LabBook connect v1

Connection middleware for LabBook

January 2026

1.   Introduction

LabBook Connect is middleware designed to ensure secure communication, integration, and data management between various medical analysis devices and the LabBook laboratory management system. It facilitates the secure exchange of results and information between instruments and the LIS (laboratory information system), thereby reducing the risk of errors associated with manual data entry and increasing laboratory productivity.

2.   List of Supported Devices

Analyzer management is based on independently developed plugins/connectors. Several connectors have already been developed; however, only those that have been tested and validated in real laboratory conditions are currently published on the Mérieux Foundation's GitHub: https://github.com/fondationmerieux and are available for download on the LabBook website: https://www.lab-book.org/telechargements/.

This list will be gradually expanded as other connectors are evaluated and validated in a real-world environment.

3.   Installation and Configuration

The LabBook software is provided as a complete, ready-to-use system (Ubuntu-based ISO image). To ensure maximum stability and enhanced security, the software architecture is segmented into separate containers using **Podman** technology.

Version 3.6 is the minimum version required to benefit from this infrastructure and the latest functional developments, including mapping, new analyses, and pop-up interfaces. In this configuration, **LabBook Connect resides in its own application container**, independent of the LabBook container. This separation allows the middleware to be operational as soon as the system is launched to communicate with your PLCs.

However, if you are using an earlier version of LabBook (< 3.6) that does not yet have this architecture for the middleware, you will need to **update** the **LabBook** and **Connect** containers via their respective tar files and replace the Ubuntu /etc/init.d/labbook file with a version from 3.6 (because this file contains the port mapping when the Connect container is launched).

3.a) Configuration

To begin, we will look at the steps to follow to configure the addition of controllers on the LabBook interface:

→ Log in with an administrator account (root) on LabBook

→ In the "Integrations" menu, go to "Analyzer Configuration."

Or directly via the shortcut on the home page

→ Now on the "List of analyzers" page, start by preparing the analyzer configuration by clicking the "Connect" button

→ On the "Connect" page, you must first enter the IP address of the server hosting LabBook, as shown in the image below. Click on the "Test" button to see if the connection is working properly before clicking the "Save" button

→ Next, you need to import three separate files from the following fields (click the "Send" button for each file selected):

- **Plugin**: Allows you to interpret messages exchanged with the PLC and ensure communication according to the supported protocol. Each PLC has a specific plugin.

- **Configuration**: Contains the connection parameters for the PLC, including the LabBook URL, the mapping file path, the communication mode, and the IP address and port used.

- **Mapping**: Defines the correspondence between the analysis codes sent by the PLC and those used in LabBook.

Important: The configuration file provided cannot be used as is. It must be edited before initial use, as its settings are unique and must be adapted to the specific characteristics of each installation.

Now that all the files have been sent to LabBook Connect, you can click on the "Reload plugins" button to refresh and reload the files that have just been imported.

→ Click on the "Back" button to return to the "List of analyzers" page. Now add an analyzer by clicking on the "Add" button.

Here is an example for the GeneXpert analyzer:

→ Enter the name you want for the analyzer. In the list of analyzers, if you have successfully added the three required files from Connect, you should have all your analyzers, then select an analyzer from the list.

→ In the ID field, enter the same name used for the ID in your configuration file.

```
# Analyzer Configuration File
# Contains general settings and connection details for one ROCHE analyzer.

version = "0.9.15"  # Configuration file version

[analyzer]
brand = "GENEXPERT"                       # Analyzer brand
name = "GeneXpert"                        # Analyzer model name
id = "GX_01"                      # Unique analyzer identifier
plugin = "AnalyzerGeneXpert"              # Plugin Java class name for this analyzer
url_lis = "http://localhost/sigl"   # URL of the Laboratory Information System (LIS) upstream endpoint
operation_mode = "batch"                  # Operation mode (batch or query)
archive_msg = "Y"                         # Enable ('Y') or disable ('N') message archiving
type_cnx = "socket"                       # Connection type (socket, ...)
type_msg = "ASTM"                         # Message type (HL7 or ASTM)
mapping = "/storage/resource/connect/analyzer/mapping/mapping_genexpert.toml"  # Mapping configuration to use

[analyzer.socket]
mode = "server"                # Socket mode (client or server)
ip = "192.168.10.61"        # Analyzer IP address
port = 12345        # Port number for socket communication
```

For example, if you have id = "GX_01," enter GX_01 as the ID and test whether the analyzer is loaded correctly.

- Set the mode to "Query." The position allows you to define the display order of the analyzer in the analyzer list page.

- Click "Save" to confirm the addition of this analyzer

3.b) Physical connection of controllers and LabBook

Whether LabBook is installed on a server or a standard computer, here is a diagram explaining how LabBook communicates with the automated system. In this diagram, we will use the connection with a GeneXpert as an example, but the principle remains the same for other automated systems:
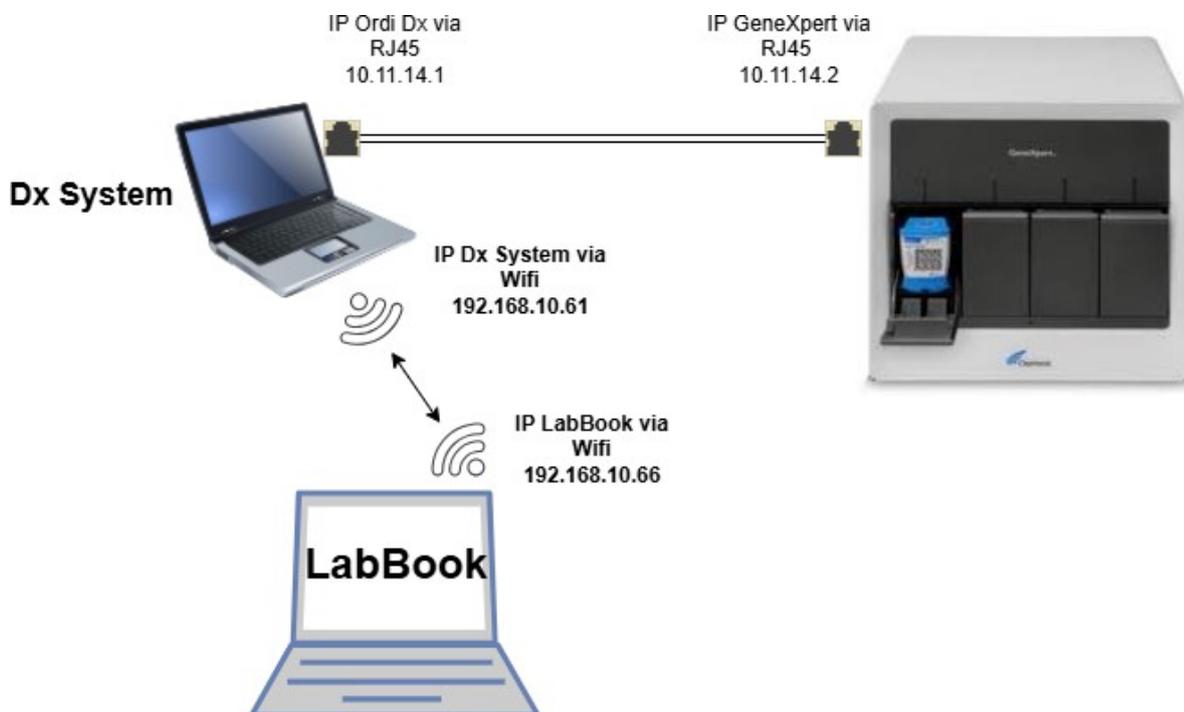

Figure 1: Communication diagram between GeneXpert, Dx System, and LabBook Connect

It is important to note that **LabBook Connect does not communicate directly with the automated system,** but rather through the system already used by the automated system.

In this diagram, the Dx System is connected to the GeneXpert automated system via an RJ45 cable. LabBook Connect then establishes communication with the Dx System via the Wi-Fi network.

LabBook Connect will therefore receive the results sent from the Dx System (automatically or manually, depending on the configuration chosen).

4. Practical example: Configuration and simulation (GeneXpert)

This section details how to configure communication between LabBook connect and the Cepheid GeneXpert system, whether for a test phase (simulation) or actual production.

All equipment must be on the same network segment or interconnected in a routed and authorized manner.

According to the diagram above, for the devices to communicate, they must be connected to the same network. Here is an example of a typical IP configuration:

→ **LabBook server** (with LabBook connect): 192.168.10.66 (via Wi-Fi)

→ **Dx System computer**: 192.168.10.61

→ **Direct Dx – GeneXpert link**: The two devices often communicate via a dedicated local network (e.g., 10.11.14.1 and 10.11.14.2)

4.a) Dx System software configuration

On the computer controlling the GeneXpert, configure the sending of data to LabBook:

1. Go to **Setup** > **System Configuration** > **Host Communication Settings** tab

2. Check **Enable Host Communication**

3. Host ID: Enter the exact ID defined in LabBook (e.g., GX_01)

4. Communication Settings:

    a. Protocol: **ASTM**

    b. Run Host As: **Server**

    c. Server IP Address: The IP address of your LabBook (192.168.10.66)

    d. Port: 12345 (must be set to a **value between 1024 and 65535**, in accordance with the recommendations in the official GeneXpert manual for LIS communication)

Fondation **Mérieux**
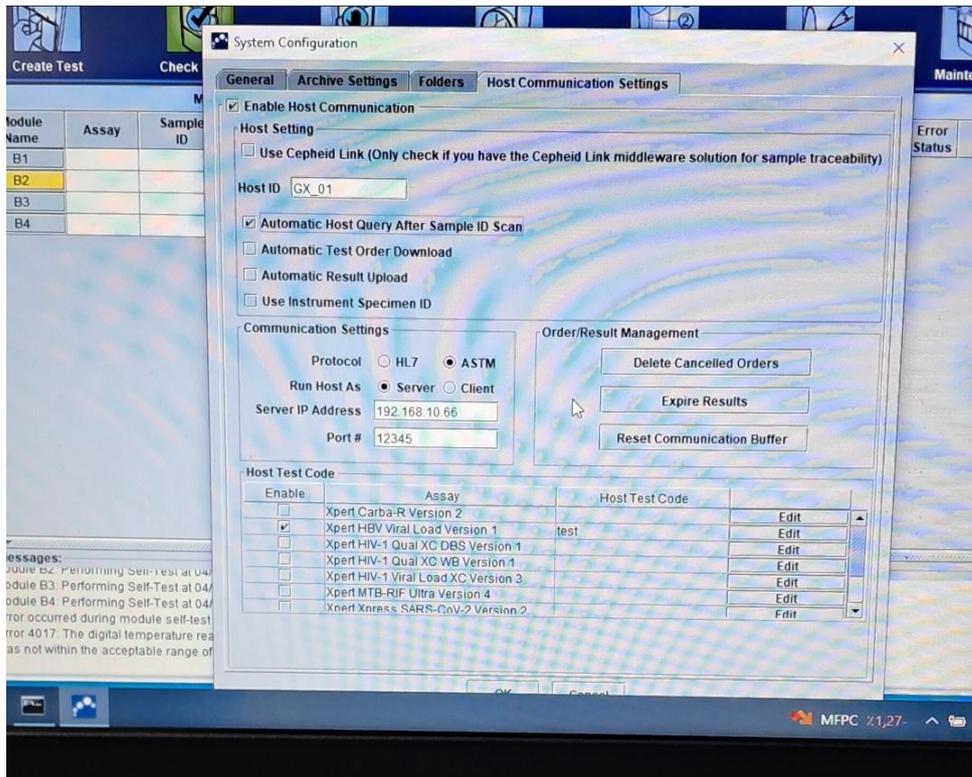
5. Host Test Code: Click **Edit** to map the analysis codes (e.g., hbv_viral_load).

   **Verification**: Once validated, the message "Host is connected" should appear in the status bar at the bottom of the Dx System screen.

4.b) LabBook Connect Operation

We have already seen how to configure LabBook, but now we will look at the rules to follow to ensure that LabBook is ready to receive results from GeneXpert.

If you need to grant specific permissions in the LabBook server firewall, you can enter the following command in the server terminal:

   **sudo ufw allow 12345/tcp**

→ First, add a new test request (external or inpatient) as usual in LabBook.

→ Go to the page where you add tests and make sure that the tests you add are tests that exist in GeneXpert (you can always add a new one in LabBook if this is not the case).

→ Then, in the "Pathological Products" section, it is **very important** to enter the **"Code" for the LabBook sample** with **the same code in the "Sample ID" field in GeneXpert**: This correspondence is essential to ensure that the results are correctly linked to the patient file.

→ Now save your new request and you can manually test sending the result from the GeneXpert Dx System interface.

4.c) Sending results to LabBook

To transfer completed results:

1. On the Dx System, click **View Results**

2. Click on the **Upload Test** button at the bottom of the screen

3. **Check the tests to be sent** in the list and click **Upload**
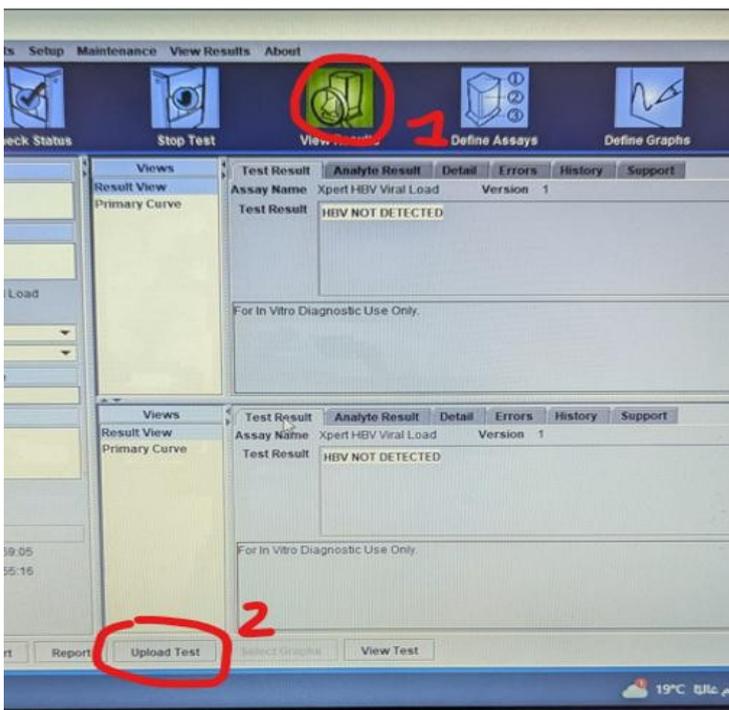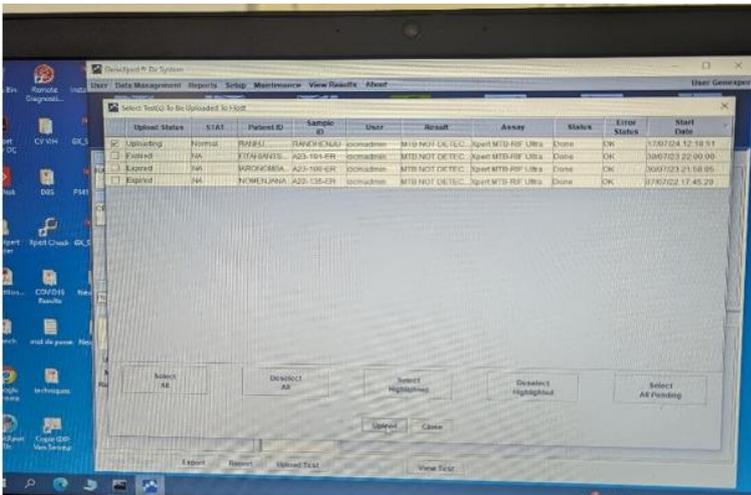
4.d) Correct Receipt of Results on LabBook

If all parameters are correctly set, the results sent by GeneXpert will be automatically inserted into the corresponding variable fields in "Results Entry."

As shown in this image, the blue "i" icon indicates that data has been received for this file and the corresponding values are automatically entered into the five variables of the Carba-R analysis. If you want to see the details of the results of this analysis, you can click on the blue "i" and you will see the table below:

Fondation Mérieux

Fighting infectious diseases since 1967

www.fondation—merieux.org