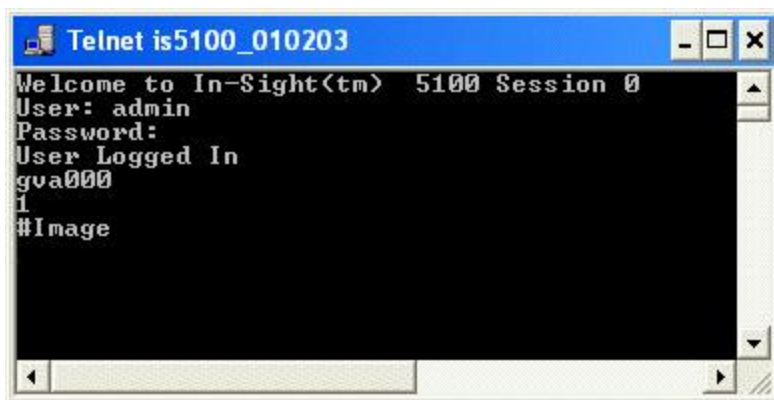
Note: When using Native Mode commands over a serial port, a different terminator character may be specified in the Native Mode Details, which is accessed from the Serial Port dialog.

Communicating with In-Sight Sensors via Telnet Using Native Mode Commands

Telnet is a built-in Microsoft Windows client that can be used to remotely communicate with the sensor using In-Sight Native Mode commands.

1. Open the telnet client application on the remote host.
2. In the telnet application, connect to the In-Sight sensor by either its Host Name or IP address. If the connection is successful, a "Welcome to In-Sight(tm)" message appears, along with the hardware sensor type (for example: 1751) and the session number.
3. Enter a valid user name and password for the In-Sight sensor. The user name and password entered must exist in the User List for that sensor. If the log on is successful, the message "User Logged In" will appear.

You can test the telnet connection by typing the Native Mode [Get Value](#) command, "gva000", at the telnet prompt. If successful, the command returns a 1, followed by the value in cell A0 in the spreadsheet of the In-Sight sensor, **#WaferID**:



Custom applications can be written (to run on a remote host) to automate the telnet logon process, as well as to send and receive Native Mode data.

Notes:

- By default, when connected via telnet, there is no default timeout. You can configure the timeout value in the WebUI.
- Under typical use, password echo is disabled in the telnet client application for security purposes.
- When using telnet to log on to an In-Sight sensor from a Unix-based platform, the telnet command line must be in the following format:
> telnet <host> <port>
 where <host> is either the In-Sight sensor's Host Name or IP Address, and <port> is the remote telnet port (usually 23). Supplying the port number disables the Unix username/password authentication and forces the Unix system to prompt you for a user name and password.

Basic Native Mode Commands

Note: The examples in the following subsections contain screenshots from the previous generation of In-Sight wafer readers. The extension of job files for In-Sight 1750 wafer readers is **.JOBX**, and not **.JOB** as the screenshots depict.

File & Job Commands

Load File

Loads the specified job from flash memory on the In-Sight sensor, making it the active job.

Syntax: LF[Filename]

Notes:

- The In-Sight sensor must be Offline.

Load File Inputs

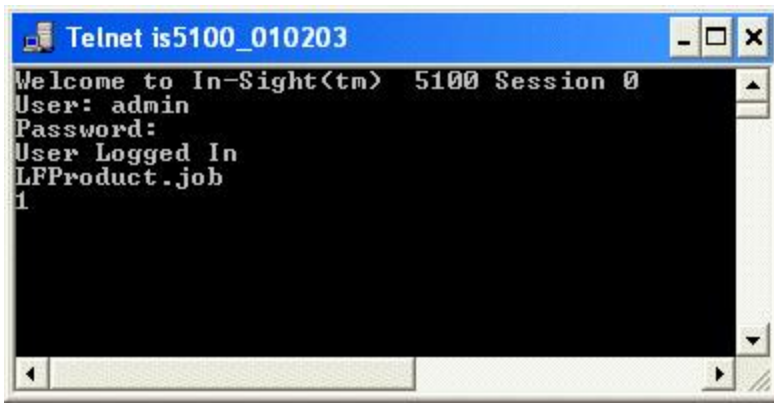
Input	Description
Filename	The name of the job. The command will execute with or without the .JOBX extension.

Load File Outputs

Returns	1	The command was executed successfully.
	0	Unrecognized command.
	-1	The filename is missing.
	-2	The job failed to load, the sensor is Online, or the file was not found.
	-4	The job to be loaded was not found, or the sensor is out of memory.
	-6	User does not have Full Access to execute the command.

Load File Example

To load "Product" as the active job on the In-Sight sensor, issue the following command: **LFProduct.jobx**



The In-Sight sensor responds with 1, indicating that the job was successfully loaded and made active.

Related Topics

[Store File](#)

[Read File](#)

[Write File](#)

[Delete File](#)

[Get File](#)

Store File

Saves the current job in flash memory on the In-Sight sensor.

Syntax: TF[Filename]

Notes:

- The In-Sight sensor must be Offline.

Store File Inputs

Input	Description
Filename	The name of the job. The file extension must be .JOBX .

Store File Outputs

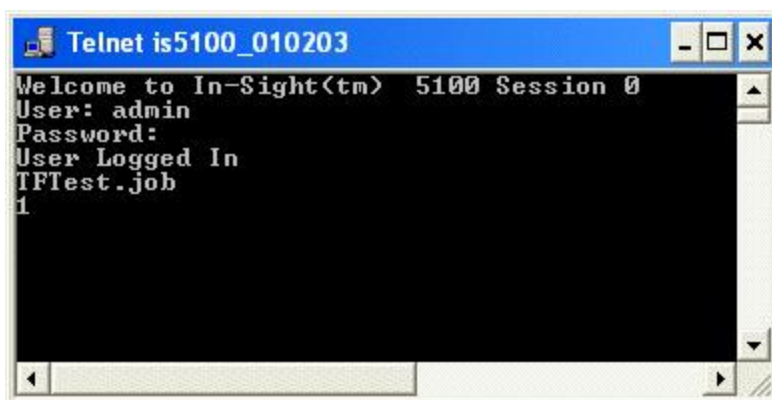
Returns A status code, followed by a line terminator. The status codes are:

- 1 The command was executed successfully.
- 0 Unrecognized command.
- 1 The filename is missing.
- 2 The job failed to save, the sensor is Online or the file was not found, therefore the command could not be executed.
- 6 User does not have Full Access to execute the command.

Store File Example

To name the current job "Test" and save it in flash memory on the In-Sight sensor, issue the following command:

TFTest.jobx



The In-Sight sensor responds with **1**, indicating that the current job was successfully saved.

Related Topics

[Load File](#)

[Read File](#)

[Write File](#)

[Delete File](#)

[Get File](#)

Read File

Reads a job from the specified In-Sight flash memory.

Syntax: RF[Filename]

Read File Inputs

Input	Description
Filename	The name of the job. The command will execute with or without the .JOBX extension.

Read File Outputs

Returns A status code, followed by a line terminator, followed by lines of job data.
The status codes are:

- 1 The command was executed successfully.
- 0 Unrecognized command.
- 1 The job filename is missing.
- 2 There is no job saved with the given name or the job data is invalid, therefore the command could not be executed.
- 6 User does not have Full Access to execute the command.

The lines of job data are:

- Name The first line contains the job name.
- Size The second line contains an integer value for the size (in bytes) of the job.
- Data The actual job data is encoded as ASCII hexadecimal values formatted to 80 characters per line. Each line ends with a terminating character.
- Checksum The last line contains four ASCII hexadecimal bytes that are a checksum of the job data.

Read File Example

To receive information about the job "Model.jobx", which is saved on the In-Sight sensor, issue the following command:

RFModel.jobx

```

Telnet is5100_010203
Welcome to In-Sight<tn> 5100 Session 0
User: admin
Password:
User Logged In
RFModel.job
1
Model.job
826
2756455253494F4E3A2020203236205B332E33302E30302028333434295D205B353130305D0D0A27
434841525345543A2020205B69736F2D383835392D315D205B456E676C6973685D0D0A4130203120
41637175697265496D61676528302C312C382C302C3438302C302C302C302C302C3132382C333329
0D0A2D3120342042494E20303020300D0A000000000D0A53657455707065724C656674284130290D
0A5365744C6F776572526967687428473238290D0A53657441637443656C6C28463132290D0A4F70
536574506C6163656D656E7428302C302C302C30290D0A4F705365745363726F6C6C52616E676528
302C3339392C302C3235290D0A4F7053657455707065724C65667428302C30290D0A4F705365744C
6F776572526967687428302C30290D0A4F7053657453656C28322C30290D0A4F70536574466C6167
732831290D0A4F705365745265667265736828302C312C312C31290D0A2743656C6C4E616D65
496E666F0D0A2743656C6C4E616D65496E666F456E640D0A2743454C4C5441473A0D0A0D0A274345
4C4C544147454E443A0D0A0D0A
8767

```

After issuing the Read File command, the first line outputted to the telnet window is the status code (1), followed by the job name (**Model.jobx**) and size (**826**) in bytes, then the job data (in ASCII hexadecimal format). Finally, the four-byte ASCII hexadecimal checksum is displayed on the last line (**8767**).

Note: The size reported by the **RF** command will be twice the number of bytes of the actual .JOBX file size stored on the In-Sight sensor, due to the ASCII hexadecimal encoding of the data.

Related Topics

[Load File](#)

[Store File](#)

[Write File](#)

[Delete File](#)

[Get File](#)

Write File

Sends a job to the flash memory on the In-Sight sensor.

Syntax: WF
 [Filename]
 [Size]
 [Data]
 [Checksum]

Notes:

- The In-Sight sensor must be Offline.

Write File Inputs

The command is sent on a separate line, followed by the lines of job data, which should be an exact copy of the data received from a [Read File](#) command. The lines of job data are:

Input	Description
Filename	The name of the job. The command will execute with or without the .JOBX extension.
Size	The second line contains an integer value for the size (in bytes) of the job.
Data	The actual job data is encoded as ASCII hexadecimal values formatted to 80 characters per line. Each line ends with a terminating character.
CheckSum	The last line contains four ASCII hexadecimal bytes that are a checksum of the job data.

Write File Outputs

Returns A status code, followed by a line terminator. The status codes are:

- 1 The command was executed successfully.
- 0 Unrecognized command.
- 2 The job could not be written, or the job data is invalid.
- 3 The checksum failed. The checksum does not match the job data.
- 4 The In-Sight sensor is out of memory.
- 6 User does not have Full Access to execute the command.

Write File Example

To write job data to the job "NewModel.jobx", issue the following command: **WF**. After issuing the Write File command, the first input is the job name (**NewModel.jobx**), followed by the size (**826**) in bytes. Next, each ASCII hexadecimal byte of the job file is input. Finally, the four ASCII hexadecimal checksum bytes are input on the last line (**8767**).

```

Telnet is5100_010203
Welcome to In-Sight<tn> 5100 Session 0
User: admin
Password:
User Logged In
WF
NewModel.job
826
2756455253494F4E3A2020203236205B332E33302E30302028333434295D205B353130305D0D0A27
434841525345543A2020205B69736F2D383835392D315D205B456E676C6973685D0D0A4130203120
41637175697265496D61676528302C312C382C302C3438302C302C302C302C302C3132382C333329
0D0A2D3120342042494E20303020300D0A000000000D0A53657455707065724C656674284130290D
0A5365744C6F776572526967687428473238290D0A53657441637443656C6C28463132290D0A4F70
536574506C6163656D656E7428302C302C302C30290D0A4F705365745363726F6C6C52616E676528
302C3339392C302C3235290D0A4F7053657455707065724C65667428302C30290D0A4F705365744C
6F776572526967687428302C30290D0A4F7053657453656C28322C30290D0A4F70536574466C6167
732831290D0A4F705365745265667265736828302C312C312C312C31290D0A2743656C6C4E616D65
496E666F0D0A2743656C6C4E616D65496E666F456E640D0A2743454C4C5441473A0D0A0D0A274345
4C4C544147454E443A0D0A0D0A
8767
1

```

The In-Sight sensor responds with **1**, indicating that the job was successfully sent to the sensor's flash memory.

Notes:

- If the job file already exists on the sensor, the new file is saved with a numeric suffix.
- If the job file does not already exist on the sensor, a new job file is created with the specified name.

Related Topics

[Load File](#)

[Store File](#)

[Read File](#)

[Delete File](#)

[Get File](#)

Delete File

Deletes the specified job file from flash memory on the In-Sight sensor.

Syntax: DF[Filename]

Notes:

- The In-Sight sensor must be Offline.

Delete File Inputs

Input	Description
Filename	The name of the job file. The file extension must be .JOBX .

Delete File Outputs

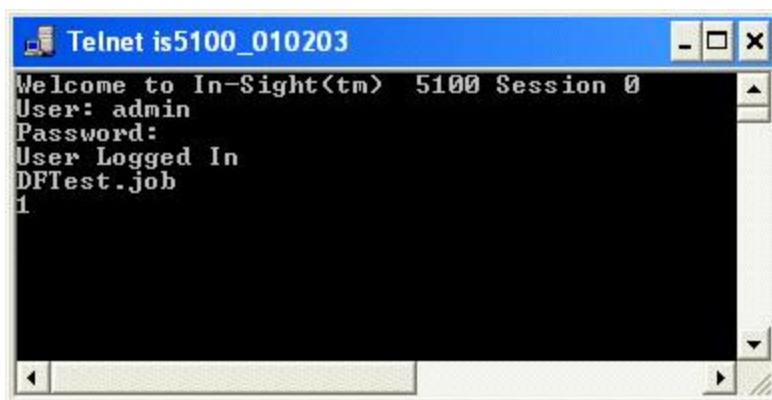
Returns A status code, followed by a line terminator. The status codes are:

- | | |
|----|--|
| 1 | The command was executed successfully. |
| 0 | Unrecognized command. |
| -1 | The filename is missing. |
| -2 | The file could not be deleted, the sensor is Online, a file does not exist with the given name, or the job data is invalid, therefore the command could not be executed. |
| -6 | User does not have Full Access to execute the command. |

Delete File Examples

Example: Delete a Job File

To delete the job "Test.jobx" from the In-Sight sensor, issue the following command: **DFTest.jobx**



The In-Sight sensor responds on the following line with the status code of **1**, indicating that the job was successfully deleted.

Related Topics

[Load File](#)

[Store File](#)

[Read File](#)

[Write File](#)

[Get File](#)

Get File

Returns the file name of the active job on the In-Sight sensor.

Syntax: GF

Notes:

- The active job must be saved before this command can be executed successfully.

Get File Inputs

None.

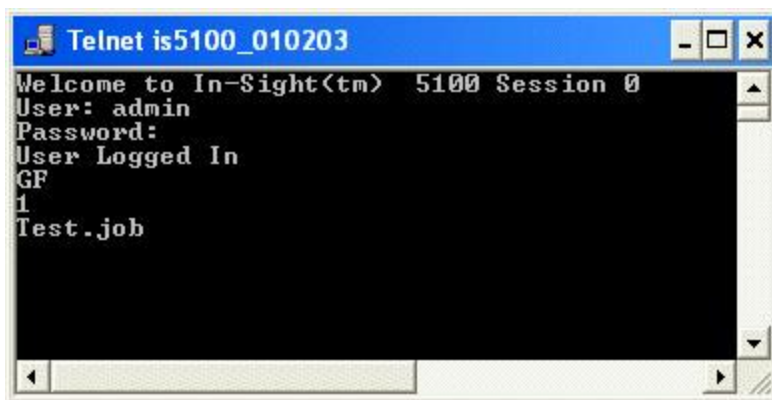
Get File Outputs

Returns A status code, followed by the name of the file. The status codes are:

- 1 The command was executed successfully.
- 0 Unrecognized command.
- 2 The active job has not been saved, therefore the command could not be executed.

Get File Example

Assume that the active, saved job on an In-Sight sensor is named "Test.jobx". To return the filename of the active job, issue the following command: **GF**



```
Telnet is5100_010203
Welcome to In-Sight(tm) 5100 Session 0
User: admin
Password:
User Logged In
GF
1
Test.job
```

The In-Sight sensor responds with **1**, indicating that the command executed successfully. On the following line, the active job filename (**Test.jobx**) is displayed.

Related Topics

[Load File](#)

[Store File](#)

[Read File](#)

[Write File](#)

[Delete File](#)

Set Job

Loads a job from one of the job slots in flash memory on the In-Sight sensor, making it the active job.

Syntax: SJ[ID]

Notes:

- The In-Sight sensor must be Offline.
- To use the job ID number feature, the job to be loaded must be saved with a numerical prefix of 0 to 999.

Set Job Inputs

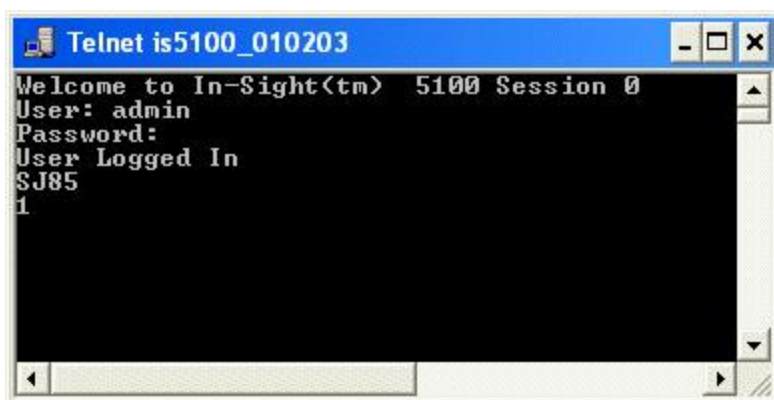
Input	Description
ID	The ID number of the job to load (0 to 999).

Set Job Outputs

Returns	1	The command was executed successfully.
	0	Unrecognized command.
	-1	The ID is less than 0, or is not an integer.
	-2	The job failed to load, the sensor is Online or the file was not found, therefore the command could not be executed.
	-4	The In-Sight sensor is out of memory.
	-6	User does not have Full Access to execute the command.

Set Job Example

To load "85Test.jobx" as the active job on the In-Sight sensor, issue the following command: **SJ85**.



The In-Sight sensor responds with **1**, indicating that the job was successfully loaded and made active.

Related Topics

[Store Job](#)

[Read Job](#)

[Write Job](#)

[Delete Job](#)

[Get Job](#)

Store Job

Saves the current job into the specified slot in flash memory on the In-Sight sensor.

Syntax: TJ[ID] [Name]

Notes:

- The In-Sight sensor must be Offline.
- To use the job ID number feature, the job to be saved must be saved with a numerical prefix of 0 to 999.

Store Job Inputs

Input	Description
ID	The job ID number (0 to 999).
Name	The name of the job. The command will execute with or without the .JOBX file extension.

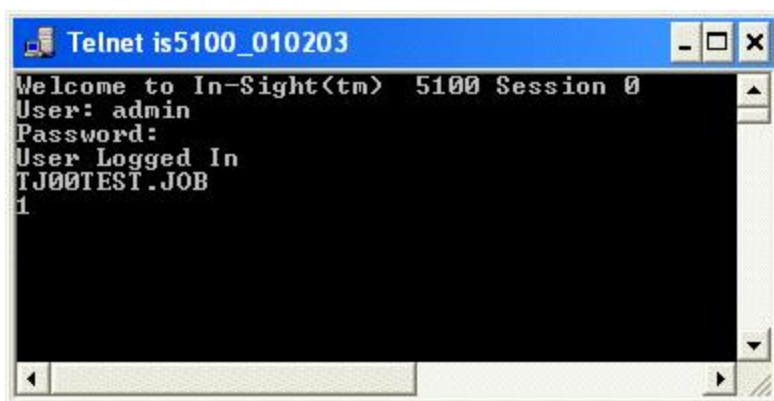
Store Job Outputs

Returns A status code, followed by a line terminator. The status codes are:

- 1 The command was executed successfully.
- 0 Unrecognized command.
- 1 The job ID number is invalid or it is not an integer.
- 2 The sensor is Online, therefore the command could not be executed.
- 6 User does not have Full Access to execute the command.

Store Job Example

To name the current job "Test" and save it into slot 0 on the In-Sight sensor, issue the following command: **TJ00Test.jobx**



The In-Sight sensor responds with 1, indicating that the current job was successfully saved.

Related Topics

[Set Job](#)

[Read Job](#)

[Write Job](#)

[Delete Job](#)

[Get Job](#)

Read Job

Reads a job from the specified In-Sight job slot.

Syntax: RJ[ID]

Read Job Inputs

Input	Description
ID	The job ID number (0 to 999).

Read Job Outputs

Returns A status code, followed by a line terminator, followed by lines of job data.
The status codes are:

- 1 The command was executed successfully.
- 0 Unrecognized command.
- 1 The job ID number is outside the allowable range (**0 to 999**).
- 2 The job could not be read, or the job slot is empty, therefore the command could not be executed.
- 4 The In-Sight sensor is out of memory.
- 6 User does not have Full Access to execute the command.

The lines of job data are:

Name	The first line contains the job name.
Size	The second line contains an integer value for the size (in bytes) of the job.
Data	The actual job data is encoded as ASCII hexadecimal values formatted to 80 characters per line. Each line ends with a terminating character.
Checksum	The last line contains four ASCII hexadecimal bytes that are a checksum of the job data.

Read Job Example

To receive information about the job named "85Test.jobx", which is saved on the In-Sight sensor, issue the following command: **RJ85**

```

Telnet is5100_010203
Welcome to In-Sight<tn> 5100 Session 0
User: admin
Password:
User Logged In
RJ85
1
85Test.job
826
2756455253494F4E3A2020203236205B332E33302E30302028333434295D205B353130305D0D0A27
434841525345543A2020205B69736F2D383835392D315D205B456E676C6973685D0D0A4130203120
41637175697265496D61676528302C312C382C302C3438302C302C302C302C302C3132382C333329
0D0A2D3120342042494E20303020300D0A000000000D0A53657455707065724C656674284130290D
0A5365744C6F7765725269676874284D3431290D0A53657441637443656C6C284C3135290D0A4F70
536574506C6163656D656E7428302C302C302C30290D0A4F705365745363726F6C6C52616E676528
302C3339392C302C3235290D0A4F7053657455707065724C65667428302C30290D0A4F705365744C
6F776572526967687428302C30290D0A4F7053657453656C28322C30290D0A4F70536574466C6167
732831290D0A4F705365745265667265736828302C312C312C31290D0A2743656C6C4E616D65
496E666F0D0A2743656C6C4E616D65496E666F456E640D0A2743454C4C5441473A0D0A0D0A274345
4C4C544147454E443A0D0A0D0A
5384

```

After issuing the Read Job command, the first line outputted to the telnet window is the status code (1), followed by the job name (**85Test.jobx**) and size (**826**) in bytes, then the job data (in ASCII hexadecimal format). Finally, the four-byte ASCII hexadecimal checksum is displayed on the last line (**5384**).

Note: The size reported by the **RJ** command will be twice the number of bytes of the actual .JOB file size stored on the In-Sight sensor, due to the ASCII hexadecimal encoding of the data.

Related Topics

[Set Job](#)

[Store Job](#)

[Write Job](#)

[Delete Job](#)

[Get Job](#)

Write Job

Sends a job to the specified In-Sight job slot in flash memory on the In-Sight sensor.

Syntax: WJ[ID]
 [Filename]
 [Size]
 [Data]
 [Checksum]

Notes:

- The In-Sight sensor must be Offline.

Write Job Inputs

The command and ID are sent on a separate line, followed by the lines of job data, which should be an exact copy of the data received from a [Read Job](#) command.

Input	Description
ID	The job ID number (0 to 999).

The lines of job data are:

Input	Description
Name	The first line contains the job name.
Size	The second line contains an integer value for the size, in bytes, of the job.
Data	The actual job data is encoded as ASCII hexadecimal values formatted to 80 characters per line. Each line ends with a terminating character.
Checksum	The last line contains four ASCII hexadecimal bytes that are a checksum of the job data.

Write Job Outputs

Returns A status code, followed by a line terminator. The status codes are:

- 1 The command was executed successfully.
- 0 Unrecognized command.
- 1 The job ID number is outside the allowable range (0 to 999).
- 2 The job could not be written, or the job data is invalid.
- 3 The checksum failed. The checksum does not match the job data.
- 4 The In-Sight sensor is out of memory.
- 6 User does not have Full Access to execute the command.

Write Job Example

To write job data to the job named "86Test.jobx", issue the following command: **WJ86**

After issuing the command, input the job name (**Test.jobx**), followed by the size (**826**) in bytes. Next, each ASCII hexadecimal byte of the job file is input. Finally, the four ASCII hexadecimal checksum bytes are input on the last line (**5384**).

```

Telnet is5100_010203
Welcome to In-Sight(tm) 5100 Session 0
User: admin
Password:
User Logged In
WJ86
Test.job
826
2756455253494F4E3A2020203236205B332E33302E30302028333434295D205B353130305D0D0A27
434841525345543A2020205B69736F2D383835392D315D205B456E676C6973685D0D0A4130203120
41637175697265496D61676528302C312C382C302C3438302C302C302C302C302C302C3132382C333329
0D0A2D3120342042494E20303020300D0A000000000D0A53657455707065724C656674284130290D
0A5365744C6F7765725269676874284D3431290D0A53657441637443656C6C284C3135290D0A4F70
536574506C6163656D656E7428302C302C302C30290D0A4F705365745363726F6C6C52616E676528
302C3339392C302C3235290D0A4F7053657455707065724C65667428302C30290D0A4F705365744C
6F776572526967687428302C30290D0A4F7053657453656C28322C30290D0A4F70536574466C6167
732831290D0A4F705365745265667265736828302C312C312C312C31290D0A2743656C6C4E616D65
496E666F0D0A2743656C6C4E616D65496E666F456E640D0A2743454C4C5441473A0D0A0D0A274345
4C4C544147454E443A0D0A0D0A
5384
1

```

The In-Sight sensor responds with **1**, indicating that the job was successfully sent to the sensor's flash memory.

Notes:

- If the job file already exists on the sensor, the new job file is saved with a numerical suffix.
- If the job file does not already exist on the sensor, a new job file is created with the specified name.
- The size will be twice the number of bytes of the actual .JOBX file size stored on the In-Sight sensor, due to the ASCII hexadecimal encoding of the data.

Related Topics

[Set Job](#)

[Store Job](#)

[Read Job](#)

[Delete Job](#)

[Get Job](#)

Delete Job

Deletes the job from the specified slot in flash memory on the In-Sight sensor.

Syntax: DJ[ID]

Notes:

- The In-Sight sensor must be Offline.

Delete Job Inputs

Input	Description
ID	The job ID number (0 to 999).

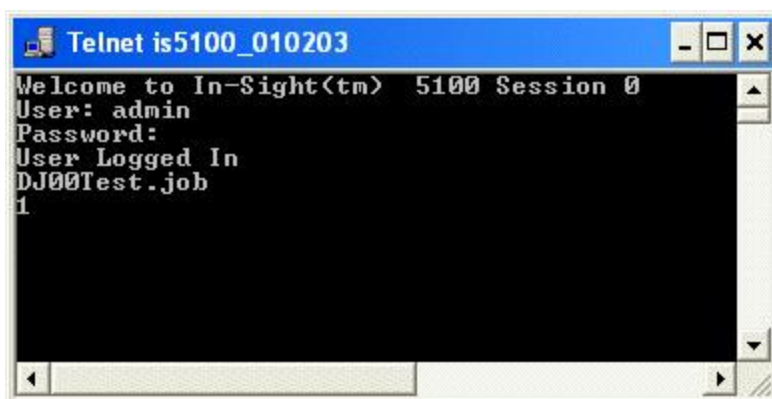
Delete Job Outputs

Returns A status code, followed by a line terminator. The status codes are:

- 1 The command was executed successfully.
- 0 Unrecognized command.
- 1 The job ID number is outside the allowable range (**0 to 999**).
- 2 The job could not be deleted, the sensor is Online, or the job slot is empty, therefore the command could not be executed.
- 6 User does not have Full Access to execute the command.

Delete Job Example

To delete the job "Test.jobx" from slot 0 on the In-Sight sensor, issue the following command: **DJ00Test.jobx**



The In-Sight sensor responds on the following line with the status code of **1**, indicating that the job was deleted successfully.

Related Topics

[Store Job](#)

[Set Job](#)

[Read Job](#)

[Write Job](#)

[Get Job](#)

Get Job

Returns the ID of the active job on the In-Sight sensor.

Syntax: GJ

Notes:

- To use the job ID number feature, the job to be loaded must be saved with a numerical prefix of 0 to 999.
- The active job must be saved with a numerical prefix before this command can be executed successfully. If the job has been dragged and dropped, the file name must have a numerical prefix before this command can be executed successfully.

Get Job Inputs

None.

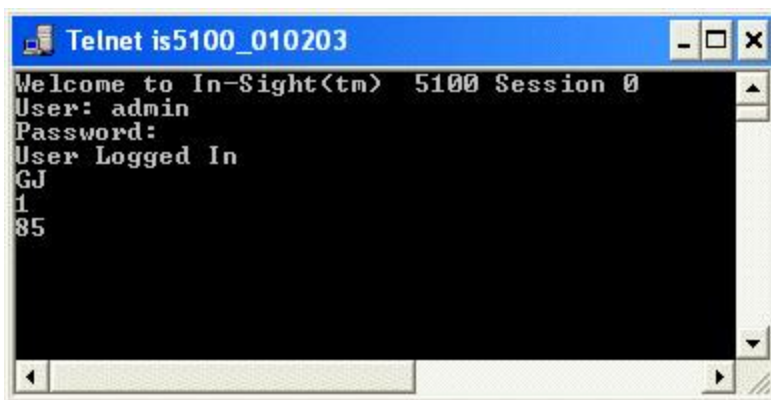
Get Job Outputs

Returns A status code, followed by the integer ID of the active job. The status codes are:

- 1 The command was executed successfully.
- 0 Unrecognized command.
- 2 The active job has not been saved or does not have a numerical prefix, therefore the command could not be executed.

Get Job Example

To return the ID for the active job, issue the following command: **GJ**



```
Telnet is5100_010203
Welcome to In-Sight(tm) 5100 Session 0
User: admin
Password:
User Logged In
GJ
1
85
```

The In-Sight sensor responds with **1** to indicate that the command was executed successfully. The active job's name is "85.JOBX".

Related Topics

[Store Job](#)

[Set Job](#)

[Read Job](#)

[Write Job](#)

[Delete Job](#)

Image Commands

Read BMP

Sends the current image, in ASCII hexadecimal format (formatted to 80 characters per line), from an In-Sight sensor out to a remote device. When converted to binary, the resulting data is in standard BMP image format.

Syntax: RB

Note: Executing the Read BMP command is the equivalent of executing the [Read Image](#) command.

Read BMP Inputs

None.

Read BMP Outputs

Returns A status code, followed by a line terminator, followed by lines of image data. The status codes are:

- 1 The command was executed successfully.
- 0 Unrecognized command.
- 4 The In-Sight sensor is out of memory.
- 6 User does not have Full Access to execute the command.

The lines of image data are:

- Size** The second line contains an integer value for the size, in bytes, of the image.
- Image** The actual image data is encoded as ASCII hexadecimal values formatted to 80 characters per line. Each line ends with a terminating character.
- Checksum** The last line contains four ASCII hexadecimal bytes that are a checksum of the image data.

Read BMP Example

To read the current image (in ASCII hexadecimal format) from an In-Sight sensor, issue the following command: **RB**

[illegible]

The In-Sight sensor initially responds with **1**, indicating the command was successfully executed. The next line contains the size (**3436**) in bytes, followed by the image data (in ASCII hexadecimal format.)

Note: Most of the actual image data has been omitted from the preceding graphic for clarity.

Finally, the four-byte ASCII hexadecimal checksum (**E213**) is returned.

Related Topics

Write BMP

Read Image

Sends the current image, in ASCII hexadecimal format (formatted to 80 characters per line), from an In-Sight sensor out to a remote device. When converted to binary, the resulting data is in standard BMP image format.

Syntax: RI

Note: Executing the Read Image command is the equivalent of executing the [Read BMP](#) command.

Read Image Inputs

None.

Read Image Outputs

Returns A status code, followed by a line terminator, followed by lines of image data. The status codes are:

- 1 The command was executed successfully.
- 0 Unrecognized command.
- 4 The In-Sight sensor is out of memory.
- 6 User does not have Full Access to execute the command.

The lines of image data are:

- Size** The second line contains an integer value for the size, in bytes, of the image.
- Image** The actual image data is encoded as ASCII hexadecimal values formatted to 80 characters per line. Each line ends with a terminating character.
- Checksum** The last line contains four ASCII hexadecimal bytes that are a checksum of the image data.

Read Image Example

To read the current image (in ASCII hexadecimal format) from an In-Sight sensor, issue the following command: **RI**

[illegible]

The In-Sight sensor initially responds with **1**, indicating the command was successfully executed. The next line contains the data size (**3436**) in bytes, followed by the image data (in ASCII hexadecimal format.)

Note: Most of the actual image data has been omitted from the preceding graphic for clarity.

Finally, the four-byte ASCII hexadecimal checksum (**E213**) is returned.

Related Topics

Write Image

Write BMP

Sends image data from a remote device to an In-Sight sensor.

Syntax: WB
 [Size]
 [Image]
 [Checksum]

Notes:

- The In-Sight sensor must be Offline.
- Executing the Write BMP command is the equivalent of executing the [Write Image](#) command.

Write BMP Inputs

The command is sent on a separate line, followed by the lines of image data, which should be an exact copy of the data sent using a [Read BMP](#) command. The lines of image data are:

Input	Description
Size	An integer value for the size (in bytes) of the image file.
Image	The actual image data is encoded as ASCII hexadecimal values formatted to 80 characters per line. Each line ends with a terminating character.
Checksum	The last line contains four ASCII hexadecimal bytes that are a checksum of the image data.

Write BMP Outputs

Returns A status code, followed by a line terminator. The status codes are:

- 1 The command was executed successfully.
- 0 Unrecognized command.
- 2 The image could not be written, or the image data is invalid.
- 3 The checksum failed. The checksum does not match the image data.
- 4 The In-Sight sensor is out of memory.
- 6 User does not have Full Access to execute the command.

Write BMP Example

To send the system settings to the sensor, issue the following command: **WB**. After issuing the Write BMP command, the first input is the size (**3436**) in bytes. Next, each ASCII hexadecimal byte of the image data is input. Finally, the four ASCII hexadecimal checksum bytes are input on the last line (**E213**).

[illegible]

Note: Most of the actual image data has been omitted from the preceding graphic for clarity.

The In-Sight sensor responds with **1**, indicating that the image data was successfully sent to the sensor.

Related Topics

Read BMP

Write Image

Sends image data from a remote device to an In-Sight sensor.

Syntax: WI
 [Size]
 [Image]
 [Checksum]

Notes:

- The In-Sight sensor must be Offline.
- Executing the Write Image command is the equivalent of executing the [Write BMP](#) command.

Write Image Inputs

The command is sent on a separate line, followed by the lines of image data, which should be an exact copy of the data sent using a [Read Image](#) command. The lines of image data are:

Input	Description
Size	An integer value for the size, in bytes, of the image file.
Image	The actual image data is encoded as ASCII hexadecimal values formatted to 80 characters per line. Each line ends with a terminating character.
Checksum	The last line contains four ASCII hexadecimal bytes that are a checksum of the image data.

Write Image Outputs

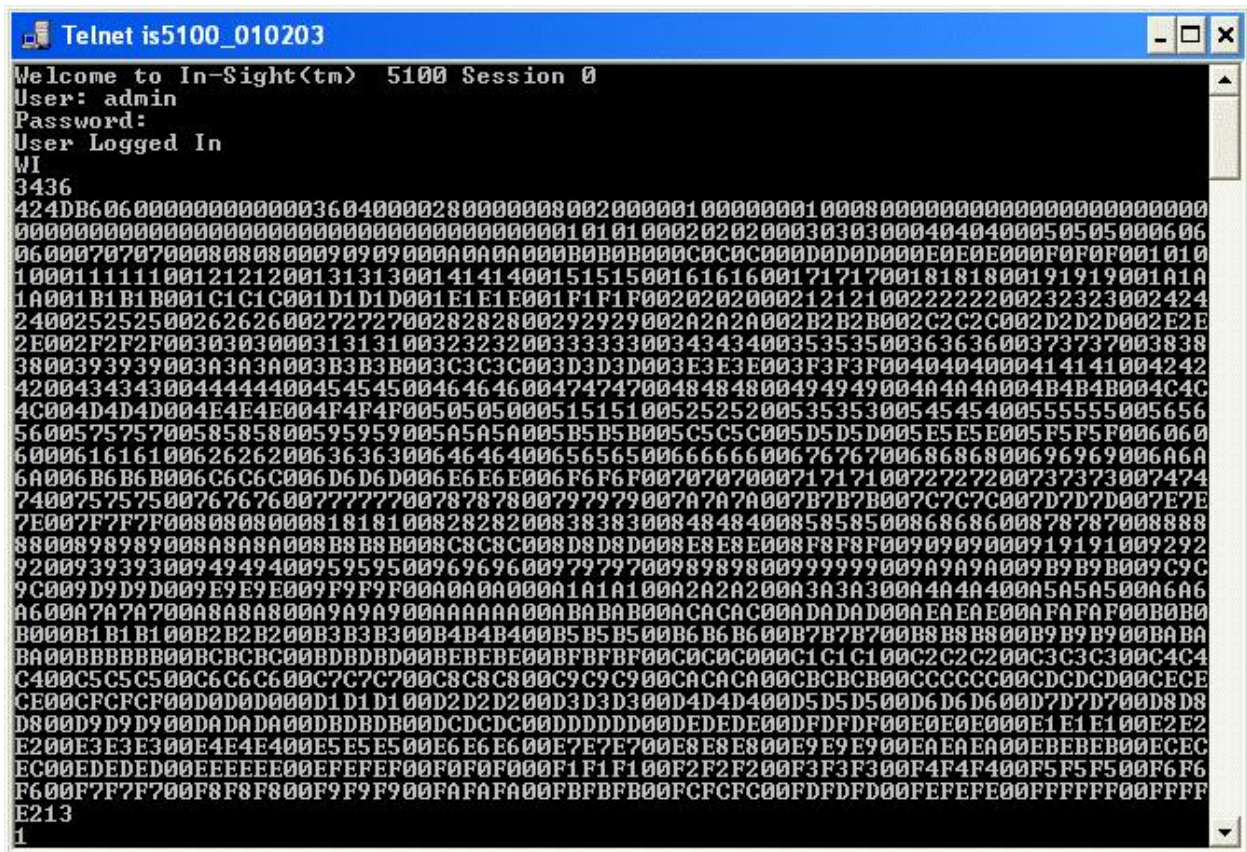
Returns A status code, followed by a line terminator. The status codes are:

- 1 The command was executed successfully.
- 0 Unrecognized command.
- 2 The image could not be written, or the image data is invalid.
- 3 The checksum failed. The checksum does not match the image data.
- 4 The In-Sight sensor is out of memory.
- 6 User does not have Full Access to execute the command.

Write Image Example

To send the system settings to the sensor, issue the following command: **WI**

After issuing the Write Image command, the first input is the size (**3436**) in bytes. Next, each ASCII hexadecimal byte of the image data is input. Finally, the four ASCII hexadecimal checksum bytes are input on the last line (**E213**).



Note: Most of the actual image data has been omitted from the preceding graphic for clarity.

The In-Sight sensor responds with **1**, indicating that the image data was successfully sent to the sensor.

Related Topics

[Read Image](#)

Settings & Cell Value Commands

Get Value

Returns the value contained in the specified cell.

Syntax: GV[Column] [Row]

Get Value Inputs

Input	Description
Column	The column letter of the cell value to get (A to Z).
Row	The row number of the cell value to get. The row number must consist of three digits (000 to 599).

Get Value Outputs

Returns A status code, followed by a line terminator, followed by the value contained in the specified cell. The status codes are:

- 1 The command was executed successfully.
- 0 Unrecognized command.
- 1 The cell ID is invalid.
- 2 The command could not be executed.

Note: An In-Sight cell containing any numeric value will return a float value formatted to 3 decimal places when requested by the Get Value command, regardless of whether the cell contains an integer or a floating-point value. If the cell contains a non-printing character, such as a Structure, a pound character (#) will replace the non-printing character. If the cell is empty, a null string will be sent.

Get Value Example

- Assume that cells A1, B0 and B1 are populated as follows:

A1 = 4.000

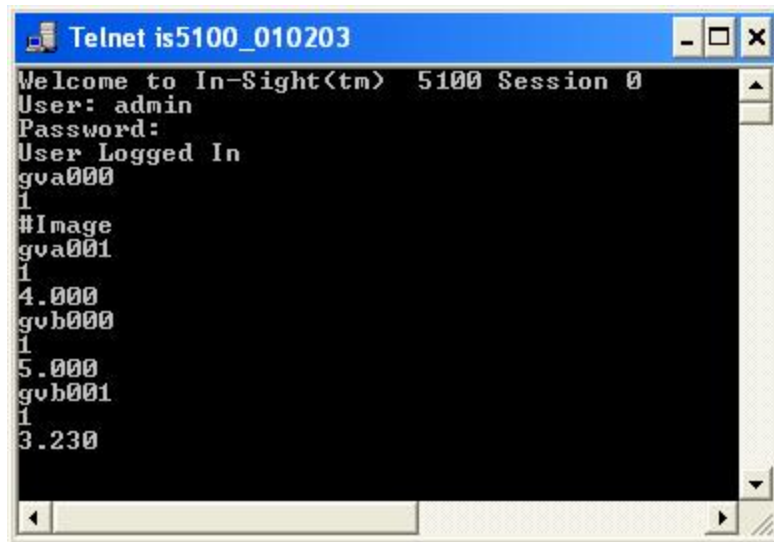
B0 = EditInt(0,10)

B1 = EditFloat(0,10)

A1 currently contains the value **4**; B0 contains the value **5**; and B1 contains the value **3.230**. The spreadsheet looks like this:

	A	B	C
0	5 Image	5	
1	4.000	3.230	
2			

Issuing various GV commands produces the following results:



```
Telnet is5100_010203
Welcome to In-Sight(tm) 5100 Session 0
User: admin
Password:
User Logged In
gva000
1
#Image
gva001
1
4.000
gva000
1
5.000
gva001
1
3.230
```

Set Integer

Sets the control contained in a cell to the specified integer value. The control must be of the types EditInt, Checkbox, or ListBox.

Syntax: SI[Column][Row][Int]

Set Integer Inputs

Input	Description
Column	The column letter of the cell value to set (A to Z).
Row	The row number of the cell value to set. The row number must contain a three-digit number (000 to 599).
Int	The integer value to set.

Set Integer Outputs

Returns	1	The command was executed successfully.
	0	Unrecognized command.
	-1	The cell ID is invalid, or was not an integer.
	-2	The command could not be executed, or the specified integer value is outside of the control's valid range. For example, the specified cell may not contain a control of the valid type.
	-6	User does not have Full Access to execute the command.

Set Integer Examples

[Example 1: Change the Value of an EditInt Control](#)

[Example 2: Toggle the State of a CheckBox Control](#)

[Example 3: Change the Selection within a ListBox Control](#)

Example 1: Change the Value of an EditInt Control

The command to set an EditInt control in cell A2 to the integer value **8** is: **SIA0028**

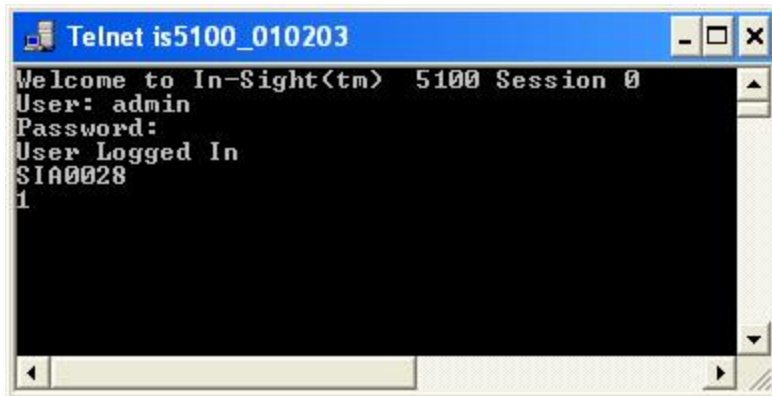
1. Assume cell A2 is populated as follows:

A2 = EditInt(0,10)

A2 currently contains the value **3**. The spreadsheet looks like this:

	A	B	C
0	Image		
1			
2	3		
3			

2. We would like to change the value in A2 to **8** using the Set Integer Native Mode command; issue the command **SIA0028** as shown in the following telnet client window:



The In-Sight sensor responds with **1**, indicating that the command was successfully executed. The EditInt control has been successfully changed from **3** to **8** within the spreadsheet:

	A	B	C
0	Image		
1			
2	8		
3			

Example 2: Toggle the State of a CheckBox Control

You can enable or disable a CheckBox control using the Set Integer command by sending a value of **1** (enable) or **0** (disable) to the cell that contains the check box.

1. Assume cells B2, D2 and E2 are populated as follows:

B2 = CheckBox("Draw circle")

D2 = 'Check box integer value =

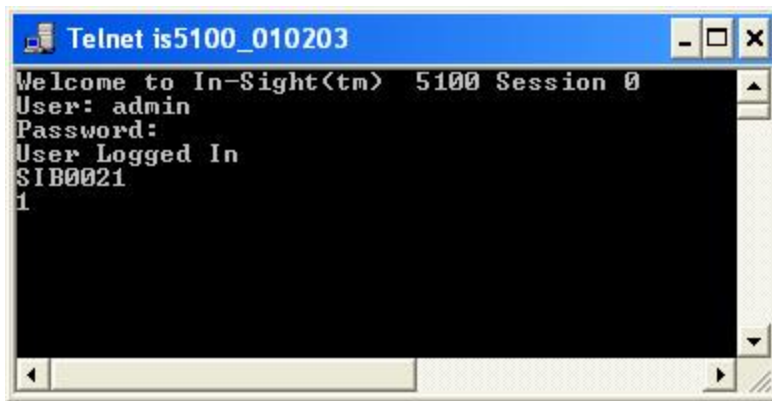
E2 = \$B\$2

The check box is currently cleared, as shown:

	A	B	C	D	E	F
0	 Image					
1						
2		☐ Draw circle		Check box integer value =	0.000	
3						

Note: Cell E2 is simply reporting the integer value of the CheckBox control.

- To select this check box using the Set Integer Native Mode command, issue the command **SIB0021**, as shown in the following telnet client window:



The In-Sight sensor responds with **1**, indicating that the command was successfully executed. The check box is now selected:

	A	B	C	D	E	F
0	 Image					
1						
2		☒ Draw circle		Check box integer value =	1.000	
3						

Cell E2 changed from **0** to **1** to reflect the new integer value of the CheckBox control.

Example 3: Change the Selection within a ListBox Control

Each item within a ListBox control has an integer value (zero-based) assigned to it. You can "choose" a different value from the list by using the Set Integer Native Mode command to assign a different value to the cell that contains the ListBox control.

- Assume cells B2, C2 and D2 are populated as follows:

B2 = ListBox("Red", "Orange", "Yellow", "Green", "Blue", "Indigo", "Violet")

C2 = Integer List Value =

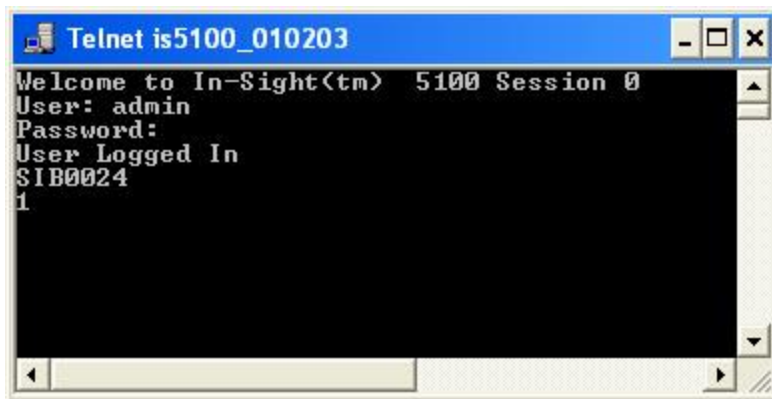
D2 = \$B\$2

The ListBox control is currently set to **Yellow**; this corresponds with an integer value of **2**, as shown in the following spreadsheet:

	A	B	C	D	E
0	Image				
1					
2		Yellow	Integer List Value =	2.000	
3					

Note: Cell D2 is simply reporting the integer value of cell B2.

- To change the current list selection to **Blue** using the Set Integer Native Mode command, issue the command **SIB0024**, as shown in the following telnet client window:



The In-Sight sensor responds with **1**, indicating that the command was successfully executed. The list selection has been changed from **Yellow** to **Blue** within the spreadsheet:

	A	B	C	D	E
0	Image				
1					
2		Blue	Integer List Value =	4.000	
3					

Cell D2 changed from **2** to **4** to reflect the new integer value of the ListBox control.

Set Float

Sets an edit box control contained in a cell to a specified floating-point value. The edit box control must be of the EditFloat type.

Syntax: SF[Column][Row][Float]

Set Float Inputs

Input	Description
Column	The column letter of the cell value to set (A to Z).
Row	The row number of the cell value to set. The row number must consist of three digits (000 to 599).
Float	The floating-point value to set, including the decimal point (.) character.

Set Float Outputs

Returns

- 1 The command was executed successfully.
- 0 Unrecognized command.
- 1 The cell ID is invalid, or the specified value does not contain a floating-point number.
- 2 The command could not be executed. For example, the specified cell may not contain an edit box control, or the edit box control was not created by the EditFloat function.
- 6 User does not have Full Access to execute the command.

Set Float Example

The command to set an edit box contained in cell A2 to the floating point value 7.2 is: **SFA0027.2**

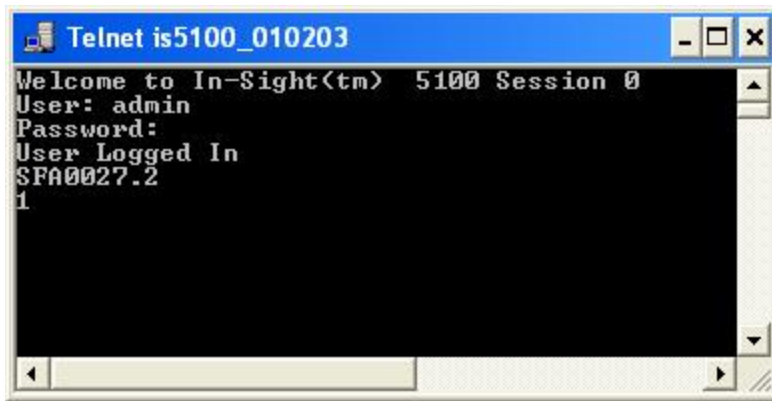
1. Assume cell A2 is populated as follows:

A2 = EditFloat(0,10)

A2 currently contains a value of **3.5**. The spreadsheet looks like this:

	A	B	C
0	Image		
1			
2	3.500		
3			

2. To change this value to **7.2** using the Set Float Native Mode command; issue the command **SFA0027.2** as shown in the following telnet client window:



The In-Sight sensor responds with **1**, indicating that the command was successfully executed. The EditFloat control has been successfully changed from **3.5** to **7.2**:

	A	B	C
0	Image		
1			
2	7.200		
3			

Set String

Sets an edit box control contained in a cell to a specified string. The edit box must be of the type EditString.

Syntax: SS[Column] [Row] [String]

Set String Inputs

Input	Description
Column	The column letter of the cell value to set (A to Z).
Row	The row number of the cell value to set. The row number must contain a three-digit number (000 to 399).
String	The string to set.

Set String Outputs

Returns	1	The command was executed successfully.
	0	Unrecognized command.
	-1	The cell ID or is invalid.
	-2	The input string is longer than the specified maximum string length in the EditString function or the cell does not contain an EditString function.
	-6	User does not have Full Access to execute the command.

Set String Example

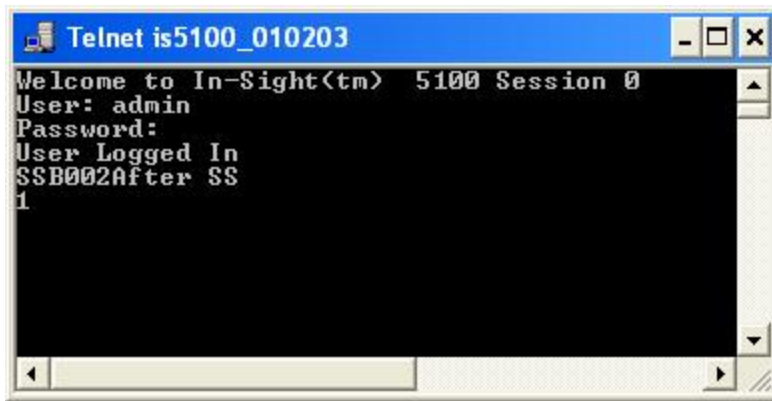
1. Assume cell B2 is populated as follows:

B2 = [EditString](#)(20)

The EditString control currently contains the text **Before SS**. The spreadsheet looks like this:

	A	B	C
0	 Image		
1			
2		Before SS	
3			

2. To change this text to **After SS** using the Set String Native Mode command, issue the command **SSB002After SS**, as shown in the following telnet client window:



The In-Sight sensor responds with **1**, indicating that the command was successfully executed. The EditString control has been successfully changed from **Before SS** to **After SS**:

	A	B	C
0	Image		
1			
2		After SS	
3			
4			

Get Info

Returns the sensor information of the In-Sight sensor.

Syntax: GI

Get Info Inputs

None.

Get Info Outputs

Returns A status code, followed by the sensor information. The status codes are:

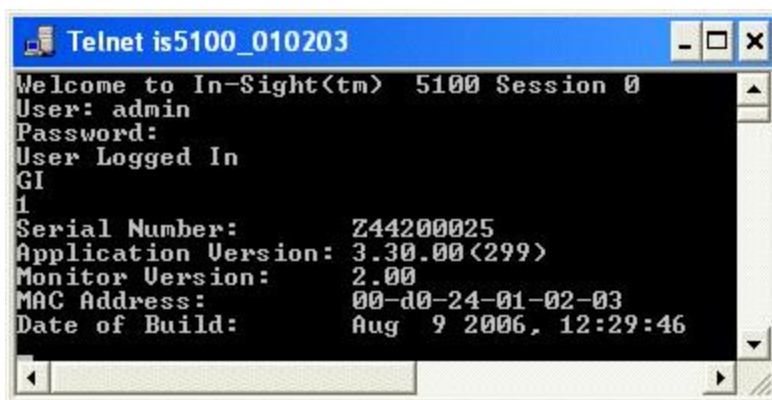
1	The command was executed successfully.
0	Unrecognized command.
-2	The command could not be executed.

System information includes:

Serial Number	The serial number of the In-Sight sensor.
Application Version	The current firmware version on the In-Sight sensor.
Monitor Version	The Monitor version of the In-Sight sensor.
MAC Address	The MAC address of the In-Sight sensor.
Date of Build	The date of build of the firmware on the In-Sight sensor.

Get Info Example

To display system information about an In-Sight sensor, issue the following command: **GI**



```

Telnet is5100_010203
Welcome to In-Sight(tm) 5100 Session 0
User: admin
Password:
User Logged In
GI
1
Serial Number:      Z44200025
Application Version: 3.30.00(299)
Monitor Version:    2.00
MAC Address:        00-d0-24-01-02-03
Date of Build:      Aug 9 2006, 12:29:46
  
```

The sensor initially responds with 1, indicating the command successfully executed. On the following lines, the system information is reported.

Execution & Online Commands

Set Online

Sets the In-Sight sensor into Online or Offline mode.

Syntax: SO[Int]

Notes:

- This command cannot place the In-Sight sensor into Online mode if the sensor has been set Offline either manually in the user interface or by a Discrete Input signal.

Set Online Inputs

Input	Description
Int	0 Sets the In-Sight sensor Offline.
	1 Sets the In-Sight sensor Online.

Set Online Outputs

Returns	1	The command was executed successfully.
	0	Unrecognized command.
	-1	The value given for Int is either out of range, or is not a valid integer.
	-2	The command could not be executed.
	-5	The communications flag was successful but the sensor did not go Online because the sensor is set Offline manually through the user interface or by a Discrete I/O signal.
	-6	User does not have Full Access to execute the command.

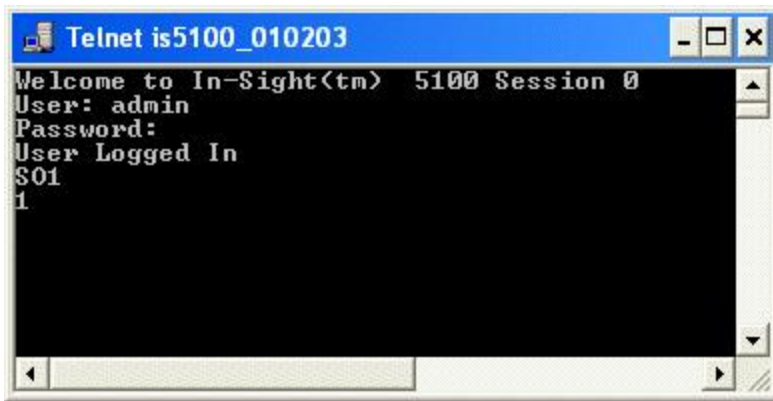
Set Online Examples

[Example 1: Set the In-Sight Sensor into Online Mode](#)

[Example 2: Set the In-Sight Sensor into Offline Mode](#)

Example 1: Set the In-Sight Sensor into Online Mode

To set an In-Sight sensor into Online mode, issue the following command: **SO1**

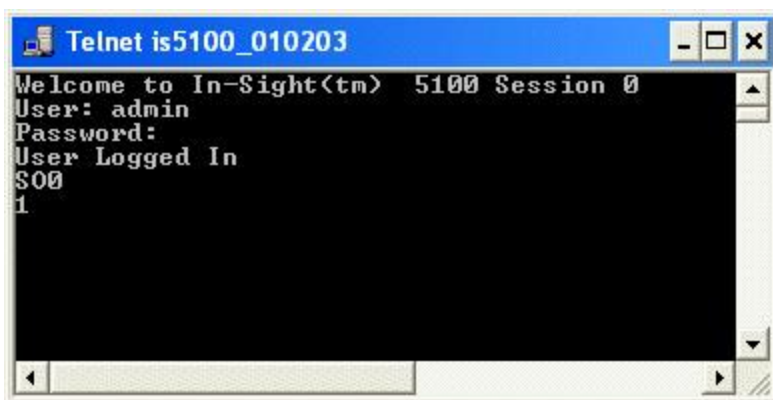
A screenshot of a Telnet window titled 'Telnet is5100_010203'. The window has a blue title bar and standard Windows window controls. The text inside the window shows a login sequence: 'Welcome to In-Sight(tm) 5100 Session 0', 'User: admin', 'Password:', 'User Logged In', 'S01', and '1'. The '1' is the status code indicating the sensor is online.

```
Telnet is5100_010203
Welcome to In-Sight(tm) 5100 Session 0
User: admin
Password:
User Logged In
S01
1
```

The In-Sight sensor responds with a status code of **1**, indicating that the sensor was successfully placed Online.

Example 2: Set the In-Sight Sensor into Offline Mode

To set an In-Sight sensor into Offline mode, issue the following command: **S00**

A screenshot of a Telnet window titled 'Telnet is5100_010203'. The window has a blue title bar and standard Windows window controls. The text inside the window shows a login sequence: 'Welcome to In-Sight(tm) 5100 Session 0', 'User: admin', 'Password:', 'User Logged In', 'S00', and '1'. The '1' is the status code indicating the sensor is offline.

```
Telnet is5100_010203
Welcome to In-Sight(tm) 5100 Session 0
User: admin
Password:
User Logged In
S00
1
```

The In-Sight sensor responds with a status code of **1**, indicating that the sensor was successfully taken Offline.

Get Online

Returns the Online state of the In-Sight sensor.

Syntax: GO

Get Online Inputs

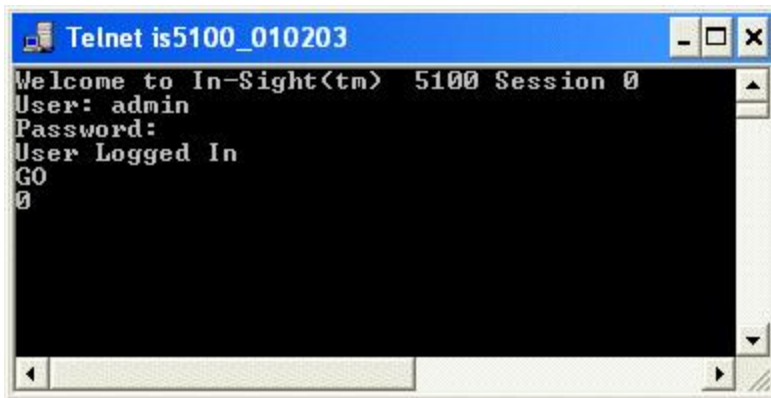
None.

Get Online Outputs

Returns	0	The In-Sight sensor is currently Offline.
	1	The In-Sight sensor is currently Online.

Get Online Example

To determine the Online state of an In-Sight sensor, issue the following command: **GO**



```
Telnet is5100_010203
Welcome to In-Sight(tm) 5100 Session 0
User: admin
Password:
User Logged In
GO
0
```

The In-Sight sensor responds with **0**, indicating that the sensor is currently Offline.

Set Event

Triggers a specified event.

Syntax: SE[Int]

Set Event Inputs

Input	Description
Int	The Event code to set.
(0 to 7)	Specifies a soft trigger (Soft 0, Soft 1, ... Soft 7).
8	Acquire an image and update the spreadsheet.

Set Event Outputs

Returns

- 1 The command was executed successfully.
- 0 Unrecognized command.
- 1 The number is either out of range (**0 to 8**) or not an integer.
- 2 The command could not be executed.
- 6 User does not have Full Access to execute the command.

Set Event Example

The command to trigger Event 0 (Soft 0) is: **SE0**

- Assume cells A2 and A3 are populated as follows:

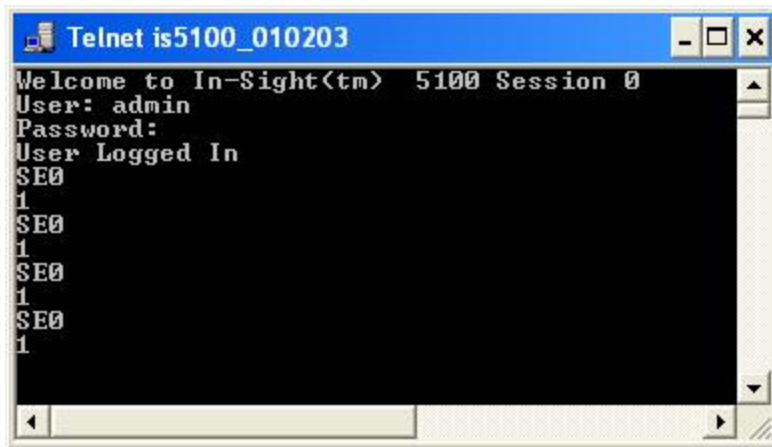
A2 = Event(80,0)

A3 = Count(\$A\$2,9999,0,0)

The spreadsheet looks like this:

	A	B	C
0	Image		
1			
2	Event		
3	0.000		

- Using the telnet client, issue 4 consecutive Set Event commands:



The counter in cell A3 is incremented by 1 for each time **SE0** is sent:

	A	B	C
0	Image		
1			
2	Event		
3	4.000		

Set Event and Wait

Triggers a specified event and waits until the command is completed to return a response.

Syntax: SW[Int]

Note: The In-Sight sensor must be Online.

Set Event and Wait Inputs

Input	Description
Int	The Event code to set.
(0 to 7)	Specifies a soft trigger (Soft 0, Soft 1, ... Soft 7).
8	Acquire an image and update the spreadsheet.

Set Event and Wait Outputs

Returns

- 1 The command was executed successfully.
- 0 Unrecognized command.
- 1 The number is either out of range (**0 to 8**) or not an integer.
- 2 The command could not be executed, or the sensor is Offline.
- 6 User does not have Full Access to execute the command.

Set Event and Wait Example

The command to trigger Event 0 (Soft 0) is **SW0**. A response is not sent until the command has been completed.

1. Assume cells A2 and A3 are populated as follows:

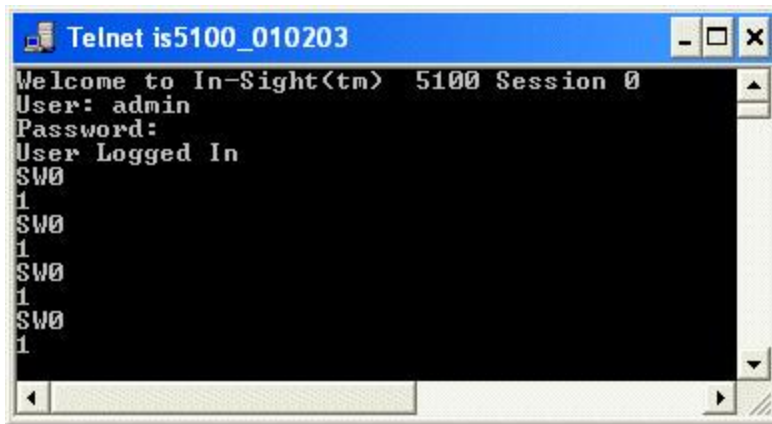
A2 = Event(80,0)

A3 = Count(\$A\$2,9999,0,0)

The spreadsheet looks like this:

	A	B	C
0	Image		
1			
2	Event		
3	0.000		

2. Place the sensor Online. Using the telnet client, issue 4 consecutive Set Event & Wait commands:



```
Telnet is5100_010203
Welcome to In-Sight(tm) 5100 Session 0
User: admin
Password:
User Logged In
SW0
1
SW0
1
SW0
1
SW0
1
SW0
1
```

The counter in cell A3 is incremented by 1 for each time **SW0** is sent:

	A	B	C
0	Image		
1			
2	Event		
3	4.000		

Reset System

Resets the In-Sight sensor. This command is similar to physically cycling power on the sensor.

Syntax: RT

Reset System Inputs

None.

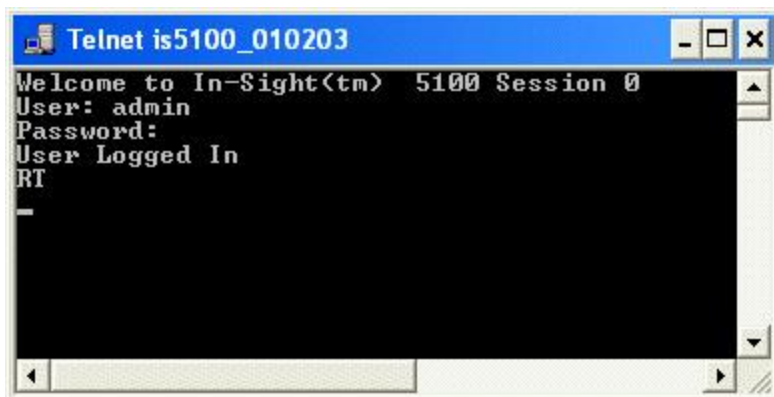
Reset System Outputs

Returns A status code, followed by a line terminator, followed by lines of settings data. The only status code returned is:

-6 User does not have Full Access to execute the command.

Reset System Example

To reset the In-Sight sensor, issue the following command: **RT**



The In-Sight sensor is successfully reset and no status code is returned.

Abort Execution

Terminates the current execution of a specified function within the spreadsheet.

Syntax: AE[int]

Abort Execution Inputs

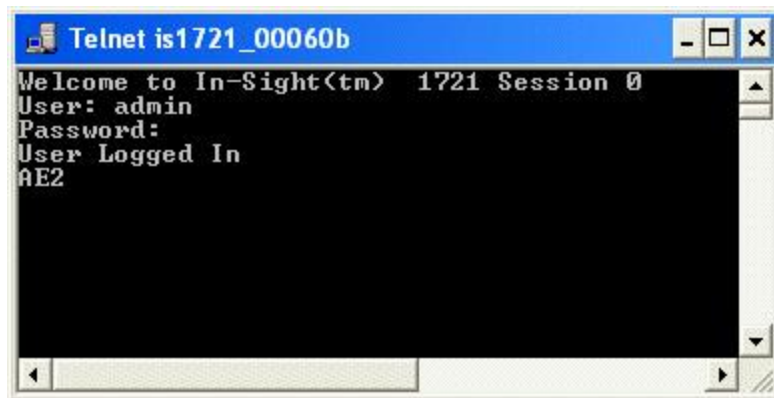
Input	Description
Int	The code used to terminate the specified function. 2 = Terminates the WaferID function. Note: WaferID is the only function currently supported.

Abort Execution Outputs

Unlike most Native Mode commands, no status code is returned after the Abort Execution command is executed; however, depending on the function that is terminated, other information may be returned.

Abort Execution Example

To terminate the WaferID function, issue the following command: **AE2**



Extended Native Mode Commands

Get

The Get Filelist command is used in conjunction with the **Get** extended Native Mode command to receive information from the In-Sight vision system and its spreadsheet:

Syntax: Get [command]

Note: A space is needed between **Get** and the command, as well as between the command and any input parameters.

Get Filelist

Returns the number of files stored in memory, and the name of each file in memory on the In-Sight vision system.

Syntax: Get Filelist

Get Filelist Inputs

None.

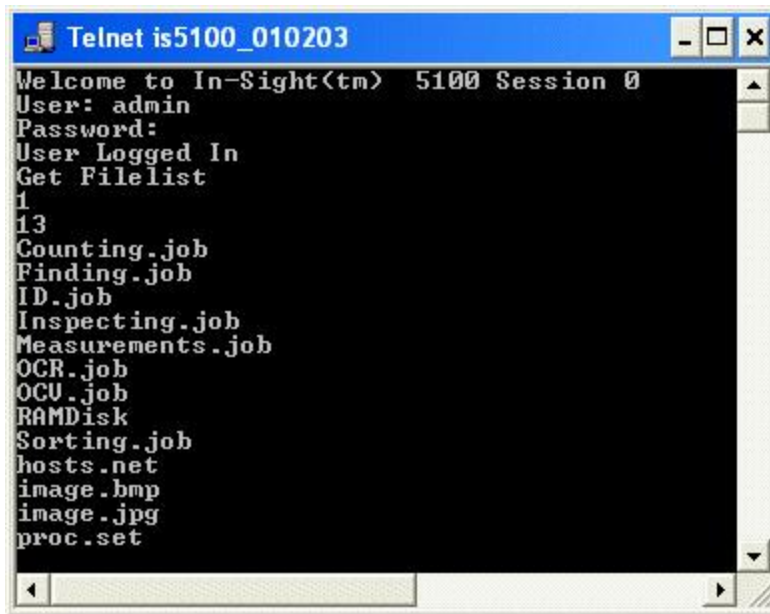
Get Filelist Outputs

Returns A status code, followed by the number of files stored in memory, followed by the name of each file in memory. The status codes are:

- 1 The command was executed successfully.
- 0 Unrecognized command.
- 1 The input is invalid or has exceeded 15 bytes in length.
- 2 The command could not be executed.

Get Filelist Example

To retrieve a count and list of files that are stored on the In-Sight vision system, issue the following command: **Get Filelist**



```
Telnet is5100_010203
Welcome to In-Sight(tm) 5100 Session 0
User: admin
Password:
User Logged In
Get Filelist
1
13
Counting.job
Finding.job
ID.job
Inspecting.job
Measurements.job
OCR.job
OCV.job
RAMDisk
Sorting.job
hosts.net
image.bmp
image.jpg
proc.set
```

After issuing the command **Get Filelist**, the In-Sight vision system responds with a status code (**1**), followed by the number of files present in memory (**13**). Finally, the individual name of each file is displayed.

Evaluate

Executes In-Sight functions as well as inserts formulas into the In-Sight spreadsheet. Evaluate executes commands for retrieving information from and making changes to In-Sight sensors.

Syntax: EV [command]

Notes:

- Changes made to the In-Sight system settings using the Evaluate command are temporary; rebooting the In-Sight sensor will lose any changes.

Evaluate Inputs

Command Any supported In-Sight Wafer ID function. See subsections for list of supported functions.

Evaluate Outputs

Returns • A status code. The status codes are:

- 1 The command was executed successfully.
- 0 Unrecognized command.
- 2 The command could not be executed.
- 6 User does not have Full Access to execute the command.

- The number of characters (in bytes) contained within the output that follows.
- The results of the expression in XML format (only shown if the status code is 1.)

Note: The results of the expression are dependent on the command that is executed.

WaferID Inputs & Outputs

The following table contains a description of inputs and outputs for WaferID Native Mode commands.

Note: Any of the following inputs that are enclosed in quotation marks (such as "FieldDef" for SetConfigFieldDef) must be enclosed by quotes when sent to the In-Sight wafer reader.

Setting	Description
Accept	The Accept Threshold setting. (40 to 100), 50*
AcqEnable	The status of the Disable Acquire setting. 0 = Disable* 1 = Enable
AcquireFlag	Specifies whether an image will be acquired when a read is performed. 0 = A read will be performed on the last image acquired. 1 = A read will be performed after a new image is acquired.

Setting	Description
AppendSEMICHkToT7	Specifies whether the SEMI checksum is automatically calculated and appended to the read result. This setting is only applicable when '2D, T7' is the selected mark. 0 = Disable* 1 = Enable
AvgScore	The average raw response score of all reads that passed. (0 to 100)
BaseScore	Returns the base score, that does not include any bonus points for passing the checksum.
BC3Digits	The status of the Three Digit setting. 0 = Disable* 1 = Enable
BCHyphen	The status of the Hyphen setting. 0 = Disable* 1 = Enable
BComp	When a barcode is the selected mark, an internally generated value is returned for the encoded state of the settings for barcode Compression option, Hyphen option and Three Digit option. The values for the individual Advanced Dialog settings can be returned or set using: GetConfigBCType GetConfigBCHyphen GetConfigBC3Digits SetConfigBCType SetConfigBCHyphen SetConfigBC3Digits
BCType	The index of the Compressed Barcode type for the specified Config. 0 = None* 1 = Base 35 2 = Compressed 1 3 = Compressed 2 4 = Compressed 3 5 = Compressed 4 6 = Compressed 5 7 = Compressed 6 8 = Compressed 7 9 = Compressed 8 11 = Compressed 10 14 = Compressed 13
BestConfig	The index of the Config that passed during the last read.
Bright	Specifies whether Bright Field Light Mode settings will be tried during tuning. 0 = Disable Brightfield tuning 1 = Enable Brightfield tuning*
Char	The ASCII value of the indexed character in a read string.
Char Index	Specifies the position number (zero-based) of a character within a string.

Setting	Description
CharSize	The settings information for the Size setting. High: (16 to 120) Wide: (8 to 480)
CharSpacing	The setting for the tolerance of the spacing between adjacent characters and the vertical alignment of the characters. 1 = Standard 2 = Relaxed
Checksum	The method by which the read result from a character string will be validated to ensure reliability, and establishes the scoring basis for determining the pass/fail status of the read. 0 = Virtual 1 = SEMI 2 = SEMI with Virtual* 3 = BC 412 with Virtual 4 = IBM 412 with Virtual
Col (Char)	The column coordinate of the indexed character in a string. (4 to 640)
Col (Fixture)	The column coordinate of the fixture. (4 to 640), 70*
Col (Region)	The column coordinate of the Config's region. 752 x 480 image resolution: (0 to 751) 1024 x 768 image resolution: (0 to 1023)
Color	The expected color of the mark to read. 0 = Black* 1 = White
CompatibilityMode	Specifies an optional Compatibility Mode to use for the In-Sight wafer reader. 0 = None* 1 = Mode 1
Config	The Config number. (0 to 49)
"ConfigName"	The name associated with the Config.
Current	The read of the current Config.
Dark	Specifies whether Dark Field Light Mode settings will be tried during tuning. 0 = Disable Darkfield tuning 1 = Enable Darkfield tuning*
Enable	Status of the Enabled setting for the specified Config. 0 = Disable 1 = Enable
ExecutionSeconds	The read execution time, in seconds.
FailCharsOutRegion	Specifies whether the read will fail if the center of any character is located outside of the region. 0 = Disable* 1 = Enable
"FieldDef"	User-defined entries for the Field String (200 characters maximum).

Setting	Description
"FieldString"	The number of characters contained in the character string or encoded in a barcode (31 characters maximum). default = "*****CS"
Fixture	The coordinates and orientation of the fixture. Col: (4 to 640) Row: (4 to 480) Theta: (-179 to 180) Note: For In-Sight 1750 wafer readers, the Col, Row, and Theta values are always 0.
Flag	Specifies whether tuning will be performed on a Config. 0 = Start tuning 1 = Get tuning status 2 = Accept current parameters
High (CharSize)	The height of a 2D symbol or the average height of the characters in a string, as measured in pixels. (16 to 120), 42*
High (Region)	The height of the region (along the Y-axis) in which the wafer mark is expected to appear in the field of view. 752 x 480 image resolution: (4 to 480), 100* 1024 x 768 image resolution: (4 to 480), 100*
LastRead	Internal Use Only.
LEDOVERRIDE	Specifies whether to use a custom Light Mode. 0 = Disable Custom Light Mode* 1 = Enable Custom Light Mode

Setting	Description
LEDOVERRIDEMask	Specifies the light rows to enable. The Mask parameter is a binary value that is converted into its decimal equivalent.

To enable more than one light row, determine the binary value for the enabled rows, then convert the value to its decimal equivalent. [Example](#)

To enable light row 1 and light row 4:

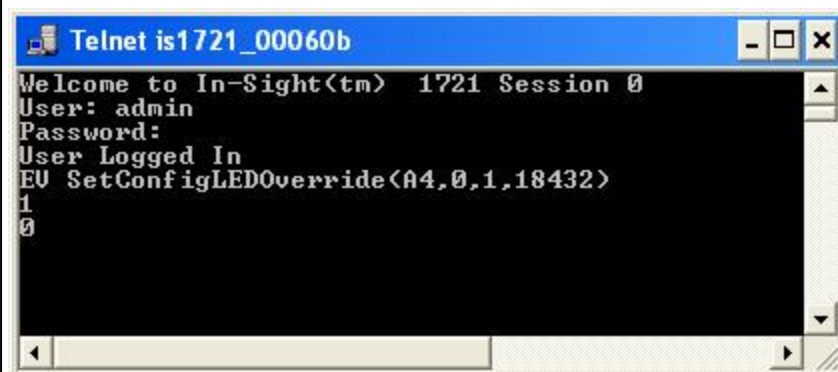
1. Determine the binary value used to enable the light rows.

Light Row	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Binary	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0

2. Convert the binary value to its decimal equivalent.

0100100000000000 = 18432

3. Send the SetConfigLEDOVERRIDE command.



Note: The intensity of light rows is shared among rows 1, 3, 5, 7, 9, 11, 13 and 15 and is shared among rows 0, 2, 4, 6, 8, 10, 12 and 14. For example if rows 0, 2, and 4 are enabled, the intensity of each of the light rows is 1/3 of what it would be if only a single light row were enabled. Or, if light rows 1, 3, 5 and 7 are enabled, the intensity of each of the light rows is 1/4

Setting	Description
Light	Specifies whether Light Mode and Light Power settings will be tried during the tuning process. 0 = Disable Light tuning 1 = Enable Light tuning*
LightMode	Specifies that a pre-programmed Light Mode will be used to illuminate the mark. 0 = All lights are disabled. (1 to 3) = A pair of Light Rows between 6 - 9 are enabled. (4 to 11) = A pair of Light Rows between 0 - 5 or between 10 - 15 are enabled. 13 = External. For use with the In-Sight 1740 series wafer reader's integrated external LEDs, for dark field lighting applications. 14 = Expansion. For use with an add-on light connected to the In-Sight 1740 series wafer reader's Expansion port, for wafer reading applications with unique illumination requirements.
LightPower	The exposure time of the wafer reader's CCD imager, in milliseconds. (0 to 127), 2* Note: The In-Sight 1720 and 1721 wafer readers are designed to work with Light Power values of 0 to 33 and the automated tuning process is optimized to return values within this range. The 1720 and 1721 wafer readers rarely require Light Power greater than 33. Light Power values of 33.17 to 127 are primarily intended for the In-Sight 1722 wafer readers. The 1722 wafer reader uses infrared LEDs that work with certain advanced wafer process coatings. The infrared LEDs do not produce as much light as the red LEDs on the 1721 wafer reader, so higher power settings may be needed. The 1722 wafer reader's automated tuning process will try Light Power values up to 127.
Mark	The type of wafer mark the selected Config will read. 0 = 2D, T7 1 = BC, BC 412 2 = BC, IBM 412 3 = Internal Use Only 4 = Chars, SEMI 5 = Chars, IBM 6 = Chars, Triple 7 = Chars, OCR-A 8 = 2D, Data Matrix 9 = Chars, Chartered 10 = 2D, QR Code 11 = SEMI M1.15
"MatchString"	A text string, which is used as a comparative match during the automated tuning process. The automated tuning process uses the Match String rather than the current Field String or Field Definition settings and returns the Light, Size and Filter settings with the highest read score that matches the string. The Match String setting can be helpful with noisy images (for example, when the mark is scribed over a die pattern, where background noise could potentially produce incorrect read results).
NumFailed	The number of failed reads for a Config.
NumRead	The total number of reads for all Configs.
NumPassed	The number of passed reads for a Config.
NumTuneFailed	The number of failed reads during a tune.

Setting	Description
NumTunePassed	The number of passed reads during a tune.
OnlinePromptToSave	Specifies whether to prompt to save the job file when placing the wafer reader Online. 0 = Disable* 1 = Enable
Order	Specifies how enabled Configs are tried during the retry sequence. 0 = Last Best* 1 = In Order 2 = All
Optimization	The index of the Optimization type for the specified Config. 0 = Mode 0* 1 = Mode 1
Orientation	The index of the Orientation setting for the specified Config. 0 = Normal* 1 = Mirrored horizontally 2 = Flipped vertically 3 = Rotated 180 degrees
Passed	The pass/fail status of the specified Config for the most recent read.
Phi (Region)	The curvature of the Config's region. (-360 to 360)  Note: In-Sight 1750 series wafer readers do not support curved regions. The Phi value is always 0, and any input is ignored.
ReadLog	A record of the most recent read that passed.
Region	The region where the wafer mark is expected to appear in the field of view. Row: 752 x 480 image resolution: (0 to 479) 1024 x 768 image resolution: (0 to 767) High: 752 x 480 image resolution: (4 to 480), 100* 1024 x 768 image resolution: (4 to 480), 100* Col: 752 x 480 image resolution: (0 to 751) 1024 x 768 image resolution: (0 to 1023) Wide: 752 x 480 image resolution: (4 to 752), 600* 1024 x 768 image resolution: (4 to 1024), 600* Theta: (-179 to 180) Phi: (-360 to 360)  Note: In-Sight 1750 series wafer readers do not support curved regions. The Phi value is always 0, and any input is ignored.

Setting	Description
RegionTolerance	Specifies the rotational tolerance of an OCR scribe within a region and whether portions of characters, or entire characters, extending beyond the region can be read. This option is only enabled for OCR mark types. 0 = Off* 1 = Low 2 = Medium 3 = High
RemoveSEMIChecksum	Specifies whether the SEMI checksum is automatically removed from the read result for the specified Config. 0 = Disable* 1 = Enable
Retry	Specifies how the Config is used during the retry sequence. All enabled Configs will be included in the retry sequence. No Config will be tuned during the retry sequence unless all enabled Configs fail to read on the first attempt. 0 = Do Not Tune* 1 = Tune 2 = Tune & Save
Row (Char)	The row coordinate of the indexed character in a string. (4 to 256)
Row (Fixture)	The row coordinate of the fixture. (4 to 480), 190*
Row (Region)	The Row coordinate of the Config's region. 752 x 480 image resolution: (0 to 479) 1024 x 768 image resolution: (0 to 767)
Score	The average response scores for all Configs that passed. The score range depends on the Mark and Checksum that are selected.
ScoreAs100	The scoring method for character positions fielded as dashes (-), dots (.), pound characters (#) and spaces. 0 = Standard * 1 = Score 100 and include 2 = Score 100 and exclude
Size	Specifies whether the mark size will be modified during tuning. 0 = Disable Size tuning 1 = Enable Size tuning*
String	The character results from the read.
StringLen	The number of characters in the string.
StrokeThickness	The index of Stroke Thickness setting for the specified Config. 0 = Normal* 1 = Thick 2 = Thin  Note: In-Sight 1750 series wafer readers do not support stroke thickness. The StrokeThickness value is always 0, and any input is ignored.
Theta (CharSize)	The orientation of a 2D symbol or of the characters in a string, as measured in pixels.

Setting	Description
Theta (Fixture)	The orientation of the fixture. (0 to 359)
Theta (Region)	The angle rotation of the Config's region. (-179 to 180)
<u>Thresh</u>	The character threshold setting. (25 to 100), 25*
Timeout	The total amount of time (in seconds) to allow for a read, including retries and tuning of enabled Configs. (0 to 15000), 30*
TryLog	The Retry status of the Config during the last read.
TuneLog	The Retry and Tune status of the Config during the last read.
Value	The raw response score of the specified character in a read string.
WaferID	References the WaferID data structure in the spreadsheet.
Wide (CharSize)	The width of a 2D symbol or the average width of the characters in a string, as measured in pixels. (8 to 480), 20*
Wide (Region)	The width of the region (along the X-axis) where the wafer mark is expected to appear in the field of view. 752 x 480 image resolution: (4 to 752), 600* 1024 x 768 image resolution: (4 to 1024), 600*
*denotes default value	

AcquireConfig

Acquires an image using the light settings in the specified Config. This command does not read.

Syntax: EV AcquireConfig(WaferID, Config)

AcquireConfig Inputs

[WaferID](#)

[Config](#)

AcquireConfig Outputs

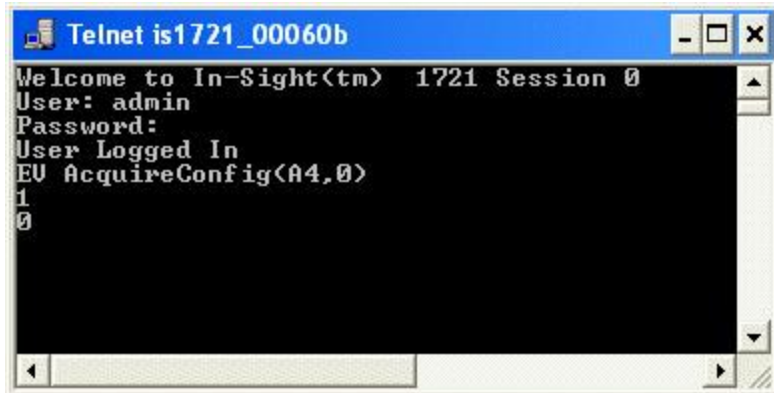
Returns A status code, followed by the number of characters (in bytes) for the results.

The status codes are:

- 1 The command was executed successfully.
- 0 Unrecognized command.
- 2 The command could not be executed.

AcquireConfig Example

To acquire an image using the light setting in Config 0, issue the following command: **EV AcquireConfig(A4,0)**



The In-Sight wafer reader responds with a status code (1), followed by the number of characters, in bytes (0).

AcquireConfigAndRunSheet

Acquires a new image in the WaferID cell in Spreadsheet using the acquisition settings in the config with the specified index. This command does not perform a read.

```
Syntax: EV AcquireConfigAndRunSheet (WaferID, [ConfigIndex])
```

AcquireConfigAndRunSheet Inputs

[WaferID](#)

[Config](#)

AcquireConfigAndRunSheet Outputs

Returns A status code, followed by the number of characters (in bytes) for the results.

The status codes are:

- 1 The command was executed successfully.
- 0 Unrecognized command.
- 2 The command could not be executed.

AcquireConfig Example

To acquire an image using the acquisition settings of Config 0, issue the following command: **EV**

AcquireConfigAndRunSheet(A0,0)

The In-Sight wafer reader responds with a status code (1), followed by the number of characters, in bytes (0).

CopyConfig

Copies all settings and read results from the source Config to the target Config.

Syntax: EV CopyConfig(WaferID,TargetConfig,SourceConfig)

CopyConfig Inputs

[WaferID](#)

[TargetConfig](#)

[SourceConfig](#)

CopyConfig Outputs

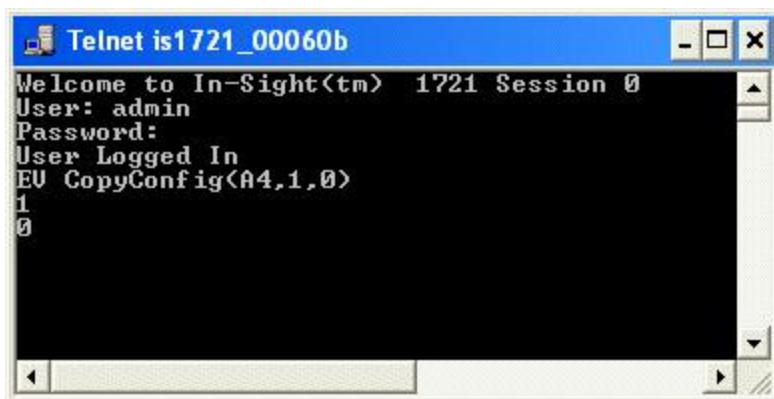
Returns A status code, followed by the number of characters (in bytes) for the results.

The status codes are:

- 1 The command was executed successfully.
- 0 Unrecognized command.
- 2 The command could not be executed.

CopyConfig Example

To copy Config 0's settings and read results to Config 1, issue the following command: **EV CopyConfig(A4,1,0)**



The In-Sight wafer reader responds with a status code (1), followed by the number of characters, in bytes (0).

GetConfig

Returns the settings for the specified Config.

```
Syntax: EV GetConfig(WaferID,Config)
```

GetConfig Inputs

[WaferID](#)

[Config](#)

GetConfig Outputs

Returns A status code, followed by the number of characters (in bytes) for the results, then the results in XML format.

The status codes are:

- 1 The command was executed successfully.
- 0 Unrecognized command.
- 2 The command could not be executed.
- 6 User does not have Full Access to execute the command.

Example XML tag returns

```
<Config Id="0">
  <Mark>4</Mark>
  <Checksum>2</Checksum>
  <LightMode>5</LightMode>
  <LightPower>3.333</LightPower>
  <Orientation>0</Orientation>
  <Region>
    <Row>187</Row>
    <Col>48</Col>
    <High>100</High>
    <Wide>500</Wide>
    <Theta>0</Theta>
    <Phi>0</Phi>
  </Region>
  <CharSize>
    <High>42</High>
    <Wide>21</Wide>
  </CharSize>
  <CharSpacing>1</CharSpacing>
  <FieldString>*****CS</FieldString>
  <FieldDef></FieldDef>
  <Color>1</Color>
  <Accept>50</Accept>
  <Enable>1</Enable>
  <Retry>0</Retry>
  <BComp>0</BComp>
  <Thresh>25</Thresh>
  <Optimization>0</Optimization>
</Config>
```

GetConfig Example

To return the settings for Config 0, issue the following command: **EV GetConfig(A4,0)**

```

Telnet is1721_00060b
Welcome to In-Sight(tm) 1721 Session 0
User: admin
Password:
User Logged In
EU GetConfig(A4,0)
1
643
<Config Id="0">
  <Mark>4</Mark>
  <Checksum>2</Checksum>
  <LightMode>1</LightMode>
  <LightPower>2</LightPower>
  <Orientation>0</Orientation>
  <Region>
    <Row>190</Row>
    <Col>70</Col>
    <High>100</High>
    <Wide>500</Wide>
    <Theta>0</Theta>
    <Phi>0</Phi>
  </Region>
  <CharSize>
    <High>42</High>
    <Wide>20</Wide>
  </CharSize>
  <CharSpacing>1</CharSpacing>
  <FieldString>*****CS</FieldString>
  <FieldDef></FieldDef>
  <Color>0</Color>
  <Accept>50</Accept>
  <Enable>1</Enable>
  <Retry>0</Retry>
  <BComp>0</BComp>
  <Thresh>25</Thresh>
  <Optimization>0</Optimization>
</Config>

```

The In-Sight wafer reader responds with a status code (1), followed by the number of characters, in bytes (643). Finally, the Config's settings are returned.

Other GetConfig Commands

```

GetConfigMark(WaferID,Config)
GetConfigChecksum(WaferID,Config)
GetConfigLightMode(WaferID,Config)
GetConfigLightPower(WaferID,Config)
GetConfigOrientation(WaferID,Config)
GetConfigRegionRow(WaferID,Config)
GetConfigRegionCol(WaferID,Config)
GetConfigRegionHigh(WaferID,Config)
GetConfigRegionWide(WaferID,Config)
GetConfigRegionTheta(WaferID,Config)
GetConfigRegionPhi(WaferID,Config)
GetConfigCharHigh(WaferID,Config)
GetConfigCharWide(WaferID,Config)

```

GetConfigCharSpacing(WaferID,Config)
GetConfigFieldString(WaferID,Config)
GetConfigFieldDef(WaferID,Config)
GetConfigColor(WaferID,Config)
GetConfigAccept(WaferID,Config)
GetConfigEnable(WaferID,Config)
GetConfigRetry(WaferID,Config)
GetConfigBCType(WaferID,Config)
GetConfigThresh(WaferID,Config)
GetConfigOptimization(WaferID,Config)

GetConfigAppendSEMIChkToT7

Returns the Append SEMI Checksum to T7 setting.

```
Syntax: EV GetConfigAppendSEMIChkToT7 (WaferID, Config)
```

GetConfigAppendSEMIChkToT7 Inputs

[WaferID](#)

[Config](#)

GetConfigAppendSEMIChkToT7 Outputs

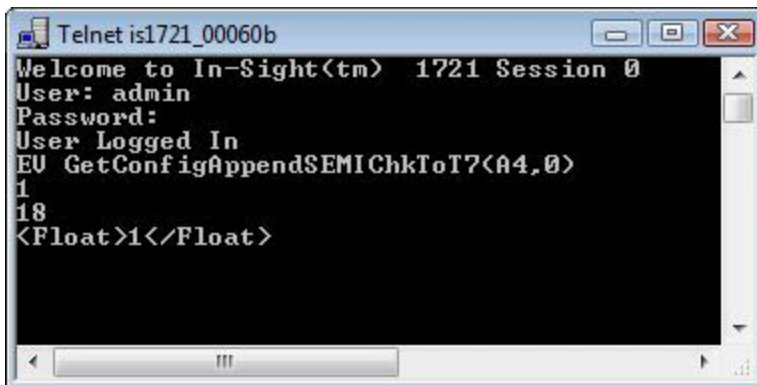
Returns A status code, followed by the number of characters (in bytes) for the results, then the results in XML format.

- 1 The command was executed successfully.
- 0 Unrecognized command.
- 2 The command could not be executed.
- 6 User does not have Full Access to execute the command.

GetConfigAppendSEMIChkToT7 Example

To retrieve the Append SEMI Checksum to T7 setting for Config 0, issue the following command: **EV**

GetConfigAppendSEMIChkToT7(A4,0)



```
Telnet is1721_00060b
Welcome to In-Sight(tm) 1721 Session 0
User: admin
Password:
User Logged In
EV GetConfigAppendSEMIChkToT7(A4,0)
1
18
<Float>1</Float>
```

The In-Sight wafer reader responds with a status code (1), followed by the number of characters, in bytes (18). Finally, the Append SEMI Checksum to T7 setting is returned (1), indicating that the SEMI checksum is automatically calculated and appended to the read result.

GetConfigEx

Returns the settings for the specified Config.

```
Syntax: EV GetConfigEx(WaferID,Config)
```

GetConfigEx Inputs

[WaferID](#)

[Config](#)

GetConfigEx Outputs

Returns A status code, followed by the number of characters (in bytes) for the results, then the results in [XML format](#).

The status codes are:

- 1 The command was executed successfully.
- 0 Unrecognized command.
- 2 The command could not be executed.
- 6 User does not have [Full Access](#) to execute the command.

Example XML tag returns

```
<Config Id="0">
  <Mark>4</Mark>
  <Checksum>2</Checksum>
  <LightMode>5</LightMode>
  <LightPower>3.333</LightPower>
  <Orientation>0</Orientation>
  <Region>
    <Row>187</Row>
    <Col>48</Col>
    <High>100</High>
    <Wide>500</Wide>
    <Theta>0</Theta>
    <Phi>0</Phi>
  </Region>
  <CharSize>
    <High>42</High>
    <Wide>21</Wide>
  </CharSize>
  <CharSpacing>1</CharSpacing>
  <FieldString>*****CS</FieldString>
  <FieldDef></FieldDef>
  <Color>1</Color>
  <Accept>50</Accept>
  <Enable>1</Enable>
  <Retry>0</Retry>
  <BComp>0</BComp>
  <Thresh>25</Thresh>
  <Optimization>0</Optimization>
  <StrokeThickness>0</StrokeThickness>
  <MatchString>0</MatchString>
  <ConfigName>0</ConfigName>
  <LEDOVERRIDE>0</LEDOVERRIDE>
  <LEDOVERRIDEMask>0</LEDOVERRIDEMask>
</Config>
```

GetConfigEx Example

To return the settings for Config 0, issue the following command: **EV GetConfigEx(A4,0)**

```

Telnet is1721_00060b
Welcome to In-Sight(tm) 1721 Session 0
User: admin
Password:
User Logged In
EV GetConfigEx(A4,0)
1
823
<Config Id="0">
  <Mark>4</Mark>
  <Checksum>2</Checksum>
  <LightMode>1</LightMode>
  <LightPower>2</LightPower>
  <Orientation>0</Orientation>
  <Region>
    <Row>190</Row>
    <Col>70</Col>
    <High>100</High>
    <Wide>500</Wide>
    <Theta>0</Theta>
    <Phi>0</Phi>
  </Region>
  <CharSize>
    <High>42</High>
    <Wide>20</Wide>
  </CharSize>
  <CharSpacing>1</CharSpacing>
  <FieldString>*****CS</FieldString>
  <FieldDef></FieldDef>
  <Color>0</Color>
  <Accept>50</Accept>
  <Enable>0</Enable>
  <Retry>0</Retry>
  <BComp>0</BComp>
  <Thresh>25</Thresh>
  <Optimization>0</Optimization>
  <StrokeThickness>0</StrokeThickness>
  <MatchString></MatchString>
  <ConfigName>Config 0</ConfigName>
  <LEDOverride>0</LEDOverride>
  <LEDOverrideMask>0</LEDOverrideMask>
</Config>

```

The In-Sight wafer reader responds with a status code (1), followed by the number of characters, in bytes (823). Finally, the Config's settings are returned.

Other GetConfigEx Commands

```

GetConfigMark(WaferID,Config)
GetConfigChecksum(WaferID,Config)
GetConfigLightMode(WaferID,Config)
GetConfigLightPower(WaferID,Config)
GetConfigOrientation(WaferID,Config)
GetConfigRegionRow(WaferID,Config)

```

GetConfigRegionCol(WaferID,Config)
GetConfigRegionHigh(WaferID,Config)
GetConfigRegionWide(WaferID,Config)
GetConfigRegionTheta(WaferID,Config)
GetConfigRegionPhi(WaferID,Config)
GetConfigCharHigh(WaferID,Config)
GetConfigCharWide(WaferID,Config)
GetConfigCharSpacing(WaferID,Config)
GetConfigFieldString(WaferID,Config)
GetConfigFieldDef(WaferID,Config)
GetConfigColor(WaferID,Config)
GetConfigAccept(WaferID,Config)
GetConfigEnable(WaferID,Config)
GetConfigRetry(WaferID,Config)
GetConfigBCType(WaferID,Config)
GetConfigThresh(WaferID,Config)
GetConfigOptimization(WaferID,Config)

GetConfigStrokeThickness(WaferID,Config)
GetConfigMatchString(WaferID,Config)
GetConfigName(WaferID,Config)
GetConfigLEDOVERRIDE(WaferID,Config)
GetConfigLEDOVERRIDEMask(WaferID,Config)

GetConfigRemoveSEMIChecksum

Returns the Remove SEMI Checksum setting.

```
Syntax: EV GetConfigRemoveSEMIChecksum(WaferID,Config)
```

GetConfigRemoveSEMIChecksum Inputs

[WaferID](#)

[Config](#)

GetConfigRemoveSEMIChecksum Outputs

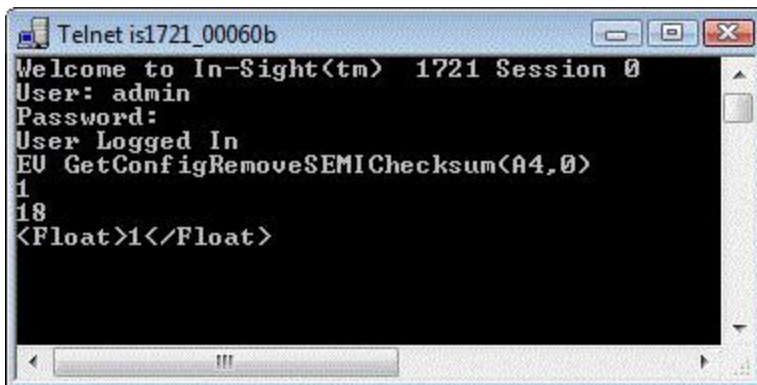
Returns A status code, followed by the number of characters (in bytes) for the results, then the results in XML format.

- 1 The command was executed successfully.
- 0 Unrecognized command.
- 2 The command could not be executed.
- 6 User does not have Full Access to execute the command.

GetConfigRemoveSEMIChecksum Example

To automatically calculate the SEMI checksum and append it to the read result for Config 0, issue the following command:

EV GetConfigRemoveSEMIChecksum(A4,0)



```
Telnet is1721_00060b
Welcome to In-Sight(tm) 1721 Session 0
User: admin
Password:
User Logged In
EV GetConfigRemoveSEMIChecksum(A4,0)
1
18
<Float>1</Float>
```

The In-Sight wafer reader responds with a status code (1), followed by the number of characters, in bytes (18). Finally, the Remove SEMI Checksum setting is returned (1), indicating that the SEMI checksum was automatically removed from the read result.

GetConfigResult

Returns the results of the most recent read performed by the specified Config.

```
Syntax: EV GetConfigResult (WaferID, Config)
```

GetConfigResult Inputs

[WaferID](#)

[Config](#)

GetConfigResult Outputs

Returns A status code, followed by the number of characters (in bytes) for the results, then the results in XML format.

The status codes are:

- 1 The command was executed successfully.
- 0 Unrecognized command.
- 2 The command could not be executed.
- 6 User does not have Full Access to execute the command.

GetConfigResult Example

To return the most recent read results for Config 0, issue the following command: **EV GetConfigResult(A4,0)**

```

Telnet is1721_00060b
Welcome to In-Sight(tm) 1721 Session 1
User: admin
Password:
User Logged In
EU GetConfigResult(A4,0)
1
2144
<Result Id="0">
  <Passed>1</Passed>
  <String>XE850FA4CGC0</String>
  <Score>397</Score>
  <TryLog>0</TryLog>
  <TuneLog>0</TuneLog>
  <LastRead>10</LastRead>
  <NumPassed>0</NumPassed>
  <NumFailed>0</NumFailed>
  <NumTunedPassed>0</NumTunedPassed>
  <NumTuneFailed>0</NumTuneFailed>
  <AvgScore>97</AvgScore>
  <High>41</High>
  <Wide>22</Wide>
  <Theta>0</Theta>
  <BaseScore>97</BaseScore>
  <ExecutionSeconds>0.185</ExecutionSeconds>
  <Orc Id="0">
    <Char>87</Char>
    <Value>92</Value>
    <Row>41</Row>
    <Col>29</Col>
  </Orc>
  <Orc Id="1">
    <Char>80</Char>
    <Value>97</Value>
    <Row>41</Row>
    <Col>70</Col>
  </Orc>

```

Note: <Orc Id="2"> through <Orc Id="11"> have been omitted from the graphic for clarity.

The In-Sight wafer reader responds with a status code (1), followed by the number of characters, in bytes (**2144**). Finally, the specified Config's most recent read results are returned.

Other GetConfigResult Commands

```

GetConfigString(WaferID,Config)
GetConfigStringLen(WaferID,Config)
GetConfigScore(WaferID,Config)
GetConfigPassed(WaferID,Config)
GetConfigHigh(WaferID,Config)
GetConfigWide(WaferID,Config)
GetConfigTheta(WaferID,Config)
GetConfigLastRead(WaferID,Config)
GetConfigNumPassed(WaferID,Config)
GetConfigNumFailed(WaferID,Config)
GetConfigNumTunePassed(WaferID,Config)
GetConfigNumTuneFailed(WaferID,Config)
GetConfigAvgScore(WaferID,Config)
GetConfigTryLog(WaferID,Config)
GetConfigTuneLog(WaferID,Config)

```

GetConfigTune

Returns the settings used during tuning.

```
Syntax: EV GetConfigTune (WaferID, Config)
```

GetConfigTune Inputs

[WaferID](#)

[Config](#)

GetConfigTune Outputs

Returns A status code, followed by the number of characters (in bytes) for the results, then the results in XML format.

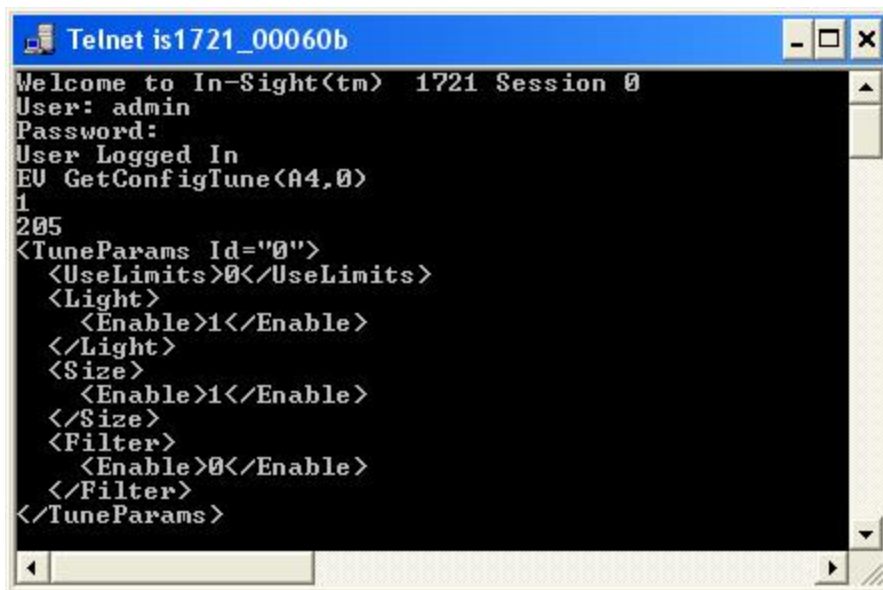
The status codes are:

- 1 The command was executed successfully.
- 0 Unrecognized command.
- 2 The command could not be executed.

GetConfigTune Example

Default Tuning

To retrieve the tuning settings for Config 0, issue the following command: **EV GetConfigTune(A4,0)**

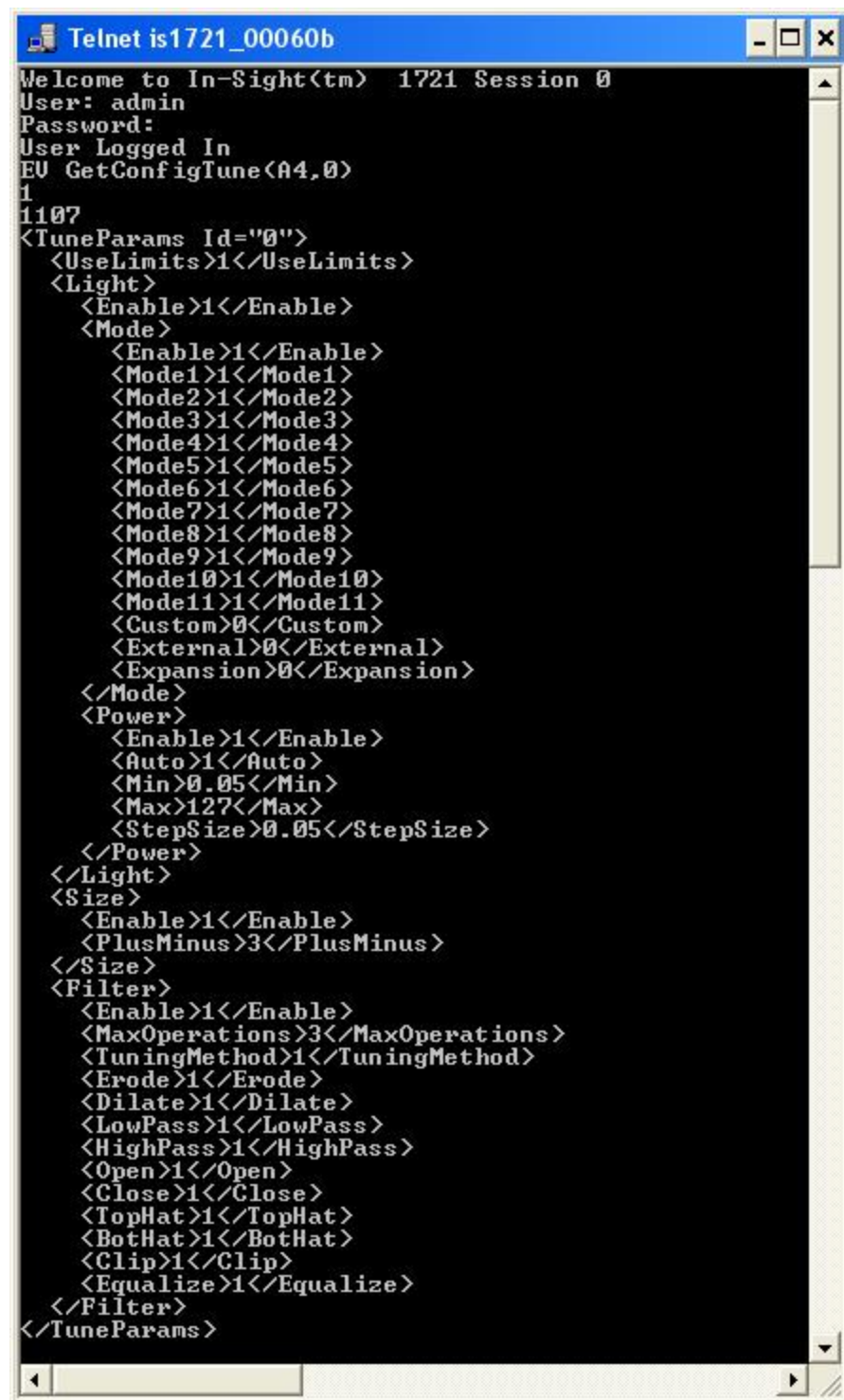


```
Telnet is1721_00060b
Welcome to In-Sight(tm) 1721 Session 0
User: admin
Password:
User Logged In
EV GetConfigTune(A4,0)
1
205
<TuneParams Id="0">
  <UseLimits>0</UseLimits>
  <Light>
    <Enable>1</Enable>
  </Light>
  <Size>
    <Enable>1</Enable>
  </Size>
  <Filter>
    <Enable>0</Enable>
  </Filter>
</TuneParams>
```

The In-Sight wafer reader responds with a status code (1), followed by the number of characters, in bytes (205). Finally, the tuning settings are returned, indicating that default tuning is enabled, and Light and Size settings are tried during tuning.

Custom Tuning

To retrieve the tuning settings for Config 0, issue the following command: `EV GetConfigTune(A4,0)`



```

Welcome to In-Sight(tm) 1721 Session 0
User: admin
Password:
User Logged In
EV GetConfigTune(A4,0)
1
1107
<TuneParams Id="0">
  <UseLimits>1</UseLimits>
  <Light>
    <Enable>1</Enable>
    <Mode>
      <Enable>1</Enable>
      <Mode1>1</Mode1>
      <Mode2>1</Mode2>
      <Mode3>1</Mode3>
      <Mode4>1</Mode4>
      <Mode5>1</Mode5>
      <Mode6>1</Mode6>
      <Mode7>1</Mode7>
      <Mode8>1</Mode8>
      <Mode9>1</Mode9>
      <Mode10>1</Mode10>
      <Mode11>1</Mode11>
      <Custom>0</Custom>
      <External>0</External>
      <Expansion>0</Expansion>
    </Mode>
    <Power>
      <Enable>1</Enable>
      <Auto>1</Auto>
      <Min>0.05</Min>
      <Max>127</Max>
      <StepSize>0.05</StepSize>
    </Power>
  </Light>
  <Size>
    <Enable>1</Enable>
    <PlusMinus>3</PlusMinus>
  </Size>
  <Filter>
    <Enable>1</Enable>
    <MaxOperations>3</MaxOperations>
    <TuningMethod>1</TuningMethod>
    <Erode>1</Erode>
    <Dilate>1</Dilate>
    <LowPass>1</LowPass>
    <HighPass>1</HighPass>
    <Open>1</Open>
    <Close>1</Close>
    <TopHat>1</TopHat>
    <BotHat>1</BotHat>
    <Clip>1</Clip>
    <Equalize>1</Equalize>
  </Filter>
</TuneParams>

```

The In-Sight wafer reader responds with a status code (1), followed by the number of characters, in bytes (**1107**). Finally, the tuning settings are returned, indicating that custom tuning limits are enabled, and the custom Light Mode, Light Power, Size and Filter settings that are tried during tuning.

Related Topics

[SetConfigTune](#)

GetFilterList

Returns a list of all filter names.

```
Syntax: EV GetFilterList (WaferID)
```

GetFilterList Inputs

[WaferID](#)

GetFilterList Outputs

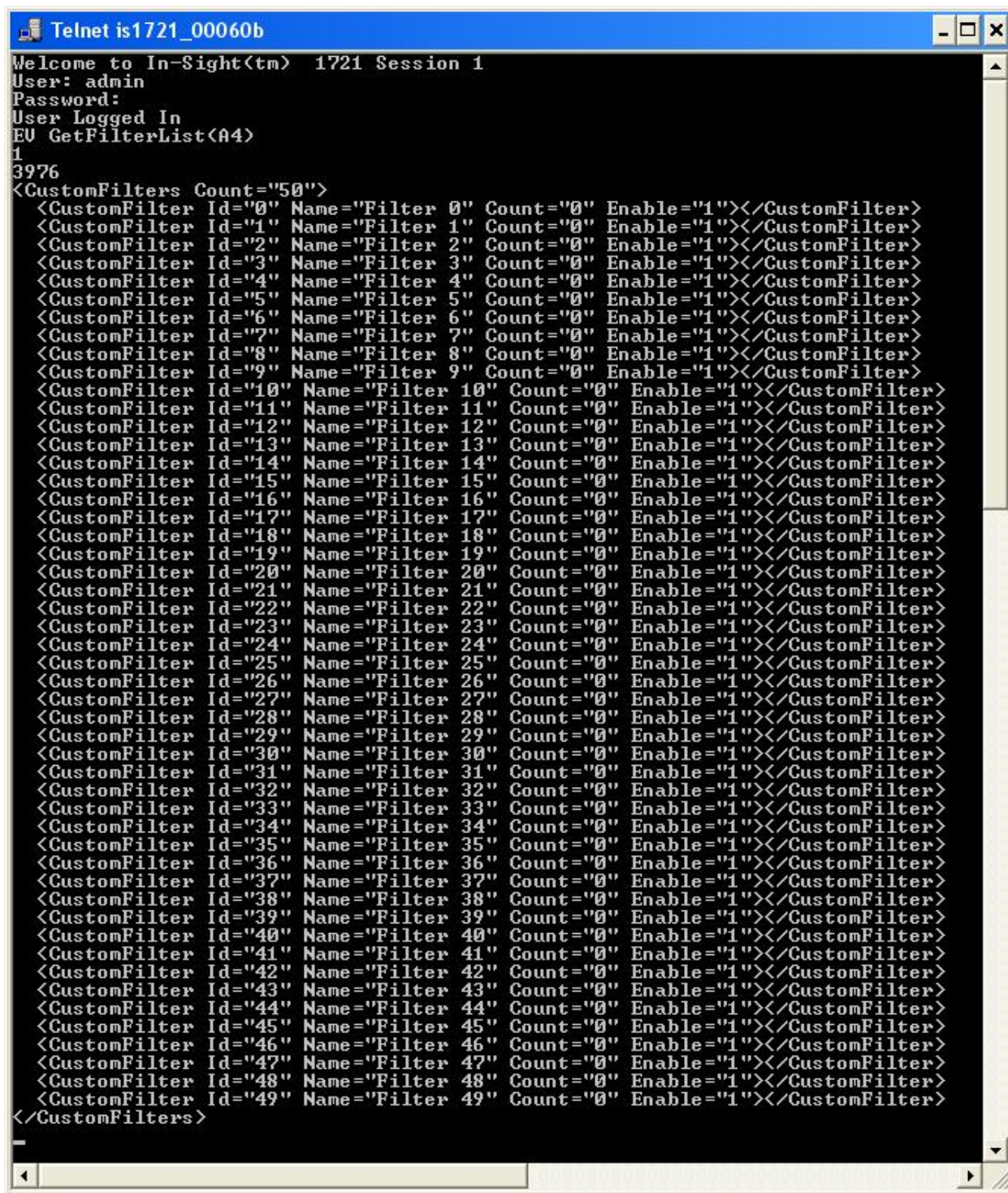
Returns A status code, followed by the number of characters (in bytes) for the results, then the results in XML format.

The status codes are:

- 1 The command was executed successfully.
- 0 Unrecognized command.
- 2 The command could not be executed.
- 6 User does not have Full Access to execute the command.

GetFilterList Example

To return the list of filter names, issue the following command: **EV GetFilterList(A4)**



```

Telnet is1721_00060b
Welcome to In-Sight(tm) 1721 Session 1
User: admin
Password:
User Logged In
EU GetFilterList(A4)
1
3976
<CustomFilters Count="50">
  <CustomFilter Id="0" Name="Filter 0" Count="0" Enable="1"></CustomFilter>
  <CustomFilter Id="1" Name="Filter 1" Count="0" Enable="1"></CustomFilter>
  <CustomFilter Id="2" Name="Filter 2" Count="0" Enable="1"></CustomFilter>
  <CustomFilter Id="3" Name="Filter 3" Count="0" Enable="1"></CustomFilter>
  <CustomFilter Id="4" Name="Filter 4" Count="0" Enable="1"></CustomFilter>
  <CustomFilter Id="5" Name="Filter 5" Count="0" Enable="1"></CustomFilter>
  <CustomFilter Id="6" Name="Filter 6" Count="0" Enable="1"></CustomFilter>
  <CustomFilter Id="7" Name="Filter 7" Count="0" Enable="1"></CustomFilter>
  <CustomFilter Id="8" Name="Filter 8" Count="0" Enable="1"></CustomFilter>
  <CustomFilter Id="9" Name="Filter 9" Count="0" Enable="1"></CustomFilter>
  <CustomFilter Id="10" Name="Filter 10" Count="0" Enable="1"></CustomFilter>
  <CustomFilter Id="11" Name="Filter 11" Count="0" Enable="1"></CustomFilter>
  <CustomFilter Id="12" Name="Filter 12" Count="0" Enable="1"></CustomFilter>
  <CustomFilter Id="13" Name="Filter 13" Count="0" Enable="1"></CustomFilter>
  <CustomFilter Id="14" Name="Filter 14" Count="0" Enable="1"></CustomFilter>
  <CustomFilter Id="15" Name="Filter 15" Count="0" Enable="1"></CustomFilter>
  <CustomFilter Id="16" Name="Filter 16" Count="0" Enable="1"></CustomFilter>
  <CustomFilter Id="17" Name="Filter 17" Count="0" Enable="1"></CustomFilter>
  <CustomFilter Id="18" Name="Filter 18" Count="0" Enable="1"></CustomFilter>
  <CustomFilter Id="19" Name="Filter 19" Count="0" Enable="1"></CustomFilter>
  <CustomFilter Id="20" Name="Filter 20" Count="0" Enable="1"></CustomFilter>
  <CustomFilter Id="21" Name="Filter 21" Count="0" Enable="1"></CustomFilter>
  <CustomFilter Id="22" Name="Filter 22" Count="0" Enable="1"></CustomFilter>
  <CustomFilter Id="23" Name="Filter 23" Count="0" Enable="1"></CustomFilter>
  <CustomFilter Id="24" Name="Filter 24" Count="0" Enable="1"></CustomFilter>
  <CustomFilter Id="25" Name="Filter 25" Count="0" Enable="1"></CustomFilter>
  <CustomFilter Id="26" Name="Filter 26" Count="0" Enable="1"></CustomFilter>
  <CustomFilter Id="27" Name="Filter 27" Count="0" Enable="1"></CustomFilter>
  <CustomFilter Id="28" Name="Filter 28" Count="0" Enable="1"></CustomFilter>
  <CustomFilter Id="29" Name="Filter 29" Count="0" Enable="1"></CustomFilter>
  <CustomFilter Id="30" Name="Filter 30" Count="0" Enable="1"></CustomFilter>
  <CustomFilter Id="31" Name="Filter 31" Count="0" Enable="1"></CustomFilter>
  <CustomFilter Id="32" Name="Filter 32" Count="0" Enable="1"></CustomFilter>
  <CustomFilter Id="33" Name="Filter 33" Count="0" Enable="1"></CustomFilter>
  <CustomFilter Id="34" Name="Filter 34" Count="0" Enable="1"></CustomFilter>
  <CustomFilter Id="35" Name="Filter 35" Count="0" Enable="1"></CustomFilter>
  <CustomFilter Id="36" Name="Filter 36" Count="0" Enable="1"></CustomFilter>
  <CustomFilter Id="37" Name="Filter 37" Count="0" Enable="1"></CustomFilter>
  <CustomFilter Id="38" Name="Filter 38" Count="0" Enable="1"></CustomFilter>
  <CustomFilter Id="39" Name="Filter 39" Count="0" Enable="1"></CustomFilter>
  <CustomFilter Id="40" Name="Filter 40" Count="0" Enable="1"></CustomFilter>
  <CustomFilter Id="41" Name="Filter 41" Count="0" Enable="1"></CustomFilter>
  <CustomFilter Id="42" Name="Filter 42" Count="0" Enable="1"></CustomFilter>
  <CustomFilter Id="43" Name="Filter 43" Count="0" Enable="1"></CustomFilter>
  <CustomFilter Id="44" Name="Filter 44" Count="0" Enable="1"></CustomFilter>
  <CustomFilter Id="45" Name="Filter 45" Count="0" Enable="1"></CustomFilter>
  <CustomFilter Id="46" Name="Filter 46" Count="0" Enable="1"></CustomFilter>
  <CustomFilter Id="47" Name="Filter 47" Count="0" Enable="1"></CustomFilter>
  <CustomFilter Id="48" Name="Filter 48" Count="0" Enable="1"></CustomFilter>
  <CustomFilter Id="49" Name="Filter 49" Count="0" Enable="1"></CustomFilter>
</CustomFilters>

```

The In-Sight wafer reader responds with a status code (1), followed by the number of characters, in bytes (3976). Finally, the list of filters is returned.

GetWaferAll

Returns all results for the most recent read, and settings for all Configs.

```
Syntax: EV GetWaferAll (WaferID)
```

Note: The GetWaferAll command is for backwards compatibility only; use the [GetWaferAllEx](#) command for all 1720 and 1740 series wafer readers running firmware version 3.2.5 and higher.

GetWaferAll Inputs

[WaferID](#)

GetWaferAll Outputs

Returns A status code, followed by the number of characters (in bytes) for the results, then the results in XML format.

The status codes are:

- 1 The command was executed successfully.
- 0 Unrecognized command.
- 2 The command could not be executed.
- 6 User does not have Full Access to execute the command.

GetWaferAll Example

To return the results for the most recent read and settings for all Configs, issue the following command: **EV GetWaferAll(A4)**

```

Telnet is1721_00060b
Welcome to In-Sight(tm) 1721 Session 0
User: admin
Password:
User Logged In
EU GetWaferAll(A4)
1
26528
<Wafer>
  <BestConfig>0</BestConfig>
  <Timeout>30</Timeout>
  <Order>0</Order>
  <AcqEnable>1</AcqEnable>
  <NumPassed>16</NumPassed>
  <NumFailed>1</NumFailed>
  <AvgScore>97</AvgScore>
  <Fixture>
    <Row>0</Row>
    <Col>0</Col>
    <Theta>0</Theta>
  </Fixture>
  <Config Id="0">
    <Mark>4</Mark>
    <Checksum>2</Checksum>
    <LightMode>3</LightMode>
    <LightPower>0.25</LightPower>
    <Orientation>0</Orientation>
    <Region>
      <Row>230</Row>
      <Col>225.199997</Col>
      <High>100</High>
      <Wide>669.599976</Wide>
      <Theta>0</Theta>
      <Phi>0</Phi>
    </Region>
    <CharSize>
      <High>42</High>
      <Wide>20</Wide>
    </CharSize>
    <CharSpacing>1</CharSpacing>
    <FieldString>*****</FieldString>
    <FieldDef></FieldDef>
    <Color>0</Color>
    <Accept>50</Accept>
    <Enable>1</Enable>
    <Retry>0</Retry>
    <BComp>0</BComp>
    <Thresh>25</Thresh>
    <Optimization>0</Optimization>
  </Config>
  <Result Id="0">
    <Passed>1</Passed>
    <String>XE850FA4CGC0</String>
    <Score>397</Score>
    <TryLog>1</TryLog>
    <TuneLog>0</TuneLog>
    <LastRead>10</LastRead>
    <NumPassed>16</NumPassed>
    <NumFailed>1</NumFailed>
    <NumTunedPassed>0</NumTunedPassed>
    <NumTuneFailed>0</NumTuneFailed>
    <AvgScore>97</AvgScore>
    <High>41</High>
    <Wide>20</Wide>
    <Theta>0</Theta>
    <Orc Id="0">
      <Char>87</Char>
      <Value>84</Value>
      <Row>52</Row>
      <Col>29</Col>
    </Orc>
    <Orc Id="1">
      <Char>80</Char>
      <Value>85</Value>
      <Row>53</Row>
      <Col>70</Col>
    </Orc>
  </Result Id="0">

```

Note: <Orc Id ="2"> through <Orc Id ="11"> have been omitted from the graphic for clarity.

The In-Sight wafer reader responds with a status code (1), followed by the number of characters, in bytes (**26528**). Finally, all results for the most recent read and the settings for all Configs are returned.

Other GetWaferAll Commands

GetWaferTimeout(WaferID)

GetWaferOrder(WaferID)

GetWaferAcqEnable(WaferID)

GetWaferBestConfig(WaferID)

GetWaferFixRow(WaferID)

GetWaferFixCol(WaferID)

GetWaferFixTheta(WaferID)

GetWaferNumPassed(WaferID)

GetWaferNumFailed(WaferID)

GetWaferAvgScore(WaferID)

GetWaferResponse(WaferID)

GetWaferAllEx

Returns all results for the most recent read, and settings for all Configs.

```
Syntax: EV GetWaferAllEx(WaferID)
```

Note: The [GetWaferAll](#) command is for backwards compatibility only; use the GetWaferAllEx command for all 1750 series wafer readers.

GetWaferAllEx Inputs

[WaferID](#)

GetWaferAllEx Outputs

Returns A status code, followed by the number of characters (in bytes) for the results, then the results in XML format.

The status codes are:

- 1 The command was executed successfully.
- 0 Unrecognized command.
- 2 The command could not be executed.
- 6 User does not have Full Access to execute the command.

GetWaferAllEx Example

To return the results for the most recent read and settings for all Configs, issue the following command: **EV GetWaferAllEx (A4)**

```

Telnet is1721_00060b
Welcome to In-Sight(tm) 1721 Session 0
User: admin
Password:
User Logged In
EU GetWaferAllEx(A4)
1
26494
<Wafer>
  <BestConfig>0</BestConfig>
  <Timeout>30</Timeout>
  <Order>0</Order>
  <AcqEnable>1</AcqEnable>
  <NumPassed>16</NumPassed>
  <NumFailed>1</NumFailed>
  <AvgScore>97</AvgScore>
  <ScoreAs100>0</ScoreAs100>
  <Config Id="0">
    <Mark>4</Mark>
    <Checksum>2</Checksum>
    <LightMode>3</LightMode>
    <LightPower>0.25</LightPower>
    <Orientation>0</Orientation>
    <Region>
      <Row>230</Row>
      <Col>225.199997</Col>
      <High>100</High>
      <Wide>669.599976</Wide>
      <Theta>0</Theta>
      <Phi>0</Phi>
    </Region>
    <CharSize>
      <High>42</High>
      <Wide>20</Wide>
    </CharSize>
    <CharSpacing>1</CharSpacing>
    <FieldString>*****</FieldString>
    <FieldDef></FieldDef>
    <Color>0</Color>
    <Accept>50</Accept>
    <Enable>1</Enable>
    <Retry>0</Retry>
    <BComp>0</BComp>
    <Thresh>25</Thresh>
    <Optimization>0</Optimization>
    <StrokeThickness>0</StrokeThickness>
    <MatchString>XE850FA4CGC0</MatchString>
    <ConfigName>Config 0</ConfigName>
    <LEDOverride>0</LEDOverride>
    <LEDOverrideMask>0</LEDOverrideMask>
  </Config>
  <Result Id="0">
    <Passed>1</Passed>
    <String>XE850FA4CGC0</String>
    <Score>397</Score>
    <TryLog>1</TryLog>
    <TuneLog>0</TuneLog>
    <LastRead>10</LastRead>
    <NumPassed>16</NumPassed>
    <NumFailed>1</NumFailed>
    <NumTunedPassed>0</NumTunedPassed>
    <NumTuneFailed>0</NumTuneFailed>
    <AvgScore>97</AvgScore>
    <High>41</High>
    <Wide>20</Wide>
    <Theta>0</Theta>
    <BaseScore>97</BaseScore>
    <ExecutionSeconds>0.267</ExecutionSeconds>
    <Orc Id="0">
      <Char>87</Char>
      <Value>84</Value>
      <Row>52</Row>
      <Col>29</Col>
    </Orc>
    <Orc Id="1">
      <Char>80</Char>
      <Value>85</Value>
      <Row>53</Row>
      <Col>70</Col>
    </Orc>
  </Result Id="0">

```

Note: <Orc Id ="2"> through <Orc Id ="11"> have been omitted from the graphic for clarity.

The In-Sight wafer reader responds with a status code (1), followed by the number of characters, in bytes (**26494**). Finally, all results for the most recent read and the settings for all Configs are returned.

Other GetWafer Commands

GetWaferTimeout(WaferID)

GetWaferOrder(WaferID)

GetWaferAcqEnable(WaferID)

GetWaferBestConfig(WaferID)

GetWaferFixRow(WaferID)

GetWaferFixCol(WaferID)

GetWaferFixTheta(WaferID)

GetWaferNumPassed(WaferID)

GetWaferNumFailed(WaferID)

GetWaferAvgScore(WaferID)

GetWaferResponse(WaferID)

GetWaferScoreAs100(WaferID)

GetWaferCompatibilityMode

Returns the Compatibility Mode setting.

```
Syntax: EV GetWaferCompatibilityMode()
```

GetWaferCompatibilityMode Inputs

None.

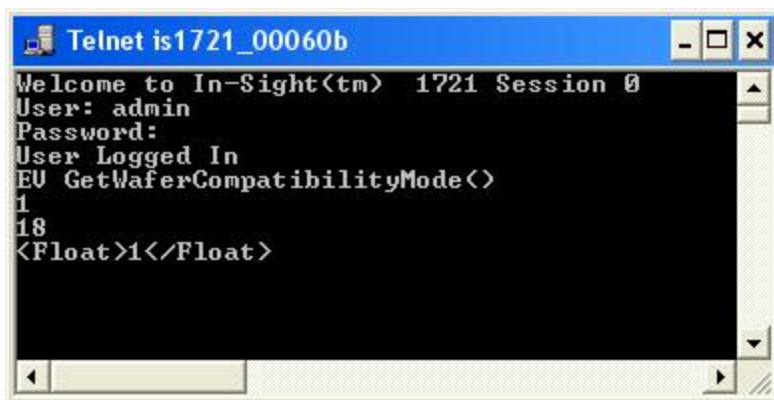
GetWaferCompatibilityMode Outputs

Returns A status code, followed by the number of characters (in bytes) for the results, then the results in XML format.

- 1 The command was executed successfully.
- 0 Unrecognized command.
- 2 The command could not be executed.
- 6 User does not have Full Access to execute the command.

GetWaferCompatibilityMode Example

To retrieve the Compatibility Mode setting, issue the following command: **EV GetWaferCompatibilityMode()**



The In-Sight wafer reader responds with a status code (1), followed by the number of characters, in bytes (18). Finally, the Compatibility Mode setting is returned (1), indicating Mode 1.

GetWaferFailCharsOutRegion

Returns the Fail Characters Outside of Region state of the wafer reader. This setting is only enabled for OCR mark types.

Syntax: EV GetWaferFailCharsOutRegion (WaferID)

GetWaferFailCharsOutRegion Inputs

[WaferID](#)

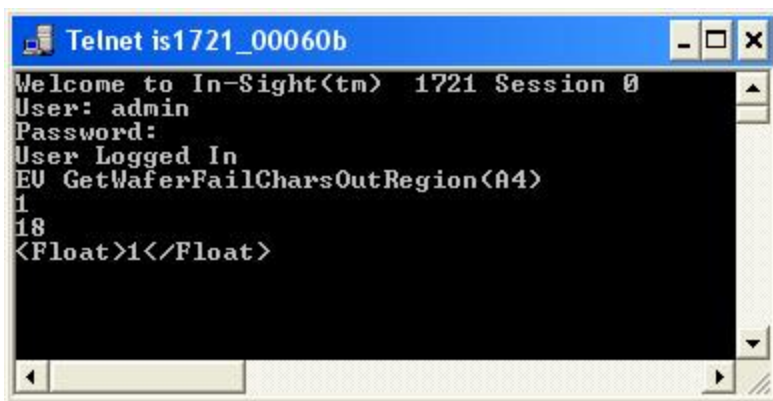
GetWaferFailCharsOutRegion Outputs

Returns A status code, followed by the number of characters (in bytes) for the results, then the results in XML format.

- 1 The command was executed successfully.
- 0 Unrecognized command.
- 2 The command could not be executed.
- 6 User does not have Full Access to execute the command.

GetWaferFailCharsOutRegion Example

To retrieve the Fail Characters Outside of Region setting, issue the following command: **EV GetWaferFailCharsOutRegion (A4)**



```
Telnet is1721_00060b
Welcome to In-Sight(tm) 1721 Session 0
User: admin
Password:
User Logged In
EV GetWaferFailCharsOutRegion(A4)
1
18
<Float>1</Float>
```

The In-Sight wafer reader responds with a status code (1), followed by the number of characters, in bytes (18). Finally, the Fail Characters Outside of Region setting is returned (1), indicating that the setting is enabled.

GetWaferRegionTolerance

Returns the Region Tolerance state of the wafer reader. This setting is only enabled for OCR mark types.

Syntax: EV GetWaferRegionTolerance (WaferID)

GetWaferRegionTolerance Inputs

[WaferID](#)

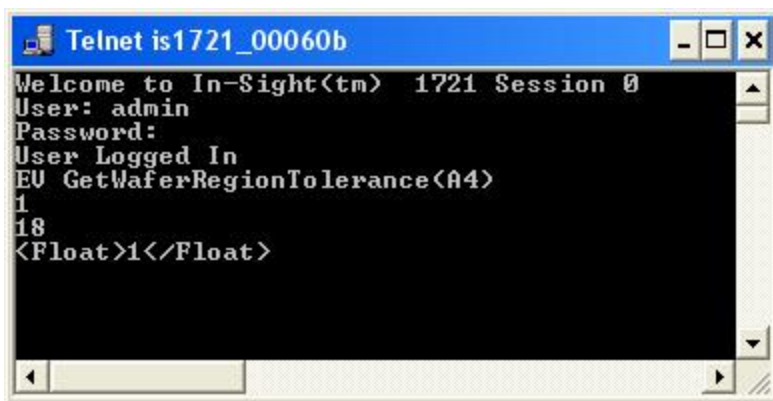
GetWaferRegionTolerance Outputs

Returns A status code, followed by the number of characters (in bytes) for the results, then the results in XML format.

- 1 The command was executed successfully.
- 0 Unrecognized command.
- 2 The command could not be executed.
- 6 User does not have Full Access to execute the command.

GetWaferRegionTolerance Example

To determine the Region Tolerance state of the wafer reader, issue the following command: **EV GetWaferRegionTolerance (A4)**



```
Telnet is1721_00060b
Welcome to In-Sight(tm) 1721 Session 0
User: admin
Password:
User Logged In
EV GetWaferRegionTolerance(A4)
1
18
<Float>1</Float>
```

The In-Sight wafer reader responds with a status code (1), followed by the number of characters, in bytes (18). Finally, the Region Tolerance state is returned (1), indicating that it is set to **Low**.

GetWaferSummary

Returns a summary of the most recent read results.

```
Syntax: EV GetWaferSummary (WaferID)
```

GetWaferSummary Inputs

[WaferID](#)

GetWaferSummary Outputs

Returns A status code, followed by the number of characters (in bytes) for the results, then the results in XML format.

The status codes are:

- 1 The command was executed successfully.
- 0 Unrecognized command.
- 2 The command could not be executed.
- 6 User does not have Full Access to execute the command.

GetWaferSummary Example

To return a summary of the most recent read results, issue the following command: **EV GetWaferSummary(A4)**



```

Telnet is1721_00060b
Welcome to In-Sight(tm) 1721 Session 0
User: admin
Password:
User Logged In
EU GetWaferSummary(A4)
1
1265
<WaferSummary>
  <Fixture>
    <Row>0</Row>
    <Col>0</Col>
    <Theta>0</Theta>
  </Fixture>
  <BestConfig>0</BestConfig>
  <Passed>1</Passed>
  <String>XE850FA4CGC0</String>
  <Score>397</Score>
  <Region>
    <Row>230</Row>
    <Col>225.199997</Col>
    <High>100</High>
    <Wide>669.599976</Wide>
    <Theta>0</Theta>
    <Phi>0</Phi>
  </Region>
  <NumPassed>16</NumPassed>
  <NumFailed>1</NumFailed>
  <AvgScore>97</AvgScore>
  <ReadLog></ReadLog>
  <ConfigLog Id="0">
    <Passed>1</Passed>
    <String>XE850FA4CGC0</String>
    <Score>397</Score>
    <TryLog>1</TryLog>
    <TuneLog>0</TuneLog>
    <NumPassed>16</NumPassed>
    <NumFailed>1</NumFailed>
    <NumTunedPassed>0</NumTunedPassed>
    <NumTuneFailed>0</NumTuneFailed>
    <AvgScore>97</AvgScore>
  </ConfigLog>
</WaferSummary>

```

The In-Sight wafer reader responds with a status code (1), followed by the number of characters, in bytes (1265). Finally, a summary of the most recent read results are returned.

ReadConfig

Reads using the specified Config.

```
Syntax: EV ReadConfig(WaferID,Config,AcquireFlag)
```

ReadConfig Inputs

[WaferID](#)

[Config](#)

[AcquireFlag](#)

Notes:

- To execute a read of all enabled Configs, set the Config input to "-1".
- If acquisition is disabled on the wafer reader (by unchecking the *Acquire new image on read* checkbox or by issuing the [SetWaferAcqEnable](#) command), the AcquireFlag input is ignored.

ReadConfig Outputs

Returns A status code, followed by the number of characters (in bytes) for the results, then the results in XML format.

The status codes are:

- 1 The command was executed successfully.
- 0 Unrecognized command.
- 2 The command could not be executed.
- 6 User does not have Full Access to execute the command.

ReadConfig Example

To read Config 0, issue the following command: **EV ReadConfig(A4,0,0)**

```

Telnet is1721_00060b
Welcome to In-Sight(tm) 1721 Session 1
User: admin
Password:
User Logged In
EU ReadConfig(A4,0,0)
1
2153
<Result Id="0">
  <Passed>1</Passed>
  <String>XE850FA4CGC0</String>
  <Score>397</Score>
  <TryLog>0</TryLog>
  <TuneLog>0</TuneLog>
  <LastRead>10</LastRead>
  <NumPassed>24</NumPassed>
  <NumFailed>0</NumFailed>
  <NumTunedPassed>0</NumTunedPassed>
  <NumTuneFailed>0</NumTuneFailed>
  <AvgScore>96.294121</AvgScore>
  <High>40</High>
  <Wide>21</Wide>
  <Theta>0</Theta>
  <BaseScore>97</BaseScore>
  <ExecutionSeconds>0.186</ExecutionSeconds>
  <Orc Id="0">
    <Char>87</Char>
    <Value>94</Value>
    <Row>51</Row>
    <Col>29</Col>
  </Orc>
  <Orc Id="1">
    <Char>80</Char>
    <Value>96</Value>
    <Row>50</Row>
    <Col>70</Col>
  </Orc>

```

Note: <Orc Id="2"> through <Orc Id="11"> have been omitted from the graphic for clarity.

The In-Sight wafer reader responds with a status code (1), followed by the number of characters, in bytes (2153). Finally, the specified Config's read results are returned.

Other ReadConfig Commands

GetConfigOrcChar(WaferID,Config,[Char Index](#))

GetConfigOrcRow(WaferID,Config,Char Index)

GetConfigOrcCol(WaferID,Config,Char Index)

GetConfigOrcVal(WaferID,Config,Char Index)

ResetStats

Reset all accumulated read statistics.

Syntax: EV ResetStats(WaferID)

ResetStats Inputs

[WaferID](#)

ResetStats Outputs

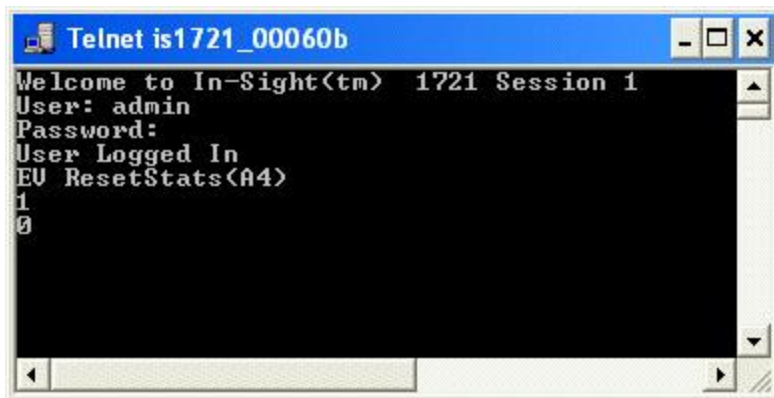
Returns A status code, followed by the number of characters (in bytes) for the results.

The status codes are:

- 1 The command was executed successfully.
- 0 Unrecognized command.
- 2 The command could not be executed.

ResetStats Example

To reset all accumulated read statistics, issue the following command: **EV ResetStats(A4)**



The In-Sight wafer reader responds with a status code (1), followed by the number of characters, in bytes (0).

SetConfig

Specifies all settings for the specified Config.

```
Syntax: EV SetConfig  
(WaferID,Config,Mark,Checksum,LightMode,LightPower,Orientation,Row,Col,High,Wide,Theta,Phi,High,Wide,CharSpacing,"FieldString",  
"FieldDef",Color,Accept,Enable,Retry,BComp,Thresh)
```

SetConfig Inputs

[WaferID](#)

[Config](#)

[Mark](#)

[Checksum](#)

[LightMode](#)

[LightPower](#)

[Orientation](#)

[Row \(Region\)](#)

[Col \(Region\)](#)

[High \(Region\)](#)

[Wide \(Region\)](#)

[Theta \(Region\)](#)

[Phi \(Region\)](#)

[High \(CharSize\)](#)

[Wide \(CharSize\)](#)

[CharSpacing](#)

["FieldString"](#)

["FieldDef"](#)

[Color](#)

[Accept](#)

[Enable](#)

[Retry](#)

[BComp](#)

[Thresh](#)

SetConfig Outputs

Returns A status code, followed by the number of characters (in bytes) for the results.

The status codes are:

- 1 The command was executed successfully.
- 0 Unrecognized command.
- 2 The command could not be executed.

SetConfig Example

To specify the following settings for Config 0:

Mark Type: Chars, SEMI

Checksum: SEMI with Virtual

LightMode: The mark will appear dark on a light background.

LightPower: 2.000

Orientation: Normal

Row (Region): 280

Col (Region): 45

High (Region): 100

Wide (Region): 500

Theta (Region): 0

Phi (Region): 0

High (Charsize): 42

Wide (Charsize): 22

Charspacing: Standard

FieldString: *****CS

FieldDef: "" (blank)

Color: Black

Accept: 50

Enable: Enable

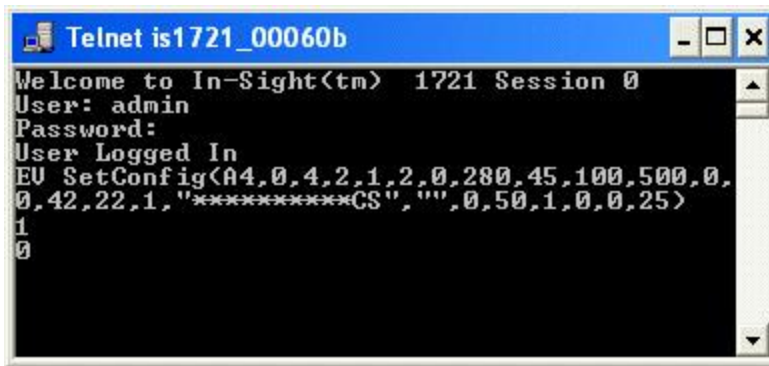
Retry: Disable

BComp: 0

Thresh: 25

Issue the following command:

EV SetConfig(A4,0,4,2,1,2,0,280,45,100,500,0,0,42,22,1,"***CS","",0,50,1,0,0,25)**



```
Telnet is1721_00060b
Welcome to In-Sight(tm) 1721 Session 0
User: admin
Password:
User Logged In
EU SetConfig(04,0,4,2,1,2,0,280,45,100,500,0,
0,42,22,1,"*****CS","",0,50,1,0,0,25)
1
0
```

The In-Sight wafer reader responds with a status code (1), followed by the number of characters, in bytes (0).

SetConfigAccept

Specifies the minimum response threshold for the raw score of the specified Config. This setting is only enabled for OCR mark types.

Syntax: EV SetConfigAccept (WaferID, Config, Accept)

SetConfigAccept Inputs

[WaferID](#)

[Config](#)

[Accept](#)

SetConfigAccept Outputs

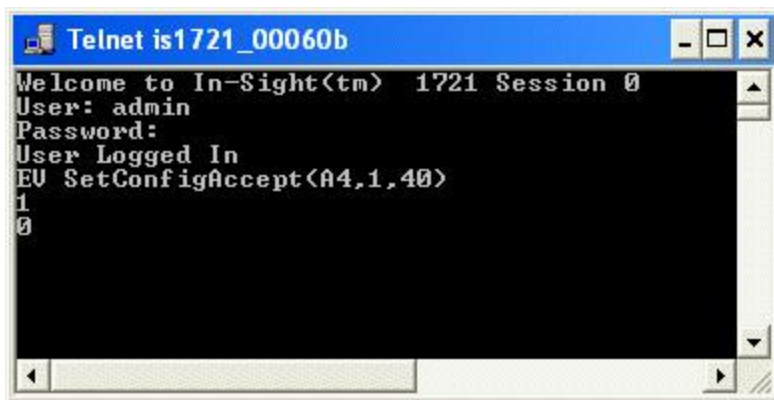
Returns A status code, followed by the number of characters (in bytes) for the results.

The status codes are:

- 1 The command was executed successfully.
- 0 Unrecognized command.
- 2 The command could not be executed.

SetConfigAccept Example

To set the [Accept Threshold](#) setting to 40 for Config 1, issue the following command: **EV SetConfigAccept(A4,1,40)**



The In-Sight wafer reader responds with a status code (1), followed by the number of characters, in bytes (0).

SetConfigAppendSEMICHkToT7

Specifies whether the SEMI checksum is automatically calculated and appended to the read result for the specified Config. This feature is only supported for '2D, T7' and 'SEMI M1.15' mark types.

```
Syntax: EV SetConfigAppendSEMICHkToT7 (WaferID, Config, AppendSEMICHkToT7)
```

SetConfigAppendSEMICHkToT7 Inputs

[WaferID](#)

[Config](#)

[AppendSEMICHkToT7](#)

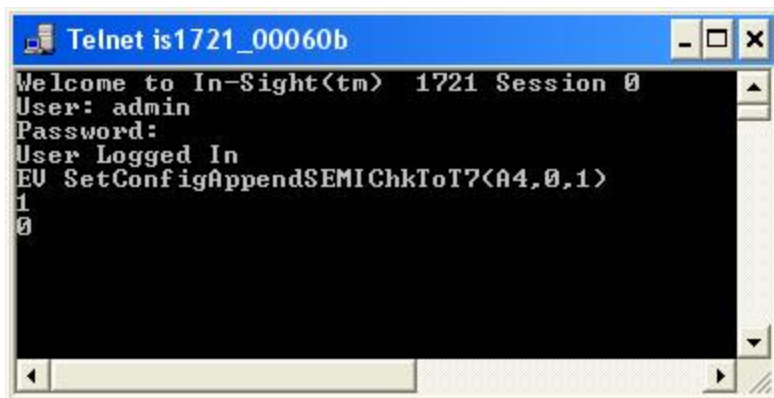
SetConfigAppendSEMICHkToT7 Outputs

Returns A status code, followed by the number of characters (in bytes) for the results, then the results in XML format.

- 1 The command was executed successfully.
- 0 Unrecognized command.
- 2 The command could not be executed.
- 6 User does not have Full Access to execute the command.

SetConfigAppendSEMICHkToT7 Example

To automatically calculate the SEMI checksum and append it to the read result for Config 0, issue the following command:
EV SetConfigAppendSEMICHktoT7(A4,0,1)



The In-Sight wafer reader responds with a status code (1), followed by the number of characters, in bytes (0).

SetConfigBC3Digits

Specifies the status of the Three Digit setting. This setting is only enabled for barcode mark types.

Syntax: EV SetConfigBC3Digits (WaferID, Config, BC3Digits)

SetConfigBC3Digits Inputs

[WaferID](#)

[Config](#)

[BC3Digits](#)

SetConfigBC3Digits Outputs

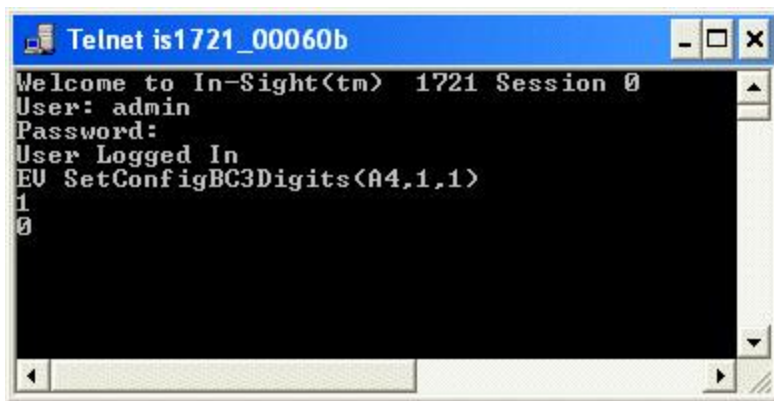
Returns A status code, followed by the number of characters (in bytes) for the results.

The status codes are:

- 1 The command was executed successfully.
- 0 Unrecognized command.
- 2 The command could not be executed.

SetConfigBC3Digits Example

To enable the Three Digit setting for Config 1, issue the following command: **EV SetConfigBC3Digits(A4,1,1)**



The In-Sight wafer reader responds with a status code (1), followed by the number of characters, in bytes (0).

SetConfigBCHyphen

Specifies the status of the Hyphen setting. This setting is only enabled for barcode mark types.

Syntax: EV SetConfigBCHyphen (WaferID, Config, BCHyphen)

SetConfigBCHyphen Inputs

[WaferID](#)

[Config](#)

[BCHyphen](#)

SetConfigBCHyphen Outputs

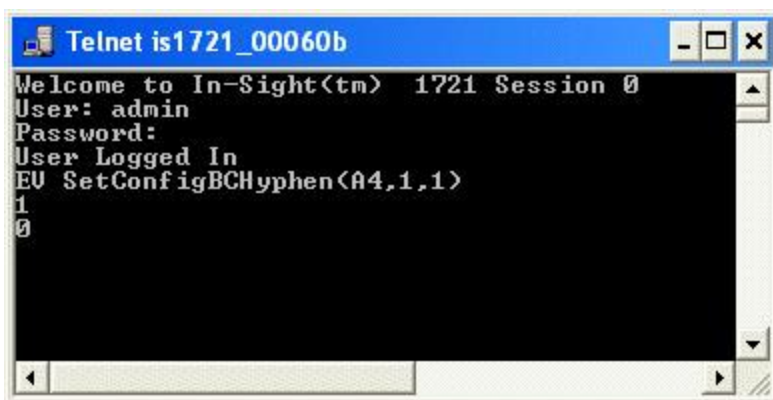
Returns A status code, followed by the number of characters (in bytes) for the results.

The status codes are:

- 1 The command was executed successfully.
- 0 Unrecognized command.
- 2 The command could not be executed.

SetConfigBCHyphen Example

To enable the Hyphen setting for Config 1, issue the following command: **EV SetConfigBCHyphen(A4,1,1)**



The In-Sight wafer reader responds with a status code (1), followed by the number of characters, in bytes (0).

SetConfigBCType

Specifies the index of the Compressed Barcode type. This setting is only enabled for barcode mark types.

Syntax: EV SetConfigBCType (WaferID, Config, BCType)

SetConfigBCType Inputs

[WaferID](#)

[Config](#)

[BCType](#)

SetConfigBCType Outputs

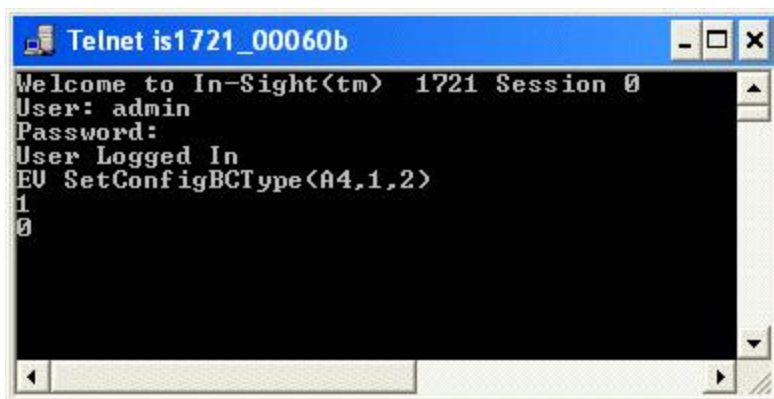
Returns A status code, followed by the number of characters (in bytes) for the results.

The status codes are:

- 1 The command was executed successfully.
- 0 Unrecognized command.
- 2 The command could not be executed.

SetConfigBCType Example

To specify a Compressed 1 setting for Config 1, issue the following command: **EV SetConfigBCType(A4,1,2)**



The In-Sight wafer reader responds with a status code (1), followed by the number of characters, in bytes (0).

SetConfigCharSize

Sets the expected size of an individual character or 2D mark for the specified Config.

Syntax: EV SetConfigCharSize (WaferID, Config, High, Wide)

SetConfigCharSize Inputs

[WaferID](#)

[Config](#)

[High](#)

[Wide](#)

SetConfigCharSize Outputs

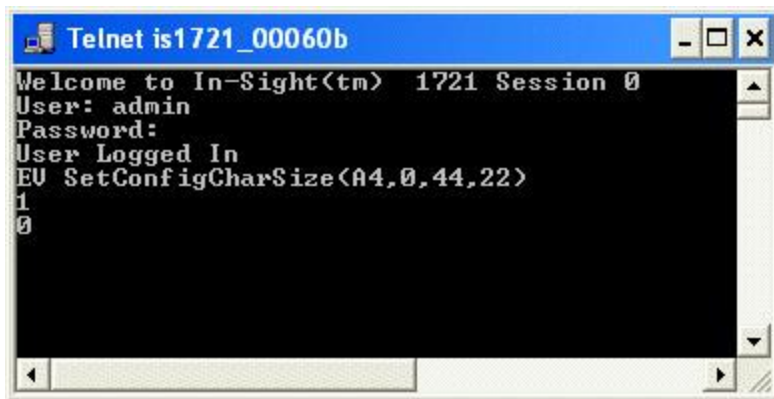
Returns A status code, followed by the number of characters (in bytes) for the results.

The status codes are:

- 1 The command was executed successfully.
- 0 Unrecognized command.
- 2 The command could not be executed.

SetConfigCharSize Example

To set the expected size of an individual character for Config 0, issue the following command: **EV SetConfigCharSize (A4,0,44,22)**



The In-Sight wafer reader responds with a status code (1), followed by the number of characters, in bytes (0).

SetConfigChecksum

Selects the method by which the read result from a character string will be validated to ensure reliability, and establishes the scoring basis for determining the pass/fail status of the read. This setting is only enabled for 'SEMI M1.15', 'Chars, SEMI', 'Chars, Triple', 'Chars, OCR-A' or 'Chars, Chartered' mark types.

```
Syntax: EV SetConfigChecksum(WaferID, Config, Checksum)
```

SetConfigChecksum Inputs

[WaferID](#)

[Config](#)

[Checksum](#)

SetConfigChecksum Outputs

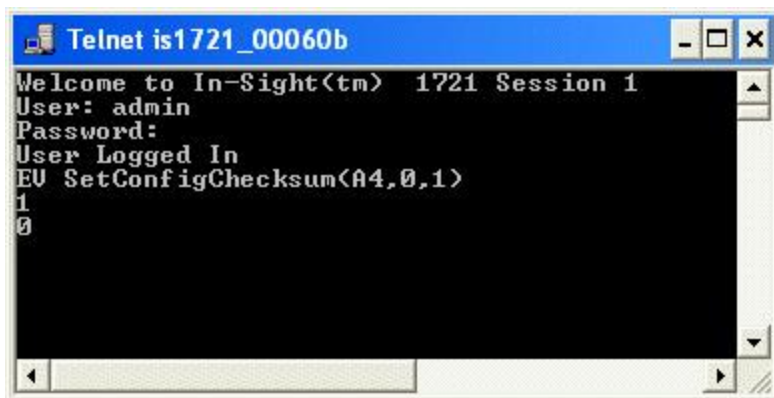
Returns A status code, followed by the number of characters (in bytes) for the results.

The status codes are:

- 1 The command was executed successfully.
- 0 Unrecognized command.
- 2 The command could not be executed.

SetConfigChecksum Example

To specify a SEMI Checksum for Config 0, issue the following command: **EV SetConfigChecksum(A4,0,1)**



The In-Sight wafer reader responds with a status code (1), followed by the number of characters, in bytes (0).

SetConfigCharSpacing

The Spacing setting controls the tolerance of the spacing between adjacent characters and the vertical alignment of the characters. This setting is only relevant for OCR mark types.

Syntax: EV SetConfigCharSpacing(WaferID,Config,CharSpacing)

SetConfigCharSpacing Inputs

[WaferID](#)

[Config](#)

[CharSpacing](#)

SetConfigCharSpacing Outputs

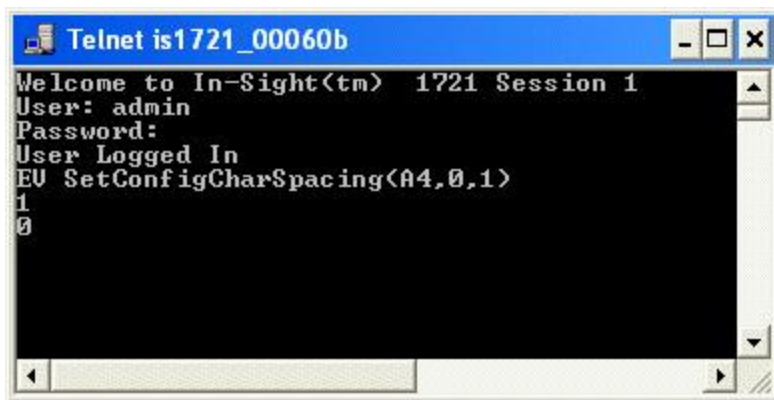
Returns A status code, followed by the number of characters (in bytes) for the results.

The status codes are:

- 1 The command was executed successfully.
- 0 Unrecognized command.
- 2 The command could not be executed.

SetConfigCharSpacing Example

To specify standard spacing for Config 0, issue the following command: **EV SetConfigCharSpacing(A4,0,1)**



The In-Sight wafer reader responds with a status code (1), followed by the number of characters, in bytes (0).

SetConfigColor

Sets the expected color of the wafer mark. For use when the Mark Color checkbox (or Color checkbox in the Java applet) will be disabled during tuning.

```
Syntax: EV SetConfigColor (WaferID, Config, Color)
```

SetConfigColor Inputs

[WaferID](#)

[Config](#)

[Color](#)

SetConfigColor Outputs

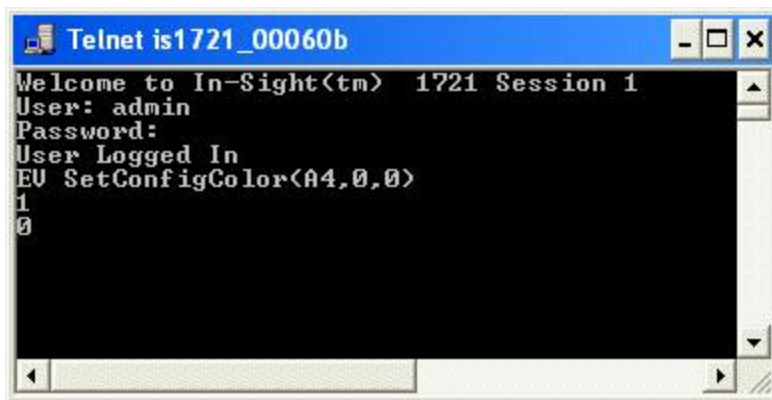
Returns A status code, followed by the number of characters (in bytes) for the results.

The status codes are:

- 1 The command was executed successfully.
- 0 Unrecognized command.
- 2 The command could not be executed.

SetConfigColor Example

To specify a black mark color for Config 0, issue the following command: **EV SetConfigColor(A4,0,0)**



The In-Sight wafer reader responds with a status code (1), followed by the number of characters, in bytes (0).

SetConfigEnable

Specifies whether the Config(s) is enabled or disabled, which determines if the specified Config(s) will be used during the retry sequence.

```
Syntax: EV SetConfigEnable (WaferID, Config, Enable, [Config, Enable], ...
[Config, Enable])
```

Notes:

- Configs 0 - 49 can be enabled or disabled, in any order using the SetConfigEnable command.
- In-Sight Native Mode commands can contain a maximum of 255 characters, including spaces. If setting the enable/disable state of more than 25 Configs at once, it is recommended that two separate SetConfigEnable commands be issued to ensure the 255 character limit is not exceeded.

SetConfigEnable Inputs

[WaferID](#)

[Config](#)

[Enable](#)

[Config,Enable]...

[Config,Enable]

SetConfigEnable Outputs

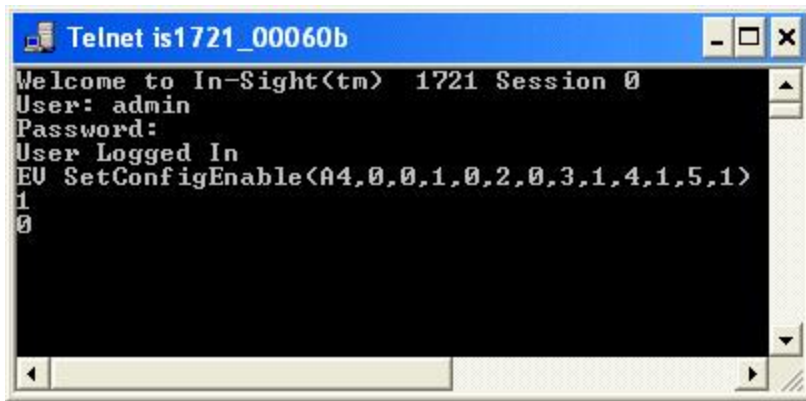
Returns A status code, followed by the number of characters (in bytes) for the results.

The status codes are:

- 1 The command was executed successfully.
- 0 Unrecognized command.
- 2 The command could not be executed.

SetConfigEnable Example

To disable Config 0, 1 and 2 and enable Config 3, 4, 5, issue the following command: **EV SetConfigEnable (A4,0,0,1,0,2,0,3,1,4,1,5,1)**



A screenshot of a Telnet session window titled "Telnet is1721_00060b". The window has a blue title bar and standard Windows window controls (minimize, maximize, close). The main area is black with white text. The text shows a login sequence: "Welcome to In-Sight(tm) 1721 Session 0", "User: admin", "Password:", "User Logged In", and "EU SetConfigEnable(A4,0,0,1,0,2,0,3,1,4,1,5,1)". Below the command, the response "1" is shown on one line and "0" on the next line. A scrollbar is visible on the right side of the text area.

```
Telnet is1721_00060b
Welcome to In-Sight(tm) 1721 Session 0
User: admin
Password:
User Logged In
EU SetConfigEnable(A4,0,0,1,0,2,0,3,1,4,1,5,1)
1
0
```

The In-Sight wafer reader responds with a status code (1), followed by the number of characters, in bytes (0).

SetConfigEx

Specifies all settings for the specified Config.

```
Syntax: EV SetConfigEx
(WaferID,Config,Mark,Checksum,LightMode,LightPower,Orientation,Row,Col,
High,Wide,Theta,Phi,High,Wide,CharSpacing,"FieldString","FieldDef",Color,Accept,En
able,Retry,
BComp,Thresh,Optimization,StrokeThickness,LEDOVERRIDE,LEDOVERRIDEMask,"MatchString
","ConfigName")
```

SetConfigEx Inputs

[WaferID](#)

[Config](#)

[Mark](#)

[Checksum](#)

[LightMode](#)

[LightPower](#)

[Orientation](#)

[Row \(Region\)](#)

[Col \(Region\)](#)

[High \(Region\)](#)

[Wide \(Region\)](#)

[Theta \(Region\)](#)

[Phi \(Region\)](#)

[High \(CharSize\)](#)

[Wide \(CharSize\)](#)

[Charspacing](#)

["FieldString"](#)

["FieldDef"](#)

[Color](#)

[Accept](#)

[Enable](#)

[Retry](#)

[BComp](#)

[Thresh](#)

[Optimization](#)

[StrokeThickness](#)

[LEDOVERRIDE](#)

[LEDOVERRIDEMASK](#)["MATCHSTRING"](#)["CONFIGNAME"](#)

SetConfigEx Outputs

Returns A status code, followed by the number of characters (in bytes) for the results.

The status codes are:

- 1 The command was executed successfully.
- 0 Unrecognized command.
- 2 The command could not be executed.

SetConfigEx Example

To specify the following settings for Config 0:

Mark Type: Chars, SEMI

Checksum: SEMI with Virtual

LightMode: The mark will appear dark on a light background.

LightPower: 2.000

Orientation: Normal

Row (Region): 280

Col (Region): 45

High (Region): 100

Wide (Region): 500

Theta (Region): 0

Phi (Region): 0

High (Charsize): 42

Wide (Charsize): 22

Charspacing: Standard

FieldString: *****CS

FieldDef: "" (blank)

Color: Black

Accept: 50

Enable: Enable

Retry: Disable

BComp: 0

Thresh: 25

Optimization: 0

StrokeThickness: 0

LEDOVERRIDE: 0

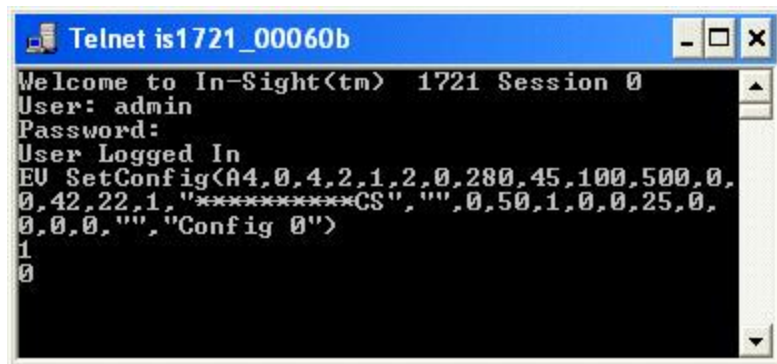
LEDOVERRIDEMASK: 0

MATCHSTRING: "" (blank)

CONFIGNAME: "Config 0"

Issue the following command:

EV SetConfigEx(A4,0,4,2,1,2,0,280,45,100,500,0,0,42,22,1,"*****CS", "",0,50,1,0,0,25,0,0,0,"", "Config 0")



The In-Sight wafer reader responds with a status code (1), followed by the number of characters, in bytes (0).

SetConfigFieldDef

Optionally, the Field Definition setting can be used to create user-defined entries for the Field String if the mark specifies a character string or barcode.

Syntax: EV SetConfigFieldDef (WaferID, Config, "FieldDef")

SetConfigFieldDef Inputs

[WaferID](#)

[Config](#)

["FieldDef"](#)

Note: The FieldDef input must be enclosed by quotation marks.

SetConfigFieldDef Outputs

Returns A status code, followed by the number of characters (in bytes) for the results.

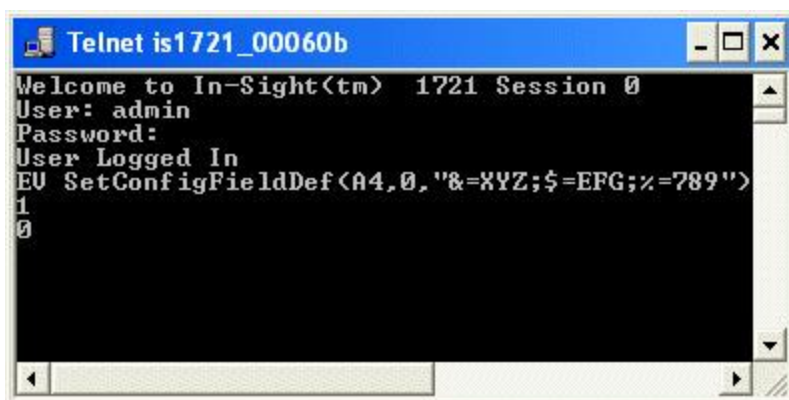
The status codes are:

- 1 The command was executed successfully.
- 0 Unrecognized command.
- 2 The command could not be executed.

SetConfigFieldDef Example

To create the following Field Definition: **&=XYZ;\$=EFG;%=789**, issue the following command: **EV SetConfigFieldDef (A4,0,"&=XYZ;\$=EFG;%=789")**

If the Field String is **&\$%*****CS**, then the only characters that are valid in the first position are X, Y or Z. The only valid characters in the second position are E, F or G. And in the third position, only 7, 8, or 9 are valid.



The In-Sight wafer reader responds with a status code (1), followed by the number of characters, in bytes (0).

SetConfigFieldString

Specifies the number of characters contained in the character string or encoded in a barcode.

Syntax: EV SetConfigFieldString(WaferID,Config,"FieldString")

SetConfigFieldString Inputs

[WaferID](#)

[Config](#)

["FieldString"](#)

Note: The FieldString input must be enclosed by quotation marks.

SetConfigFieldString Outputs

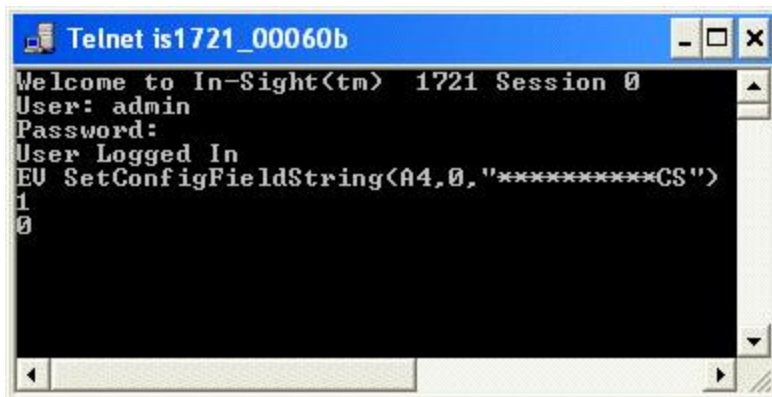
Returns A status code, followed by the number of characters (in bytes) for the results.

The status codes are:

- 1 The command was executed successfully.
- 0 Unrecognized command.
- 2 The command could not be executed.

SetConfigFieldString Example

To specify the number of characters contained in the character string for Config 0, issue the following command: EV SetConfigFieldString(A4,0,"*****CS")



The In-Sight wafer reader responds with a status code (1), followed by the number of characters, in bytes (0).

SetConfigLEDOVERRIDE

Specifies whether a custom Light Mode is used and the light row(s) to enable.

Syntax: EV SetConfigLEDOVERRIDE(WaferID,Config,LEDOVERRIDE,LEDOVERRIDEMask)

SetConfigLEDOVERRIDE Inputs

[WaferID](#)

[Config](#)

[LEDOVERRIDE](#)

[LEDOVERRIDEMask](#)

SetConfigLEDOVERRIDE Outputs

Returns A status code, followed by the number of characters (in bytes) for the results.

The status codes are:

- 1 The command was executed successfully.
- 0 Unrecognized command.
- 2 The command could not be executed.

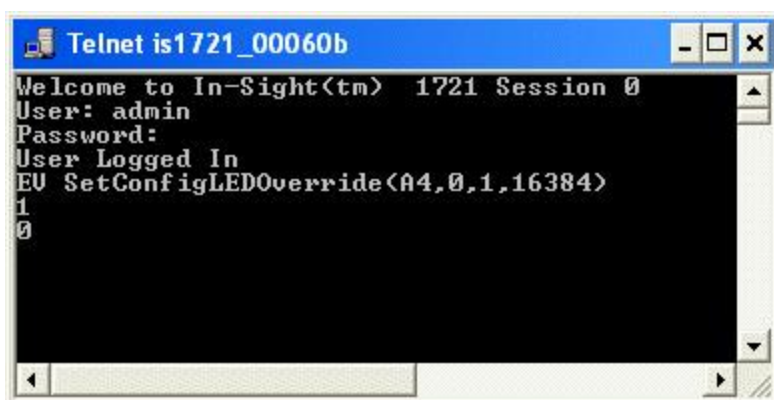
SetConfigLEDOVERRIDE Examples

[Example 1: Enable a Single Light Row](#)

[Example 2: Enable Multiple Light Rows](#)

Example 1: Enable a Single Light Row

To enable light row 1 for Config 0, issue the following command: **EV SetConfigLEDOVERRIDE(A4,0,1,16384)**



The In-Sight wafer reader responds with a status code (1), followed the number of characters, in bytes (0).

Example 2: Enable Multiple Light Rows

To enable light row 1 and light row 4:

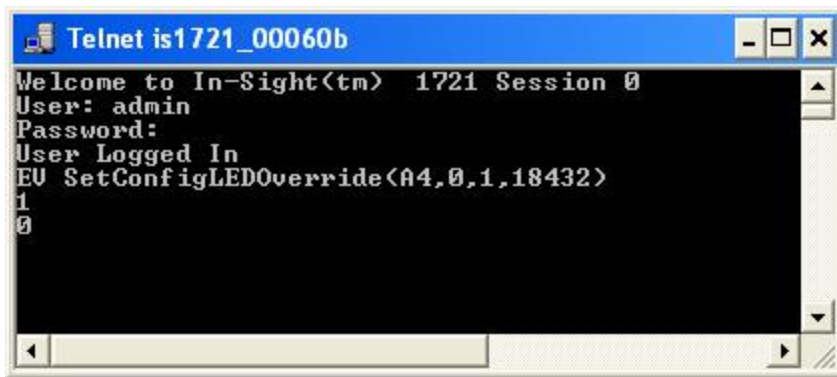
1. Determine the binary value used to enable the light rows.

Light Row	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Binary	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0

2. Convert the binary value to its decimal equivalent.

0100100000000000 = 18432

3. Issue the following command: **EV SetConfigLEDOVERRIDE(A4,0,1,18432)**



The In-Sight wafer reader responds with a status code (1), followed by the number of characters, in bytes (0).

SetConfigLightMode

Specifies either a Bright Field or a Dark Field illumination effect on the mark.

Syntax: EV SetConfigLightMode (WaferID, Config, LightMode)

SetConfigLightMode Inputs

[WaferID](#)

[Config](#)

[LightMode](#)

SetConfigLightMode Outputs

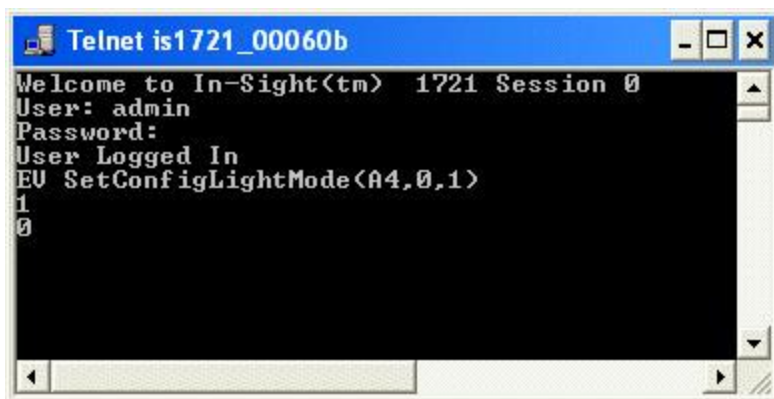
Returns A status code, followed by the number of characters (in bytes) for the results.

The status codes are:

- 1 The command was executed successfully.
- 0 Unrecognized command.
- 2 The command could not be executed.

SetConfigLightMode Example

To set Config 0's Light Mode to 1, issue the following command: **EV SetConfigLightMode(A4,0,1)**



The In-Sight wafer reader responds with a status code (1), followed by the number of characters, in bytes (0).

SetConfigLightPower

Controls the effective intensity of the LEDs on the mark by increasing the exposure time of the wafer reader's CCD imager.

Syntax: EV SetConfigLightPower (WaferID, Config, LightPower)

SetConfigLightPower Inputs

[WaferID](#)

[Config](#)

[LightPower](#)

SetConfigLightPower Outputs

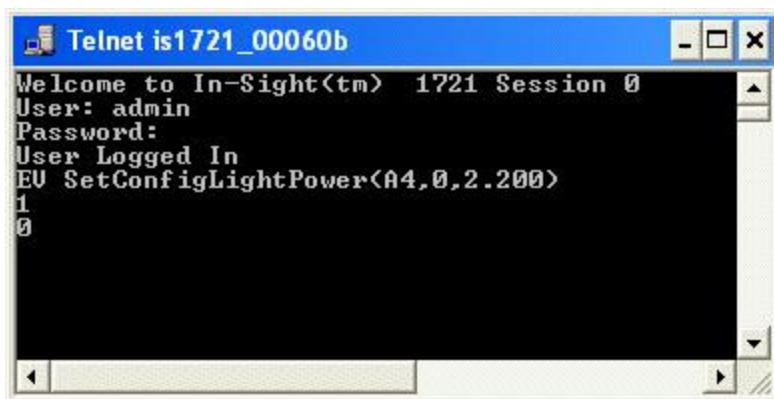
Returns A status code, followed by the number of characters (in bytes) for the results.

The status codes are:

- 1 The command was executed successfully.
- 0 Unrecognized command.
- 2 The command could not be executed.

SetConfigLightPower Example

To set Config 0's Light Power to 2.200, issue the following command: **EV SetConfigLightPower(A4,0,2.200)**



The In-Sight wafer reader responds with a status code (1), followed by the number of characters, in bytes (0).

SetConfigMark

Specifies the type of wafer mark the selected Config will read.

Syntax: EV SetConfigMark(WaferID, Config, Mark)

SetConfigMark Inputs

[WaferID](#)

[Config](#)

[Mark](#)

SetConfigMark Outputs

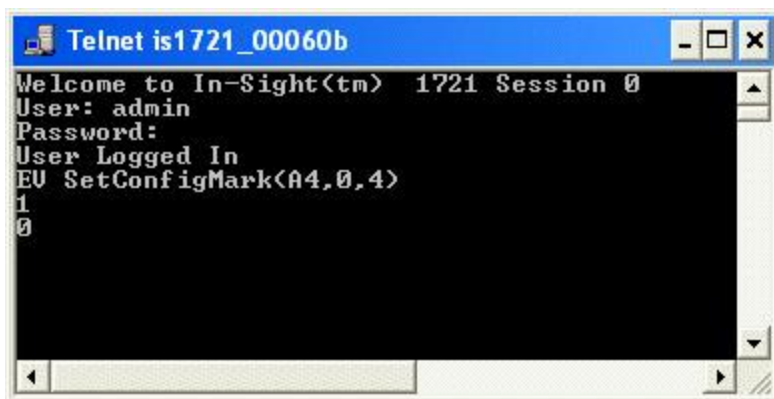
Returns A status code, followed by the number of characters (in bytes) for the results.

The status codes are:

- 1 The command was executed successfully.
- 0 Unrecognized command.
- 2 The command could not be executed.

SetConfigMark Example

To specify a 'Chars, SEMI' mark type for Config 0, issue the following command: **EV SetConfigMark(A4,0,4)**



The In-Sight wafer reader responds with a status code (1), followed by the number of characters, in bytes (0).

SetConfigMatchString

Sends a text string, which is used as a comparative match during the tuning process. The tuning process uses the Match String rather than the current Field String or Field Definition settings and returns the Light, Size and Filter settings with the highest read score that matches the string. The Match String setting can be helpful with noisy images (for example, when the mark is scribed over a die pattern, where background noise could potentially produce incorrect read results). This option is only enabled for OCR mark types.

```
Syntax: EV SetConfigMatchString(WaferID,Config,"MatchString")
```

SetConfigMatchString Inputs

[WaferID](#)

[Config](#)

["MatchString"](#)

Notes:

- The MatchString input must be enclosed by quotation marks.
- The MatchString input is case sensitive.

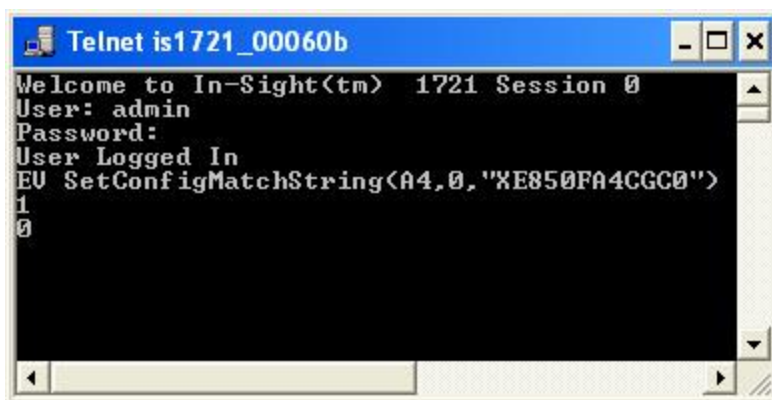
SetConfigMatchString Outputs

Returns A status code, followed by the number of characters (in bytes) for the results, then the results in XML format.

- 1 The command was executed successfully.
- 0 Unrecognized command.
- 2 The command could not be executed.
- 6 User does not have Full Access to execute the command.

SetConfigMatchString Examples

To set the Match String to "XE850FA4CGC0", issue the following command: **EV SetConfigMatchString (A4,0,"XE850FA4CGC0")**



The In-Sight wafer reader responds with a status code (1), followed by the number of characters, in bytes (0).

SetConfigName

Sets the name of an individual Config.

Syntax: EV SetConfigName(WaferID, Config, "ConfigName")

Notes:

- The "ConfigName" input must be enclosed by quotes.
- Config names can contain up to 15 alphanumeric characters; symbols are not valid.
- Only users with Full Access can set the Config name by issuing the SetConfigName Native Mode command.

SetConfigName Inputs

[WaferID](#)

[Config](#)

["ConfigName"](#)

SetConfigName Outputs

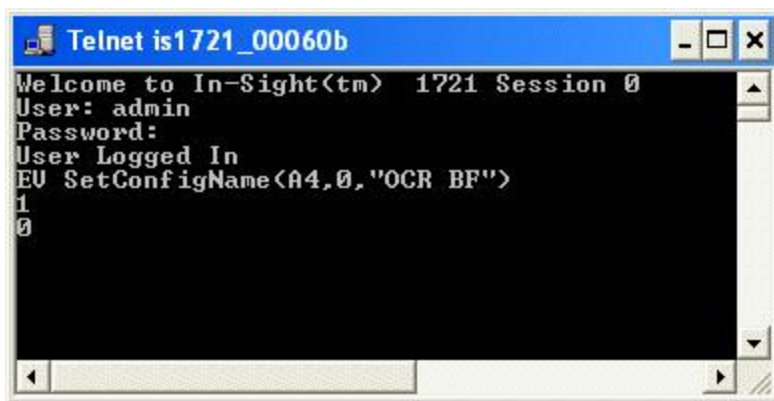
Returns A status code, followed by the number of characters (in bytes) for the results.

The status codes are:

- 1 The command was executed successfully.
- 0 Unrecognized command.
- 2 The command could not be executed.
- 6 User does not have Full Access to execute the command.

SetConfigName Example

To rename Config 0 to **OCR BF**, issue the following command: **EV SetConfigName(A4,0,"OCR BF")**



The In-Sight wafer reader responds with a status code (1), followed by the number of characters, in bytes (0).

SetConfigOptimization

When 'SEMI M1.15', 'Chars, SEMI' or 'Chars, OCR-A' is the selected Mark, SetConfigOptimization specifies the optimization of the image.

```
Syntax: EV SetConfigOptimization(WaferID,Config,Optimization)
```

SetConfigOptimization Inputs

[WaferID](#)

[Config](#)

[Optimization](#)

SetConfigOptimization Outputs

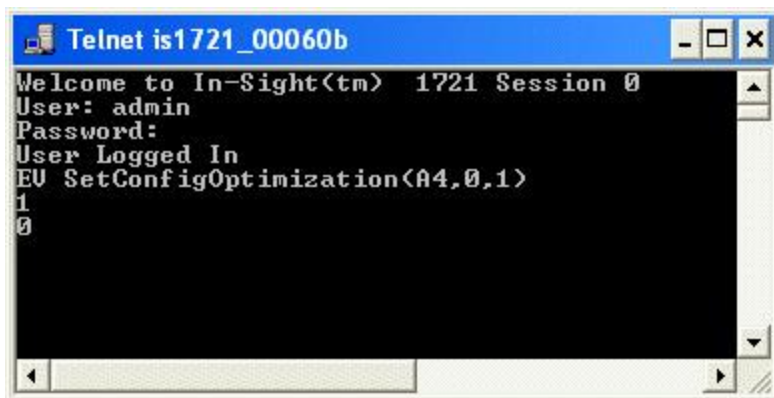
Returns A status code, followed by the number of characters (in bytes) for the results.

The status codes are:

- 1 The command was executed successfully.
- 0 Unrecognized command.
- 2 The command could not be executed.
- 6 User does not have [Full Access](#) to execute the command.

SetConfigOptimization Example

To specify Mode 1 as the Optimization type for Config 0, issue the following command: **EV SetConfigOptimization(A4,0,1)**



The In-Sight wafer reader responds with a status code (1), followed by the number of characters, in bytes (0).

SetConfigOrientation

Specifies the orientation of the image.

Syntax: EV SetConfigOrientation(WaferID,Config,Orientation)

SetConfigOrientation Inputs

[WaferID](#)

[Config](#)

[Orientation](#)

SetConfigOrientation Outputs

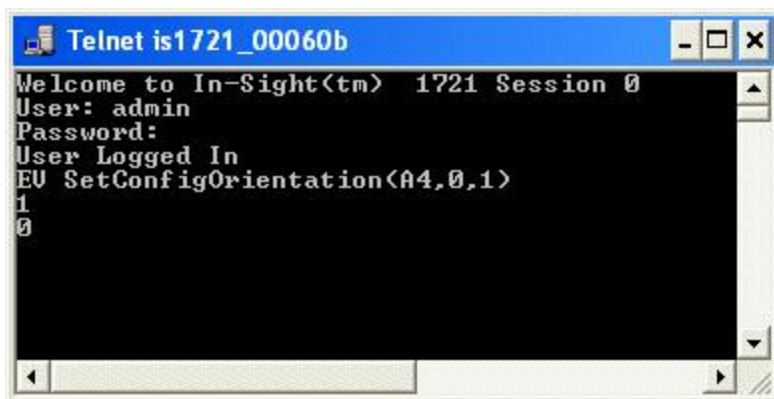
Returns A status code, followed by the number of characters (in bytes) for the results.

The status codes are:

- 1 The command was executed successfully.
- 0 Unrecognized command.
- 2 The command could not be executed.

SetConfigOrientation Example

To specify a **Mirrored horizontally** orientation for Config 0, issue the following command: **EV SetConfigOrientation(A4,0,1)**



The In-Sight wafer reader responds with a status code (1), followed by the number of characters, in bytes (0).

SetConfigRegion

Sets the region in which the wafer mark is expected to appear in the field of view for the specified Config.

```
Syntax: EV SetConfigRegion (WaferID, Config, Row, Col, High, Wide, Theta, Phi)
```

SetConfigRegion Inputs

[WaferID](#)

[Config](#)

[Row](#)

[Col](#)

[High](#)

[Wide](#)

[Theta](#)

[Phi](#)

SetConfigRegion Outputs

Returns A status code, followed by the number of characters (in bytes) for the results.

The status codes are:

- 1 The command was executed successfully.
- 0 Unrecognized command.
- 2 The command could not be executed.

SetConfigRegion Example

To specify the following settings for Config 0:

Row (Region): 280

Col (Region): 45

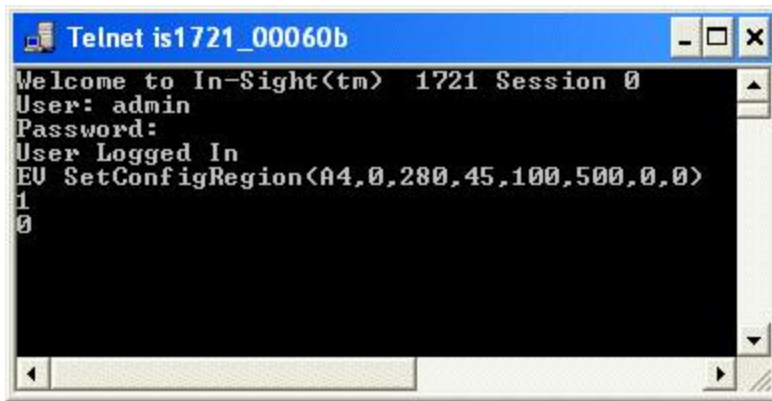
High (Region): 100

Wide (Region): 500

Theta (Region): 0

Phi (Region): 0

Issue the following command: **EV SetConfigRegion(A4,0,280,45,100,500,0,0)**



A screenshot of a Telnet session window titled "Telnet is1721_00060b". The window has a blue title bar and standard Windows window controls (minimize, maximize, close). The main area is black with white text. The text shows a login sequence: "Welcome to In-Sight(tm) 1721 Session 0", "User: admin", "Password:", "User Logged In", and a command "EU SetConfigRegion(A4,0,280,45,100,500,0,0)". The response is "1" followed by "0" on the next line. A scrollbar is visible on the right side of the text area.

```
Telnet is1721_00060b
Welcome to In-Sight(tm) 1721 Session 0
User: admin
Password:
User Logged In
EU SetConfigRegion(A4,0,280,45,100,500,0,0)
1
0
```

The In-Sight wafer reader responds with a status code (1), followed by the number of characters, in bytes (0).

SetConfigRemoveSEMIChecksum

Specifies whether the 2-character SEMI checksum and any "special" character immediately preceding the checksum ("-", ".", "#" or space) is automatically removed from the read result for the specified Config.. This feature is only supported for 'SEMI, M1.5', 'Chars, SEMI', 'Chars, Triple', 'Chars, OCR-A', 'Chars, Chartered' or 'Chars, Cannon' mark types with a SEMI or SEMI with Virtual checksum.

Syntax: EV SetConfigRemoveSEMIChecksum (WaferID,Config,RemoveSemiChecksum)

SetConfigRemoveSEMIChecksum Inputs

[WaferID](#)

[Config](#)

[RemoveSemiChecksum](#)

SetConfigRemoveSEMIChecksum Outputs

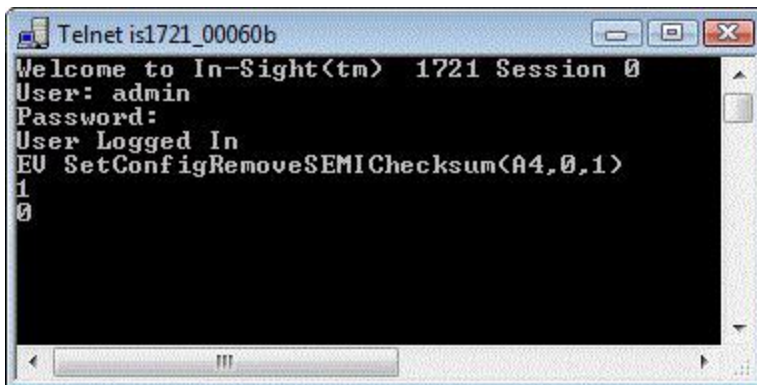
Returns A status code, followed by the number of characters (in bytes) for the results, then the results in XML format.

- 1 The command was executed successfully.
- 0 Unrecognized command.
- 2 The command could not be executed.
- 6 User does not have Full Access to execute the command.

SetConfigRemoveSEMIChecksum Example

To automatically calculate the SEMI checksum and append it to the read result for Config 0, issue the following command:

EV SetConfigRemoveSEMIChecksum (A4,0,1)



The In-Sight wafer reader responds with a status code (1), followed by the number of characters, in bytes (0).

SetConfigRetry

Specifies how the Config will be used during the retry sequence.

Syntax: EV SetConfigRetry (WaferID, Config, Retry)

SetConfigRetry Inputs

[WaferID](#)

[Config](#)

[Retry](#)

SetConfigRetry Outputs

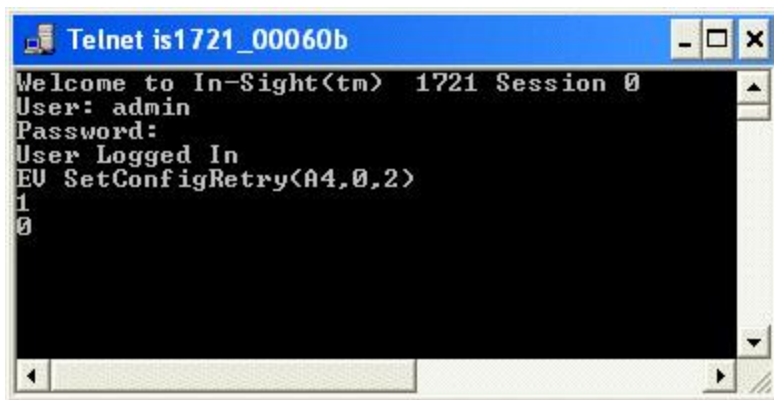
Returns A status code, followed by the number of characters (in bytes) for the results.

The status codes are:

- 1 The command was executed successfully.
- 0 Unrecognized command.
- 2 The command could not be executed.

SetConfigRetry Example

To set Config 0's tuning behavior to **Tune & Save**, issue the following command: **EV SetConfigRetry(A4,0,2)**



The In-Sight wafer reader responds with a status code (1), followed by the number of characters, in bytes (0).

SetConfigStrokeThickness

The Stroke Thickness setting can be used to improve the reading performance of thick or thin characters. This option is only enabled for OCR mark types.

Note: This command is only available in In-Sight 1750 wafer readers for backwards compatibility, and returns success without performing any operations.

Syntax: `EV SetConfigStrokeThickness (WaferID, Config, StrokeThickness)`

SetConfigStrokeThickness Inputs

[WaferID](#)

[Config](#)

[StrokeThickness](#)

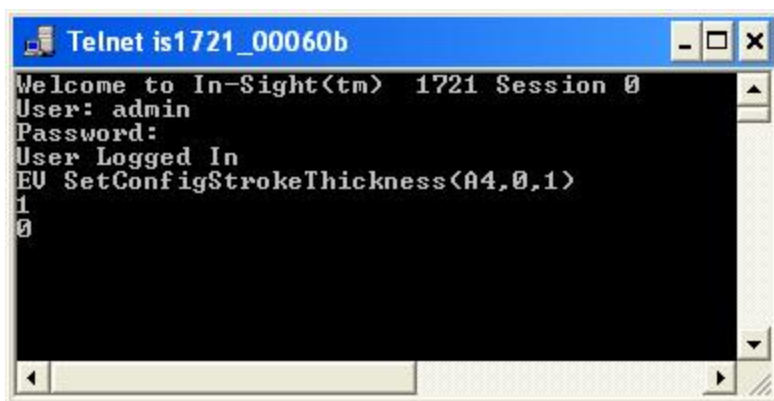
SetConfigStrokeThickness Outputs

Returns A status code, followed by the number of characters (in bytes) for the results, then the results in XML format.

- 1 The command was executed successfully.
- 0 Unrecognized command.
- 2 The command could not be executed.
- 6 User does not have Full Access to execute the command.

SetConfigStrokeThickness Examples

To set the Stroke Thickness to Thick, issue the following command: `EV SetConfigStrokeThickness(A4,0,1)`



The In-Sight wafer reader responds with a status code (1), followed by the number of characters, in bytes (0).

SetConfigThresh

Specifies the minimum response threshold for each individual character score. Each individual character score must be greater than the Character Threshold or the read will fail. This setting is only enabled for OCR mark types.

Syntax: EV SetConfigThresh(WaferID,Config,Thresh)

SetConfigThresh Inputs

[WaferID](#)

[Config](#)

[Thresh](#)

SetConfigThresh Outputs

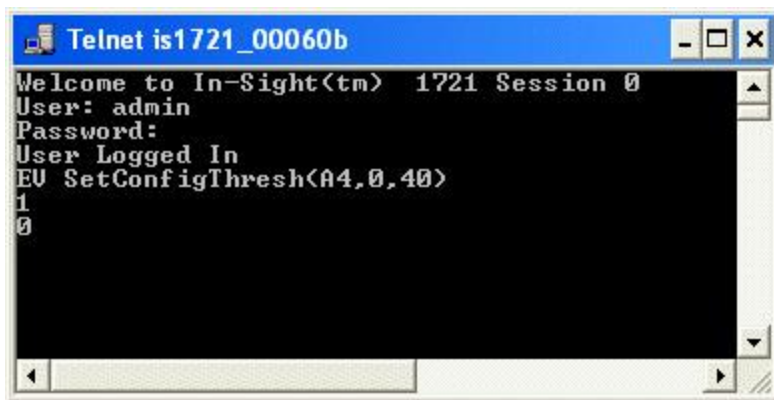
Returns A status code, followed by the number of characters (in bytes) for the results.

The status codes are:

- 1 The command was executed successfully.
- 0 Unrecognized command.
- 2 The command could not be executed.

SetConfigThresh Example

To specify a Character Threshold of **40**, issue the following command: **EV SetConfigThresh(A4,0,40)**



The In-Sight wafer reader responds with a status code (1), followed by the number of characters, in bytes (0).

SetConfigTune

Specifies the settings to try during tuning.

```
Syntax: EV SetConfigTune (WaferID, Config, ParameterSet, Enable, [SetCommand])
```

SetConfigTune Inputs

Setting	Description
WaferID	References the  WaferID data structure in the spreadsheet.
Config	The Config number. (0 to 49)
ParameterSet	Specifies the tuning parameter to set.
Notes: - A separate SetConfigTune command must be issued for each individual <i>ParameterSet</i>. - The <i>ParameterSet</i> input must be enclosed in quotation marks.	
"UseLimits"	Specifies whether custom tuning limits will be used during tuning. If enabled, custom tuning settings are used to define custom Light Mode, Light Power, Size and Filter settings to use during tuning. If disabled, the default tuning settings are used.
"Light"	Specifies whether Light Mode and Light Power settings will be tried during tuning.
"Mode"	Specifies whether Mode settings will be tried during custom tuning.
"Power"	Specifies whether Power settings will be tried during custom tuning.
"Filter"	Specifies whether Filter settings will be tried during tuning.
"Size"	Specifies whether the Size settings will be tried during tuning.
Enable	Specifies whether the <i>ParameterSet</i> is enabled or disabled during tuning. 0 = Disable 1 = Enable*
[SetCommand]	Optional parameter that specifies custom tuning limits for the specified <i>ParameterSet</i> . The <i>SetCommand</i> input must be preceded by the corresponding <i>ParameterSet</i> input. For example, if specifying custom tuning limits for Filters, first specify the "Filter" <i>ParameterSet</i> input, followed by one or more of the "Filter" <i>Inputs</i> . When specifying more than one <i>SetCommand</i> input, the <i>SetCommand</i> inputs can be listed in any order.

Notes:

- The *SetCommand* input must be enclosed in quotation marks.

Setting	Description
<i>"Mode" Inputs</i> "ModeX=x"	Specifies the Light Mode to try during tuning.
	ModeX Mode1 Mode2 Mode3 Mode4 Mode5 Mode6 Mode7 Mode8 Mode9 Mode10 Mode11
	x 0 = Disable 1 = Enable*
"ModeX-Y=x"	Specifies the Light Mode range (inclusive) to try during tuning.
"Custom=x"	Specifies whether to use a Custom light mode during tuning.
	x 0 = Disable Custom Light Mode 1 = Enable Custom Light Mode*
"External=x"	Specifies whether the In-Sight 1740 wafer reader's integrated external LEDs (intended for dark field lighting applications) will be tried during tuning.
	x 0 = Disable External Light* 1 = Enable External Light
"Expansion=x"	Specifies whether an add-on light connected to the In-Sight 1740 wafer reader's Expansion port (intended for wafer reading applications with unique illumination requirements) will be tried during tuning.
	x 0 = Disable Expansion Light* 1 = Enable Expansion Light

Setting	Description
<i>"Power" Inputs</i> "Auto=x"	Specifies whether Auto Light Power tuning is used. If enabled, all Light Power settings are tried. If enabled, the <i>Min</i>, <i>Max</i> and <i>StepSize</i> parameter values are not used during tuning. x 0 = Disable 1 = Enable*
"Min=x"	Specifies the minimum Light Power setting.
"Max=x"	x 0.05 - 127; 0.05* Specifies the maximum Light Power setting.
"StepSize=x"	x 0.05 - 127; 127* Defines the increments of Light Power values to try during tuning.
<i>"Size" Inputs</i> "PlusMinus=x"	x 0.05 - 20; 0.25* Specifies the Size settings (in pixels) to try during tuning. During tuning, the Size region is adjusted based on the selected pixel size. x ±1 - 10; ±3*

Setting	Description
"Filter" Inputs "MaxOperations=x"	The maximum number of filter operations to use when constructing the filter during tuning. (1 - 3); default = 3
"TuningMethod=x"	Specifies whether the existing filter should be optimized or replaced during subsequent tuning cycles. Specify Optimize to preserve the existing filter operations during tuning, and automatically select the optimal parameters for these operations. Specify Replace to overwrite the existing filter and replace it with a new filter during tuning.
"AllFilters=x"	x 1 = Optimize* 2 = Replace Specifies that all filter operations should be tried when constructing the filter during tuning.
"OperationNameX=x"	x 0 = Disable 1 = Enable* Specifies individual filter operations to try when constructing the filter during tuning.
	OperationName Erode LowPass Open TopHat Clipping Dilate HighPass Close BottomHat Equalize
	x 0 = Disable 1 = Enable*

SetConfigTune Outputs

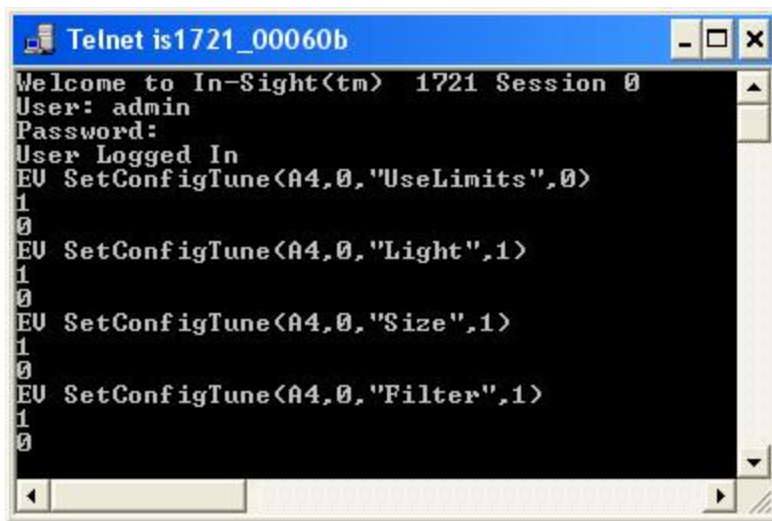
Returns A status code, followed by the number of characters (in bytes) for the results, then the results in XML format.

The status codes are:

- 1 The command was executed successfully.
- 0 Unrecognized command.
- 2 The command could not be executed.

SetConfigTune Example - Default Tuning

1. To disable custom tuning, issue the following command:
EV SetConfigTune(A4,0,"UseLimits",0)
2. To enable Light Mode and Light Power tuning, issue the following command:
EV SetConfigTune(A4,0,"Light",1)
3. To enable Size tuning, issue the following command:
EV SetConfigTune(A4,0,"Size",1)
4. To enable Filter tuning, issue the following command:
EV SetConfigTune(A4,0,"Filter",1)



5. Once the tuning settings have been specified, issue the [TuneConfigEx](#) command to initiate the tuning process.

SetConfigTune Examples - Custom Tuning

Example 1

1. To enable custom tuning, issue the following command:
EV SetConfigTune(A4,0,"UseLimits",1)
2. To enable Light Mode and Light Power tuning, issue the following command:
EV SetConfigTune(A4,0,"Light",1)
3. To enable Light Mode tuning and to enable Light Modes 1 - 8 and disable Light Modes 9 - 11, issue the following command:
EV SetConfigTune(A4,0,"Mode",1,"Mode1-8=1","Mode9-11=0")
4. To enable Light Power tuning and enable Auto Power tuning, issue the following command:
EV SetConfigTune(A4,0,"Power",1,"Auto=1")
5. To enable Size tuning and to adjust the Size region ± 7 pixels, issue the following command:
EV SetConfigTune(A4,0,"Size",1,"PlusMinus=7")

- To enable Filter tuning and use a maximum of three filter operations when constructing the filter, try all filter operations except Low Pass and High Pass and specify the tuning behavior to optimize the existing filter on subsequent tuning cycles, issue the following command:

EV SetConfigTune

(A4,0,"Filter",1,"MaxOperations=3","AllFilters=1","LowPass=0","HighPass=0","TuningMethod=1")

```

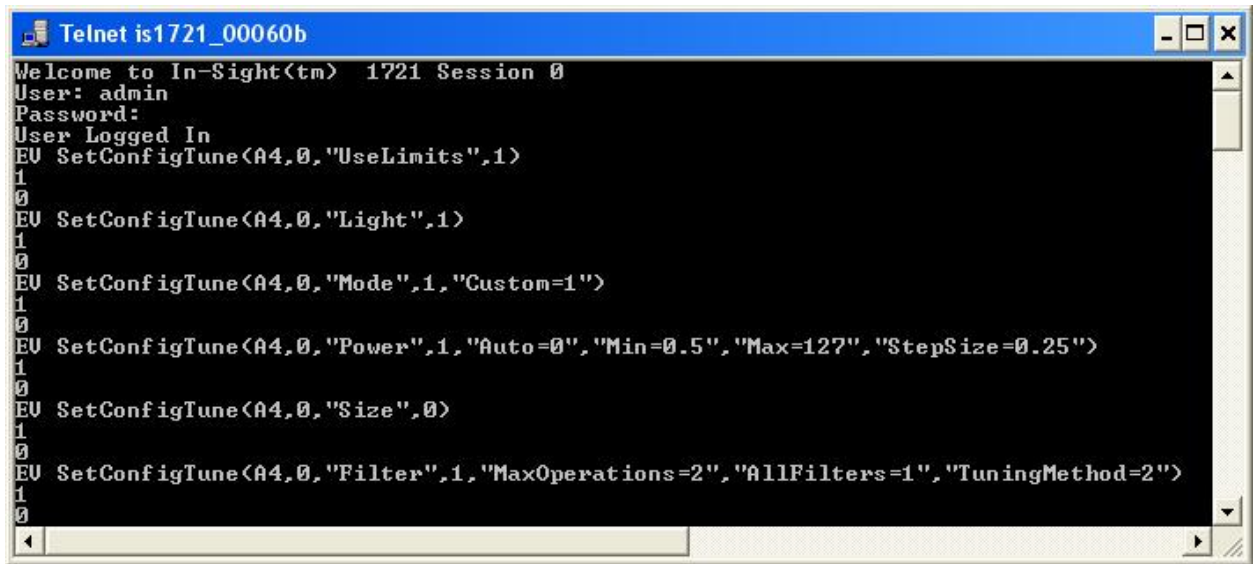
Telnet is1721_00060b
Welcome to In-Sight(tm) 1721 Session 0
User: admin
Password:
User Logged In
EV SetConfigTune(A4,0,"UseLimits",1)
1
EV SetConfigTune(A4,0,"Light",1)
1
EV SetConfigTune(A4,0,"Mode",1,"Mode1-8=1","Mode9-11=0")
1
EV SetConfigTune(A4,0,"Power",1,"Auto=1")
1
EV SetConfigTune(A4,0,"Size",1,"PlusMinus=7")
1
EV SetConfigTune(A4,0,"Filter",1,"MaxOperations=3","AllFilters=1","LowPass=0","HighPass=0","TuningMethod=1")
1

```

- Once the tuning settings have been specified, issue the [TuneConfigEx](#) command to initiate the tuning process.

Example 2

- To enable custom tuning, issue the following command:
EV SetConfigTune(A4,0,"UseLimits",1)
- To enable Light Mode and Light Power tuning, issue the following command:
EV SetConfigTune(A4,0,"Light",1)
- To enable Light Mode tuning and use the Custom Light Mode, issue the following command:
EV SetConfigTune(A4,0,"Mode",1,"Custom=1")
- To enable Light Power tuning, specify the Manual Light Power settings to use and disable Auto Power tuning, issue the following command:
EV SetConfigTune(A4,0,"Power",1,"Auto=0","Min=0.5","Max=127","StepSize=0.25")
- To disable Size tuning, issue the following command:
EV SetConfigTune(A4,0,"Size",0)
- To enable Filter tuning and use a maximum of two filter operations when constructing the filter, try all filter operations and specify the tuning behavior to replace the existing filter on subsequent tuning cycles, issue the following command:
EV SetConfigTune(A4,0,"Filter",1,"MaxOperations=2","AllFilters=1","TuningMethod=2")



```
Telnet is1721_00060b
Welcome to In-Sight(tm) 1721 Session 0
User: admin
Password:
User Logged In
EU SetConfigTune(A4,0,"UseLimits",1)
1
0
EU SetConfigTune(A4,0,"Light",1)
1
0
EU SetConfigTune(A4,0,"Mode",1,"Custom=1")
1
0
EU SetConfigTune(A4,0,"Power",1,"Auto=0","Min=0.5","Max=127","StepSize=0.25")
1
0
EU SetConfigTune(A4,0,"Size",0)
1
0
EU SetConfigTune(A4,0,"Filter",1,"MaxOperations=2","AllFilters=1","TuningMethod=2")
1
0
```

7. Once the tuning settings have been specified, issue the [TuneConfigEx](#) command to initiate the tuning process.

SetWaferAcqEnable

Disables image acquisition to allow a [test read](#) from the last image acquired, or from an image file, when the reader is Offline. The Light Mode and Light Power settings are temporarily disabled.

Syntax: EV SetWaferAcqEnable (WaferID, AcqEnable)

SetWaferAcqEnable Inputs

[WaferID](#)

[AcqEnable](#)

SetWaferAcqEnable Outputs

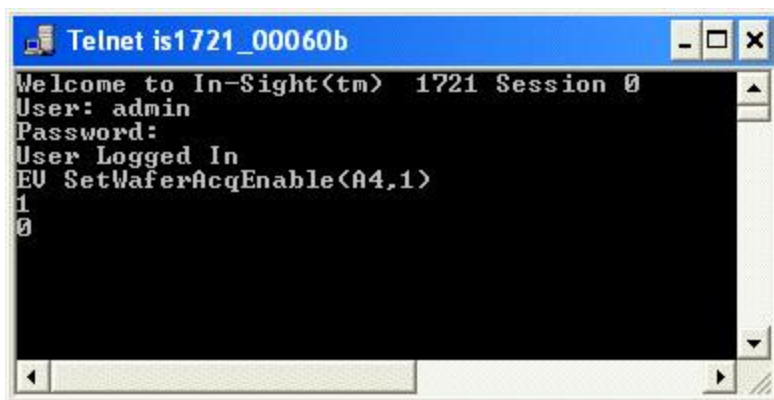
Returns A status code, followed by the number of characters (in bytes) for the results.

The status codes are:

- 1 The command was executed successfully.
- 0 Unrecognized command.
- 2 The command could not be executed.

SetWaferAcqEnable Example

To disable image acquisition to allow a test read from an image file, issue the following command: **EV SetWaferAcqEnable (A4,1)**



The In-Sight wafer reader responds with a status code (1), followed by the number of characters, in bytes (0).

SetWaferCompatibilityMode

Specifies an optional Compatibility Mode to use for the In-Sight wafer reader.

```
Syntax: EV SetWaferCompatibilityMode (CompatibilityMode)
```

SetWaferCompatibilityMode Inputs

[CompatibilityMode](#)

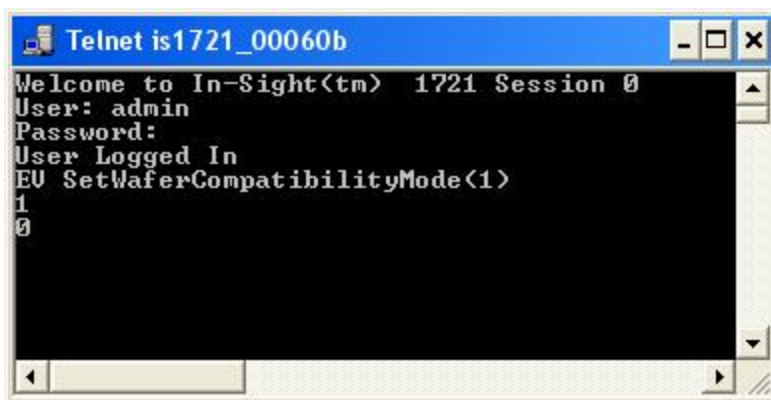
SetWaferCompatibilityMode Outputs

Returns A status code, followed by the number of characters (in bytes) for the results, then the results in XML format.

- 1 The command was executed successfully.
- 0 Unrecognized command.
- 2 The command could not be executed.
- 6 User does not have Full Access to execute the command.

SetWaferCompatibilityMode Example

To set the Compatibility Mode to Mode 1, issue the following command: **EV SetWaferCompatibilityMode(1)**



The In-Sight wafer reader responds with a status code (1), followed by the number of characters, in bytes (0).

SetWaferFailCharsOutRegion

Specifies whether the read will fail if the center of any character is located outside of the region. This setting is only enabled for OCR mark types.

```
Syntax: EV SetWaferFailCharsOutRegion(WaferID, FailCharsOutRegion)
```

SetWaferFailCharsOutRegion Inputs

[WaferID](#)

[FailCharsOutRegion](#)

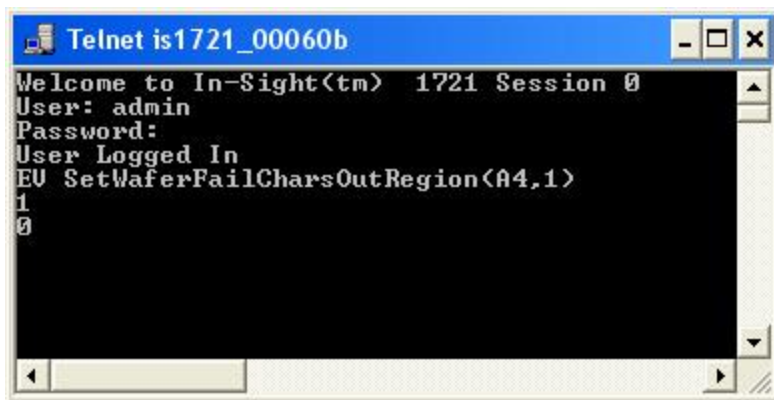
SetWaferFailCharsOutRegion Outputs

Returns A status code, followed by the number of characters (in bytes) for the results, then the results in XML format.

- 1 The command was executed successfully.
- 0 Unrecognized command.
- 2 The command could not be executed.
- 6 User does not have Full Access to execute the command.

SetWaferFailCharsOutRegion Example

To enable the Fail Characters Outside of Region setting, issue the following command: **EV SetWaferFailCharsOutRegion (A4,1)**



The In-Sight wafer reader responds with a status code (1), followed by the number of characters, in bytes (0).

SetWaferOrder

Specifies the order in which enabled Configs will be tried until a successful read can be obtained.

```
Syntax: EV SetWaferOrder(WaferID,Order)
```

SetWaferOrder Inputs

[WaferID](#)

[Order](#)

SetWaferOrder Outputs

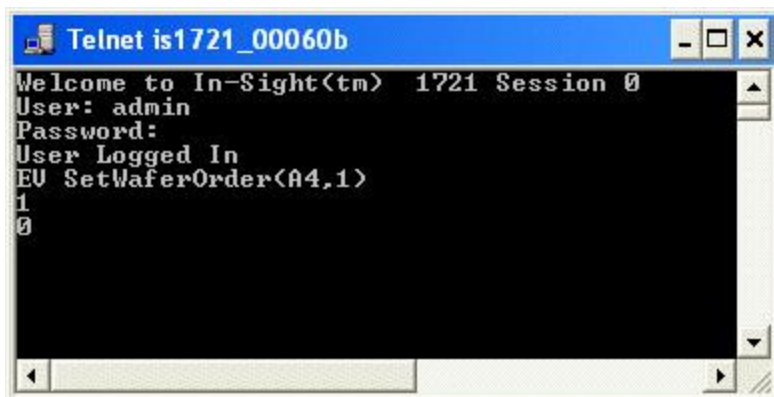
Returns A status code, followed by the number of characters (in bytes) for the results.

The status codes are:

- 1 The command was executed successfully.
- 0 Unrecognized command.
- 2 The command could not be executed.

SetWaferOrder Example

To try the Configs in numerical order (0 - 49), regardless of how they scored during the most recent read, issue the following command: **EV SetWaferOrder(A4,1)**



The In-Sight wafer reader responds with a status code (1), followed by the number of characters, in bytes (0).

SetWaferRegionTolerance

Specifies the rotational tolerance of an OCR scribe within a region and whether portions of characters, or entire characters, extending beyond the region can be read. This setting is only enabled for OCR mark types.

```
Syntax: EV SetWaferRegionTolerance(WaferID,RegionTolerance)
```

SetWaferRegionTolerance Inputs

[WaferID](#)

[RegionTolerance](#)

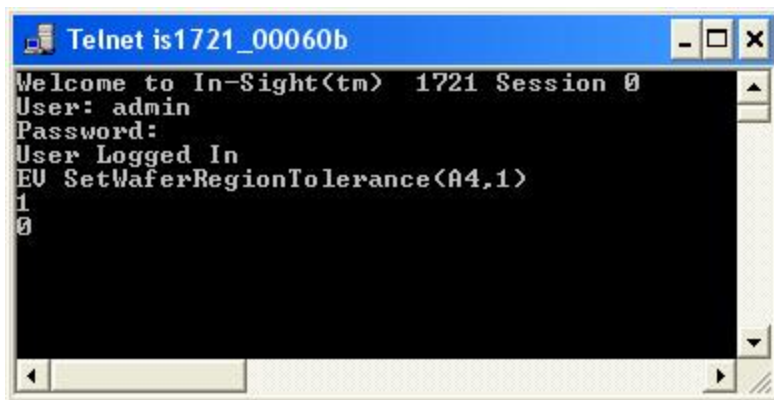
SetWaferRegionTolerance Outputs

Returns A status code, followed by the number of characters (in bytes) for the results, then the results in XML format.

- 1 The command was executed successfully.
- 0 Unrecognized command.
- 2 The command could not be executed.
- 6 User does not have Full Access to execute the command.

SetWaferRegionTolerance Example

To set the Region Tolerance to Low, issue the following command: **EV SetWaferRegionTolerance(A4,1)**



The In-Sight wafer reader responds with a status code (1), followed by the number of characters, in bytes (0).

SetWaferScoreAs100

Specifies how character positions fielded as dashes (-), dots (.), pound characters (#) and spaces are scored. This option is only enabled for OCR mark types.

Syntax: EV SetWaferScoreAs100 (WaferID, ScoreAs100)

SetWaferScoreAs100 Inputs

[WaferID](#)

[ScoreAs100](#)

SetWaferScoreAs100 Outputs

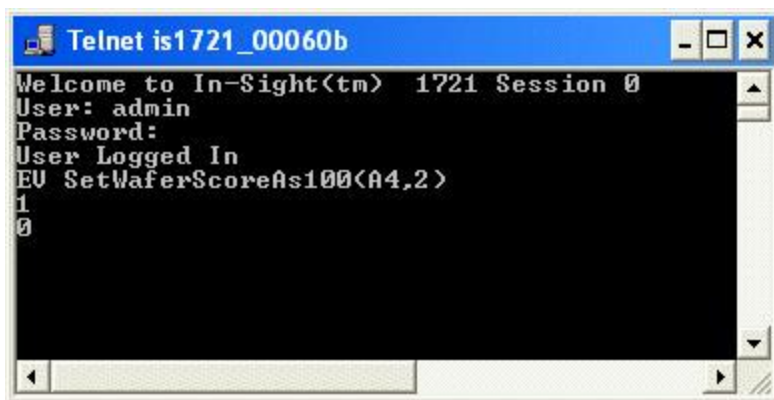
Returns A status code, followed by the number of characters (in bytes) for the results.

The status codes are:

- 1 The command was executed successfully.
- 0 Unrecognized command.
- 2 The command could not be executed.

SetWaferScoreAs100 Example

To specify that all characters fielded as dashes (-), dots (.), pound characters (#) and spaces automatically score 100 points and the scores are used in the calculation of the read score, issue the following command: **EV SetWaferScoreAs100(A4,2)**



The In-Sight wafer reader responds with a status code (1), followed by the number of characters, in bytes (0).

SetWaferTimeout

Specifies the total amount of time (in seconds) to allow for a read, including retries and tuning of enabled Configs.

Syntax: EV SetWaferTimeout (WaferID,Timeout)

SetWaferTimeout Inputs

[WaferID](#)

[Timeout](#)

SetWaferTimeout Outputs

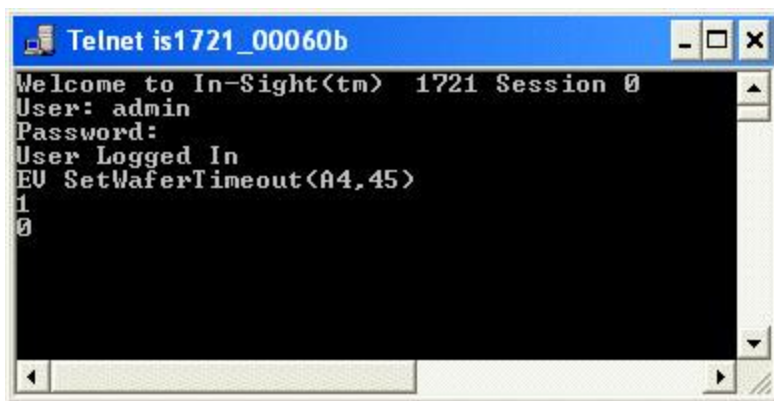
Returns A status code, followed by the number of characters (in bytes) for the results.

The status codes are:

- 1 The command was executed successfully.
- 0 Unrecognized command.
- 2 The command could not be executed.

SetWaferTimeout Example

To specify a timeout of 45 seconds, issue the following command: **EV SetWaferTimeout(A4,45)**



The In-Sight wafer reader responds with a status code (1), followed by the number of characters, in bytes (0).

TuneConfig

Tunes the specified Config.

```
Syntax: EV TuneConfig(WaferID,Config,Flag,Bright,Dark,Size)
```

Note: The TuneConfig command is for backwards compatibility only; use the [TuneConfigEx](#) command for In-Sight 1750 wafer readers.

TuneConfig Inputs

[WaferID](#)

[Config](#)

[Flag](#)

[Bright](#)

[Dark](#)

[Size](#)

TuneConfig Outputs

Return A status code, followed by the number of characters (in bytes) for the results, then the results in XML format.
s

The status codes are:

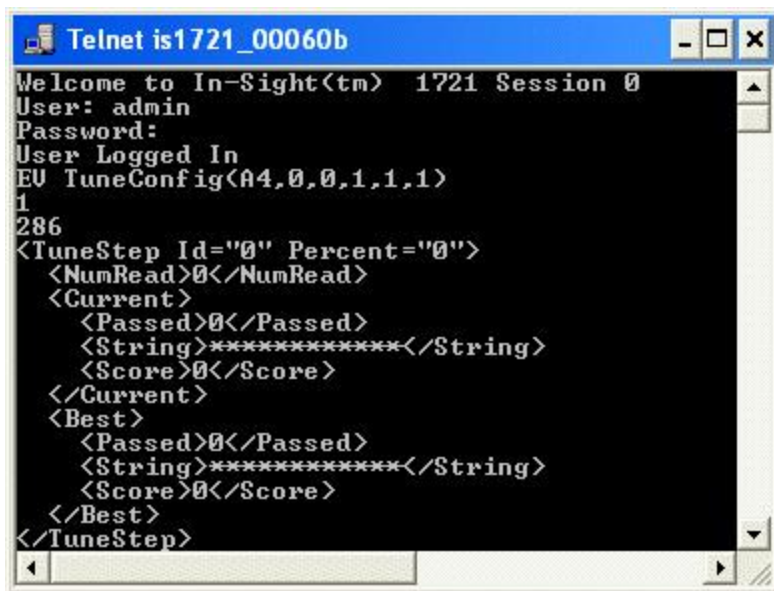
- 1 The command was executed successfully.
- 0 Unrecognized command.
- 2 The command could not be executed.

Example XML tag returns

```
<TuneStep Id="0" Percent="2">
  <NumRead>1</NumRead>
  <Current>
    <Passed>1</Passed>
    <String>6713H129S6A1</String>
    <Score>396</Score>
  </Current>
  <Best>
    <Passed>1</Passed>
    <String>6713H129S6A1</String>
    <Score>396</Score>
  </Best>
</TuneStep>
```

Use the TuneConfig Command

1. To initiate tuning for Config 0, send the following command: **EV TuneConfig(A4,0,0,1,1,1)**, setting the Flag input to 0.

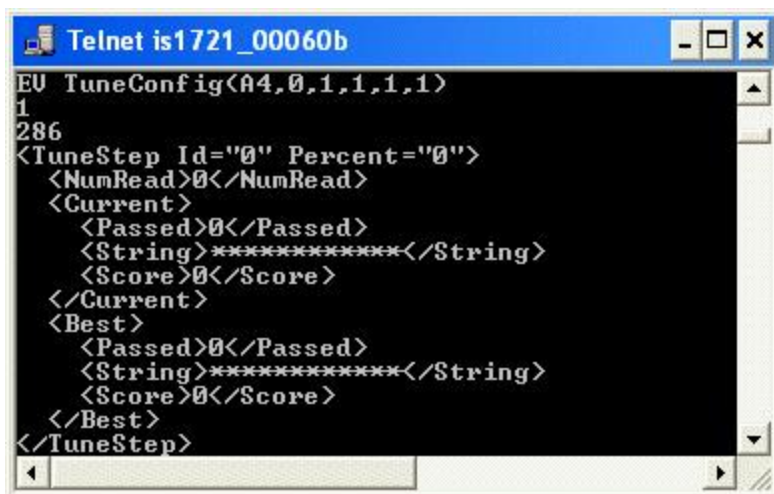


```

Telnet is1721_00060b
Welcome to In-Sight(tm) 1721 Session 0
User: admin
Password:
User Logged In
EV TuneConfig(A4,0,0,1,1,1)
1
286
<TuneStep Id="0" Percent="0">
  <NumRead>0</NumRead>
  <Current>
    <Passed>0</Passed>
    <String>*****</String>
    <Score>0</Score>
  </Current>
  <Best>
    <Passed>0</Passed>
    <String>*****</String>
    <Score>0</Score>
  </Best>
</TuneStep>

```

2. To continue tuning, send the following command: **EV TuneConfig(A4,0,1,1,1,1)**, setting the Flag input to 1.

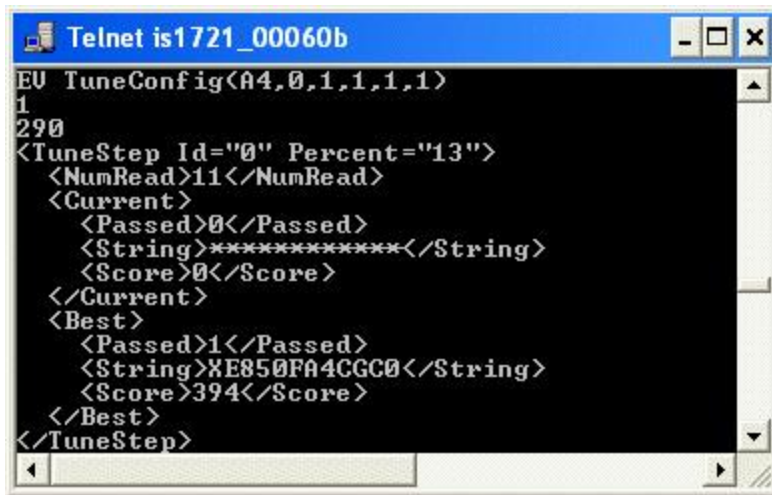


```

Telnet is1721_00060b
EV TuneConfig(A4,0,1,1,1,1)
1
286
<TuneStep Id="0" Percent="0">
  <NumRead>0</NumRead>
  <Current>
    <Passed>0</Passed>
    <String>*****</String>
    <Score>0</Score>
  </Current>
  <Best>
    <Passed>0</Passed>
    <String>*****</String>
    <Score>0</Score>
  </Best>
</TuneStep>

```

3. Reissue the **EV TuneConfig(A4,0,1,1,1,1)** command with the Flag input to set to 1 until an acceptable passing score is reached.

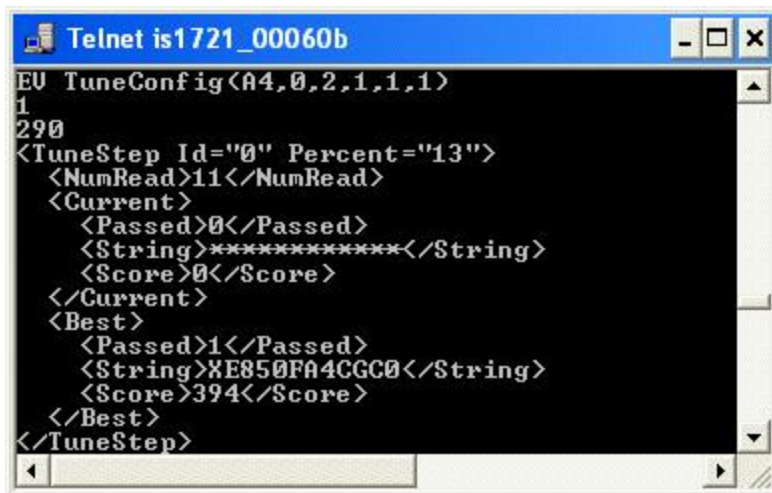


```

Telnet is1721_00060b
EU TuneConfig(A4,0,1,1,1,1)
1
290
<TuneStep Id="0" Percent="13">
  <NumRead>11</NumRead>
  <Current>
    <Passed>0</Passed>
    <String>*****</String>
    <Score>0</Score>
  </Current>
  <Best>
    <Passed>1</Passed>
    <String>XE850FA4CGC0</String>
    <Score>394</Score>
  </Best>
</TuneStep>

```

4. To accept the optimized settings when tuning is completed, send the following command: **EV TuneConfig (A4,0,2,1,1,1)**, setting the Flag input to 2.



```

Telnet is1721_00060b
EU TuneConfig(A4,0,2,1,1,1)
1
290
<TuneStep Id="0" Percent="13">
  <NumRead>11</NumRead>
  <Current>
    <Passed>0</Passed>
    <String>*****</String>
    <Score>0</Score>
  </Current>
  <Best>
    <Passed>1</Passed>
    <String>XE850FA4CGC0</String>
    <Score>394</Score>
  </Best>
</TuneStep>

```

TuneConfigEx

Tunes the specified Config.

```
Syntax: EV TuneConfigEx(WaferID, Config, Flag)
```

Note: The [TuneConfig](#) command is for backwards compatibility only; use the TuneConfigEx command for In-Sight 1750 wafer readers.

TuneConfigEx Inputs

[WaferID](#)

[Config](#)

[Flag](#)

TuneConfigEx Outputs

Returns A status code, followed by the number of characters (in bytes) for the results, then the results in XML format.

The status codes are:

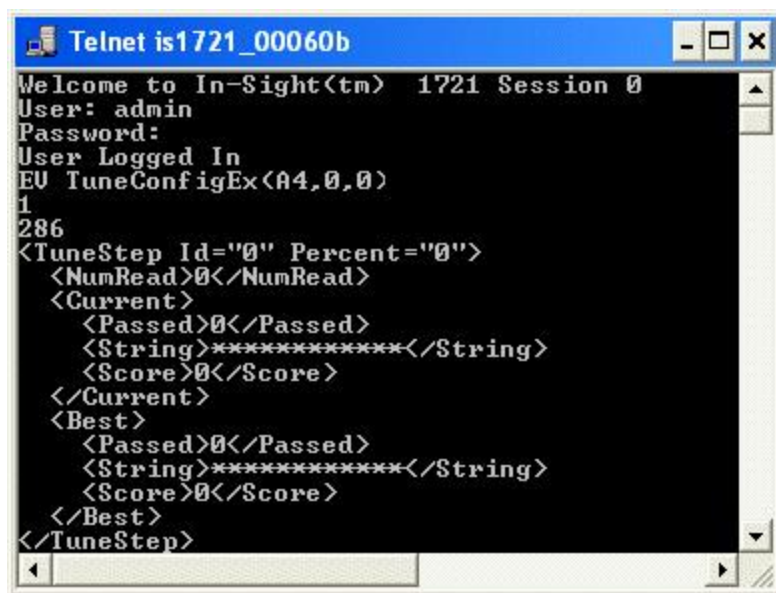
- 1 The command was executed successfully.
- 0 Unrecognized command.
- 2 The command could not be executed.

Example XML tag returns

```
<TuneStep Id="0" Percent="2">
  <NumRead>1</NumRead>
  <Current>
    <Passed>1</Passed>
    <String>6713H129S6A1</String>
    <Score>396</Score>
  </Current>
  <Best>
    <Passed>1</Passed>
    <String>6713H129S6A1</String>
    <Score>396</Score>
  </Best>
</TuneStep>
```

Use the TuneConfigEx Command

1. To initiate tuning for Config 0, send the following command: **EV TuneConfigEx(A4,0,0)**, setting the Flag input to **0**.

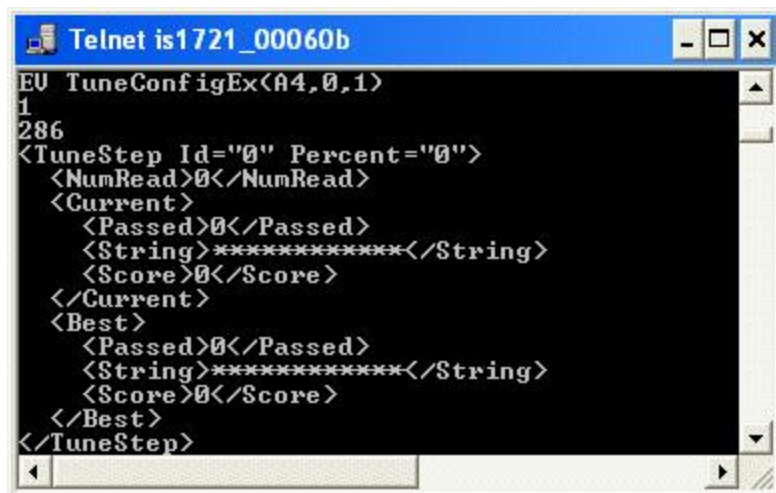


```

Telnet is1721_00060b
Welcome to In-Sight(tm) 1721 Session 0
User: admin
Password:
User Logged In
EV TuneConfigEx(A4,0,0)
1
286
<TuneStep Id="0" Percent="0">
  <NumRead>0</NumRead>
  <Current>
    <Passed>0</Passed>
    <String>*****</String>
    <Score>0</Score>
  </Current>
  <Best>
    <Passed>0</Passed>
    <String>*****</String>
    <Score>0</Score>
  </Best>
</TuneStep>

```

2. To continue tuning, send the following command: **EV TuneConfigEx(A4,0,1)**, setting the Flag input to 1.



```

Telnet is1721_00060b
EV TuneConfigEx(A4,0,1)
1
286
<TuneStep Id="0" Percent="0">
  <NumRead>0</NumRead>
  <Current>
    <Passed>0</Passed>
    <String>*****</String>
    <Score>0</Score>
  </Current>
  <Best>
    <Passed>0</Passed>
    <String>*****</String>
    <Score>0</Score>
  </Best>
</TuneStep>

```

3. Reissue the **EV TuneConfigEx(A4,0,1)** command with the Flag input set to 1 until an acceptable passing score is reached.

```

Telnet is1721_00060b
EU TuneConfigEx(A4,0,1)
1
286
<TuneStep Id="0" Percent="0">
  <NumRead>0</NumRead>
  <Current>
    <Passed>0</Passed>
    <String>*****</String>
    <Score>0</Score>
  </Current>
  <Best>
    <Passed>0</Passed>
    <String>*****</String>
    <Score>0</Score>
  </Best>
</TuneStep>

```

4. To accept the optimized settings when tuning is completed, send the following command: **EV TuneConfigEx (A4,0,2)**, setting the Flag input to 2.

```

Telnet is1721_00060b
EU TuneConfigEx(A4,0,2)
1
290
<TuneStep Id="0" Percent="13">
  <NumRead>11</NumRead>
  <Current>
    <Passed>1</Passed>
    <String>XE850FA4CGC0</String>
    <Score>394</Score>
  </Current>
  <Best>
    <Passed>1</Passed>
    <String>XE850FA4CGC0</String>
    <Score>394</Score>
  </Best>
</TuneStep>

```

Protocol Commands

- [About Protocol Commands](#)
- [In-Sight Protocol](#)
- [TEL Protocol](#)
- [TSK Protocol](#)

About Protocol Commands

Protocol commands can be sent to the wafer reader to trigger an event. To send Protocol commands, you must configure the wafer reader settings from the Communications application step in the WebUI. If used with TCP/IP, a valid Port number must be specified. When using an RS-232 serial device, the wafer reader serial port settings must be configured to correspond with the serial device the wafer reader will be communicating with.

Note: To trigger an event, the wafer reader must be Online.

In-Sight Protocol

In-Sight is the default protocol for the wafer reader. This protocol can be used with either a TCP/IP or RS-232 connection.

Command	Description	Outputs
READ (Config)	Reads the specified Config (0 - 49). Note: To execute a read of all enabled Configs, send the READ command, setting the <Config> input to "-1". For example, READ(-1).	
READ	Reads all enabled Configs. This command uses the Read Order selected in the Read tab. If the READ command needs to be terminated before the read Timeout is exhausted, the Abort Execution Native Mode command can terminate the READ execution.	For all In-Sight Protocol commands, the Output String is a character string for a successful read. For failed reads, the output is either a string of asterisks (***) or a partial read (where unreadable characters are returned as question marks). The type of output is dependent on the score.
TUNE (Config)	Tunes the specified Config (0 - 49).	

Example

TEL Protocol

Protocol for use with a TEL wafer prober. This protocol can be used with either a TCP/IP or RS-232 connection.

Command	Description	Outputs
READ	Reads all enabled Configs. This command uses the Read Order selected in the Read tab.	The Output String is a character string for a successful read. For failed reads, the output is either a string of asterisks (***) or a partial read (where unreadable characters are returned as question marks). The type of output is dependent on the score.

TSK Protocol

Protocol for use with a TSK wafer prober. This protocol can be used with either a TCP/IP or RS-232 connection.

Command	Description
NBC	Request to read a barcode mark that does not match the OCR string.
READ	Request to read any mark in this order: OCR, DDM, BC.
SET_HW	Echoes the input argument.
VER	Request the firmware version of the wafer reader.

