PlateLoc ActiveX v3.1

User Guide

Agilent Technologies

Notices

User Guide Part Number

G5402-90000

June 2009 rebranded edition
October 2001 second edition of the PlateLoc User Guide

Contact Information

Agilent Technologies Inc.
Automation Solutions
5301 Stevens Creek Blvd.
Santa Clara, CA 95051
USA

Technical Support: 1.800.979.4811
or +1.408.345.8011
service.automation@agilent.com

Customer Service: 1.866.428.9811
or +1.408.345.8356
orders.automation@agilent.com

European Service: +44 (0)1763853638
euroservice.automation@agilent.com

Documentation feedback:
documentation.automation@agilent.com

Web:
www.agilent.com/lifesciences/automation

Acknowledgements

Microsoft and Windows are registered trademarks of the Microsoft Corporation in the United States and other countries.

Warranty

The material contained in this document is provided "as is," and is subject to being changed, without notice, in future editions. Further, to the maximum extent permitted by applicable law, Agilent disclaims all warranties, either express or implied, with regard to this manual and any information contained herein, including but not limited to the implied warranties of merchantability and fitness for a particular purpose. Agilent shall not be liable for errors or for incidental or consequential damages in connection with the furnishing, use, or performance of this document or of any information contained herein. Should Agilent and the user have a separate written agreement with warranty terms covering the material in this document that conflict with these terms, the warranty terms in the separate agreement shall control.

Technology Licenses

The hardware and/or software described in this document are furnished under a license and may be used or copied only in accordance with the terms of such license.

Restricted Rights Legend

If software is for use in the performance of a U.S. Government prime contract or subcontract, Software is delivered and licensed as "Commercial computer software" as defined in DFAR 252.227-7014 (June 1995), or as a "commercial item" as defined in FAR 2.101(a) or as "Restricted computer software" as defined in FAR 52.227-19 (June 1987) or any equivalent agency regulation or contract clause. Use, duplication or disclosure of Software is subject to Agilent Technologies' standard commercial license terms, and non-DOD Departments and Agencies of the U.S. Government will receive no greater than Restricted Rights as defined in FAR 52.227-19(c)(1-2) (June 1987). U.S. Government users will receive no greater than Limited Rights as defined in FAR 52.227-14 (June1987) or DFAR 252.227-7015 (b)(2) (November 1995), as applicable in any technical data.

Safety Noticies

A WARNING notice denotes a hazard. It calls attention to an operating procedure, practice, or the like that, if not correctly performed or adhered to, could result in personal injury or death. Do not proceed beyond a WARNING notice until the indicated conditions are fully understood and met.

A **CAUTION** notice denotes a hazard. It calls attention to an operating procedure, practice, or the like that, if not correctly performed or adhered to, could result in damage to the product or loss of important data. Do not proceed beyond a **CAUTION** notice until the indicated conditions are fully understood and met.

Chapter 1: Active X Documentation

The PlateLoc is designed to allow easy integration into an automated system. An ActiveX control is provided that installs on your computer, simplifying the programming needed to control the PlateLoc from your system. This document provides information on how to use Velocity11's ActiveX control within your Visual Basic or Visual C++ application. Please note that this documentation is for ActiveX version 3.1, and that all example code assumes that the PlateLoc control is instantiated as PlateLoc1.

NOTE

For manual operation of the PlateLoc, please see "Manually Sealing a Microplate" on page 20.

Installing PlateLoc Software

1) Take the compact disc marked "PlateLoc ActiveX" from its case and insert it into the CD drive on your PC.

NOTE

For the highest performance from your PlateLoc, we recommend a Pentium 166 or better based PC running Windows NT 4.0, Service Pack 6 or Windows 2000

2) Double-click **Setup.exe**. The **InstallShield Wizard** will appear.
3) Click **Next**. The **Customer Information** screen will appear.
4) Enter your name, company name, and PlateLoc serial number in the appropriate fields.
5) Click **Next**. The **Choose Destination Location** screen will appear.
6) The installation program will install the program on your C: drive unless you otherwise specify a location. To specify a location, click **Browse** and choose the desired destination folder. Otherwise, click **Next** and skip to step 9.
7) When the folder has been specified, click **Next**. The **Select Program Folder** screen will appear.
8) Click **Next** if you want the specified folder. If not, select a new folder and then click **Next**.
9) The **Setup Status** Screen will appear. When **InstallShield Wizard Complete** appears, click **Finish**.
10) Follow the directions for your particular development environment for adding an ActiveX to your application.

The PlateLoc ActiveX has been installed in the folder you specified. You are now ready to run the PlateLoc ActiveX from your computer.

Properties

short DesiredTemperature

`DesiredTemperature` is the temperature (in Celsius) needed to seal a plate.

GetDesiredTemperature

Description: Retrieves the current temperature set point (in Celsius) saved in the PlateLoc. If an error occurs, `GetDesiredTemperature()` will return -1 and an error event will be fired.

Parameters: None

Returns: short

Example:

Visual Basic	**Visual C++**

`PlateLoc1.Initialize 2`
`' sets the COM port to COM2`
`Dim desiredtemp As Short`
`desiredtemp =`
`  PlateLoc1.DesiredTemperature`
	`PlateLoc1.Initialize(2);`
	`// sets the COM port to COM2`
	`short iDesiredTemp;`
	`iDesiredTemp =`
	`  PlateLoc1.GetDesiredTemperature();`

SetDesiredTemperature

Description: Sets the temperature (in Celsius) used for a sealing cycle. Valid numbers are from 20 to 230, inclusive. If the user tries to set a temperature that is too low, the temperature is automatically set to 20 °C. Similarly, if the user tries to set a temperature that is too high, the temperature is automatically set to 230 °C. In either case, an error event is fired.

Parameters: short

Returns: None

Example:

Visual Basic	**Visual C++**

`PlateLoc1.Initialize 2`
`' sets the COM port to COM2`
`Dim temp As Short`
`temp = 196`
`PlateLoc1.DesiredTemperature = temp`
	`PlateLoc1.Initialize(2);`
	`// sets the COM port to COM2`
	`short itemp;`
	`itemp = 196;`
	`PlateLoc1.SetDesiredTemperature(itemp);`

short CurrentTemperature

`CurrentTemperature` is the present temperature (in Celsius) of the hot plate.

GetCurrentTemperature

Description: Retrieves the current temperature (in Celsius) of the PlateLoc's hot plate. If an error occurs, `GetCurrentTemperature()` will return -1 and an error event will be fired.

Parameters: None

Returns: short

Example:

Visual Basic	**Visual C++**

`PlateLoc1.Initialize 2`
`' sets the COM port to COM2`
`Dim temp As Short`
`temp = PlateLoc1.CurrentTemperature`
	`PlateLoc1.Initialize(2);`
	`// sets the COM port to COM2`
	`short iTemp;`
	`iTemp = PlateLoc1.GetCurrentTemperature();`

SetCurrentTemperature

Not implemented.

double SealingTime

`SealingTime` refers to the length of time needed to seal a microplate in seconds.

GetSealingTime

Description: Retrieves the current sealing time set in the PlateLoc's memory. If an error occurs, `GetSealingTime()` will return -1 and an error event will be fired.

Parameters: None

Returns: double

`Example:`

Visual Basic	**Visual C++**

`PlateLoc1.Initialize 2`
`' sets the COM port to COM2`
`Dim sealingtime As Double`
`sealingtime = PlateLoc1.SealingTime`
	`PlateLoc1.Initialize(2);`
	`// sets the COM port to COM2`
	`double fSealingTime;`
	`fSealingTime = PlateLoc1.GetSealingTime();`

SetSealingTime

Description: Sets the amount of time that the hot plate contacts the sealing foil. Valid numbers are from 0.5 to 12.0 seconds, inclusive. If a sealing time less than 0.5 seconds is specified, the sealing time will automatically be set to 0.5 seconds and an error event will be fired. Similarly, if a sealing time greater than 12.0 seconds is specified, the sealing time will automatically be set to 12.0 seconds and an error event will be fired.

Parameters: double

Returns: None

Example:

Visual Basic	**Visual C++**

`PlateLoc1.Initialize 2`
`' sets the COM port to COM2`
`Dim sealingtime As Double`
`sealingtime = 2.3`
`PlateLoc1.SealingTime = sealingtime`
	`PlateLoc1.Initialize(2);`
	`// sets the COM port to COM2`
	`double fSealingTime = 2.3;`
	`PlateLoc1.SetSealingTime(fSealingTime);`

short State

`State` contains information about the PlateLoc's current state of operation.

GetState

Description: Retrieves the PlateLoc state code. Refer to table 3 for a list of valid state codes.

Parameters: None

Returns: short

Example:

Visual Basic	**Visual C++**

`PlateLoc1.Initialize 2`
`' sets the COM port to COM2`
`Dim state As Short`
`state = PlateLoc1.State`
	`PlateLoc1.Initialize(2);`
	`// sets the COM port to COM2`
	`short iState;`
	`iState = PlateLoc1.GetState();`

SetState

Not implemented

State Codes

Table 3: State Protocol Definitions

Status Code	Definition
0	Ready
1	Running
2	OutOfTemp
3	Error
4	Setup Screen
5	Password Screen
6	Service Screen (service use only)
7	AD Values Screen (service use only)
8	Actuation Screen (service use only)
10	Threshold Screen (service use only)
11	Timing Screen (service use only)
12	Return Actuators
13	Debounce
14	Waiting to Run
15	Config Screen (service use only)
16	Fatal Error
17	Temperature Screen (service use only)

`short ErrorFlags`

`ErrorFlags` contain information about the last error asserted by the PlateLoc. A summary of the error flags is given in table 4.

`GetErrorFlags`

Description: Retrieves the PlateLoc's error flags.

Parameters: None

Returns: short

Example:

Visual Basic	Visual C++

`PlateLoc1.Initialize 2`
`Dim iErrorFlags As Short`
`iErrorFlags = PlateLoc1.ErrorFlags`
	`PlateLoc1.Initialize(2);`
	`short iErrorFlags;`
	`iErrorFlags = PlateLoc1.GetErrorFlags();`

SetErrorFlags

Not implemented

Table 4: Flag Bit Protocol Definitions

Bit	Definition
Bit 0	Set— Cycle Finished Clear— Cycle Not Finished
Bit 1	Set—Fatal Error
Bit 2	Set— Non-Fatal Error
Bit 3	Set— Insufficient Air/Vacuum Error
Bit 4	Set— Sensor Error
Bit 5-8	Spare

VARIANT ErrorString

`ErrorString` contains a text explanation of the last error asserted by the PlateLoc. A summary of the error strings is given in table 5.

GetErrorString

Description: Retrieves the PlateLoc's error strings.

Parameters: None

Returns: VARIANT

Example:

Visual Basic	Visual C++

`PlateLoc1.Initialize 2`
`' sets the COM port to COM2`
`Dim ErrorString As String`
`ErrorString = PlateLoc1.ErrorString`
	`PlateLoc1.Initialize(2);`
	`// sets the COM port to COM2`
	`VARIANT vErrorString;`
	`vErrorString = PlateLoc1.GetErrorString();`
	`CString sStatus;`
	`sStatus = vErrorString.bstrVal;`

SetErrorString

Not implemented

Table 5: Error Strings

Error String	Flag Bits Set
Transfer Plate Vacuum Error	2,3
Keystone Error	2
Low Air Pressure Error	2,3
Gripper Plate Vacuum Error	2,3
Hot Plate Vacuum Error	2,3
Overheat Error	2
Cycle Stopped Manually	2
No Plate in Holder	2,4
Temperature Sensor Error	2,4
Memory Access Error	1
Actuator Timeout Error	2
Serial Communications Error	2
Door Open During Cycle	2,4
Insufficient Vacuum Pressure	2,3

long ProcessTimeout

ProcessTimeout represents the allowable length of a sealing cycle in mS. If the cycle exceeds this length, an error is asserted. The default timeout is 10000 ms (10 seconds).

GetProcessTimeout

Description: Returns the current timeout in ms.

Parameters: None

Returns: long

Example:

Visual Basic	**Visual C++**

`PlateLoc1.Initialize 2`
`' sets the COM port to COM2`
`Dim iProcessTimeout As Long`
`iProcessTimeout =`
`  PlateLoc1.ProcessTimeout`
	`PlateLoc1.Initialize(2);`
	`// sets the COM port to COM2`
	`long iProcessTimeout;`
	`iProcessTimeout =`
	`  PlateLoc1.GetProcessTimeout();`

SetProcessTimeout

Description: Sets a new process length in ms.

Parameters: long

Returns: None

Example:

Visual Basic	**Visual C++**

`PlateLoc1.Initialize 2`
`' sets the COM port to COM2`
`PlateLoc1.ProcessTimeout = 10000`
`'Set the process timeout to 10 seconds`
	`PlateLoc1.Initialize(2);`
	`// sets the COM port to COM2`
	`PlateLoc1.SetProcessTimeout(10000);`
	`//Set the process timeout to 10 seconds`

ControlPicture

Description: Retrieves a picture of the PlateLoc bitmap that can be used in the container's application.

Parameters: None

Returns: IPictureDisp

Example: In this example, we will paint the PlateLoc bitmap over a button.

Visual Basic	**Visual C++**

``' Assume that there is a button named
' Command1 on the current form. You
' must set the Style property of
' Command1 to "Graphical"
Command1.Picture =
   PlateLoc1.ControlPicture```
	``/* The CPicture class will be imported into
	   your project when the ActiveX is
	   installed */
	CButton button; // Create a button
	CPicture PlateLocPic;
	PlateLocPic = PlateLoc.GetControlPicture();
	  // Retrieve the picture
	button.SetBitmap((HBITMAP)
	   PlateLocPic.GetHandle());
	// Paint the bitmap onto the button```

Methods

Initialize

Description: Initializes the COM port that the PlateLoc is attached to. This method must be called before any other commands can be issued to the PlateLoc.

Parameters: short

Returns: None

Example:

Visual Basic	**Visual C++**
```PlateLoc1.Initialize 2 ' sets the COM port to COM2```	```PlateLoc1.Initialize(2); // sets the COM port to COM2```

ShowDiagsDialog

Description: Opens a dialog box that allows you to easily test the PlateLoc. From this dialog, you may set the sealing time, sealing temperature, read the PlateLoc status codes, and start a sealing cycle.

Parameters: None

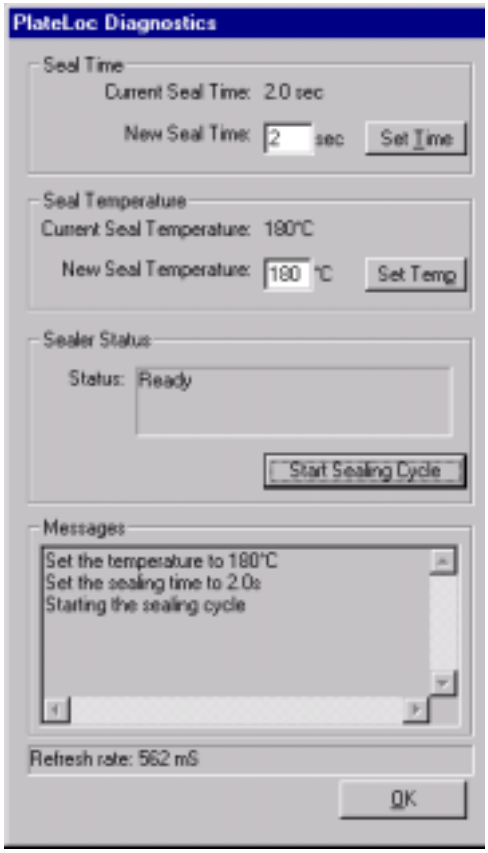
Returns: None

Example:

Visual Basic	**Visual C++**
```PlateLoc1.Initialize 2 ' sets the COM port to COM2 PlateLoc1.ShowDiagsDialog```	```PlateLoc1.Initialize(2); ' sets the COM port to COM2 PlateLoc1.ShowDiagsDialog();```

The following screen will appear when `ShowDiagsDialog` is called (see figure 13).

Figure 13: PlateLoc Diagnostics Screen



`Close`

Description: Closes serial port used by the PlateLoc ActiveX. After calling `Close`, the method `Initialize` must be called again before resuming communications with the PlateLoc.

Parameters: None

Returns: None

Example:

Visual Basic	**Visual C++**

`PlateLoc1.Initialize 2`
`' sets the COM port to COM2`
`PlateLoc1.SealingTime = 2.3`
`PlateLoc1.Close`
`' Free up the serial port`
	`PlateLoc1.Initialize(2);`
	`// sets the COM port to COM2`
	`PlateLoc1.SetSealingTime(2.3);`
	`PlateLoc1.Close();`
	`// Free up the serial port`

StartCycle

Description: Starts the PlateLoc sealing cycle.

Parameters: None

Returns: None

Example:

Visual Basic	**Visual C++**

`PlateLoc1.Initialize 2`
`' sets the COM port to COM2`
`PlateLoc1.StartCycle`
	`PlateLoc1.Initialize(2);`
	`// sets the COM port to COM2`
	`PlateLoc1.StartCycle();`

StopCycle

Description: Aborts the current cycle.

Parameters: None

Returns: None

Example:

Visual Basic	**Visual C++**

`Public Declare Sub Sleep Lib "kernel32" Alias "Sleep" (ByVal dwMilliseconds As Long)`
 	
`PlateLoc1.Initialize 2`
`'Set up on COM 2`
`PlateLoc1.StartCycle`
`'Start up a seal cycle`
`Sleep 1000`
`'Wait 1 second`
`PlateLoc1.StopCycle`
`'Kill cycle`
	`PlateLoc1.Initialize(2);//Set up on COM 2`
	`PlateLoc1.StartCycle();`
	`// Start up a seal cycle`
	`Sleep (1000); // Wait 1 second`
	`PlateLoc1.StopCycle(); //Kill cycle`

Events

Events are fired asynchronously by the ActiveX to notify the container that a procedure has finished or an error has occurred. Consult Microsoft's ActiveX documentation on how to handle events in your Visual C++ or Visual Basic code.

`CycleCompleted`

Description: This event fires whenever the sealing cycle has completed.

Parameters: None

`Error`

Description: This stock event fires whenever any fatal or non-fatal PlateLoc error has occurred. The SCODE parameter can be used by the container application to determine what kind of error occurred.

Parameters: (short Number, BSTR FAR* Description, SCODE Scode, LPCTSTR Source, LPCTSTR HelpFile, long HelpContext, BOOL FAR* CancelDisplay)

Table 6: SCodes

Description	SCode
Communication port failed to open	32767
Time is invalid	32766
Temperature is invalid	32765
Could not create status thread	32764
Could not create temperature thread	32763
Communication port is not open	32762
Waiting for status	32761
Waiting for temperature	32760
Could not create cycle thread	32759
Cycle error	32758
No dialog thread	32757
Waiting for current sealing time	32756
Waiting for desired temperature	32755
Cycle never started	32754
No response to cycle start command	32753
Cycle start command not acknowledged	32752
No response to `Settime` command	32751

Table 6: SCodes

Description	SCode
`Settime` not acknowledged	32750
Error with `Settime` command	32749
No response to `Settime` command	32748
`Settemp` command not acknowledged	32747
Error with `Settemp` command	32746
No response to `Clearerror` command	32745
`Clearerror` command was not acknowledged	32744
Error with `Clearerror` command	32743
Cycle did not finish	32742
Non runtime error	32741
No response	32740
Error with `Abortcycle` command	32739
`Abortcycle` command not acknowledged	32738
No response to `Abortcycle` command	32737

Agilent Technologies

User Guide

G5402-90000