LightCycler® 480
LIMS Interface Module

Version 1.5

www.roche-applied-science.com

Prologue 5

I **Revision History** **5**

II **Contact Addresses** **6**

III **Trademarks** **6**

IV **Intended Use** **6**

V **Software License Agreement** **7**

1 Program License Agreement 7
2 Grant of Software License 7
3 Limited Warranty 8
4 Disclaimer of Warranties 8
5 Limitations of Remedies 9
6 General Information 9
7 Intellectual Property Rights 9
8 Duration and Termination 10
9 Import, Export and Use of the Software 10
10 Miscellaneous 10
11 Governing Law and Place of Jurisdiction 10

VI **Conventions Used in this Manual** **11**

VII **Warnings and Precautions** **12**

LightCycler® 480 LIMS Interface Module **13**

1 **Installation of the LightCycler® 480 LIMS Interface Module** 16

2 **Interfacing a Host and the LightCycler® 480 LIMS Interface Module** 18

3 **Workflow for Using LightCycler® 480 LIMS Interface Module** 20

4 **Reference Information for the API** 24

Prologue

I Revision History

Version	Revision Date
1.0	September 2005
1.1	June 2006
2.0	February 2008

Information in this document is subject to change without notice. No part of this document may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without the express written permission of Roche Diagnostics GmbH.

Questions or comments regarding the contents of this manual can be directed to the address below or to your Roche representative.

Roche Diagnostics GmbH
Roche Applied Science
Customer Support
Nonnenwald 2
82372 Penzberg, Germany

Every effort has been made to ensure that all the information contained in this manual is correct at the time of printing.

However, Roche Diagnostics GmbH reserves the right to make any changes necessary without notice as part of ongoing product development.

II Contact Addresses

	Manufacturer	Roche Diagnostics Ltd. Forrenstrasse CH-6343 Rotkreuz Switzerland
	Distribution	Roche Diagnostics GmbH Sandhofer Straße 116 D-68305 Mannheim Germany
	Distribution in the US	Roche Diagnostics 9115 Hague Road PO Box 50457 Indianapolis, IN 46250 USA

III Trademarks

LIGHTCYCLER and LC are trademarks of Roche.

Other brands or product names are trademarks of their respective holders.

IV Intended Use

The LightCycler® 480 LIMS Interfaced Module enables the user to integrate the LightCycler® 480 System into an in-house Laboratory Information Management System (LIMS) environment and to employ the system's internal multiwell plate bar-code reader.

The LightCycler® 480 LIMS Interface Module is intended for general laboratory use in combination with the LightCycler® 480 Instrument and LightCycler® 480 Software.

V Software License Agreement

Read the following terms and conditions of this Software License Agreement ("Agreement") carefully before installing the LightCycler® 480 Software, hereinafter referred to as ("Software"). Proceeding with the installation of the Software will constitute acceptance of the terms and conditions of this Agreement. By accepting the terms and conditions of this Agreement, the end-user ("Licensee") assumes all responsibility and liability for the selection of this Software to achieve the intended results, and for its installation and subsequent use. If Licensee is not willing to be bound by the terms and conditions of this Agreement, the Software package must be promptly returned to Roche ("Supplier") with a copy of the receipt against refunding of the purchase price for this Software.

1 Program License Agreement

Licensee assumes all responsibility and liability for the selection of this Software to achieve the intended results, and for its installation and subsequent use. The Software is protected by copyright.

2 Grant of Software License

Supplier grants to Licensee subject to continuous compliance with all the provisions hereinafter, a non-exclusive, single-use license to use the Software upon the terms and conditions contained in this Agreement.

Licensee may:

a. Use the Software only on one workstation at a time and such workstations have to be owned, leased or otherwise controlled by Licensee, whether in a network or other configuration.

b. Transfer the Software by assigning the rights under this Agreement to another party, provided that the other party agrees in writing to accept the terms and conditions of this Agreement. In addition, Licensee must ensure that the copyright notice is maintained on the Software transferred.

Licensee may not:

a. Use the Software, in whole or in part, except as expressly provided in this Agreement.

b. Use the Software on more than one workstation at a time.

c. Copy, sell, or otherwise transfer the Software or assign its rights under this Agreement, in whole or in part, to another party, except as expressly provided in this Agreement.

d. Rent, distribute, license or sublicense the Software.

e. Create derivative works based on Software.

f. Modify, adapt, translate, reverse engineer, decompile or disassemble the Software.

Supplier reserves all rights not expressly granted herein, including, but not limited to, the rights to market the Software either directly or through affiliates, distributors and/or third parties.

For further information, please contact your local Roche Applied Science support organization. You will find the contact information on the following webpage: www.roche-applied-science.com.

3 Limited Warranty

The Software is provided "as is" without warranty of any kind, either expressed or implied, including, but not limited to the implied warranties of merchantability and fitness for a particular purpose. The entire risk as to the quality and performance of the Software is with Licensee, should the Software prove to be defective. Licensee assumes the entire costs of all necessary servicing, repair, or correction. However, Supplier warrants that the program media on which the Software is furnished is free from defects in materials and workmanship under normal use for a period of ninety (90) days from the date of delivery as evidenced by a copy of your receipt. SUPPLIER MAKES NO FURTHER WARRANTIES OR GUARANTEES NOR EXPLICIT NOR IMPLIED

4 Disclaimer of Warranties

THE WARRANTY SET FORTH IN THE PREVIOUS PARAGRAPH, IS IN LIEU OF ALL OTHER WARRANTIES, EXPRESS OR IMPLIED, ARISING BY LAW, FROM A COURSE OF PERFORMANCE, A COURSE OF DEALING, TRADE USAGE, OR OTHERWISE. SUPPLIER AND ANY ENTITY CONTROLLING, CONTROLLED BY OR UNDER COMMON CONTROL WITH SUPPLIER ("SUPPLIER'S AFFILIATE") SPECIFICALLY DISCLAIM, WITHOUT LIMITATION, ALL WARRANTIES OF ANY KIND, WHETHER EXPRESS OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NON-INFRINGEMENT. SUPPLIER AND SUPPLIER'S AFFILIATES MAKE NO REPRESENTATION OR WARRANTY AS TO THE SOFTWARE OR AS TO THE RESULTS TO BE ATTAINED BY LICENSEE OR ANY THIRD PARTY FROM THE SOFTWARE. LICENSEE ACKNOWLEDGES THAT IT HAS NOT RELIED UPON ANY REPRESENTATIONS OR WARRANTIES MADE BY SUPPLIER OR A SUPPLIER'S AFFILIATE EXCEPT FOR THOSE EXPRESSLY AND SPECIFICALLY SET FORTH IN THIS AGREEMENT.

5 Limitations of Remedies

Supplier's sole liability and Licensee's sole remedy shall be:

a. The replacement of the program media not meeting Supplier's limited warranty and which is returned to Supplier with a copy of Licensee's receipt;

b. If Supplier is unable to deliver replacement of program media which is free of defects in material and workmanship, Licensee may terminate this Agreement by returning the Software and a copy of Licensee's receipt to Supplier, and Licensee's money will be refunded.

IN NO EVENT WILL SUPPLIER OR ANY OF SUPPLIER'S AFFILIATES (OR THEIR RESPECTIVE OFFICERS, EMPLOYEES, CONSULTANTS, ATTORNEYS OR AGENTS), BE LIABLE FOR ANY SPECIAL, INDIRECT, INCIDENTAL, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, LOST PROFITS, LOST DATA OR INFORMATION, LOSS OF USE OF THE SOFTWARE, BUSINESS INTERRUPTION, LOSS OF BUSINESS REPUTATION OR GOODWILL, OR DOWNTIME COSTS) WHICH THE LICENSEE OR THIRD PARTIES MAY INCUR OR EXPERIENCE, DIRECTLY OR INDIRECTLY ARISING OUT OF OR RELATING TO THE SOFTWARE, THIS AGREEMENT, OR THE TERMINATION OF THIS AGREEMENT, EVEN IF SUPPLIER OR A SUPPLIER'S AFFILIATE HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES AND NOTWITHSTANDING ANY FAILURE OF ESSENTIAL PURPOSE. THE AGGREGATE LIABILITY, ON A COMBINED BASIS, OF SUPPLIER AND SUPPLIER'S AFFILIATES (AND THEIR RESPECTIVE OFFICERS, EMPLOYEES CONSULTANTS, ATTORNEYS, AND AGENTS) FOR DAMAGES FOR ANY CAUSE WHATSOEVER DIRECTLY OR INDIRECTLY RELATING TO OR ARISING OUT OF THIS AGREEMENT OR THE SOFTWARE, AND REGARDLESS OF THE FORM OF ACTION, SHALL BE LIMITED TO, AT SUPPLIER'S OPTION, REPLACEMENT OF THE SOFTWARE OR REFUND OF THE FEES RECEIVED BY SUPPLIER OR A SUPPLIER'S AFFILIATE FROM LICENSEE WITH RESPECT TO THE SOFTWARE.

6 General Information

Licensee may not sublicense, assign or transfer the license or the Software, in whole or in part, except as expressly provided in this Agreement. Any attempt otherwise to sublicense, assign or transfer any of the rights, duties or obligations hereunder is void.

7 Intellectual Property Rights

Licensee shall only hold those rights to the Software that are expressly described in Section 2 of this Agreement. Any other rights with regard to the Software, including without limitation, ownership rights and patent, copyright, trademark, trade secret and other intellectual property rights, shall remain the sole property of Supplier. Licensee will not remove from the Software any references to copyrights, trademarks or other ownership rights, or cover up or alter any such references. Licensee will take all reasonable steps to prevent any unauthorized use, reproduction, sale, or publication of the Software or the unauthorized provision of access thereto. Licensee will indemnify and hold harmless Supplier from any losses, damages, claims and expenses (including, without limitation, reasonable legal expenses) relating to any infringement of the rights of Supplier caused by Licensee, Licensee's breach of this Agreement or Licensee's use of the Software in a manner not authorized under this Agreement.

8 Duration and Termination

The Agreement is effective until terminated. Licensee may terminate this Agreement at any time by destroying the Software and documentation relating to the Software in any form. The Agreement will terminate automatically and without notice from Supplier, if Licensee fails to comply with any term or condition of this Agreement. Licensee agrees to destroy the Software upon termination of this Agreement by Supplier. On any termination of this Agreement, all rights of use of the Software held by Licensee shall expire.

9 Import, Export and Use of the Software

Licensee shall be exclusively responsible for ensuring compliance with the relevant legislation relating to its rights to import, export or use the Software.

10 Miscellaneous

Should any part of this Agreement be declared void or unenforceable by a court of competent jurisdiction, the remaining terms shall remain in full force and effect.

Failure of Supplier to enforce any of its rights in this Agreement shall not be considered a waiver of its rights, including but not limited to its rights to respond to subsequent breaches.

By opening and using this Software Licensee acknowledges that he has read this Agreement, understands it, and agrees to be bound by its terms and conditions. Licensee further agrees that this Agreement is the complete and exclusive statement of the Agreement between Licensee and Supplier and supersedes any proposal or prior agreement, oral or written, any other communications between Licensee and Supplier relating to the subject matter of this Agreement.

The headings of the several Sections of this Agreement are intended for convenience of reference only and are not intended to be a part of or to affect the meaning or interpretation of this Agreement.

11 Governing Law and Place of Jurisdiction

This Agreement shall be governed by and construed in accordance with the laws of the State of Indiana, without giving effect to any choice of law principles thereof. The parties agree that the United Nations Convention on Contracts for the International Sale of Goods (1980) is specifically excluded from application to this Agreement.

VI Conventions Used in this Manual

Text Conventions

To impart information that is consistent and memorable, the following text conventions are used in this Operator's Manual:

Numbered Listing	Steps in a procedure that must be performed in the order listed.
Italic type, blue	Points to a different chapter in this Operator's Manual which should be consulted.
Italic type	Describes how to proceed when operating the LightCycler® 480 LIMS Interface Module.

Symbols

In this Operator's Manual symbols are used as an optical signal to point out important things.

Symbol	Heading	Description
	IMPORTANT NOTE	Information critical to the success of the procedure or use of the product.
	INFORMATION NOTE	Additional information about the current topic or procedure.

VII Warnings and Precautions

	The LightCycler® 480 System ("Product") is equipped with software, enabling the user of the Product to connect it with a network. Roche draws the atten tion of the user to the fact that such connection may have an adverse effect on the Product's integrity, *e.g.*, due to an infection of the Product with malicious code (viruses, Trojan horses, etc.) or access by unauthorized third parties (*e.g.*, intrusion by attackers). Roche therefore highly recommends to protecting the Product against such risks by taking appropriate and state of the art action. As the Product is not intended to be used within networks without an appropriate firewall and has not been designed for such use, Roche assumes no liability in that regard. Roche offers the user the cobas IT firewall to be installed prior to the first connection of the Product to any network. For further information on this cobas IT firewall and/or the Roche network security concept please contact your local Roche representative. In the event the user connects the Product with any network without using the cobas IT Firewall, Roche cannot offer any Product support regarding any problem resulting from such network connection.
	Microsoft Office and Norton Antivirus software are tested not to interfere with LightCycler® 480 Software and LightCycler® 480 software modules. Any other additional software must not be installed on the LightCycler® 480 Instrument PC. Installation of any other additional software on the LightCycler® 480 Instrument PC presents the risk of interference with LightCycler® 480 Software and LightCycler® 480 software modules, and could affect result security.
	**Anti-virus software is not provided.** Therefore, it is essential to take precautions to ensure that any software loaded onto the LightCycler® 480 Instrument PC is virus-free.
	If a LIMS server is installed on a LightCycler® 480 control unit, no remote database connection must be used from this control unit!

LightCycler® 480
LIMS Interface Module

The LightCycler® 480 LIMS Interface Module is an accessory module to the LightCycler® 480 System that enables the user to integrate the LightCycler® 480 System into an in-house Laboratory Information Management System (LIMS) environment. Furthermore, it allows the user to employ the system's internal bar-code reader which scans the bar-code information on the LightCycler® 480 Multiwell Plates, permitting easy assay tracking.

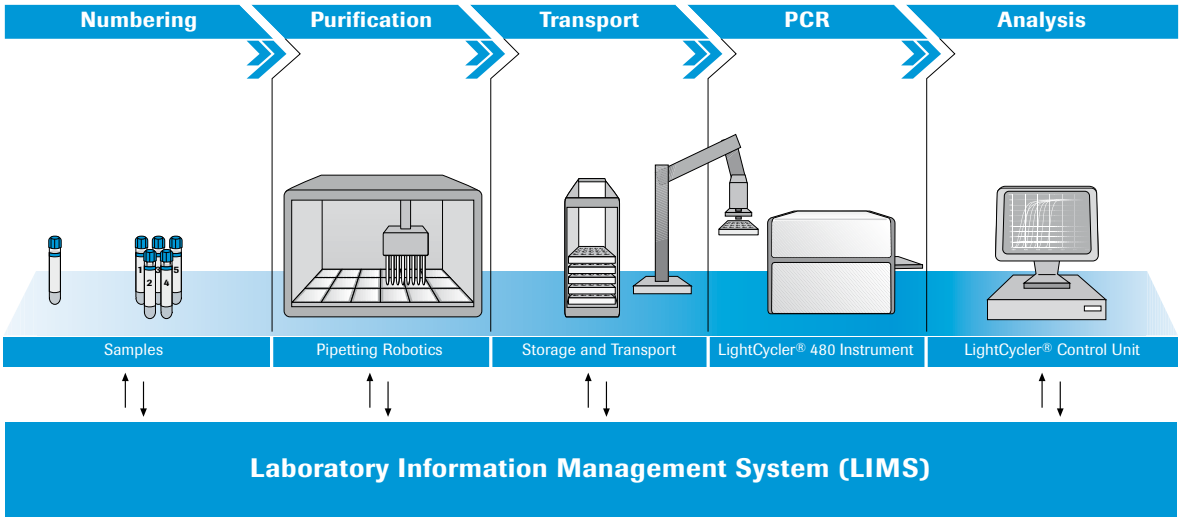


Figure 1 Schematic view of the LightCycler® 480 System integrated into an exemplary workflow via a Laboratory Information Management System (LIMS)

LightCycler® 480 Software provides an application programming interface (API) to enable interfacing with a laboratory information management system (LIMS).

A typical application for the LightCycler® LIMS Interface Module is the integration of the LightCycler® 480 System into a robotic environment. Such an application could include the following:

- LightCycler® 480 LIMS Interface Module (provided by Roche as an additional software module to the LightCycler® 480 Software, which is loaded on a control unit connected to the LightCycler® 480 Instrument)
- Robot (*i.e.*, a multiwell plate handler) with control software
- Host software which coordinates tasks between the robot and the LightCycler® 480 System
- External handheld bar-code scanner (optional) connected to the host software

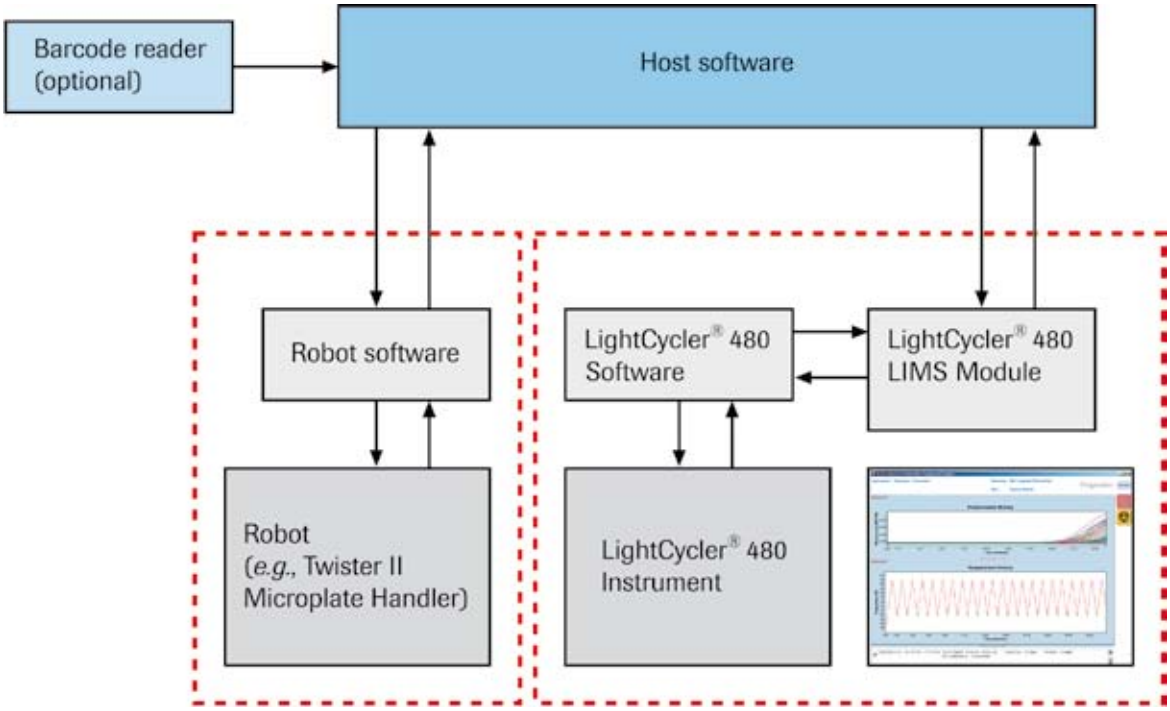


Figure 2 Illustration of an example LightCycler® 480 LIMS environment

Only the LightCycler® 480 Instrument, the LightCycler® 480 Software, the LightCycler® 480 LIMS Interface Module, and the (optional) handheld bar-code scanner are provided by Roche.

Via the LIMS, the LightCycler® 480 System can be integrated into a laboratory workflow. An external bar-code scanner can be connected to the LightCycler® 480 Instrument to enter sample identification information, while the integrated bar-code scanner enables recognition of the multiwell plate. Via the LightCycler® 480 LIMS Interface Module, the instrument and software can be controlled by the LIMS: In combination with suitable robotic equipment, the LightCycler® 480 LIMS Interface Module communicates with the LightCycler® 480 System, thus enabling you to load, control and analyze experiments based on predefined experiment templates called macros. The results are stored and transferred back into the LIMS.

This combination allows the following:

- LIMS-initiated download of test order
- LIMS-initiated request for final results

The following section provides detailed information about the following topics:

1. Installation of LightCycler® 480 LIMS Interface Module software
2. Interfacing the LightCycler® 480 LIMS Interface Module
3. Description of the workflow using the LightCycler® 480 LIMS Interface Module
4. Example for programming a “LIMS control program” using the LightCycler® 480 LIMS API
5. Reference information for the API

1 Installation of the LightCycler® 480 LIMS Interface Module

In order to assure the functionality and performance of the LightCycler® 480 LIMS Interface Module described in this manual, the PC system should fulfill the following minimum requirements:

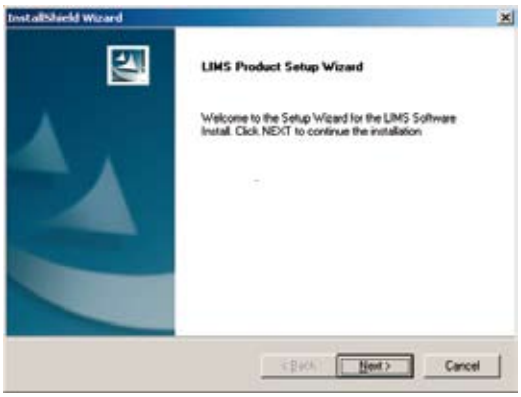
- processor: Intel Pentium 4 (or equivalent), 3.0 GHz
- main memory: 512 MB RAM
- hard disk: 40 GB
- communication: LAN network card
- display: 1284 × 1024 resolution
- operating system: Windows XP Professional, Service Pack 2

To be able to run the LightCycler® 480 LIMS Interface Module you must have a valid software user license file from Roche Applied Science installed. Usually, the license file is generated and installed by a Roche specialist during installation of the complete system. In case a Roche specialist is not available for software installation, please follow the steps below to obtain the license file:

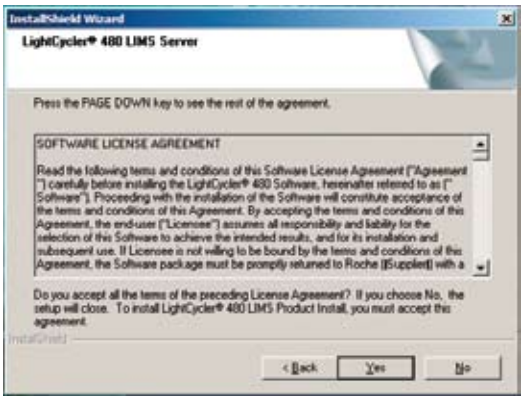
To install the LightCycler® 480 LIMS Interface Module software:

1. Insert the LightCycler® 480 LIMS Interface Module installation CD into the CD-ROM drive of the control unit connected to your LightCycler® 480 Instrument.

2. Open the software CD in Windows Explorer and double-click *LIMS_Product_Install.exe* to start the LIMS product installation program. The setup wizard Welcome window is displayed. Click *Next* to continue.

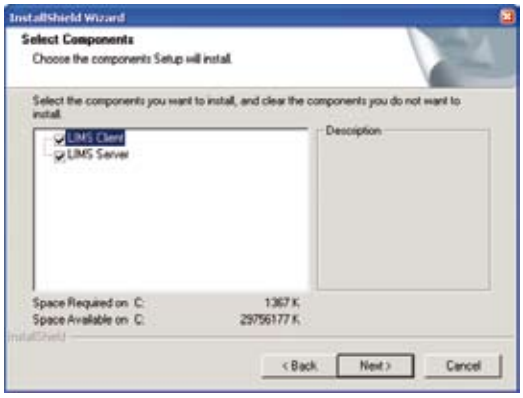


3. You are prompted to agree to the license conditions. Click *Yes*.



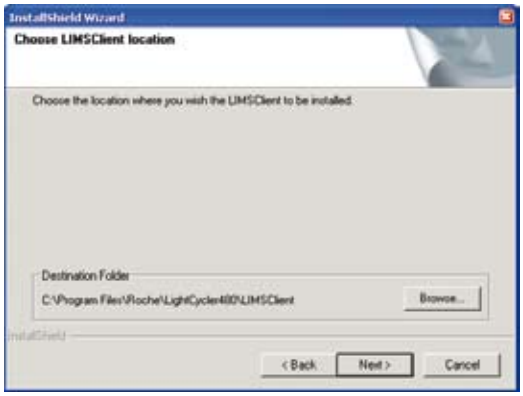
►►►

4 In case the LightCycler®480 SW is already installed on your system the *Select Components* window will appear.

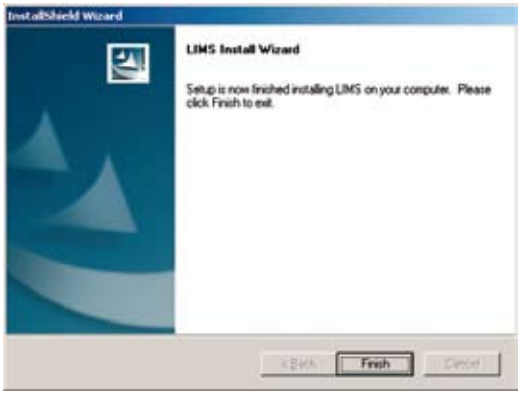


Please select both components by ticking the checkboxes *Lims Server* and *Lims Client*. Click *Next*

5 In the Choose LIMSClient location window, use the default settings to install the LIMSClient or browse to select the location of the database engine. Click *Next*.



6 When the installation is finished, the wizard displays the following window. Click *Finish* to complete the installation.



2 Interfacing a Host and the LightCycler® 480 LIMS Interface Module

The LightCycler® 480 LIMS programming interface is delivered as a dynamic link library (DLL). This interface can be addressed using Delphi, Visual Basic, C/C++/C#, VBScript or any other programming language supported by the Windows XP platform.

Communication between the LIMS software and the LightCycler® 480 LIMS Interface Module is only possible through the DLL. Two options are available for establishing communication between the LIMS software and the LightCycler® 480 LIMS Interface Module.

a. Direct communication using the DLL. (This option requires that your LIMS software is running on Windows XP.)

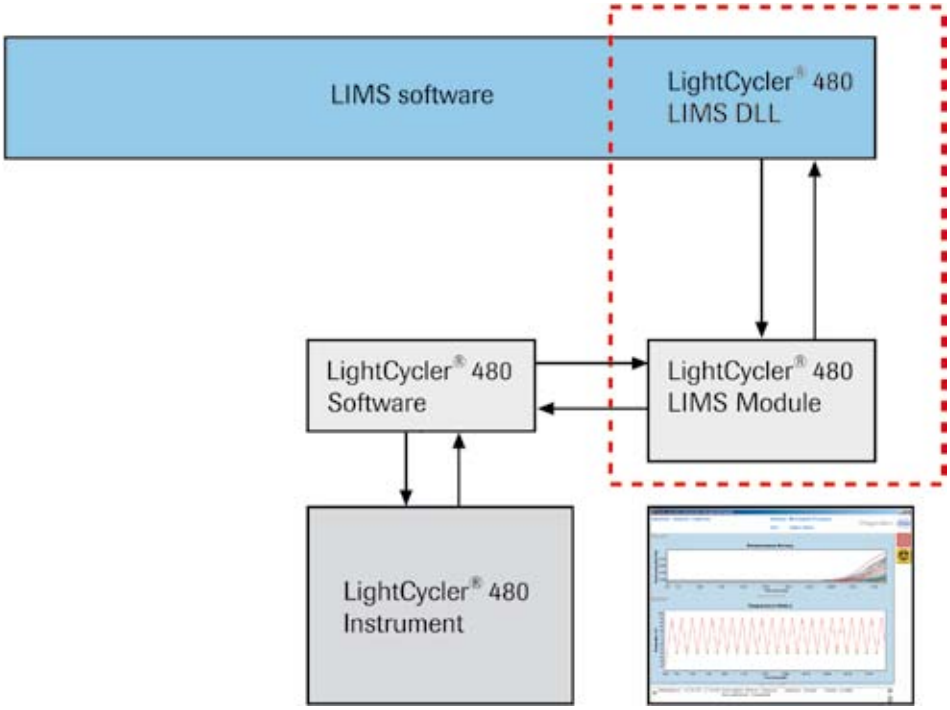


Figure 3 Direct communication between the LIMS software and LightCycler® 480 LIMS Interface Module using the DLL.

b. Indirect communication using an intermediate program that interfaces the DLL. The following example uses scripts to interface the LightCycler® 480 LIMS Interface Module. The scripts read the files created by the LIMS software, control experiments using the LightCycler® 480 LIMS DLL and write the results back to files.

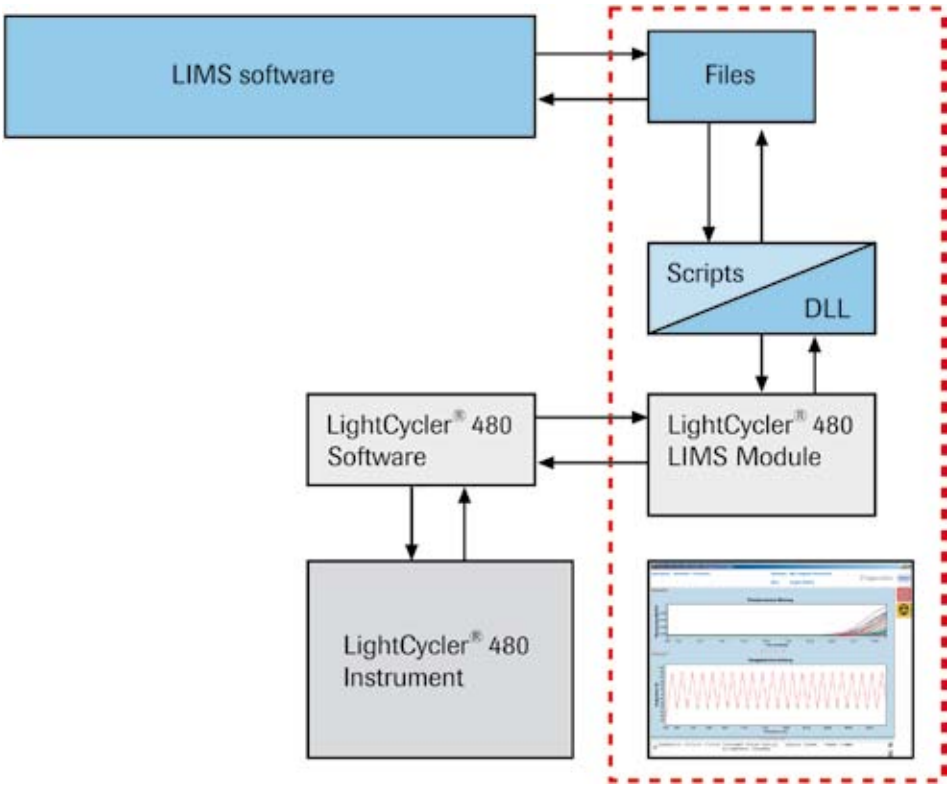


Figure 4 Indirect communication between the LIMS software and LightCycler® 480 LIMS Interface Module using text files

Since scripts are specific for the LIMS software, they must to be written for each project.

3 Workflow for Using LightCycler® 480 LIMS Interface Module

In this section, "host" refers to the part of the LIMS software that directly interfaces the LightCycler® 480 LIMS DLL.

To set up your workflow using LightCycler® 480 LIMS Interface Module, ensure that the following prerequisites are fulfilled:

- The host is connected to LightCycler® 480 LIMS Interface Module
- The host is logged onto LightCycler® 480 LIMS Interface Module

The complete workflow consists of the following steps:

1. Order is prepared
2. Workflow is completed
3. Results are received

Preparing the order:

1	Prepare your plate as described in Section *Operation* of the LightCycler® 480 Instrument Operator's Manual.
2	Create a new order in the host software: ▶ Define the experiment template to use (macro) and define an experiment name for the order. ▶ Scan the bar code (plate ID) and assign it to the experiment (optional). ▶ Define a sample list (optional).
3	Place the prepared plate in the robot (optional).
4	Confirm the order.

■

Completion of the workflow:

1	The host sends a command to the LightCycler® 480 LIMS Interface Module to open the Multiwell Plate Loader of the LightCycler® 480 Instrument.
2	The plate is loaded with one of the following options. **Option A:** The host sends a command to the robot to load the multiwell plate into the LightCycler® 480 Instrument. (If a plate is already loaded, the host sends a command to the robot to remove the old plate before loading the new plate.) The robot returns the ID of the loaded plate. **Option B:** The user loads the multiwell plate manually.
3	The host sends a command to the LightCycler® 480 LIMS Interface Module to close the Multiwell Plate Loader of the LightCycler® 480 Instrument.
4	The host retrieves the order assigned to the plate ID.
5	The host sends a command to the LightCycler® 480 LIMS Interface Module to start the experiment with the following information: ▶ Macro to be executed ▶ Experiment name ▶ Plate ID (optional) ▶ Sample list (optional)
6	If a plate ID was specified, the LightCycler® 480 LIMS Interface Module compares the plate ID with the loaded plate in the LightCycler® 480 Instrument. The experiment is started only if the plate IDs match.
7	The host waits until the experiment is finished. While the experiment is being performed, the host sends commands to request the status of the experiment.
8	After the experiment is finished, the LightCycler® 480 LIMS Interface Module saves the experiment.
9	The LightCycler® 480 LIMS Interface Module returns the completed experiment status to the next status request from the host.
10	If there are more plates to process, the host restarts with Step 1.

■

Retrieving the results of an order:

1	Select an order from which you want to retrieve the experiment results.
2	The host sends commands to the LightCycler® 480 LIMS Interface Module to request the status of the experiment matching the order you have selected according to the experiment name.
3	If the experiment is finished, the host can request the results.

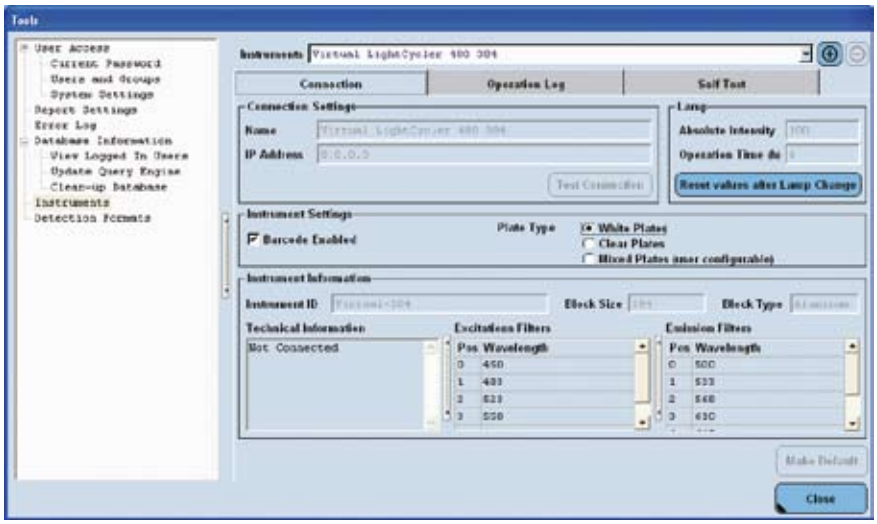
■

Using the bar-code module:

The LightCycler® 480 Software enables the user to use the multiwell plate bar-code scanner of the LightCycler® 480 Instrument. The multiwell plate bar-code scanner is an integral part of the block cycler unit. It is used for automated identification and identifier (ID) tracking of PCR multiwell plates. During plate loading, the linear bar-code present on the LightCycler® 480 Multiwell Plates is scanned.

To use the bar-code module proceed as follows:

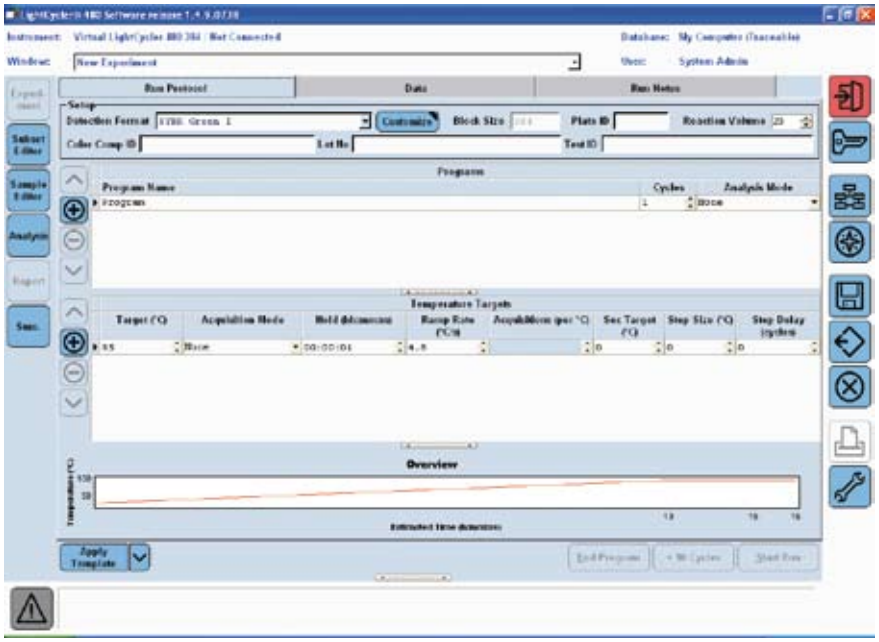
1. First you have to check if the internal multiwell bar-code scanner is enabled:
   - Open the *Tools* window and select *Instruments*.
   - On the *Connection* tab, the checkbox *Barcode Enabled* must be checked.



   - When the multiwell bar-code scanner is enabled, the ID label present on the LightCycler® 480 Multiwell Plates is scanned automatically during plate loading. The following two situations are possible to occur:

▶ ▶ ▶

2 ▶ If you program a new experiment, the *Plate ID* field on the *Run Protocol* screen is initially empty.



▶ When a LightCycler® 480 Multiwell Plate is loaded, the plate ID is scanned and shown in the *Plate ID* field.

3 ▶ If you program a new experiment and you enter a plate ID manually into the *Plate ID* field, the LightCycler® 480 Software compares the entered plate ID to the scanned plate ID after loading the LightCycler® 480 Multiwell Plate.

▶ If the user defined plate ID matches the scanned plate ID, the run is started.

▶ If the user defined plate ID does not match the scanned plate ID, a message window opens asking whether you want to overwrite the scanned plate ID. If you accept overwriting, the run is started. If you do not accept overwriting, the run is aborted.

■

4 Reference Information for the API

The LightCycler® 480 LIMS application program interface (API) supports various control commands (*e.g.*, open and close the Multiwell Plate Loader, start and stop the experiment) and several features related to data transfer, such as download of sample information and return of results and experiments information. This section provides examples of the reference information for the API.

A detailed description of all codes are given in electrical form on the installation CD in the document "Programming Reference.pdf"

LIMS Connection	ExperimentInfo	Returns an interface for accessing the experiment API	[C/C++] **HRESULT get_ExperimentInfo(ILIMSExperiment-Info** Value);** [Visual Basic] **Public ReadOnly Property ExperimentInfo As Object** [C#] **public object ILIMSConnection.ExperimentInfo { get; }**
	Host	Gets the IP address or machine name of the computer on which the LIMS server is running	[C/C++] **HRESULT get_Host(BSTR* Value);** **HRESULT put_Host(BSTR Value);** [Visual Basic] **Public Overloads Property Host As String** [C#] **public ref string ILIMSConnection.Host { get; set; }**
	Instrument	Returns an interface for controlling the instrument	[C/C++] **HRESULT get_Instrument(ILIMSInstrument** Value);** [Visual Basic] **Public ReadOnly Property Instrument As Object** [C#] **public object ILIMSConnection.Instrument { get; }**
	LoggedIn	Checks to see if you are currently logged into the LIMS server	[C/C++] **HRESULT get_LoggedIn(VARIANT_BOOL* Value);** [Visual Basic] **Public ReadOnly Property LoggedIn As Boolean** [C#] **public ref bool ILIMSConnection.LoggedIn { get; }**

► ► ►

LIMS Connection	Login	Establishes a login with the LIMS server if not already logged in	[C/C++] **HRESULT Login(** **BSTR User,** **BSTR Password,** **ILIMSOperationResult** Value** **);** [Visual Basic] **object.Login(** **ByVal User As String,** **ByVal Password As String,** **) As Object** [C#] **void ILIMSConnection.Login(** **string User,** **string Password,** **);**
	Logout	Terminates the login with the LIMS server	[C/C++] **HRESULT Logout(** **ILIMSOperationResult** Value** **);** [Visual Basic] **object.Logout() As Object** [C#] **object ILIMSConnection.Logout();**
	Port	Gets the TCP port number on which the LIMS server is running (default = 1776)	[C/C++] **HRESULT get_Port(LONG* Value);** **HRESULT put_Port(LONG Value);** [Visual Basic] **Public Overloads Property Port As Long** [C#] **public ref int ILIMSConnection.Port { get; set; }**

► ► ►

<table>
<tr>
<td rowspan="2">LIMS Experiment Info</td>
<td>GetCompleted Experiment Summary</td>
<td>Generates and returns summary information for the specified experiment</td>
<td>

```
       [C/C++]
HRESULT   GetCompletedExperimentSummary(

    BSTR                    ExperimentName,
    BSTR*                   Summary,
    ILIMSOperationResult**  Value

);
[Visual Basic]
object.GetCompletedExperimentSummary(

   ByVal ExperimentName As String,
   ByRef Summary        As String,

) As Object
[C#]
void   ILIMSExperimentInfo.GetCompletedExperi
mentSummary(

    string     ExperimentName,
    ref string Summary,

);
```

</td>
</tr>
<tr>
<td>GetStatus</td>
<td>Returns the current state of the specified experiment</td>
<td>

```
       [C/C++]
HRESULT   GetStatus(

    BSTR                      ExperimentName,
    BSTR*                     Status,
    ILIMSOperationResult**    Value

);
[Visual Basic]
object.GetStatus(

   ByVal ExperimentName As String,
   ByRef Status         As String,

) As Object
[C#]
void   ILIMSExperimentInfo.GetStatus(

    string                    ExperimentName,
    ref string                Status,

);
Status:
const
LIMS_EXP_STATUS_NO_ANALYSES = 'No analyses';
LIMS_EXP_STATUS_ANALIZED = 'Has analyses';
LIMS_EXP_STATUS_NOT_STARTED = 'Not started';
LIMS_EXP_STATUS_ACTIVE = 'Running';
LIMS_EXP_STATUS_ABORTED = 'Aborted';
LIMS_EXP_STATUS_ERROR = 'Error';
```

</td>
</tr>
</table>

► ► ►

LIMS Experiment Info	Export Experiment		[C/C++] **HRESULT ExportExperiment(** **BSTR ExperimentName <>,** **BSTR Filename <>,** **ILIMSOperationResult** Value <>** **);** [Visual Basic] **object.ExportExperiment(** **ByVal ExperimentName <> As String,** **ByVal Filename <> As String,** **) As Object** [C#] **void ILIMSExperimentInfo.ExportExperiment(** **string ExperimentName <>,** **string Filename <>,**
LIMS Instrument	Close	Instructs the instrument to remit access to the physical sample container	[C/C++] **HRESULT Close(** **ILIMSOperationResult** Value** **);** [Visual Basic] **object.Close() As Object** [C#] **object ILIMSInstrument.Close();**
	Open	Instructs the instrument to permit access to the physical sample container	[C/C++] **HRESULT Open(** **ILIMSOperationResult** Value** **);** [Visual Basic] **object.Open() As Object** [C#] **object ILIMSInstrument.Open();**
	Reserve	Attempts to obtain exclusive control of the instrument	[C/C++] **HRESULT Reserve(** **ILIMSOperationResult** Value** **);** [Visual Basic] **object.Reserve() As Object** [C#] **object ILIMSInstrument.Reserve();**

► ► ►

LIMS Instrument	StartExperiment	Runs the specified experiment with the given parameters	[C/C++] **HRESULT StartExperiment(** **BSTR ExperimentName,** **BSTR ContainerBarCode,** **BSTR MacroName,** **ILIMSSampleDefinition* SampleDefinition,** **ILIMSOperationResult** Value** **);** [Visual Basic] **object.StartExperiment(** **ByVal ExperimentName As String,** **ByVal ContainerBarCode As String,** **ByVal MacroName As String,** **ByRef SampleDefinition As ILIMSSampleDefinition,** **) As Object** [C#] **void ILIMSInstrument.StartExperiment(** **string ExperimentName,** **string ContainerBarCode,** **string MacroName,** **ref ILIMSSampleDefinition SampleDefinition,**
	Unreserve	Relinquishes exclusive control of the instrument	[C/C++] **HRESULT Unreserve(** **ILIMSOperationResult** Value** **);** [Visual Basic] **object.Unreserve() As Object** [C#] **object ILIMSInstrument.Unreserve();**
	GetStatus		[C/C++] **HRESULT GetStatus(** **BSTR* Status <>,** **ILIMSOperationResult** Value <>** **);** [Visual Basic] **object.GetStatus(** **ByRef Status <> As String,** **) As Object** [C#] **void ILIMSInstrument.GetStatus(** **ref string Status <>,** **);**

▶ ▶ ▶

LIMS Instrument	Abort Experiment		[C/C++] **HRESULT AbortExperiment(** **ILIMSOperationResult** Value <>** **);** [Visual Basic] **object.AbortExperiment() As Object** [C#] **object ILIMSInstrument.AbortExperiment();**
	GetContainer Barcode		[C/C++] **HRESULT GetContainerBarcode(** **BSTR* Barcode <>,** **ILIMSOperationResult** Value <>** **);** [Visual Basic] **object.GetContainerBarcode(** **ByRef Barcode <> As String,** **) As Object** [C#] **void ILIMSInstrument.GetContainerBarcode(** **ref string Barcode <>,** **);**

► ► ►

LIMS Operation Result	DateTime		[C/C++] **HRESULT get_DateTime(DATE* Value);** [Visual Basic] **Public ReadOnly Property DateTime As Date** [C#] **public ref System.DateTime ILIMSOperation Result.DateTime { get; }**
	ErrorNumber		[C/C++] **HRESULT get_ErrorNumber(LONG* Value);** [Visual Basic] **Public ReadOnly Property ErrorNumber As Long** [C#] **public ref int ILIMSOperationResult.Error-Number { get; }**
	Message	If failed, provides a user-readable description of the error	[C/C++] **HRESULT get_Message(BSTR* Value);** [Visual Basic] **Public ReadOnly Property Message As String** [C#] **public ref string ILIMSOperationResult.Mes-sage { get; }**
	Server Error	If failed, indicates whether the failure occurred on the server	[C/C++] **HRESULT get_ServerError(VARIANT_BOOL* Value);** [Visual Basic] **Public ReadOnly Property ServerError As Bool-ean** [C#] **public ref bool ILIMSOperationResult.Server-Error { get; }**
	Successful	Indicates the overall success of the operation	[C/C++] **HRESULT get_Successful(VARIANT_BOOL* Value);** [Visual Basic] **Public ReadOnly Property Successful As Boolean** [C#] **public ref bool ILIMSOperationResult.Suc cessful { get; }**
	UserMessage		[C/C++] **HRESULT get_UserMessage(BSTR* Value);** [Visual Basic] **Public ReadOnly Property UserMessage As String** [C#] **public ref string ILIMSOperationResult.User Message { get; }**

► ► ►

LIMS Sample Definition	AddSample		[C/C++] **HRESULT AddSample(** **ILIMSSampleInfo* Sample** **);** [Visual Basic] **object.AddSample(** **ByRef Sample As ILIMSSampleInfo** **)** [C#] **void ILIMSSampleDefinition.AddSample(** **ref ILIMSSampleInfo Sample** **);**
	Clear		[C/C++] **HRESULT Clear();** [Visual Basic] **object.Clear()** [C#] **void ILIMSSampleDefinition.Clear();**
	DeleteSample		[C/C++] **HRESULT DeleteSample(** **LONG Index** **);** [Visual Basic] **object.DeleteSample(** **ByVal Index As Long** **)** [C#] **void ILIMSSampleDefinition.DeleteSample(** **int Index** **);**
LIMS Sample Definition	GetSample		[C/C++] **HRESULT GetSample(** **LONG Index,** **ILIMSSampleInfo** Value** **);** [Visual Basic] **object.GetSample(** **ByVal Index As Long,** **) As Object** [C#] **void ILIMSSampleDefinition.GetSample(** **int Index,** **);**

►►►

	SampleCount		[C/C++] **HRESULT get_SampleCount(LONG* Value);** [Visual Basic] **Public ReadOnly Property SampleCount As Long** [C#] **public ref int ILIMSSampleDefinition.SampleCount { get; }**
LIMS Sample Info	Comments		[C/C++] **HRESULT get_Comments(BSTR* Value);** **HRESULT put_Comments(BSTR Value);** [Visual Basic] **Public Overloads Property Comments As String** [C#] **public ref string ILIMSSampleInfo.Comments { get; set; }**
	ID		[C/C++] **HRESULT get_ID(BSTR* Value);** **HRESULT put_ID(BSTR Value);** [Visual Basic] **Public Overloads Property ID As String** [C#] **public ref string ILIMSSampleInfo.ID { get; set; }**
	Name		[C/C++] **HRESULT get_Name(BSTR* Value);** **HRESULT put_Name(BSTR Value);** [Visual Basic] **Public Overloads Property Name As String** [C#] **public ref string ILIMSSampleInfo.Name { get; set; }**
	Position		[C/C++] **HRESULT get_Position(BSTR* Value);** **HRESULT put_Position(BSTR Value);** [Visual Basic] **Public Overloads Property Position As String** [C#] **public ref string ILIMSSampleInfo.Position { get; set; }**
	Replicate Position		[C/C++] **HRESULT get_ReplicatePosition(BSTR* Value);** **HRESULT put_ReplicatePosition(BSTR Value);** [Visual Basic] **Public Overloads Property ReplicatePosition As String** [C#] **public ref string ILIMSSampleInfo.ReplicatePosition { get; set; }**

Published by

Roche Diagnostics GmbH
Roche Applied Science
68298 Mannheim
Germany

05152089001 0208