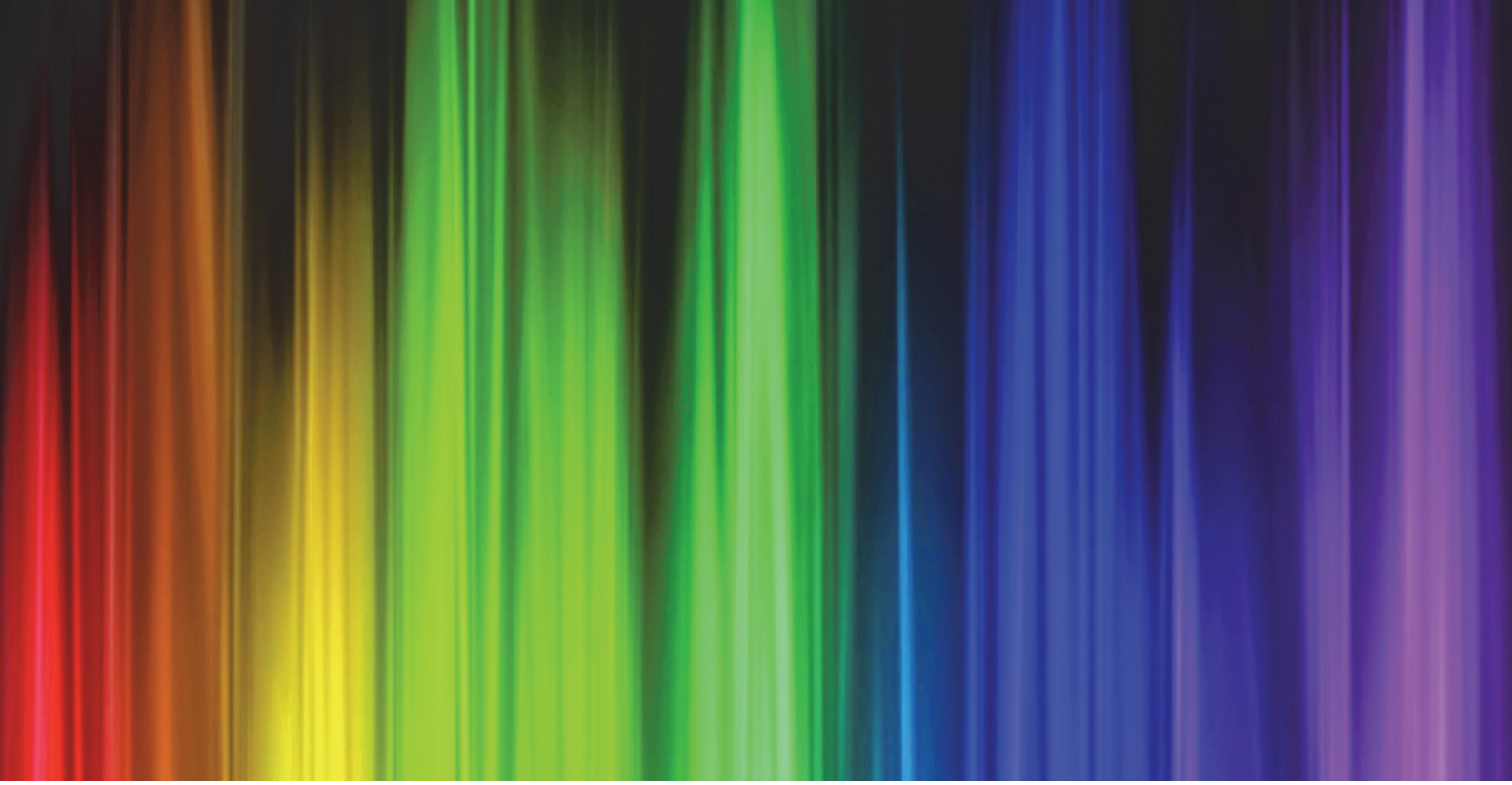


Thermo Scientific

SkanIt™ Automation Interface

User Manual

API Version 3.0

Cat. No. N19215 Rev. 3.0 2022

thermo scientific

Manufacturer:

Life Technologies Holdings Pte. Ltd.

(a part of Thermo Fisher Scientific Inc.)

33, Marsiling Industrial Estate Road 3, #7-06,

Singapore 739256

These are the original instructions.

For Research Use Only. Not for use in diagnostic procedures.

Preface

About this guide

This user manual has been written for the software engineer planning to integrate a supported Thermo Fisher Scientific Microplate Reader (hereinafter ‘microplate reader’ or ‘instrument’ for short) into an automation environment. It provides all information needed to operate a multiplate reader in an automated fashion which means operation without the SkanIt Software user interface. The SkanIt Automation Interface (hereinafter ‘Automation Interface’) supports the following devices:

- Varioskan LUX
- Multiskan SkyHigh
- Multiskan Sky
- Multiskan GO
- Multiskan FC
- Fluoroskan
- Fluoroskan FL
- Luminoskan

In this document, Automation Interface refers to the complete software package found on the installation media of SkanIt Software for Microplate Readers, Research Edition (hereinafter 'SkanIt Software'), containing both the automation server and a sample client.

Contents

1

Introduction

The Automation Interface provides high-level control for Thermo Fisher Scientific microplate instruments in automation environments. Setting up an automation environment requires:

- the Automation Server (provided by the SkanIt Automation Interface)
- the Automation Client (developed or acquired by the integrator), and
- the plate handling instrument and respective software to load and unload microplates (developed or acquired by the integrator).

The SkanIt Automation Interface is designed in a client-server architecture. The automation server application (hereinafter Automation Server) provides access to one microplate reader. A client application (hereinafter Automation Client) connects to the Automation Server and executes commands exposed by the Automation Server, for example, opening and closing the plate tray and executing predefined SkanIt sessions.

The microplate reader needs to be connected to the PC where the Automation Server is running. The Automation Client can be run either locally on the same PC, or on another PC in the same network. It is recommended to start by running everything locally to rule out communication issues coming from the local IT infrastructure.

The SkanIt Automation Server is implemented using gRPC, Google's RPC technology of choice. An Automation Client can be implemented using any programming language supported by gRPC. To help the integrator's work, the SkanIt Automation Interface includes a sample GUI client called the AutomationTester written in C#. Its source code can be used as a starting point to develop a custom client application, but it is not a ready client solution.

The Automation Interface is a part of the SkanIt Software which must be installed before using the Automation Interface. In a typical workflow, the end-user creates a session (this includes layout definitions, measurement steps, possible result steps etc.) using SkanIt Software. After this step, SkanIt Software is closed, but it still needs to remain installed. Through calls to the Automation Server, the Automation Client then takes control of the instrument, loads the desired plates using plate handlers, and starts the session execution. The measurement results are stored in the resulting session file but also returned from the Automation Server as XML data. Session files created through the Automation Interface can be opened in SkanIt Software for further analysis. Full reports of the execution can be generated on demand (PDF, Excel, TXT, XML), including all calculation results given that the calculation steps had been defined in the session.

The Automation Interface 3.0 as described here and included in SkanIt Software 7.0 and later is not compatible with Automation clients developed against version 1.0-2.0 (included in SkanIt Software 4.x-6.x). Due to the adoption of SiLA2 and a new communication technology there is no simple upgrade path for previous clients. However, the commands and events in the API itself are still very similar. Most of the required work will be in the communication layer.

SiLA2 and AniML Compliance

The Automation Interface has been implemented to conform to the standard SiLA2 v1.0 (hereinafter SiLA2).

In terms used by the SiLA Standard, the Automation Server application is the SiLA Server. Any SiLA2-compliant Process Management System (PMS) or the provided Automation Tester application can access and control an instrument, therefore acting as a SiLA Client.

For more information, visit the web page of the SiLA Standard at: http://www.sila-standard.org/. The implemented features are available in the features folder of the SkanIt installation folder.

Measurement results coming from the instrument are returned to the server in a response containing XML data. The data is compatible with the ASTM standard AniML v0.90. For details, visit the AniML webpage https://www.animl.org/.

Requirements

The Automation Interface requires the installation of the SkanIt Software for Microplate Readers, Research Edition on the PC. For its system requirements, refer to the *SkanIt™ Software for Microplate Readers User Manual*.

Note The Automation Interface is not supported for the Drug Discovery Edition (DDE) version of SkanIt Software. DDE-only features, for example, user authentication and audit trails as typically required in regulated environments, have to be implemented by the client application or PMS.

It is recommended to familiarize yourself with the SiLA2 Standard prior to developing your own client.

Some basic knowledge about Google gRPC is also recommended, especially if you want to implement your client in a language other than C#. For more information https://grpc.io/.

Contents of the Automation Interface

The Automation Interface for SkanIt consists of the following components which can be found in the 'Automation' folder of the SkanIt installation media.

User Manual

This document.

Client API

The documentation of the Client API (C#).

Automation Server

The Automation Server is a .NET console application providing access to a USB-connected instrument using gRPC calls. The command interface is compatible with the SiLA2 Standard (see SiLA2 and AniML Compliance).

Note Only one instance of the Automation Server can be running at a time. Therefore only one device can be controlled by the Automation Interface.

Automation Client (AutomationTester)

The AutomationTester is a .NET sample application that acts as a client to connect to the Automation Server. The integrator has to implement its own version of an Automation Client, the AutomationTester is only meant for initial testing. However, through its own GUI it offers access to the complete functionality that is provided by the Automation API.

The AutomationTester is available both as compiled binaries and source code in a Visual Studio 2019 solution.

The AutomationTester uses the Client API which is a custom C# wrapper around the generic SiLA2 API. Only one AutomationTester can be connected to an Automation Server at a time.

Note The AutomationTester sample application is provided as is, it has not been fully tested for stability and may malfunction when used in an unexpected way.

LIMS compatibility

SkanIt Automation Interface supports laboratory information management software such as LIS/LIMS (hereinafter LIMS) with the following functionality:

- Measurement and calculation results are returned after execution in machine-readable XML/AnIML format.
- On-demand exporting of reports in XML/XLSX/PDF/TXT format, see command GetReport() in the Client API (C#) documentation.
- Possibility to define the location and filenames of both SkanIt result files and automatically exported reports.

Typically LIMS is used to store information about SkanIt sessions, for example, sample IDs and lot numbers. If needed, SkanIt Software can be used to enter all of this information upfront to a SkanIt session. SkanIt can be used to export report files either on demand or automatically after executing a session. Refer to the *SkanIt Software for Microplate Readers User Manual* for details on editing the layout and generating reports, especially auto-generating reports to a user-defined location. This is useful if the LIMS can be configured to automatically detect new files in a certain folder.

Sometimes it may be required to feed data to a session without opening SkanIt, for example adding the bar code of a microplate before starting the execution. SkanIt Automation Interface supports bi-directional communication through the executedFilePath parameter in ExecuteSession(). This means that an arbitrary identification string such as a bar code can be passed in, and in combination with an auto-generated report this bar code is also used for the XML reports.

Installation and usage

Setting up the Automation Server

1. Install the SkanIt Software, this will also install the Automation Server. To initialize SkanIt Software and the automation interface, SkanIt Software must be started at least once before using the Automation Server.
2. Create a valid session for the instrument you are going to use and save it to a preferred location.
3. Navigate to your SkanIt installation folder (C:\Program Files\Thermo\SkanIt...) and start the server by executing StartServer.bat. No errors (red lines) should appear.
4. Now the server is running and waiting for incoming connection requests.

Note You cannot run SkanIt Software and the Automation Server simultaneously. Make sure that SkanIt Software is not running before starting the Automation Server.

Testing the Automation Server with the AutomationTester

1. On your installation media navigate to the Automation\bin\AutomationTester folder.
2. Start the tester by executing AutomationTester.exe.
3. Edit “Server connection”. If you have not changed server settings simply double-click “default”.
4. Click **Open** to open the connection to the Automation Server.
5. Define the instrument.
   - For simulation mode, select "Simulator" and specify the instrument type.
   - For using a real instrument, unselect "Simulator" and specify the serial number of the instrument.
6. Click **Connect** to establish the connection to the instrument or simulator.

Configuration files and logging

The logging configuration file for the Automation Server can be found from the SkanIt installation folder (nlog.config). In this configuration file, the user can define where the log file is saved (the default is \ProgramData\Thermo\MIP\Readers\Automation\Logs) and configure other logging related settings.

The server config file (ProgramData\Thermo\MIP\Readers\Automation\config.json) is used by the server application to persist the server ID and server name (the user can set the server name utilizing the provided SiLA2 or Client APIs).

3

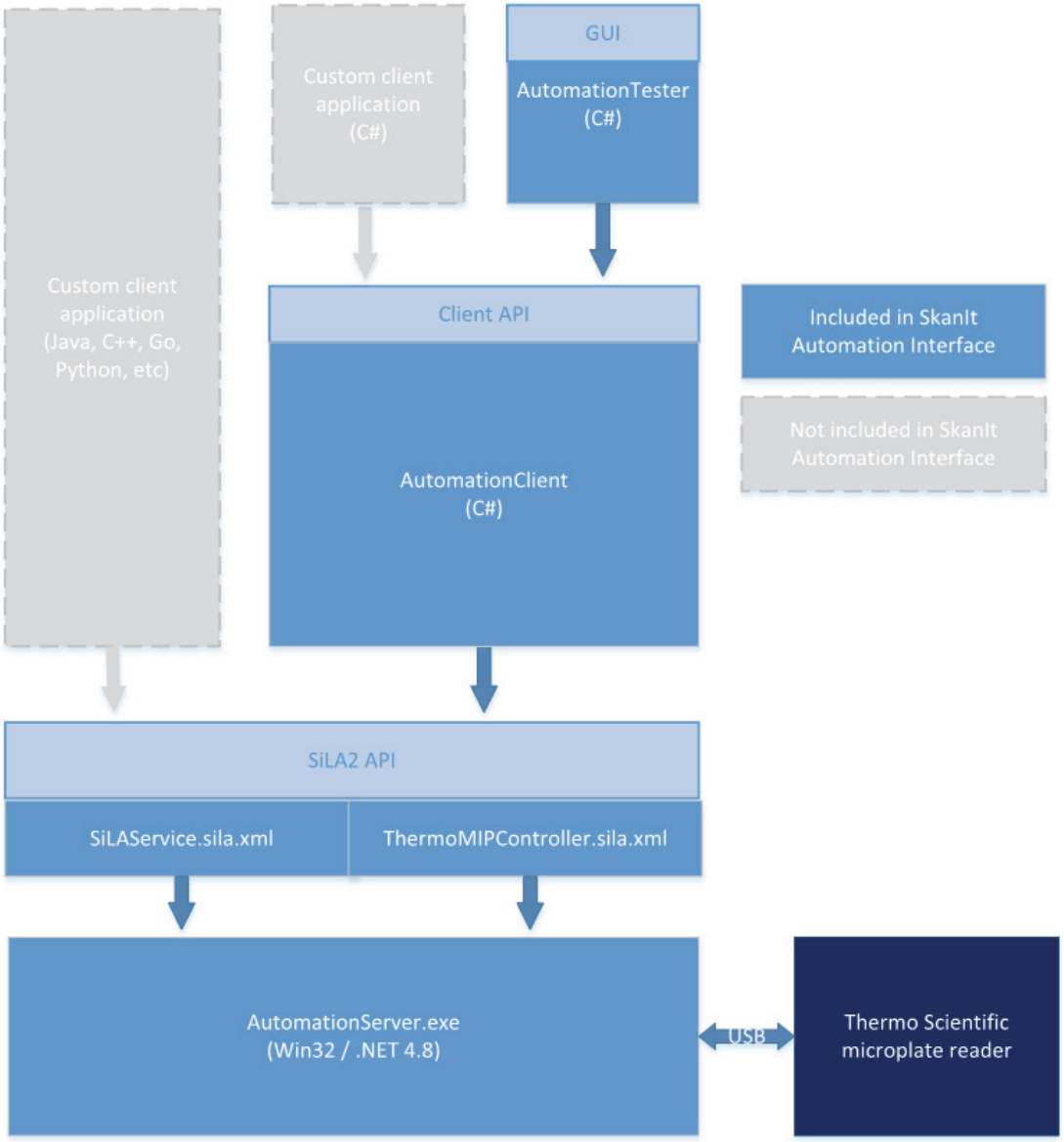
Automation Interface API

The Automation Interface API consists of two levels: a lower level cross-platform SiLA2 API and a higher level C#-specific Client API. It is up to the integrator which API is used to develop a client application.

The recommended path is to use the C# Client API which is also demonstrated in the AutomationTester sample client. Also, technical support can only be provided for implementations against the C# Client API.

The specification of the Client API can be found in a separate *Client API v3.0* document.

Figure 1. API diagram.



SiLA2 API

Based on the SiLA2 Standard and gRPC as communication protocol, the SiLA2 API level is not specific to any programming language. It utilizes XML files to specify so-called features. Features define the interface to the server and thus to the instrument. Currently the API consists of two feature XML files:

- **SiLAService.sila.xml** - A set of commands as defined in the SiLA2 standard.
- **ThermoMIPController.sila.xml** - A set of commands specific to Thermo Fisher Scientific microplate readers.

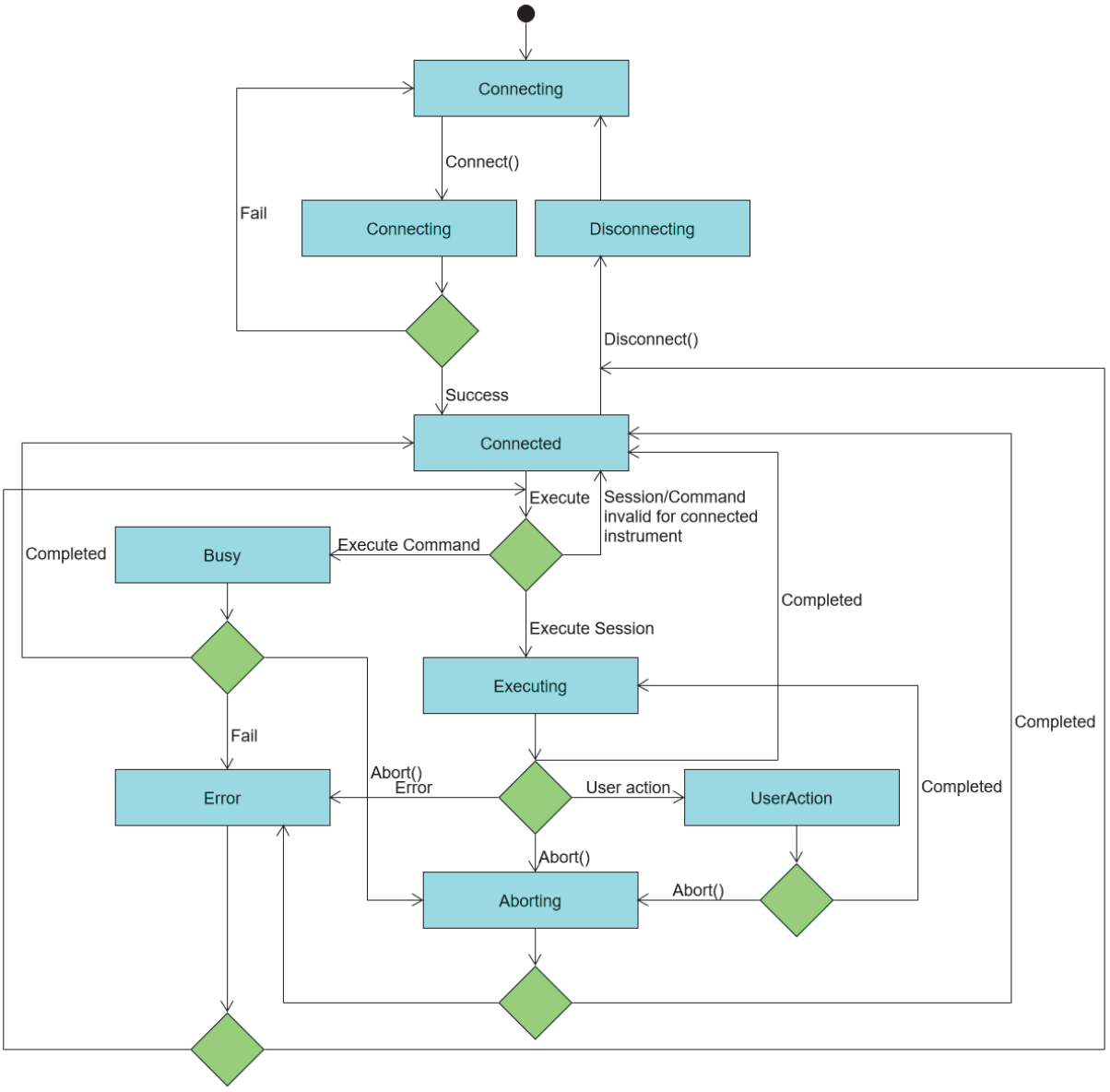
For instructions on using this API, refer to SiLA2 and gRPC documentation and the contents of the XML files. The XML files can be found in the SkanIt installation folder.

Client API

The Client API is a C#-implementation built on top of the SiLA2 API. This is the recommended API if the final client applications will be written in C#. The included AutomationTester also uses the Client API.

The following diagram shows the states and the possible transitions of the server.

Figure 2. Automation states.



For details on the Client API, see the *Client API v3.0* document found in the package that contains the Automation Interface for SkanIt.

Troubleshooting and known issues

Operating the plate tray

It is not possible for the Automation Interface to know whether the plate tray is currently outside or inside the instrument.

Use the PlateOut command to drive the plate tray out of the instrument, or use the PlateIn command to drive the plate tray into the instrument.

Both commands will send the instructions to the instrument, but the instrument does NOT tell the Automation Server when the actual movement has finished. Therefore, the client cannot determine when the movement has been performed. The response to the command that the client receives is merely an acknowledgment that the instrument has accepted the command and starts moving. The Automation Server is in IDLE state during the time of the physical movement of the tray. Any command issued during this time will be queued and executed when the movement is finished. However, it is recommended to wait for the instrument to finish before executing any further commands.

If you want to ensure that the plate tray is ejected, you have to issue a PlateOut command and wait for the time it takes to move the plate tray. The time needed for the movement differs from instrument to instrument. The same applies to the information whether the plate tray is inside the instrument.

Note If the plate is already driven out and the instrument gets the PlateOut command, depending on the instrument it may drive the plate in and then out again. Thus you will have to take this extra time into account.

Connect/Disconnect

Fast cycles of Disconnect() and Connect() can cause the establishment of an USB connection to fail. It is recommended to give the instrument time between those commands. Generally, it is not necessary to perform reconnects during normal usage.

After establishing a USB connection, some instruments can report a temperature reading of 0 for a short period of time. To ensure correct readings, wait 5-7 seconds after Connect() before sending GetTemperature() commands.

Instrument communication log

The instrument communication log contains command communication between the server and the instrument over the USB cable. However, inspecting this log is only meant for advanced troubleshooting of the instrument behavior. The log is enabled in the SkanIt Software, refer to the *SkanIt™ Software for Microplate Readers User Manual* for details.