

Agilent Microplate Labeler ActiveX

Version 16.0.0

User Guide

Original Instructions



Agilent Technologies

Notices

© Agilent Technologies, Inc. 2012

No part of this manual may be reproduced in any form or by any means (including electronic storage and retrieval or translation into a foreign language) without prior agreement and written consent from Agilent Technologies, Inc. as governed by United States and international copyright laws.

User Guide Part Number

G5404-90008

Edition

Revision 00, April 2012

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)1763208826
euroservice.automation@agilent.com

Documentation feedback:
documentation.automation@agilent.com

Web:
www.agilent.com/lifesciences/automation

Acknowledgements

Microsoft® and Windows® are either registered trademarks or 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 sub-contract, 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

(June 1987) or DFAR 252.227-7015 (b)(2) (November 1995), as applicable in any technical data.

Safety Notices

 **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.

Contents

Preface	iii
About this guide	iv
Reporting problems	iv
Microplate Labeler ActiveX control	1
About ActiveX controls	2
Properties	3
Methods	7
Events	26

Contents



Preface

This preface contains the following topics:

- “About this guide” on page iv
- “Reporting problems” on page iv

About this guide

What this guide covers

This guide describes the ActiveX controls for the Agilent Microplate Labeler. This guide does not provide instructions for setting up and using the Microplate Labeler. For these details, see the [G5404B Microplate Labeler User Guide](#).

Accessing Agilent Technologies Automation Solutions user guides

You can search the online knowledge base or download the latest version of any PDF file from the Agilent Technologies website at www.agilent.com/lifesciences/automation. Safety information for the devices appears in the corresponding device user guide. You can also search the knowledge base or the PDF files for safety information.

Related topics

For information about...	See...
How to set up and use the Agilent Microplate Labeler	G5404B Microplate Labeler User Guide
Reporting problems	“Reporting problems” on page iv

Reporting problems

Contacting Automation Solutions Technical Support

If you find a problem with the Microplate Labeler, contact Automation Solutions Technical Support. For contact information, see Notices on the back of the title page.

Reporting hardware problems

When contacting Agilent Technologies, make sure you have the serial number of the device ready.

Reporting software problems

When you contact Automation Solutions Technical Support, make sure you provide the following:

- Short description of the problem

- Relevant software version number (for example, automation control software, diagnostics software, ActiveX control software, and firmware)
- Error message text (or screen capture of the error message dialog box)
- Relevant files, such as log files

Reporting user guide problems

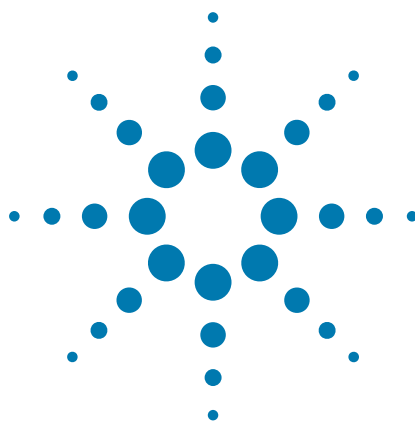
If you find a problem with this user guide or have suggestions for improvement, send your comments in an email to documentation.automation@agilent.com.

Related topics

For information about...	See...
How to set up and use the Agilent Microplate Labeler	<i>G5404B Microplate Labeler User Guide</i>
Accessing user information	“Accessing Agilent Technologies Automation Solutions user guides” on page iv

Preface

Reporting problems



Microplate Labeler ActiveX control

This chapter gives integrators the ActiveX control information required to integrate the Agilent Microplate Labeler into another company's lab automation system. The ActiveX has been verified to work with both Visual C++ and Visual Basic .NET.

This chapter contains the following topics:

- “About ActiveX controls” on page 2
- “Properties” on page 3
- “Methods” on page 7
- “Events” on page 27

About ActiveX controls

What is the Microplate Labeler ActiveX control

The Microplate Labeler ActiveX control is the software component that allows third-party lab automation systems to interact with the Microplate Labeler.

How the Microplate Labeler ActiveX control is used

In an Agilent Technologies automation system that is running the VWorks software, ActiveX interfaces are not used to communicate with devices. However, some integrations, such as those with LIMS, require that a third-party application control the Microplate Labeler. The Microplate Labeler ActiveX control enables third-party applications to interface with the Microplate Labeler.

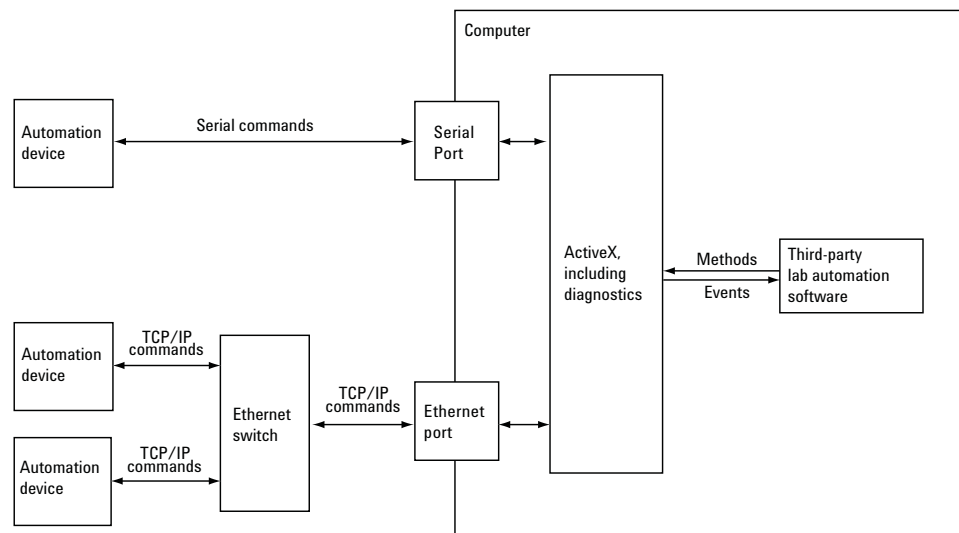
Each ActiveX control consists of a collection of the following:

- *Methods*. Functions that can be called to invoke individual operations
- *Properties*. Attributes or features of the ActiveX control
- *Events*. Notifications that methods have completed or resulted in errors

To ensure proper integration, you must know the available methods and properties for the ActiveX control.

The following diagram illustrates the use of the Microplate Labeler ActiveX control in a lab automation system environment. Actions you perform are conducted through ActiveX methods. System responses are relayed back through ActiveX events or through return values and variables passed to methods.

Note: Although the Microplate Labeler ActiveX control generates events, the third-party application must implement handlers for them.



Properties

Blocking

VARIANT_BOOL Blocking

Description

Determines whether methods should block until completion or return immediately for asynchronous operation.

Acceptable values

- VARIANT_TRUE (C++) or True (Visual Basic .NET). The ActiveX is forced to block or wait until a method completes before it returns control to the caller.
- VARIANT_FALSE (C++) or False (Visual Basic .NET). Returns control to the application immediately, and the caller should handle the events accordingly.

Default value

VARIANT_FALSE or False

Blocking affects some methods differently. See each method's description for the effect. Unless otherwise noted:

- In non-blocking mode (Block = False), a method:
 - Starts another thread of execution to perform the given method, returning control to the application immediately.
 - Returns 0 on launching new thread successfully; otherwise returns nonzero.
 - If the method is successful, an event indicating completion is sent; if unsuccessful, an Error event is sent.
- In blocking mode (Block = True), a method:
 - Is executed.
 - Returns 0 if it completes successfully; returns nonzero otherwise.
- Error message can be reviewed by calling GetLastError().

Visual C++ example

```
//set the Microplate Labeler in blocking mode
VARIANT_BOOL blocking=VARIANT_TRUE;
m_MicroplateLabeler.PutBlocking(blocking);
//set the Microplate Labeler in non-blocking mode
blocking=VARIANT_FALSE;
m_MicroplateLabeler.PutBlocking(blocking);
//returns the blocking value
blocking=m_MicroplateLabeler.GetBlocking();
//user should handle events if non-blocking mode is
selected!
```

Visual Basic .NET example

```
'set Microplate Labeler in blocking mode
MicroplateLabeler1.Blocking=True
'set Microplate Labeler in non-blocking mode
MicroplateLabeler1.Blocking=False
'returns the blocking value
Dim bMode as Boolean
bMode= MicroplateLabeler1.Blocking
'user should handle events if non-blocking mode is
selected!
```

ControlPicture

IPictureDisp*ControlPicture

Description

A read-only picture of the Microplate Labeler that can be used in the container's application.

Parameters

None

Visual C++ example

```
/*the CPicture class will be imported in to your project  
When the ActiveX is installed*/  
CButton button;  
//create button  
button.Create("Button", WS_CHILD | WS_VISIBLE | BS_BITMAP,  
CRect(10, 10, 60, 60), pParentWnd/*pointer of parent  
window*/, 1);  
CPicture MicroplateLabelerPic;  
//retrieve the picture  
MicroplateLabelerPic=m_MicroplateLabeler.GetControlPicture()  
();  
//paint the bitmap onto the button  
button.SetBitmap((HBITMAP)  
MicroplateLabelerPic.GetHandle());
```

Visual Basic .NET example

```
Dim iPicture As System.Drawing.Image=  
MicroplateLabeler1.ControlPicture()  
button.BackgroundImage=iPicture
```

NumReprintAttempts

LONG NumReprintAttempts

Description

If a barcode reader is attached to the Microplate Labeler, setting this value to a non-zero integer enables barcode verification after printing and applying labels.

If the barcodes cannot be read, the Microplate Labeler will reprint and apply up to NumReprintAttempts times before giving an error.

Parameters

None

Visual C++ example

```
m_MicroplateLabeler.SetNumReprintAttempts(5);  
LONG Reprints=  
m_MicroplateLabeler.GetNumReprintAttempts();
```

Visual Basic .NET example

```
MicroplateLabeler1.NumReprintAttempts=5  
Dim iReprints as Integer  
iReprints=MicroplateLabeler1.NumReprintAttempts
```

Related topics

For information about...	See...
Microplate Labeler ActiveX methods	“Methods” on page 7
Microplate Labeler ActiveX events	“Events” on page 27
Overview of ActiveX controls	“About ActiveX controls” on page 2
Reporting problems	“Reporting problems” on page iv

Methods

Abort

```
LONG Abort( )
```

Description

Aborts a current task that is in the error state and clears the error.

Parameters

None

Returns

0 if successful; Other value if there was an error

Visual C++ example

```
LONG lres=m_MicroplateLabeler.Abort();
```

Visual Basic .NET example

```
Dim ires as Integer  
ires=MicroplateLabeler1.Abort()
```

AboutBox

```
void AboutBox( )
```

Description

Displays the Microplate Labeler About dialog box that contains the ActiveX and firmware version numbers.

Parameters

None

Returns

None

Visual C++ example

```
m_MicroplateLabeler.AboutBox();
```

Visual Basic .NET example

```
MicroplateLabeler1.AboutBox()
```

AddToQueue

```
LONG AddToQueue(SHORT format, BYTE side, VARIANT_BOOL  
dropStage, BSTR field0, BSTR field1, BSTR field2, BSTR  
field3, BSTR field4, BSTR field5);
```

Description

If multiple label formats and data need to be printed on each side of the microplate, use AddToQueue for adding formats by their index to the queue. ClearQueue must be called after the print-and-apply cycle to prevent excess labels from being applied to the microplate.

Parameters

Name	Type	Range	Description
format	SHORT	1-20	Index of format to use
side	BYTE	1, 2, 4, or 8	1 = east side 2 = north side 4 = west side 8 = south side The side is a bitmask, so if you want to apply label on the north and south sides, the parameter is 10 (2+8).
dropStage	VARIANT_BOOL	VARIANT_TRUE/ VARIANT_FALSE	VARIANT_TRUE = drop stage before applying label VARIANT_FALSE = raise stage before applying label
field0	BSTR	Not applicable	Data for field 1
field1	BSTR	Not applicable	Data for field 2
field2	BSTR	Not applicable	Data for field 3
field3	BSTR	Not applicable	Data for field 4
field4	BSTR	Not applicable	Data for field 5
field5	BSTR	Not applicable	Data for field 6

Returns

0 if successful; Other value if there was an error

Visual C++ example

```
/* Add a label using format 1, east side, drop the stage
before printing, and print "hi" in field 1, and "there" in
field 2.*/
LONG lres= m_MicroplateLabeler1.AddToQueue( 1, 1,
VARIANT_TRUE, "hi", "there", "", "", "", "");
```

Visual Basic .NET example

```
Dim ires as Integer
'Add a label using format 1, east side, drop the stage
'before printing, and print "hi" in field 1, and "there"
'in field 2.
ires=MicroplateLabeler1.AddToQueue (1, 1, True, "hi",
"there", "", "", "", "")
```

AddToQueueByFormatName

```
LONG AddToQueueByFormatName(BSTR format, BYTE side,
VARIANT_BOOL dropStage, BSTR field0, BSTR field1, BSTR
field2, BSTR field3, BSTR field4, BSTR field5);
```


Description

If multiple label formats and data need to be printed on each side of the plate, use `AddToQueueByFormatName` for adding formats by their name to the queue. `ClearQueue` must be called after the print-and-apply cycle to prevent excess labels from being applied to the plate.

Parameters

Name	Type	Range	Description
format	BSTR	All existing label format names	Name of the label format to use
side	BYTE	1, 2, 4, or 8	1 = east side 2 = north side 4 = west side 8 = south side The side is a bitmask, so if you want to apply label on the north and south sides, the parameter is 10 (2+8).
dropStage	VARIANT_BOOL	VARIANT_TRUE/ VARIANT_FALSE	TRUE = drop stage before applying label FALSE = raise stage before applying label
field0	BSTR	Not applicable	Data for field 1
field1	BSTR	Not applicable	Data for field 2
field2	BSTR	Not applicable	Data for field 3
field3	BSTR	Not applicable	Data for field 4
field4	BSTR	Not applicable	Data for field 5
field5	BSTR	Not applicable	Data for field 6

Returns

0 if successful; Other value if there was an error

Visual C++ example

```

/* Add a label using format "Format ABC", east side, drop
the stage before printing, and print "hi" in field 1, and
"there" in field 2.*/

LONG lres=
m_MicroplateLabeler1.AddToQueueByFormatName("Format ABC",
1, VARIANT_TRUE, "hi", "there", "", "", "", "");

```

Visual Basic .NET example

```

Dim ires as Integer

'Add a label using format "Format ABC", east side, drop
the stage 'before printing, and 'print "hi" in field 1,
and "there" in field 2.

ires=MicroplateLabeler1.AddToQueueByFormatName ("Format
ABC", 1, True, "hi", "there", "", "", "", "")

```

ClearQueue

```
LONG ClearQueue( )
```

Description

Clears the label queue. It should be called after each call to PrintAndApplyQueue.

Parameters

None

Returns

0 if successful; Other value if there was an error

Visual C++ example

```
LONG lres=m_MicroplateLabeler1.ClearQueue();
```

Visual Basic .NET example

```

Dim ires as Integer

ires=MicroplateLabeler1.ClearQueue()

```

Close

```
LONG Close( )
```

Description

Disconnects from the Microplate Labeler. After successful closing, the CloseComplete event is sent.

Parameters

None

Returns

0 if successful; Other value if there was an error

Visual C++ example

```
LONG lres=m_MicroplateLabeler1.Close();
```

Visual Basic .NET example

```
Dim ires as Integer
ires=MicroplateLabeler1.Close()
```

DropStage

```
LONG DropStage(VARIANT_BOOL drop)
```

Description

Drops or raises the plate stage. After DropStage operation is successful, the DropStageComplete event is sent.

Name	Type	Range	Description
drop	VARIANT_BOOL	VARIANT_TRUE/ VARIANT_FALSE	VARIANT_TRUE = drop stage before applying label VARIANT_FALSE = raise stage before applying label

Returns

0 if successful; Other value if there was an error

Visual C++ example

```
//Drops the stage.
LONG lres=m_MicroplateLabeler.DropStage(VARIANT_TRUE);
```

Visual Basic .NET example

```
Dim ires as Integer
'Raises the stage
ires=MicroplateLabeler1.DropStage(False)
```

EnumerateProfiles

```
VARIANT EnumerateProfiles( )
```

Description

Retrieves a list of defined profiles. The strings in this array are the profile names that should be used for the Initialize method.

Parameters

None

Returns

An array of profile names

Visual C++ example

```
VARIANT  
vProfiles=m_MicroplateLabeler1.EnumerateProfiles();  
SAFEARRAY *psa=vProfiles.parray;  
BSTR *bstrArray;  
if  
(FAILED(SafeArrayAccessData(psa,reinterpret_cast<void**>( &bstrArray))))  
{  
    VariantClear(&vProfiles);  
    return;  
}  
for(ULONG i=0;i<psa->rgsabound[0].cElements;i++)  
{  
    MessageBox(CString(bstrArray[i]));  
}  
SafeArrayUnaccessData(psa);  
VariantClear(&vProfiles);
```

Visual Basic .NET example

```
Dim i as Integer  
Dim sProfiles() As String  
sProfiles=MicroplateLabeler1.EnumerateProfiles()  
For i=0 To sProfiles.GetLength(0)-1  
    MsgBox sProfiles(i)  
Next
```

EnumerateAllFormatNames

```
VARIANT EnumerateAllFormatNames( )
```

Description

Retrieves a list of format names from the printer. The strings in this array are the profile names that should be used for the AddToQueueByFormatName, PrintLabelByFormatName, and PrintAndApplyByFormatName methods.

Parameters

None

Returns

An array of profile names

Visual C++ example

```
VARIANT
vFormatNames=m_MicroplateLabeler1.EnumerateAllFormatNames
();
SAFEARRAY *psa= vFormatNames.parray;
BSTR *bstrArray;
if
(FAILED(SafeArrayAccessData(psa,reinterpret_cast<void**>(
&bstrArray))))
{
VariantClear(&vFormatNames);
return;
}
for(ULONG i=0;i<psa->rgsabound[0].cElements;i++)
{
MessageBox(CString(bstrArray[i]));
}
SafeArrayUnaccessData(psa);
VariantClear(&vFormatNames);
```

Visual Basic .NET example

```
Dim i as Integer
Dim sFormatNames () As String
sFormatNames=
MicroplateLabeler1.EnumerateAllFormatNames()
For i=0 To sFormatNames.GetLength(0)-1
MsgBox sFormatNames (i)
Next
```

EnumerateAllFormats

```
LONG EnumerateAllFormats( BSTR* strFormats)
```

Description

Returns the list of label formats and the list of fields for each format. The data is retrieved in XML format.

Parameters

Name	Type	Description
strFormats	BSTR*	XML string containing retrieved label formats with their fields

Returns

An array of proile names

Visual C++ example

```
BSTR strFormats;  
LONG  
lres=m_MicroplateLabeler.EnumerateAllFormats(&strFormats)  
;
```

Visual Basic .NET example

```
Dim ires as Integer  
Dim sXMLFormats As String  
ires=MicroplateLabeler1.EnumerateAllFormats(sXMLFormats)
```

GetActiveXVersion

```
BSTR GetActiveXVersion( )
```

Description

Retrieves the Microplate Labeler ActiveX version number.

Parameters

None

Returns

ActiveX version number (string)

Visual C++ example

```
CString  
ActiveXVer=m_MicroplateLabeler1.GetActiveXVersion();
```

Visual Basic .NET example

```
Dim sVersion As String  
sVersion=MicroplateLabeler1.GetActiveXVersion()
```

GetBarcodeField

```
LONG GetBarcodeField(SHORT format, SHORT* field);
```

Description

Allows the user to retrieve the index of the barcode field in the specified label format. Ideal for verifying the printed barcode.

Parameters

Name	Type	Range	Description
format	SHORT	1-20	Specifies the label format to check.
field	SHORT*	Not applicable	Stores the barcode field index after returning from the call.

Returns

0 if successful; Other value if there was an error

Visual C++ example

```
SHORT BarcodeFieldIndx;  
LONG lres=  
m_MicroplateLabeler1.GetBarcodeField(1,&BarcodeFieldIndx)  
;  
//Field now contains the barcode field index,0-based.
```

Visual Basic .NET example

```
Dim BarcodeFieldIndx As Short  
Dim ires as Integer=  
MicroplateLabeler1.GetBarcodeField(1,BarcodeFieldIndx)
```

GetFirmwareVersion

```
BSTR GetFirmwareVersion( )
```

Description

Retrieves the Microplate Labeler firmware version number. The firmware version is only available after profile is initialized.

Parameters

None

Returns

Firmware version number (string)

Visual C++ example

```
CString FirmwareVer=  
m_MicroplateLabeler1.GetFirmwareVersion();
```

Visual Basic .NET example

```
Dim sFirmwareVersion As String  
sFirmwareVersion= MicroplateLabeler1.GetFirmwareVersion()
```

GetLastError

```
BSTR GetLastError( )
```

Description

Retrieves the last known error condition.

Parameters

None

Returns

An error string.

Visual C++ example

```
CString str=m_MicroplateLabeler1.GetLastError();
```

Visual Basic .NET example

```
Dim sError As String=""
sError=MicroplateLabeler1.GetLastError()
```

GetRemainingLabels

```
LONG GetRemainingLabels(long *numRemaining);
```

Description

Returns number of remaining labels in the roll. This number can also be found on 'I/O Functions' tab in diagnostics.

Parameters

Name	Type	Description
*numRemaining	LONG	The number of labels remaining.

Return

0 if successful; Other value if there was an error

Visual C++ example

```
LONG numRemaining;
LONG lres=
m_MicroplateLabeler1.GetRemainingLabels(&numRemaining);
```

Visual Basic .NET example

```
Dim numRemaining As Integer
Dim lres as Integer=MicroplateLabeler1.
GetRemainingLabels (numRemaining)
```

HasBCR

```
VARIANT_BOOL HasBCR( )
```

Description

Returns whether or not the Microplate Labeler has detected a barcode reader.

Parameters

None

Return

VARIANT_TRUE = barcode reader was detected

VARIANT_FALSE = barcode reader wasn't detected

Visual C++ example

```
VARIANT_BOOL hasBCR=m_MicroplateLabeler.HasBCR();
```

Visual Basic .NET example

```
Dim hasBCR as Boolean
hasBCR=MicroplateLabeler1.HasBCR()
```

HomeStage

```
LONG HomeStage( )
```

Description

Homes the Microplate Labeler stage, and sends HomeStageComplete event when the motor has finished homing.

Parameters

None

Returns

0 if successful; Other value if there was an error

Visual C++ example

```
LONG lres=m_MicroplateLabeler.HomeStage();
```

Visual Basic .NET example

```
Dim ires as Integer
ires=MicroplateLabeler1.HomeStage()
```

Ignore

```
LONG Ignore( )
```

Description

Ignores the previously issued error and moves to the next step in the task. This is not a recommended course of action, as the errors are issued for a reason. However, ignoring some errors can be appropriate if the operator understands the implications.

Parameters

None

Returns

0 if successful; Other value if there was an error

Visual C++ example

```
LONG lres=m_MicroplateLabeler1.Ignore();
```

Visual Basic .NET example

```
Dim ires as Integer
ires=MicroplateLabeler1.Ignore()
```

Initialize

```
LONG Initialize(BSTR Profile)
```

Description

Initializes the profile and starts communication with the Microplate Labeler using the parameters set in the profile. The profile specifies the serial or Ethernet connection used to communicate with the Microplate Labeler. The parameters for each profile can be adjusted in the Diagnostics dialog box (by a call to the ShowDiagsDialog method) on the Profiles page.

Parameters

Name	Type	Description
Profile	BSTR	The name of the profile to be used for initialization.

Returns

0 if successful, and sends the InitializeComplete event; Other value if there was an error

Visual C++ example

```
//connect via serial connection specified in the profile
LONG lres=
m_MicroplateLabeler.Initialize(_bstr_t("MicroplateLabeler
Serial profile"));
```

Visual Basic .NET example

```
'connect via serial connection specified in the profile
Dim ires as Integer
ires=MicroplateLabeler1.Initialize("MicroplateLabler
Serial profile")
```

PrintAndApply

```
LONG PrintAndApply(SHORT format, BYTE sides, VARIANT_BOOL
dropStage, BSTR field0, BSTR field1, BSTR field2, BSTR
field3, BSTR field4, BSTR field5)
```

Description

Prints and applies one set of label data to specified side(s) of the microplate. After print and apply operation is successful, the PrintAndApplyComplete event is sent.

Parameters

Name	Type	Range	Description
format	SHORT	1-20	Index of format to use
side	BYTE	1, 2, 4, or 8	1 = east side 2 = north side 4 = west side 8 = south side The side is a bitmask, so if you want to apply label on the north and south sides, the parameter is 10 (2+8).

Name	Type	Range	Description
dropStage	VARIANT_BOOL	VARIANT_TRUE/ VARIANT_FALSE	VARIANT_TRUE = drop stage before applying label VARIANT_FALSE = raise stage before applying label
field0	BSTR	Not applicable	Data for field 1
field1	BSTR	Not applicable	Data for field 2
field2	BSTR	Not applicable	Data for field 3
field3	BSTR	Not applicable	Data for field 4
field4	BSTR	Not applicable	Data for field 5
field5	BSTR	Not applicable	Data for field 6

Return

0 if successful; Other value if there was an error

Visual C++ example

```
/* Print and apply "hi" and "there" using format 2 on the
west and south sides. According to the table, west = 4 and
south = 8, so pass 12 in the for the sides parameter.*/
LONG lres=
m_MicroplateLabeler1.PrintAndApply(2,12,VARIANT_TRUE,
"hi", "there", "", "", "", "");
```

Visual Basic .NET example

```
Dim ires as Integer
'Print and apply "hi" and "there" using format 2 on the
'west and south sides. According to the table, west = 4
'and south = 8, so pass 12 in the for the sides
'parameter.
ires=MicroplateLabeler1.PrintAndApply(2, 12, True, "hi",
"there", "", "", "", "")
```

PrintAndApplyByFormatName

```
LONG PrintAndApplyByFormatName(BSTR format, BYTE sides,
VARIANT_BOOL dropStage, BSTR field0, BSTR field1, BSTR
field2, BSTR field3, BSTR field4, BSTR field5)
```

Description

Prints and applies one set of label data to specified side(s) of the microplate. After print and apply operation is successful, the PrintAndApplyComplete event is sent.

Parameters

Name	Type	Range	Description
format	BSTR	All existing label format names	Name of the label format to use
side	BYTE	1, 2, 4, or 8	1 = east side 2 = north side 4 = west side 8 = south side The side is a bitmask, so if you want to apply label on the north and south sides, the parameter is 10 (2+8).
dropStage	VARIANT_BOOL	VARIANT_TRUE/ VARIANT_FALSE	VARIANT_TRUE = drop stage before applying label VARIANT_FALSE = raise stage before applying label
field0	BSTR	Not applicable	Data for field 1
field1	BSTR	Not applicable	Data for field 2
field2	BSTR	Not applicable	Data for field 3
field3	BSTR	Not applicable	Data for field 4
field4	BSTR	Not applicable	Data for field 5
field5	BSTR	Not applicable	Data for field 6

Return

0 if successful; Other value if there was an error

Visual C++ example

```
/* Print and apply "hi" and "there" using format 2 on the
west and south sides. According to the table, west = 4 and
south = 8, so pass 12 in the for the sides parameter.*/
LONG lres=
m_MicroplateLabeler1.PrintAndApplyByFormatName("Format
ABC",12,VARIANT_TRUE, "hi", "there", "", "", "", "");
```

Visual Basic .NET example

```
Dim ires as Integer
'Print and apply "hi" and "there" using format "Format
'ABC" on the west and south sides. According to the table,
```

```
'west = 4 and south = 8, so pass 12 in the for the sides
'parameter.
ires=MicroplateLabeler1.PrintAndApplyByFormatName("Format
ABC",, 12, True, "hi", "there", "", "", "", "")
```

PrintLabel

```
LONG PrintLabel(SHORT format, BSTR field0, BSTR field1, BSTR
field2, BSTR field3, BSTR field4, BSTR field5)
```

Description

Prints a label with specified parameters, where format is specified by index instead of name. After print operation is successful, the PrintLabelComplete event is sent.

Parameters

Name	Type	Range	Description
format	SHORT	1-20	Index of format to use
field0	BSTR	Not applicable	Data for field 1
field1	BSTR	Not applicable	Data for field 2
field2	BSTR	Not applicable	Data for field 3
field3	BSTR	Not applicable	Data for field 4
field4	BSTR	Not applicable	Data for field 5
field5	BSTR	Not applicable	Data for field 6

Return

0 if successful; Other value if there was an error

Visual C++ example

```
// Print "hi" and "there" using format 2  
LONG lres= m_MicroplateLabeler1.PrintLabel(2,"hi",  
"there", "", "", "", "", "");
```

Visual Basic .NET example

```
Dim ires as Integer  
'Print and apply "hi" and "there" using format 2  
ires=MicroplateLabeler1.PrintLabel(2,"hi","there", "",  
"", "", "")
```

PrintLabelByFormatName

```
LONG PrintLabelByFormatName(BSTR format, BSTR field0, BSTR  
field1, BSTR field2, BSTR field3, BSTR field4, BSTR field5)
```

Description

Prints a label with specified parameters, where format is specified by the name. After print operation is successful, the PrintLabelComplete event is sent.

Parameters

Name	Type	Range	Description
format	BSTR	All existing label format names	Name of the label format to use
field0	BSTR	Not applicable	Data for field 1
field1	BSTR	Not applicable	Data for field 2
field2	BSTR	Not applicable	Data for field 3
field3	BSTR	Not applicable	Data for field 4
field4	BSTR	Not applicable	Data for field 5
field5	BSTR	Not applicable	Data for field 6

Returns

0 if successful; Other value if there was an error

Visual C++ example

```
// Print "hi" and "there" using format "Format ABC"
LONG lres=
m_MicroplateLabeler1.PrintLabelbyFormatName("Format
ABC","hi", "there", "", "", "", "");
```

Visual Basic .NET example

```
Dim ires as Integer
'Print "hi" and "there" using format "Format ABC"
ires=MicroplateLabeler1.PrintLabelbyFormatName("Format
ABC","hi", "there", "", "", "", "")
```

PrintLabelQueue

```
LONG PrintLabelQueue()
```

Description

Prints all of the labels added to the label queue using the AddToQueue method. After print operation is done for all label formats in the queue, the PrintLabelComplete event is sent.

Parameters

None

Returns

0 if successful; Other value if there was an error

Visual C++ example

```
LONG lres=m_MicroplateLabeler.PrintLableQueue();
```

Visual Basic .NET example

```
Dim ires as Integer
ires=MicroplateLabeler1.PrintLableQueue()
```

PrintAndApplyQueue

```
LONG PrintAndApplyQueue ()
```

Description

Prints and applies all of the labels added to the label queue using the AddToQueue method. After print and apply operation is done for all label formats in the queue, the PrintAndApplylComplete event is sent.

Parameters

None

Returns

0 if successful; Other value if there was an error

Visual C++ example

```
LONG lres=m_MicroplateLabeler.PrintAndApplyQueue();
```

Visual Basic .NET example

```
Dim ires as Integer
ires=MicroplateLabeler1.PrintAndApplyQueue()
```

ReadBarcode

```
LONG ReadBarcode(BYTE side, BSTR* barcode);
```

Description

Reads a barcode on specified side of the microplate. After read barcode operation is complete, the ReadBarcodeComplete event is sent.

Parameters

Name	Type	Range	Description
side	BYTE	0, 1, 2, 4, or 8	0 = current stage position 1 = east side 2 = north side 4 = west side 8 = south side
barcode	BSTR*	Not applicable	The barcode string read by BCR in blocking mode. NULL in non-blocking mode. User can get the barcode string from the parameter in ReadBarcodeComplete event.

Return

0 if successful; Other value if there was an error

Visual C++ example

```
BSTR bc;
// Read the barcode at the current stage position.
m_MicroplateLabeler.ReadBarcode( 0, &bc);
SysFreeString( bc);
// Read the barcode on the north side of the plate.
m_MicroplateLabeler.ReadBarcode( 2, &bc);
SysFreeString( bc);
```

Visual Basic .NET example

```
Dim bc As String
Dim ires as Integer
ires=MicroplateLabeler1.GetBarcodeField(2, bc)
```

Retry

```
LONG Retry( )
```

Description

Retries the last action after an error occurred.

Parameters

None

Returns

0 if successful; Other value if there was an error

Visual C++ example

```
LONG lres=m_MicroplateLabeler1.Retry();
```

Visual Basic .NET example

```
Dim ires as Integer
ires=MicroplateLabeler1.Retry()
```

RotateStage

```
LONG RotateStage( DOUBLE angle)
```

Description

Rotates the Microplate Labeler stage to a specified degree. After RotateStage operation is successful, the RotateStageComplete event is sent.

Parameters

None

Returns

0 if successful; Other value if there was an error

Visual C++ example

```
//Rotates stage to 10 degree
LONG lres=m_MicroplateLabeler.RotateStage(10.0);
```

Visual Basic .NET example

```
Dim ires as Integer
'Rotates stage to 90 degree
ires=MicroplateLabeler1.RotateStage(90.0)
```

Rotate180

```
LONG Rotate180()
```

Description

If a microplate enters the automated system in the wrong orientation, Rotate180 will rotate the plate by 180 degrees. After Rotate180 operation is successful, the Rotate180Complete event is sent.

Parameters

None

Returns

0 if successful; Other value if there was an error

Visual C++ example

```
LONG lres=m_MicroplateLabeler.Rotate180();
```

Visual Basic .NET example

```
Dim ires as Integer
ires=MicroplateLabeler1.Rotate180()
```

ShowDiagsDialog

```
void ShowDiagsDialog (VARIANT_BOOL modal, SHORT
securityLevel)
```

Description

Displays the Diagnostics dialog box that allows the operator to troubleshoot and correct problems. This method can be called before the Initialize method to create a profile. Contents displayed are based on the operator's access level.

Parameters

Name	Type	Range	Description
modal	VARIANT_BOOL	VARIANT_TRUE/ VARIANT_FALSE	VARIANT_TRUE = display dialog box as modal (does not permit users to access the parent window) VARIANT_FALSE = display dialog box as modeless (permits users to access the parent window)

Name	Type	Range	Description
securityLevel	SHORT	-1, 0, 1, 2, 3,	The security level the operator has in the dialog box: 0 = Administrator 1 = Technician 2 = Operator 3 = Guest -1 = No access

Returns

None

Visual C++ example

```
m_MicroplateLabeler.ShowDiagsDialog(VARIANT_TRUE, 0);
```

Visual Basic .NET example

```
MicroplateLabeler1.ShowDiagsDialog(True, 0)
```

Related topics

Microplate Labeler ActiveX properties	“Properties” on page 3
Microplate Labeler ActiveX events	“Events” on page 27
Overview of ActiveX controls	“About ActiveX controls” on page 2
Reporting problems	“Reporting problems” on page iv

Events

Error

```
void Error(SHORT Number, BSTR* Description, LONG Scode, BSTR Source, BSTR HelpFile, LONG HelpContext, VARIANT_BOOL* CancelDisplay)
```

Description

This event is sent when an error occurs during any non-blocking method execution.

Parameters

Name	Type	Range	Description
Description	BSTR*		The description of the error

Name	Type	Range	Description
CancelDisplay	VARIANT_BOOL	VARIANT_TRUE/ VARIANT_FALSE	The option to hide the error message. Use VARIANT_TRUE for C++ Use True for Visual Basic .NET

Note: SHORT Number, LONG Scode, BSTR Source, BSTR HelpFile, and LONG HelpContext are not used.

Returns

None

InitializeComplete

```
void InitializeComplete( )
```

Description

This event occurs when the Initialize method is successful.

Parameters

None

Returns

None

CloseComplete

```
void CloseComplete()
```

Description

This event occurs when the Close method is successful.

Parameters

None

Returns

None

DropStageComplete

```
void DropStageComplete()
```

Description

This event occurs when the DropStage method is successful.

Parameters

None

Returns

None

Returns

None

GreenButtonPressComplete

```
void GreenButtonPressComplete()
```

Description

This event occurs when the green button is pressed.

Parameters

None

Returns

None

HomeStageComplete

```
void HomeStageComplete()
```

Description

This event occurs when the HomeStage method is successful.

Parameters

None

Returns

None

PrintAndApplyComplete

```
void PrintAndApplyComplete()
```

Description

This event occurs when the print and apply operation completes successfully using PrintAndApply, PrintAndApplyByFormatName, or PrintAndApplyQueue (after the entire queue is printed) methods.

Parameters

None

Returns

None

PrintLabelComplete

```
void PrintLabelComplete()
```

Description

This event occurs when a label is printed successfully using PrintLabel, PrintLabelByFormatName, or PrintLabelQueue (after the entire queue is printed) methods.

Parameters

None

Returns

None

ReadBarcodeComplete

```
void ReadBarcodeComplete(BSTR barcode)
```

Description

This event occurs when the ReadBarcode method is successful.

Parameters

Name	Type	Description
barcode	BSTR	Barcode read by the reader.

Returns

None

RotateStageComplete

```
void RotateStageComplete()
```

Description

This event occurs when the RotateStage method is successful.

Parameters

None

Returns

None

Rotate180Complete

```
void Rotate180Complete()
```

Description

This event occurs when the Rotate180 method is successful.

Parameters

None

Returns

None

Related topics

Microplate Labeler ActiveX methods	“Methods” on page 7
Microplate Labeler ActiveX properties	“Properties” on page 3
Overview of ActiveX controls	“About ActiveX controls” on page 2
Reporting problems	“Reporting problems” on page iv



User Guide

G5404-90008

Revision 00, April 2012