SENTINEL
ADMINISTRATOR'S GUIDE
Version 2026

The information contained in this guide is not of a contractual nature and may be subject to change without prior notice.

The software described in this guide is sold under a license agreement. The software may be used, copied or reproduced only in accordance with the terms of the agreement.

No part of this guide may be copied, reproduced or transmitted in any form, by any means or for any purpose other than the purchaser's own use without the written permission of TEKLYNX Corporation SAS.

©2026 TEKLYNX Corporation SAS,

All rights reserved.

Table of Contents

Installation . . . . . 1 - 1

System Requirements . . . . . 1 - 2
Server Requirements . . . . . 1 - 2
Workstation Requirements . . . . . 1 - 3
Software Protection . . . . . 1 - 4
Activating the software key protection . . . . . 1 - 4
Server Installation. . . . . 1 - 8
Design/Workstation Installation . . . . . 1 - 9
Running the Label Print Manager Service . . . . . 1 - 9
Setting up a user account . . . . . 1 - 10
Installing/Uninstalling the Label Print Manager Service . . . . . 1 - 11

Introduction . . . . . 2 - 13

Definition of terms . . . . . 2 - 14
Basic concepts . . . . . 2 - 15
How it works . . . . . 2 - 15

The Application Modules . . . . . 3 - 17

The application modules . . . . . 3 - 18
Web Manager . . . . . 3 - 18
User Manager . . . . . 3 - 18
Kernel . . . . . 3 - 18
Sentinel Manager . . . . . 3 - 18
Mapper . . . . . 3 - 18
Query Manager. . . . . 3 - 19
The major steps from creating to running the sentinels . . . . . 3 - 19
On your workstation . . . . . 3 - 19
On the server . . . . . 3 - 19

User Manager . . . . . 4 - 21

Setting Permissions . . . . . 4 - 22
Launching the User Manager . . . . . 4 - 22
Adding new users. . . . . 4 - 22
Defining or changing user rights. . . . . 4 - 23
Changing a password. . . . . 4 - 24

Web Manager . . . . . 5 - 25

User Interface. . . . . 5 - 26

Permissions . . . . . 5 - 28
Using the Web Manager. . . . . 5 - 29
Creating or Modifying an Existing Sentinel . . . . . 5 - 29
Sending Out the Sentinels . . . . . 5 - 30

Sentinels . . . . . 6 - 33

What are sentinels and how do they work? . . . . . 6 - 34
How do sentinels work? . . . . . 6 - 35

Sentinel Manager . . . . . 7 - 37

Launching the program . . . . . 7 - 39
Description of the main window . . . . . 7 - 39
Menu Bar . . . . . 7 - 39
Workspace . . . . . 7 - 39
The Toolbar . . . . . 7 - 40
The Sentinel list . . . . . 7 - 40
Information panel . . . . . 7 - 40
Interface settings . . . . . 7 - 41
Changing display options . . . . . 7 - 41
Changing the interface language . . . . . 7 - 41
Adjusting column width . . . . . 7 - 41
Adding a sentinel . . . . . 7 - 41
Defining the properties of a sentinel . . . . . 7 - 41
Displaying a specific group . . . . . 7 - 42
Modifying the display order. . . . . 7 - 42
Displaying sentinel properties. . . . . 7 - 42
Managing sentinels . . . . . 7 - 42
Duplicating a sentinel . . . . . 7 - 42
Deleting a sentinel . . . . . 7 - 43
Activating one or more sentinels. . . . . 7 - 43
Deactivating one or more sentinels . . . . . 7 - 43
Terminate unit. . . . . 7 - 44
Sentinel status . . . . . 7 - 45
Task management . . . . . 7 - 45
Viewing current tasks . . . . . 7 - 45
Error Management . . . . . 7 - 45
Structure of the error logfile . . . . . 7 - 46
Error messages . . . . . 7 - 46
Deleting an error logfile. . . . . 7 - 47

Plug-ins . . . . . A - 47

Input plug-ins . . . . . A - 49

File capture . . . . . A - 49
TCP/IP port listening . . . . . A - 51
Print capture . . . . . A - 58
Web Server . . . . . A - 59
Records capture . . . . . A - 62
SFTP file capture . . . . . A - 75
REST Web Server API . . . . . A - 78
Modbus Server . . . . . A - 89
Process plug-ins . . . . . A - 92
Label Printing . . . . . A - 97
Email . . . . . A - 105
Database . . . . . A - 109
Web Service Client . . . . . A - 110
Output plug-ins . . . . . A - 113
Transfer . . . . . A - 113
Email . . . . . A - 114
Report plug-ins . . . . . A - 118
Log file . . . . . A - 118
E-mail . . . . . A - 119
TCP/IP port listening . . . . . A - 120

API Capabilities . . . . . B - 127

Definitions/Introduction . . . . . B - 128
Kernel Sentinels API . . . . . B - 129
Sentinels models description . . . . . B - 130
CODESOFT Plugin Printers API . . . . . B - 132
Printers models descriptions . . . . . B - 134
REST Web Server API . . . . . B - 136

About this manual

Documents supplied

Complete documentation is provided to help you make optimum use of the full range of resources offered by your software.

The *Online Help* provides instant access to context-sensitive help.

The *Administrator's Guide* covers the basic concepts to ensure you to get the very most from the software. It is designed to provide effective help for administrators, both first-time and experienced users.

The *User Guide* provides an introduction to using the application. It is designed to get you off to a quick start with the software by familiarizing you with the basic concepts before looking at some of the more advanced functions.

The documentation is designed for use in conjunction with the integrated online help.

Typographical conventions

This manual distinguishes between different types of information using the following conventions:

- Terms taken from the interface itself, such as commands, appear in **bold**.
- Keys appear in small caps. For example: ”Press the SHIFT key.”
- Numbered lists mean there is a procedure to follow.

- When the conjunction -or- appears next to a paragraph, it means there is the choice of another procedure for carrying out a given task.
- When a menu command contains submenus, the menu name followed by the command to select appear in bold. Thus, "Go to **File > Open**" means choose the **File** menu then the **Open** command.

This symbol highlights warnings and other important information on how a particular command or procedure works.

Following this symbol you will find hints and tips for optimizing tasks, speeding up commands, and so on.

CHAPTER 1

Installation

This chapter covers the following topics:

System Requirements

- Server Requirements
- Workstation Requirements

Software Protection Key

Server Installation

Design/Workstation Installation

Running the Label Print Manager Service

- Setting up a User Account
- Installing and Uninstalling the Service

System Requirements

When installing this application, you will install two separate components of the program – the Server component and the Design/Workstation component.

Server Requirements

The Server installation allows you to install the following components on your server: the Kernel, the label print engine, the sentinel manager and the User Manager.

The following requirements must be met to ensure successful installation:

- OPERATING SYSTEM
  Windows® 11 x64
  Windows® 10 x86/x64
  Windows® Server 2025
  Windows® Server 2022
  Windows® Server 2019
  Windows® Server 2016
  Windows® Server Core 2025
  Windows® Server Core 2022
  Windows® Server Core 2019
  Windows® Server Core 2016
- VIRTUAL MACHINE PLATFORMS
  Requires VM license
- MEMORY
  4GB or more RAM (8GB recommended - depending on OS and usage)
- HARD DISK SPACE
  A hard drive with at least 10 GB free disk space (depending on the installation options).
- DISPLAY
  Video card: Windows display adapter capable of 1024x768 with True Color capabilities and DirectX® 9 (DirectX 11 compliant card recommended) .
- SUPPORTING SOFTWARE
  Microsoft .NET Framework 4.8 (provided with the installation)
  Any pdf reader
  Microsoft .NET Desktop Runtime 10 (provided with the installation)

Microsoft ASP.NET Core Runtime 10 (Hosting bundle) (provided with the installation)

- ADDITIONAL REQUIREMENTS
  Installation requires full Administrator privileges.

Note

The release note contains the latest information on the software. This information supersedes the information contained in this manual.

If you want to create labels on the workstation, you will have to purchase a separate license for the label designer to work with this software.

Workstation Requirements

The Design/Workstation installation allows you to install the Mapper module, the Label designer and the Query Manager on a workstation.

The following minimum requirements must be met to ensure successful installation.

- OPERATING SYSTEM
  Windows® 11 x64
  Windows® 10 x86/x64
  Windows® Server 2025
  Windows® Server 2022
  Windows® Server 2019
  Windows® Server 2016
  Windows® Server Core 2025
  Windows® Server Core 2022
  Windows® Server Core 2019
  Windows® Server Core 2016
- MEMORY
  2GB or more RAM (4GB recommended - depending on OS and usage)

- HARD DISK SPACE
  A hard drive with at least 1.5 GB free disk space (depending on the installation options).
- DISPLAY
  Video card: Windows display adapter capable of 1024x768 with True Color capabilities and DirectX® 9 (DirectX 11 compliant card recommended) .
- ADDITIONAL REQUIREMENTS
  Installation requires full Administrator privileges.
  A label designer license (software or dongle) if you intend to create labels on this workstation

Software Protection

Your software is protected using a software key (an electronic code) or a hardware key (a dongle).

The dongle is a small electronic device that you plug into your PC's parallel or USB port before launching the software.

The software key is an electronic code that is requested by the Activation Wizard when you launch the software for the first time or as long as you have not activated the software.

You must first install the software and then activate or connect the protection key to your computer.

Activating the software key protection

Once you have installed the labeling software, the **Activation Wizard** will be launched, taking you step-by-step through the software key protection process.

To activate the software key:

In the **Activation Wizard**, select **Activate.**

There are up to four possible activation methods: **Online activation, Using a USB key**, **Using a smartphone** and **Using our website.**

Note

If you have an Internet connection on the computer where the software is installed, online activation is automatically selected.

Online activation

Some enterprise networks are protected by a proxy server. In this case, there is a **Settings** button to setup your proxy settings to connect to the Internet.

1 Enter the **Serial number** and the **password** provided with the product and click **Next**.

2 Depending on your connectivity, Connection Test step is displayed. This allows you to setup your connection. Click Next.

3 Enter or edit your information in the User Registration form. Click **Next**.

4 Click **Finish**.

Note

If you do not have an Internet connection on the computer where the software is installed but you have access to another computer with an Internet connection, you can activate the product using a USB key.

Using a USB key

1 Enter the **Serial number** and the **password** provided with the product and click **Next**.

2 **Select Using a USB key.**

3 Select the media to be used for activation (USB key or external hard disk or network drives) and click **Next**. Files for activation will be copied onto the selected media.

4 USB devices only: Unplug the removable media and plug it into the computer connected to Internet.

5 On the computer connected to the Internet, double-click USBWizard.exe (which can be found on the root of the media). This will start the Activation wizard.

6 Select the license(s) to activate and click **Next**.

7 Enter or edit your information in the User Registration form. Click **Next**.

A message will inform you that the activation information for the selected license(s) has been successfully retrieved.

8 **USB devices only**: Unplug the removable media from this computer and plug it back into the computer where the software is installed.

9 On the computer where the software is installed, double-click **USBWizard.exe** again (the file can be found on the root of the media).

The wizard will display the license that can be activated on the computer where the software is installed.

10 Click **Next**.

11 Click **Finish**.

If any problems were encountered, a message will inform you that the activation has failed and an error will be displayed.

Note

If you do not have access to a computer with an Internet connection, you can activate the product using a smartphone.

Using a smartphone

A barcode scanner application is required to read the

barcode which allows you to activate your software using a smartphone. You can download this from your application store / market. This activation mode uses an Internet connection on your smartphone.

1 Enter the **Serial number** and **Password** provided with the software and click **Next.**

2 Select **Using a smartphone**.

3 Input your information in the User Registration form. Click on **Next**.

4 A barcode appears. Move the cursor over the barcode to zoom in.

5 Open your barcode scanner application on your smartphone.

6 Scan the barcode with your smartphone.

7 Click **Continue** or open the browser on the phone.

8 The Software Activation website is displayed. This allows you to register and activate your software. All fields must be filled in.

9 Click **Submit**.
The **Validation code** and **Installation code** are displayed at the bottom of the Software Activation website page. Please scroll down if necessary.

10 Enter the **Validation code** and **Installation code** in the wizard (using uppercase characters).

11 Click **Next**.

12 Click **Finish**.

Using our website

1 Enter the **Serial number** and **Password** provided with the software and click **Next.**

2 Select **Using our website**.

3 The link provided on this screen (http://www.teklynx.com/nsp/) can be reached from a different computer with Internet access. You or someone else with an Internet connection can connect to our website to complete the activation. The **Serial Number** and **Password** (provided with the product) and the **User Code** (displayed in the wizard) will be required in order to use this service.

4 Enter the license user's information in the User Registration form displayed on our website. Click **Next**.

5 Click **Submit**.

6 The **Validation code** and **Installation code** are displayed on our website. Enter these in the wizard (using uppercase characters) and click **Next**.

7 Click **Finish**.

Server Installation

Server installation allows you to install on your server the various modules used to build an automatic data exchange solution between your applications and devices and our labelling products.

Step 1 Place the DVD in your server's DVD drive.

Step 2 The installation program should launch automatically.
If it does not, go to Windows explorer and expand the letter of the DVD drive. Double click SETUP.EXE (for example, D:\setup.exe).

Step 3 In the tree structure containing the various installation options, select the server option of the product you've purchased.

Step 4 Follow the on-screen instructions to complete the installation process.

Design/Workstation Installation

Design/workstation installation installs the Mapper module, which is used for defining how data exchange will be done between your application and the Label Print server. In the Design/Workstation installation of the product you have purchased, the Query Manager application will also be available for installation. This installation allows you to set up queries over a database system that can be executed by the sentinel server during a data exchange between your data files and your labeling software.

Step 1 Place the DVD in your workstation's DVD drive.

Step 2 The installation program should launch automatically.
If it does not, go to Windows explorer and expand the letter of the DVD drive. Double click (e.g. D:\start.bat).

Step 3 In the tree structure containing the various installation options, select **Design/workstation** and then the required components.

Step 4 Follow the on-screen instructions to complete the installation process.

If you want to design a label and to use the label design option available with the Mapper, you will have to purchase a separate license for the label designer to work with this software.

Running the Label Print Manager Service

The application runs as a service on the host workstation. This means it will launch automatically when the workstation is started.

Step 1 To define the startup options for the Label Print Manager Service, open **Services** control panel and select **Label Print Manager**.

Step 2 Select a start method.

Choose either:

Automatic: The service launches when the workstation is started.
Manual: The service must be activated manually. You will therefore need to open a Windows work session and then launch the service in Control panel Services. Select **Label Print Manager** from the list of services and then click **Start**.

The above start methods launch the service and the sentinels at the same time. All available sentinels are thus launched when the service is started.

Whatever the start method is, you can control the service activity by launching the Kernel from the application programs group.

If the service is stopped, a start request is sent to the system. When the service is started, the application management icon displays in the system tray. Using this icon, you can start or stop the service and the different application modules.

Unlike the start method using the Windows Control panel services, the start method from the Kernel does not launch sentinels (which would instead be launched from the Manager).

Setting up a user account

The application must be able to access shared network folders and print on network printers. You must therefore select a user account to which these rights have been assigned.

Step 1 Select a user account, then enter the password.

The selected account must be declared in the local administrator group of the workstation.

To view sentinel status while the service is running, simply open a work session and then launch the application via the program group defined during installation. The default name of the group is the same as that of the application.

Installing/ Uninstalling the Label Print Manager Service

Once the application is installed, you can always uninstall the Label Print Manager service and reinstall it later.

To install the **Label Print Manager** service,

select **Start** > **Run** **C:\Windows\Microsoft.NET\Framework\v4.0.30319\ installutil.exe" [name of application installation folder] \ TKXKernel.exe"**

To uninstall the **Label Print Manager** service,

select **Start** > **Run** **C:\Windows\Microsoft.NET\Framework\v4.0.30319\ installutil.exe "[name of application installation folder] \ TKXKernel.exe" /u**

CHAPTER 2

Introduction

This chapter covers the following topics:

Definition of Terms

Basic Concepts

How it Works

This application is a server application dedicated to the data exchange between your application and our products.

It allows you to automatically print barcode labels created with your labeling software by simply exchanging information between your ERP/WMS system and the sentinels server.

Definition of terms

The application involves a number of specific concepts which are explained below.

Sentinels server: PC running supported Microsoft Windows platform on which the application is installed.

Input data: The data generated by your system. They are received by the sentinels server via a communication channel.

Bearer channel: Once launched, each sentinel "listens" to a specific bearer channel.

Sentinel printer: The printer created by the application on the print server which users can then designate as the output printer in their applications.

Sentinel: The procedure used for analyzing and processing data generated by your application.

Mapfile: Describes the method the sentinel will use for analyzing input data.

Plug-in: The process module executes a specific task during a sentinel process.

There are four types of plug-ins:

- The **Input plug-in** listens to a data bearer channel supplying a sentinel.
- The **Process plug-in** processes information retrieved from input data.
- The **Report plug-in** informs users and updates a report

about sentinel process.

- The **Output plug-in** backs up or transmits input data to a file

Plug-ins list available in your application depends on the product used.

Basic concepts

The application monitors data communication channels connected with your system. As soon as a channel receives data, the application analyzes the incoming data and processes them through one or more process plug-ins. The communication channel has to be chosen according to your application and the input plug-in you have at your disposal.

How it works

The application runs as a Windows background task. This application is considered a service by the system and can thus be launched with the system. Sentinels are used to define analysis and printing tasks. Once launched each sentinel monitors the bearer channel for which it has been configured. When data coming from your system are received on the bearer channel, the analysis and process activity begin.

Input data is filtered by the map file of the sentinel. As for each printing request, input data information is sent to the different process plug-ins configured for the sentinel.

At the end of the process, input data is sent to the output plug-in in order to be transferred or saved.

During each step of the process and after sentinel validation, information concerning the sentinel process is sent to the report plug-in.

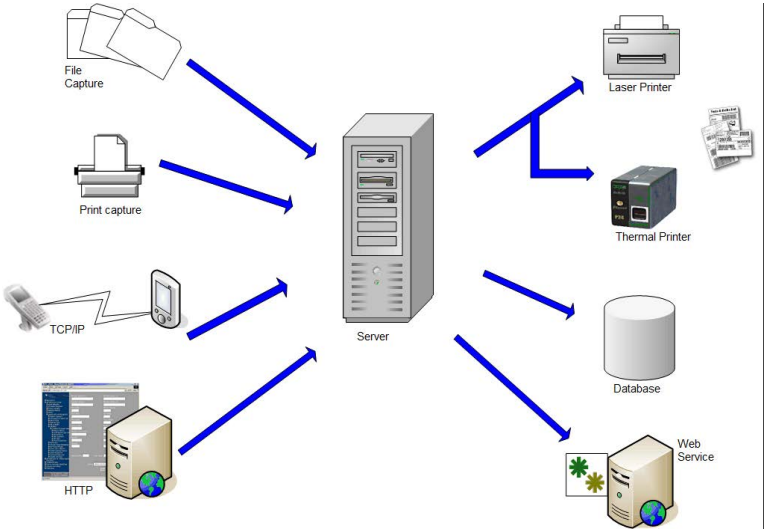


Figure 1: Input and Process Plug-in Flow

CHAPTER 3

The Application Modules

This chapter covers the following topics:

The Application Modules
- Web Manager
- User Manager
- The Kernel
- Sentinel Manager
- Mapper
- Query Manager
- Plug-ins

The major steps from creating to running the sentinels.
- On your Workstation
- On the Server

The application modules

Web Manager

The Web Manager acts as an interface to the sentinels server over the internet/intranet. It allows the user to interact with any sentinels server, regardless of its location, via their Internet browser by typing in the web application's address in the browser's address line.

User Manager

The User Manager secures the internet connections to the sentinels server, by allowing an administrator to set permissions for application users.

Kernel

The Kernel is the core application that manages all other tasks, including launching sentinels and monitoring sentinel activity. If the Kernel is closed, all sentinels will be disabled.

The Kernel operates as a background task and can be configured to launch automatically when your workstation is started up. It is therefore not necessary to open a Windows work session for the application to run.

Once launched from the application program group, the Kernel module appears as an icon on the Windows taskbar.

You can access Sentinel Manager via the Kernel menu.

Step 1 Right-click on the Kernel icon

Step 2 Choose Sentinel Manager in the menu.

Sentinel Manager

Sentinel Manager is launched via the Kernel menu. It allows you to create, configure, and manage the sentinels that will analyze your input data, to view sentinel status, activate or deactivate individual sentinels and display the error logfiles.

Mapper

The Mapper allows you to create and configure the mapfiles which will analyze your input data. A mapfile describes the structure of the input data for analysis and assigns the mapped data to the corresponding variables in the initial document. Mapper is accessed via the application program group or Sentinel Manager.

Mapper features an integrated wizard to guide you through the process of creating and configuring mapfiles. See the *User Guide* for a full description of the module and examples of how to create mapfiles.

Query Manager

This application module lets you define queries over different database systems and lets you group them into tasks that can later be invoked by a sentinel through the use of the database process plug-in.

The major steps from creating to running the sentinels

Here is a general overview of the main steps to take in order to configure a sentinel that will receive text files into a specific folder and will print the corresponding barcode label.

On your workstation

Use the Mapper module to:

- Define the structure of the files that will be used for label printing (known as “mapfiles”).
- Design the labels directly in your labeling software.
- Test the map.

On the server

Use Sentinel Manager to:

- Create sentinels that will analyze your input data from the text files.
- Define their properties and the properties of the plug-ins associated with them.
- Enable sentinels for starting.
- Enable individual sentinels.
- Launch printing.
- Monitor printing.

-or-

Use the Web Manager, via the web, to create, define, enable and launch sentinels. The Web Manager acts as an online Sentinel Manager.

CHAPTER 4

User Manager

This chapter covers the following topics:

Launching the User Manager

Adding New Users

Defining or Changing User Rights

Changing a Password

Setting Permissions

In order to secure the internet connections to the sentinels server, the User Manager will allow an administrator to set permissions for application users. The different types of permissions include:

- **Administrate**: Allows the user to launch the Security Management module and change the rights of other users.
- **Manage**: Allows the user to add, remove or configure sentinels as well as enable/disable sentinels.
- **Operate**: Allows users to start/stop sentinels.
- **Control**: Allows the user to view sentinel activity via the Web Manager without any right to alter the settings or activity of the sentinels in any way.

Launching the User Manager

The User Manager can be launched from Sentinel Manager Tools menu.

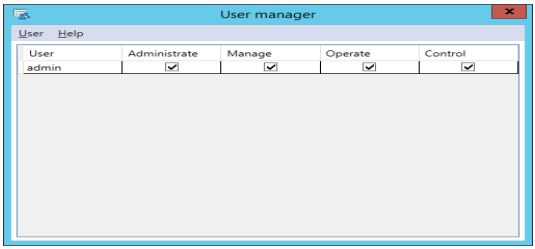


Figure 2: User Manager window

Adding new users

Once you've launched the User Manager:

Step 1 Go to **User** and select **Add**.

A box will pop-up prompting you to enter a user name, user role and password.

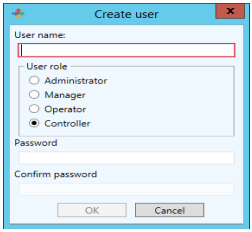


Figure 3: Create User window

Step 2 Click **OK**.

You will see that the user has been added to the list.

Defining or changing user rights

Determine the user whose rights you want to set or change.

Step 1 Go to **User**.

Step 2 Select the highest level or rights you want to assign the user.

If like to give a user the ability to start/stop sentinels, select **Operate** from the list of available permissions. You will notice that **Yes** appears for each right given to the user. In this case, you would see **Yes** under **Operate** and **Control**, but not under **Manage** and **Administrate**.

Changing a password

The User Manager also allows you to change the password assigned to users.

Step 1 Go to **User**.

Step 2 Select **Change password**.

Step 3 Enter the old password.

Step 4 Enter a new password.

CHAPTER 5

Web Manager

This chapter covers the following topics:

User Interface

Permissions

Using the Web Manager
- Creating or Modifying an Existing Sentinel
- Sending Out the Sentinels

The Web Manager is the user interface for the sentinels server. It can be accessed by any user, from any browser, anywhere in the world, as long as that user has been given permission, a user name, and a password by the administrator.

Note

The installation activates the Internet Information Services (IIS) feature if it is not already activated and it creates the "Web Manager" web application in a dedicated application pool.

The Web Manager can be installed on a different server from the one sentinels server runs on. Installing them on separate servers ensures that:

- The deployment and maintenance of the Web Manager application is easier and runs more smoothly since a single web server can administrate multiple sentinels server on the network.
- They are each running at full power by giving the Web Manager and the sentinels server their own server. Depending on the number of opened sessions, the Web Manager can consume a lot of processor time which may hinder the power of sentinels server.
- Web Manager host PC must have access to the server where SENTINEL Service is running and to the port 4504.

User Interface

Typing in the Web Manager's URL in the browser will display the Data Exchange Server's login page.

Example: http://192.0.0.1:8090

Note

By default, Web Manager installation proposes to use "All unassigned" address. It means that all free IP addresses of the server that are not specifically assigned to another website hosted on the server will lead to the Web Manager.

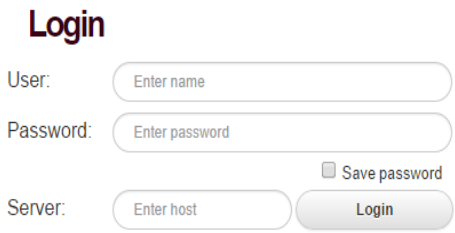


Figure 4: Web Manager Login screen

This page is the gateway to the Sentinel Control and Configuration page.

Step 1 Enter a valid user name and password (these have been configured and given to you by an administrator).

Step 2 Select the name of the sentinels server you want to work with.

Step 3 Click **Connect**.

The selection of the server is done from a combo box. This combo box is linked to a file that the administrator has configured which lists the available sentinels server names and the associated TCP/IP port to be used (if no port is specified, a default port will be used instead).

If your connection is successful, a page listing all the sentinels defined on the sentinels server will be displayed. If not, please try again, or contact your administrator to verify that your user name and password are correct.

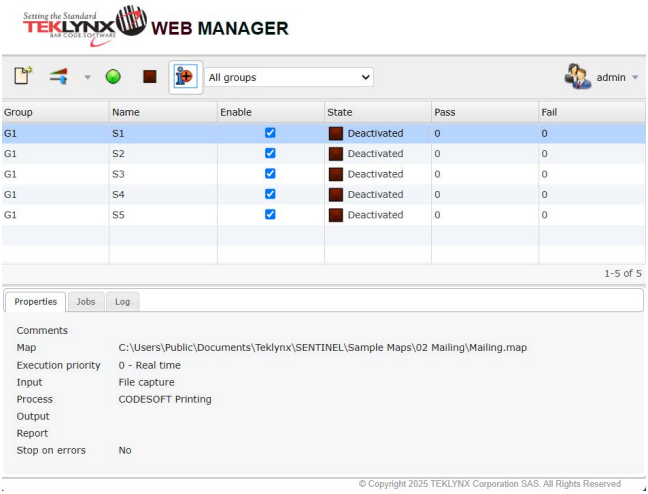


Figure 5: Sentinel list window

Permissions

The administrator can set up your account to give you some or all of the possible permissions.

Depending on the permissions you have been given, your account may allow you to:

- Create a new sentinel
- Delete one or more sentinels
- Edit or modify the settings of a particular sentinel
- Enable one or more sentinels
- Disable one or more sentinels
- Start one or more sentinels
- Stop one or more sentinels
- Refresh the Web Manager

The following icons are to be used for the above mentioned functions:

Create a new sentinel

Edit or modify the settings of a particular sentinel

Start one or more sentinels

Stop one or more sentinels

Using the Web Manager

Once the user has successfully logged in, a list of available sentinels will be displayed. Just like in the Sentinel Manager, the group name, the sentinel name, its state and whether the sentinel has been enabled will be displayed on the screen.

One or multiple sentinels can be selected by clicking in the check box - a check mark will appear to show that it has been selected and the selected sentinel(s) will change to the color orange. Clicking in a group box will automatically select all the sentinels belonging to that group. Disabling a sentinel will change the ‘enable’ box color to red. Starting the sentinel will change the ‘state’ box to green. In an unselected, enabled and stopped state, the sentinel remains in the default color - white.

The icons below the list can be used to perform a number of tasks as long as the user’s account has the permission to do so.

Creating or Modifying an Existing Sentinel

Clicking the **Create new sentinel** icon or selecting a sentinel and clicking the **Edit/Modify sentinel** icon will bring up the Sentinel Configuration screen. Here, the user can select the sentinel’s input/output/process/report plug-ins.

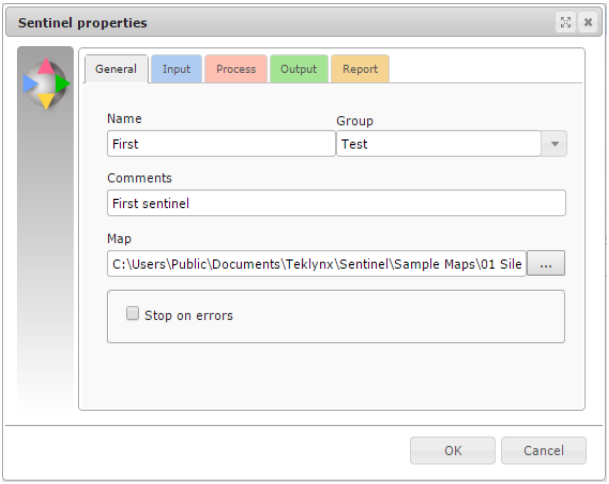


Figure 6: Plug-in settings screen

Only one input plug-in can be selected for each sentinel, however, you can select as many output, process or report plug-ins as you would like.

For more information on the different plug-ins and their settings, please see *Appendix A - Plug-ins*.

Sending Out the Sentinels

To send out the sentinels, you must:

Step 1 Select the sentinel(s) you want to activate by clicking in the appropriate check box.

Step 2 Click to start the selected sentinel(s). The screen should automatically refresh to show the state of the sentinel(s) as "running."

Clicking will stop the selected sentinel(s).

Stopping sentinels may take a few seconds. The more sentinels you stop at one time, the longer it may take.

CHAPTER 6

Sentinels

This chapter covers the following topics:

What Are Sentinels?

How Do Sentinels Work?

What are sentinels and how do they work?

A sentinel is simply a group of settings the application uses to intercept and process the information coming from your application and devices.

Once activated, each sentinel acts on your system as an analyzing and processing service giving you the ability to exchange data between your application and TEKLYNX products in order to build a complete automatic ID solution.

Each sentinel must specify:

- A bearer channel on which your printing requests will be received.
- A method for data analysis given by the map file selection associated with a sentinel.
- One or several types of processes to be implemented for each information block identified in the input data according to the sentinel map file.
- One or several types of logs to be generated when a specific event is detected during analysis or during input data process.
- One or several types of backup input data once they have been treated.

A sentinel must belong to a group. Each group can contain as many sentinels as your organization requires. The organization of sentinels within a given group is just performed for organizational purposes.

The choice of possibilities for communication channels, type of process, type of log generated, and type of backup will vary depending on the labeling software version you are using and the plug-ins available for that product version.

How do sentinels work?

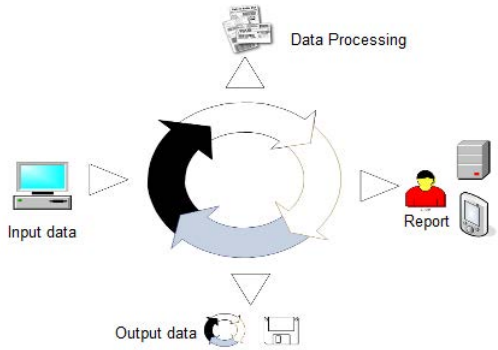
After you have selected an input plug-in and one or more output plug-ins for your sentinel, when the sentinel is activated, the input plug-in is also activated and it monitors its bearer channel. When information is received on its channel, the plug-in sends input data to the core application. Data are then analyzed according to the map file associated to the sentinel.

For each information block detected, the core application sends data to the different plug-ins associated with the sentinel. Data are treated according to the order sets up in the sentinel. Once data are treated, the analysis goes on towards the next blocks. The same process is repeated until all the data have been processed.

The core application sends data received by the input plug-in to the different output plug-ins. These plug-ins are configured to implement the different backup processes.

For each step of the process, the activity parameters of the sentinel are provided to the selected log plug-ins, by the core application.

The diagram below shows the workflow.



Note

Some sample files for the this applications are available in the document folder of the application (C:\Users\Public\Public Documents or C:\Document And Settings\All Users\Documents)

CHAPTER 7

Sentinel Manager

This chapter covers the following topics:

Launching the Program

Description of the Main Window
- Menu Bar
- Workspace
- The Sentinel List
- The Toolbar
- Selecting a Tool
- Information Panel
- The Status Bar

Interface Settings
- Changing Display Options
- Changing the Interface Language
- Adjusting Column Width
- Adding a Sentinel
- Defining the Properties of a Sentinel
- Displaying a Specific Group
- Modifying the Display Order
- Displaying Sentinel Properties
- Activating one or more Sentinels
- Deactivating one or more Sentinels
- Terminate unit

Managing Sentinels
- Duplicating a Sentinel
- Deleting a Sentinel
- Activating a Sentinel

Sentinel Status
Task Management
    Viewing Current Tasks
    Cancelling a Task

Error Management
    Structure of the Error Logfile
    Error Messages
    Deleting an Error Logfile

Launching the program

Once launched from the application program group, the Kernel module appears as an icon on the Windows taskbar.

To access Sentinel Manager:

Step 1 Right-click on the Kernel application icon

.

Step 2 Choose Sentinel Manager in the menu.

The main Sentinel Manager window appears.

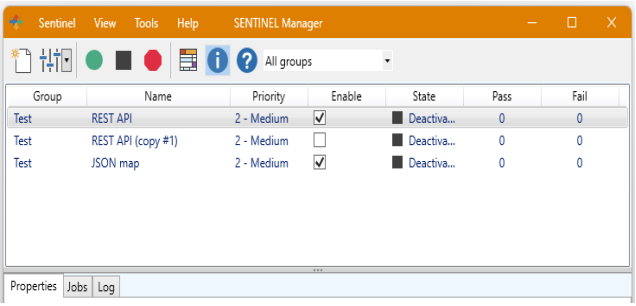


Figure 7: Sentinel Manager window

Description of the main window

This section presents a general overview of the main elements that make up the interface as they appear in the main window at the beginning of a work session. The main window is divided into two areas:

- The sentinel list
- The information panel

Menu Bar

The menu bar comprises four drop-down menus: **Sentinel**, **View**, **Tools,** and **Help**.

Workspace

The workspace occupies the central part of the main

To access commands using the keyboard, use the keyboard shortcuts. Press ALT plus the key corresponding to the letter underlined in the menu name, then the key corresponding to the letter underlined in the command name.

window. The list of sentinels is displayed in this area as a table.

The Toolbar

These tools allow you to execute routine tasks more quickly than using the menus.

To select a tool, click the button corresponding to the tool.

The Sentinel list

The sentinel list occupies the central part of the main window and is displayed as a table.

Group	Name	Priority	Enable	State	Pass	Fail
G1	S1	0 - Real time	☑	Deact...	0	0
G1	S2	1 - High	☑	Deact...	0	0
G1	S3	2 - Medium	☑	Deact...	0	0
G1	S4	3 - Low	☑	Deact...	0	0
G1	S5	3 - Low	☑	Deact...	0	0

Information panel

The information panel, located in the lower part of the window, comprises three tabs which display information on pending tasks and the printing/error logfiles respectively. You can show or hide the information panel using the command **View Information** or the **Information tool** button .

You can move the border separating the sentinel list and the information panel by dragging it with the cursor.

Interface settings

Changing display options

You can display the interface in English or French. You can also modify column widths and select a display filter.

Changing the interface language

Step 1 Choose **Tools** > **Options**.
Step 2 Select the required language in the language zone.

A checkmark shows which language is selected.

Adjusting column width

Place the cursor over the column's right-hand border in the table header and drag it to the left or right to obtain the required width.

Adding a sentinel

Choose **Sentinel** > **New**.

The new sentinel appears in the table. Its status is defined as "under construction".

Defining the properties of a sentinel

The **General** tab allows you to assign a name to the new sentinel, add any comments and associate it with a group and a map file.

Step 1 Enter a name in the zone **Name**.

Step 2 In the zone **Group**, select a group or type a group name. If needed, type comments.

Step 3 Select the map file defining the structure of your data file.

Step 4 Select the **Stop on errors** check box if you want a sentinel process to be stopped when an error occurs.

The others tabs allow you to define which plug-in you want to run for the Input, the Output, the Process and the Report.

If you want more information about the plug-ins, please refer to *Appendix A: Plug-ins* at the end of this manual.

You will also find information about the plug-in selected in

each tab of the Sentinel Properties dialog box.

Displaying a specific group

You can display sentinels belonging to a particular group.

Choose **View** > **Group** and then select the required group of sentinels.

- or -

Select the option directly from the toolbar.

Modifying the display order

Click on the header of the column by which you wish to sort the sentinels.

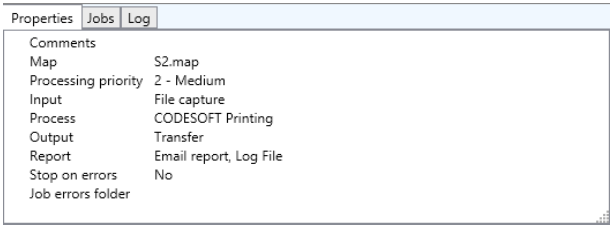
Displaying sentinel properties

You can display the properties of a sentinel.

Step 1 Select the sentinel.

Step 2 Choose **View Information**.

Step 3 Click the **Properties** tab.



Managing sentinels

Sentinel Manager allows you to carry out a number of operations, including duplicating an existing sentinel.

Duplicating a sentinel

Step 1 Select the sentinel.

Step 2 Right-click the sentinel.

Step 3 Choose **Duplicate** in the context menu.

The duplicated sentinel appears in the list with the same name followed by *copy no. x*.

Deleting a sentinel

Sentinel Manager also allows you to delete sentinels.

Step 1 Select the sentinel.

Step 2 Right-click the sentinel.

Step 3 Choose **Delete** in the context menu.

Activating one or more sentinels

In order to activate a sentinel (from the service or from the Sentinel Manager), you must first enable it.

Step 1 Select the required sentinel(s) using the mouse or the Select all and Invert selection commands.

Step 2 Check or un check the box in the **Enable** column.

Step 3 Click the button to activate the selection.

- or -

Choose **Sentinel > Activate**.

Deactivating one or more sentinels

Step 1 Select the required sentinel(s) using the mouse or the **Select all** and **Invert selection** commands.

Step 2 Click the button to deactivate the selection.

- or -

Choose **Sentinel > Deactivate**.

Unlike sentinels are activated by the service, all sentinels launched manually will stop processing when the application is closed.

Terminate unit

The existing stop mechanism performs a graceful shutdown by waiting for the currently running job to complete before stopping the unit. While this approach works well under normal conditions, it becomes ineffective in situations where job processing hangs. To address this limitation, a new **terminate unit** functionality has been introduced. This feature allows immediate interruption of ongoing job processing.

The function can be triggered from the Sentinel Manager by pressing the red octagonal stop button .

Upon activation, the application prompts the user for confirmation. Once confirmed, a termination signal is sent to the SENTINEL service.

The SENTINEL service then propagates a cancellation signal to all active plugins. Each plugin is responsible for interrupting its current processing as quickly as possible, using the best available method. Any job that was in progress at the time of termination is automatically marked as failed.

Note

Terminate function is more priority than the usual stop.

Sentinel status

A symbol appears next to each sentinel in the list to indicate its status.

This symbol indicates the sentinel is deactivated.

This symbol indicates the sentinel is about to begin analysis.

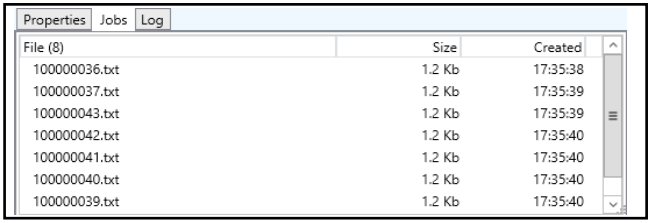
This symbol indicates the sentinel is under construction.

This symbol indicates the sentinel is activating.

This symbol indicates the sentinel is processing a datafile.

Task management

The Information panel allows you to monitor the progress of current tasks.



Viewing current tasks

Activate the Information panel and click the **Jobs** tab.

Error Management

When Sentinel Manager detects an error during datafile analysis, it updates the relevant error logfile located in the application LOGFILES folder. The filename of the logfile will be that of the sentinel. The logfile describes the nature of the error so you can rectify it if necessary.

To view the error logfile for the selected sentinel, display the Information panel, and then click the **Log** tab.

The **Fail** column and the sentinel symbols indicate any errors encountered during datafile analysis.

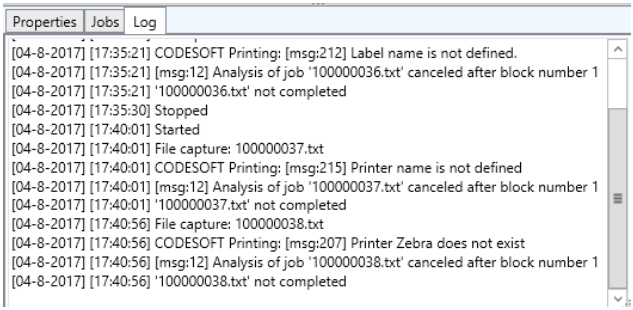
To reset failed jobs counter, select a sentinel and then choose **Reset** in Sentinel menu.

Structure of the error logfile

This file allows you to monitor file processing status.

Error messages

Error messages inform you of the nature and source of any errors so you can rectify them. Error messages include the error number, the date and the time, the name of the file concerned, and the error message.



Deleting an error logfile

You can delete error logfiles using the Information panel.

Step 1 Select the sentinel to which the error logfile is associated.

Step 2 Click on the Information panel **Log** tab then press CTRL + DEL.

Deleting an error logfile will permanently remove the .log file.

APPENDIX A

Plug-ins

This appendix covers:

Information about the configuration and selection of the different plug-ins:

- Input
  - File Capture
  - TCP/IP Port Listening
  - Print Capture
  - Web Server
  - Records Capture
  - SFTP file capture
  - REST Web Server API
  - Modbus Server
- Process
  - Label Printing
  - Database
  - Web Service Client
  - Email
- Output
  - Transfer Plug-in
  - Email
- Report
  - Log File
  - Email
  - TCP/IP Port Listening

plug-ins list available in your application depends on the product used.

Input plug-ins

You have eight input plug-ins at your disposal:

- File Capture
- TCP/IP Port Listening
- Print Capture
- Web Server
- Records Capture
- SFTP file capture
- REST Web Server API
- Modbus Server

File capture

The File Capture plug-in collects files in a specific folder.

Files are processed in chronological order – the first file to appear in the folder is the first file processed.

To be analyzed by the sentinel, the file must be accessible in read/write mode. In addition, the sentinel must have full control over the data file before it will proceed with processing (the program that is creating the data file for sentinel must be finished writing the file).

From the **Input** tab of the Sentinel Properties dialog box, select the Input plug-in and click **Settings**.

The following dialog box appears:

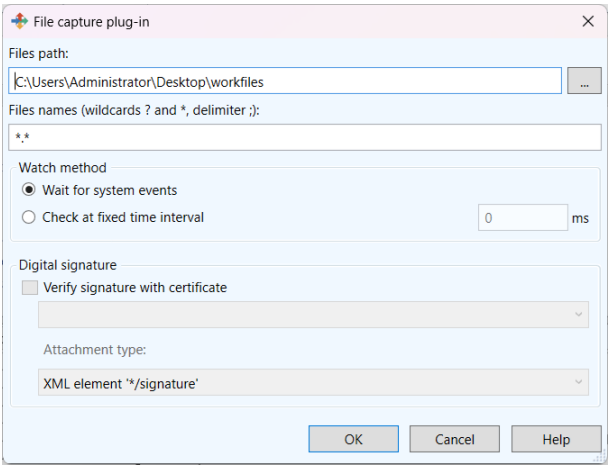


Figure 8 The Print Capture Plug-in window

Step 1 Type or select the file path of the folder to be watched.

Step 2 Type file name or file extension of the files to be captured.

You can type several files names separated by a semicolon or use wildcard characters. Asterisk (*) is used to replace zero or several characters. Question mark (?) is used to replace one and only one character in a file name.

Examples

a*.txt	All files with extension .txt starting with a (or A), like AF104.txt or a.txt.
item_n?.*	All files named item_n + one character, whatever the extension, like item_n3.dat or ITEM_NZ.txt, but not item_n24.doc
.txt;.dat	All files with extension txt and dat

Step 3 Choose your watch method:

- **Wait for file system events:** In this mode, the application is suspended until the system «awakens» when a file is in the directory. At that time, the file processing starts.

> This method is a good way to limit the resources needed by the application. However, this method cannot always be used. Some file systems, such as shared folders on AS400, are not compatible with the Windows notification system.

- **Check on time interval:** The folder is watched at regular intervals.

TCP/IP port listening

This plug-in is viewed as a Socket server and receives data from a client in a pre-defined format. For more information about this format, please refer to the example available in your installation directory. This plug-in can be considered as an input and/or a report plug-in. It allows client application to retrieve printing status. This plug-in can support any kind of platforms. This plugin supports localhost.

From the **Input** tab of the sentinel properties dialog box, select the input plug-in and click **Settings**.

The TCP/IP Plug-in window appears:

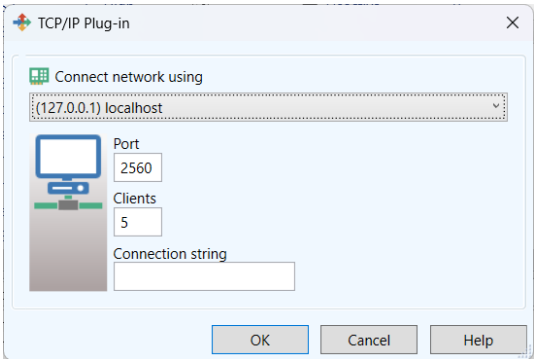


Figure 9: The TCP/IP Plug-in window

How does the TCP/IP Plug-in work?

Step 1 The client tries to connect.

Step 2 The server validates the connection.

Step 3 The client sends a data process request.

Step 4 The server sends the events linked to the file process.

Step 5 The server sends the end of process.

Step 6 (optional) Repeat steps 3-5.

Step 7 The client disconnects.

If this plug-in is considered as both an input and report plug-in, events reported by the server to the client will concern only the client that sent the input.

General message format

A message is sent to exchange information between the client and the Socket Server. The structure of the message is defined below.

Message size (including those four characters)	4 bytes (int)
Command number	4 bytes (int)
Parameter 1	4 bytes (int)
Parameter 2	4 bytes (int)
Job name	260 bytes (array of char)
Data	varying length

Data format description

Data sent will have the following format:

variable name = value CRLF

The variable name is the name of a variable on your label or that of a control variable:

@LABEL_NAME: Used to indicate the label name to be printed.
@PRINTER_NAME: Used to indicate the target printer.
@LABEL_QUANTITY: Used to indicate the number of labels to be printed.

If you want to send several process demands during the same information exchange, data blocks must be separated by CRLF (ASCII characters 10 and 13).

If there are several blocks of data, blocks are separated by CRLF:

variable name 1 = value CRLF
variable name 2 = value CRLF
variable name 3 = value CRLF
CRLF
variable name 1 = value CRLF
variable name 2 = value CRLF
variable name 3 = value CRLF
variable name 4 = value CRLF

Commands list

Login request

The connection string indicated in the plug-in configuration is aimed at allowing the client to connect to the server.

Clients connection limitation: each client that is validated with this command counts towards maximum connections count. When maximum number of connections is reached, new connections will be refused. Note than only logged in clients count.

Command number	0
Parameter 1	0 - message in ANSI (Default Windows Code Page, 1 byte)
	1 - message in UNICODE (UTF16LE, 2 byte)
Parameter 2	Client ID. Optional parameter used to check information sent from the server to the client.
Name	Not used.
Data	Connection string.

Job request

Command number	2
Parameter 1	0 - message in ANSI (Default Windows Code Page, 1 byte)
	1 - message in UNICODE (UTF16LE, 2 byte)
Parameter 2	Not used.
Name	Job name.
Data	Data to process.

Response to login request command

Once the connection demand has been made, the client must wait for an answer from the server in order to continue.

Job name	Client ID.
Command number	1

Parameter 1	0 (OK)
	1 (wrong connection string)
	2 (maximum number of clients reached)
Parameter 2	Not used.
Name	IP address or Client PC name + Client ID (value set during connection)
Data	Not used.

Response to job request

Command number	3
Parameter 1	0 (OK)
	1 (cancelled)
	2 (message)
	3 (error)
	4 (end of process)
Parameter 2	See table below.
Name	IP address or client's PC name + client ID (value set during connection).
Data	Text message linked to the receiving of an error or a message. The module name or the name of the plug-in is defined before this message.

Details about parameter 2

On message reception and according to what has been selected in the Report plug-in.	0 (Sentinel stopped: client disconnected)
	1 (Data received)
	2 (Data to be processed)
	3 (Data are sent to the output plug-in)
	4 (Free message)
On error reception	3 (Format not supported)
	4 (Process error)
On end of process	Equal to the NewData return value.
	0 (OK)
	1 Process canceled.
	6 Process error.
	7 Output plug-in error.

Communication between the client and the server can either be done in UNICODE(UTF16LE) or in ANSI (Default Windows Code Page).

Pay attention: the default Windows code page (ANSI) is not the same for different locals (for US locale it is CP-1252) and different code pages are not compatible.

If the data is sent in UNICODE, the server will answer in UNICODE and vice versa.

Logout command

This command is optional. Use it when you prefer to use plugin in "connect-process data-disconnect" mode.

Command number	4
Parameter 1	Not used
Parameter 2	Not used
Name	Not used
Data	Not used

Logout status command

Indicates logout result.

Command number	5
Parameter 1	1 (OK)
Parameter 2	Not used
Name	IP address or Client PC name + Client ID (value set during connection)
Data	Not used

Server disconnection

The client will be disconnected:

- For the input, if the client tries to send a new process without having received the end of the process of the previous request.
- For the report plug-in only, if the client tries to send data.
- For both modes, if the logout command is sent.

Print capture

This plug-in captures spooled tasks on a printer. This printer is set up during your plug-in configuration. All printed tasks printed from your system to the printer will be analyzed by the sentinel once activated.

Printing your document from a Windows application on a sentinel printer may modify the original document. For instance empty lines can be removed and tabulation characters can be replaced by a carriage return line feed sequence. In order for the map to read the reformatted data correctly, do the following:

Step 1 Start by making the sentinel printer available for printing, as it is suspended for the capture by default.

Step 2 Select the FILE port for the output.

Step 3 Print your document on this printer from your application.
At print time, a dialog box is displayed that lets you specify the output file name.

Step 4 Use the file that is printed as the work file to define the Map file.
This method allows for the modifications that could occur during the printing process.

Step 5 When the map file is done, don't forget to set the printer back to suspended mode and to select the COM1: as the output port.

When printing through the TCP/IP protocol, the original document is not modified through the use of the driver. In that case it is possible to create the map file with the original document.

From the **Input** tab of the sentinel properties dialog box, select the input plug-in and click **Settings**.

The **Print Capture Plug-in** window appears:

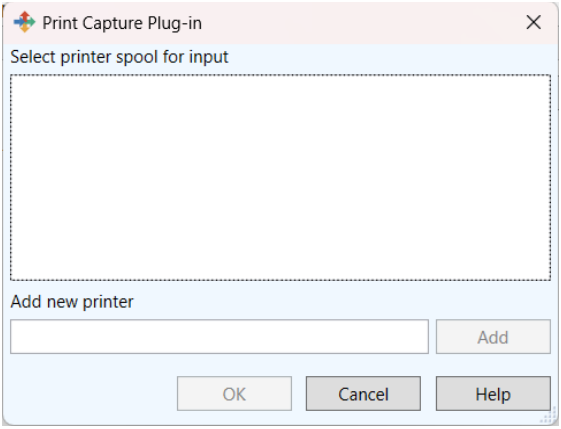


Figure 10: Print Capture Plug-in window

Step 6 Select printer spool for input.

Step 7 If you want to add another printer, enter its name and click **Add**. The new printer appears in the printer spool for input list.

Step 8 Click **OK** to validate.

Web Server

Note

Depending on the product version you have purchased, this plug-in may not be available.

The Web Server plug-in allows the sentinels server to receive information from a Web page, previously created and configured for data collection. It allows the sentinels server to be connected to any Web Application with a minimum of integration work on the Web Application itself. All it would take is for a **Submit** button on a Web page to be connected to the server's URL. This would send all the information entered in the Web page to be collected and, for example, inserted into a database or transmitted to a Web service available on the ERP system to perform a particular

transaction. Since this plug-in acts as an HTTP server, it will answer an HTTP POST message and will call the assigned process plug-in to act according to the information. The web form will need to be customized to link the data collected on the web page with the information requested by the process plug-ins used by the sentinel.

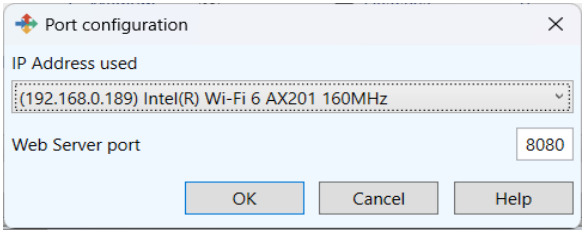


Figure 11: Port Configuration window

In order to customize a Web form, three things must be done:

Step 1 Create standard input fields, giving them the name of the variables you want to address from the fields.

Example:

<input type= "edit" name= "variable_name" value= "variable_value">

For more details, please see the HTML sample web form in the sample program directory.

Step 2 Create three input fields named Group, Sentinel and Job Name that are used to identify the sentinel which will receive the data and to provide a name for the job.

Example:
<INPUT type= "edit" name= "Group" value= "Production">
<INPUT type= "edit" name= "Sentinel" value="S01">

<INPUT type=" "edit" name "JobName" value="Job2545">
Data will be processed by the sentinel "S01" of the "Production group". The Job will be referred to as "Job2545".

Step 3 Customize the method and action parameters for the form so that it could be linked to the sentinels server.

Example:

<form method= "post" action= "http://sentinel_server:port_number/des">

...

...

...

</form>

Each input field name and value will be sent to the sentinel. Then, the sentinel will tell the caller (usually the Web browser) to call back to the web server's normal address. It allows the data to pass through the sentinel, first, and then again through the web application is it has been designed on a web server such as IIS or Apache.

Another way to exchange information between your application and the sentinels server through the web server plug-in is to directly use the web service that is published. From a development platform that supports web service calls, like .NET or J2EE, you can directly invoke the function exposed by the web service. To do that, you simply need to provide the URL address of the plug-in web service:

http://sentinel_server:portnumber/service?wsdl

Your development tool should create a wrapper containing two methods and an enumeration (function result):

Process1 (

Group as string : name of the sentinel group

Sentinel as string : name of the sentinel

JobName as string : job name to be displayed
Values as ArrayOfString : array of strings
) as FunctionResult

array of string example: array("Name", "McCarthy", "Surname", "Doris")

Process2(
Group as string : name of the sentinel group
Sentinel as string : name of the sentinel
JobName as string : job name to be displayed
Values as string : string representation of values
) as FunctionResult

Values: "Name=McCarthy\r\nSurname=Doris\r\n" where "\r\n" represents CrLf (Carriage return, Line feed).

FunctionResult enumeration:

OK : process has been done successfully
Canceled : process has been cancelled
Err_Process : process raised an error

Records capture

Principle

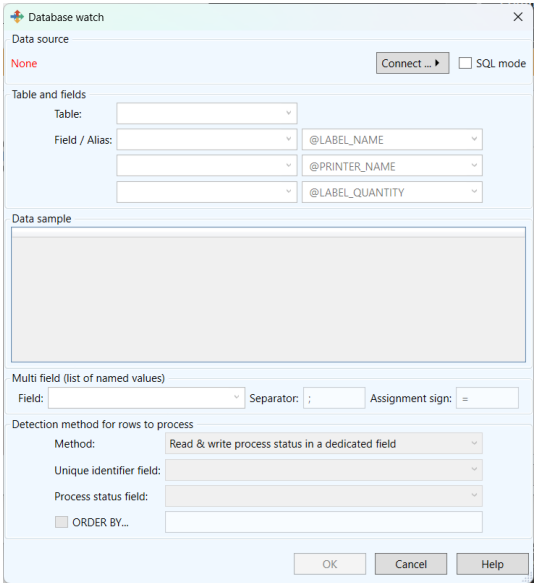
The plug-in watches the output of an SQL query and detects new records based on different user selectable strategies.

For each new record, it sends a process request to the sentinel.

Depending on the selected strategy, it writes status information in each processed record, deletes the processed records or just counts the number of records.

Sentinels using this plug-in do not need a map.

User interface



The dialog box for the plug-in is composed of a single Window with five sections:

- Data source
- Table and fields
- Data sample
- New rows detection method
- Multi field

The window is resizable to some extent.

Expansion mainly allows viewing more columns and rows from the Data Sample section.

Size is automatically saved and restored between sessions.

Data source

First step is to select a data source. Click the **Connect** button to open the standard Windows OLEDB database connection dialog.

Two types of connections are available: OLEDB and Oracle.

- OLEDB enables user to access data sources using OLE DB providers installed in the system (SQL Server, Oracle, Access etc).
- Oracle connection uses native library to retrieve data from the Oracle database server. It is recommended to use this option to work with Oracle servers newer than v11.

As soon as the data source is defined, the dialog box will update to list all the retrieved tables.

The option box labeled "SQL Mode" is also enabled. This mode is described in the SQL Mode section below.

Here are some common connection settings for known database types.

SQL Server, OLE DB provider mode:

1st tab: select Microsoft OLE DB Provider for SQL Server

2nd tab: item (1), specify the name of the machine where the SQL server resides

- item (2), enter user name and password (check the "record password" option box)
- item (3), select a Catalog from the drop-down list

Oracle, OLE DB provider mode:

1st tab: select Microsoft OLE DB Provider for Oracle

2nd tab:

- item (1), specify the published name of the Oracle server

- item (2), enter user name and password (check the "record password" option box)

Oracle, native mode:

Enter data source (usually supported format is {address}/{Oracle service name}), previously defined user name and password. Use Test button to validate the connection settings.

Access (MDB):

1st tab: select Microsoft Jet 4.0 OLE DB Provider (or latest version if available)

2nd tab: item (1), specify the data source as the path to the .MDB file

Excel (XLS):

1st tab: select Microsoft Jet 4.0 OLE DB Provider (or latest version if available)

2nd tab: item (1), specify the data source as the path to the .XLS file (*)

4th tab: double click 'Extended properties' and enter Excel 8.0

Dbase (DBF):

1st tab: select Microsoft Jet 4.0 OLE DB Provider (or latest version if available)

2nd tab: item (1), specify the data source as the path to the folder containing all the .dbf files (*)

4th tab: double click 'Extended properties' and enter DBase 5.0

Delimited text files (CSV):

1st tab: select Microsoft Jet 4.0 OLE DB Provider (or latest version if available)

2nd tab: item (1), specify the data source as the path to the folder containing all the .CSV files (*)

4th tab: double click 'Extended properties' and enter Text;Hdr=Yes;Fmt=Delimited (**)

(*) The "…" commands only allows selecting a file name whereas a folder is needed. To override this behavior,: in the "…" dialog select the file filter "Any file" and select any file from the intended folder and validate. Back from the dialog, manually edit the data source path to remove the file name, only leaving the folder name like for example c:\mybases\.

(**) For Unicode text files, enter Text;Hdr=Yes;Fmt=Delimited;CharacterSet=Unicode

Note

For other variations (no header, other delimiter char than semi-colon …etc) you should use a schema.ini file as explained here: http://msdn.microsoft.com/en-us/library/ms709353.aspx

Table and fields

Once the data source is defined, the table list shows all retrieved table names.

Select the table from which the plug-in should pick records.

The Data sample will display the first available rows from the table (up to 10).

You can resize the window if you want to display more rows or columns.

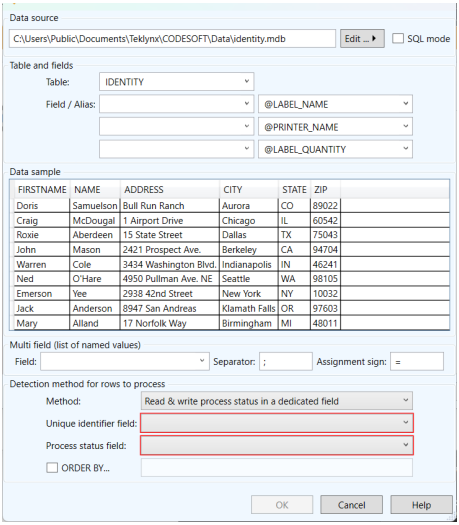
The table selection can be automated if the table contains a table or a view having a name starting with SENT_.

(For more information see the Automated Configuration section.)

Field / Alias

If the field names from the table do not match the variable names (or control variable names) from the process plug-in, you can create up to three relations there. Known control variables are listed. For variables not listed, just type their name here.

Example:



The column giving the name of the label to print spells 'Template' in the table: select field TEMPLATE and alias @LABEL_NAME.

The data from the column FIRST should go in the label variable FIRSTNAME: select field FIRST and type alias FIRSTNAME.

Note

If you need to create more than three aliases, you will have to switch to SQL mode and type the query containing the needed AS clauses.

For instance:
SELECT Template AS @LABEL_NAME, FIRST AS FIRSTNAME …etc… from JOBS

(For more information see the SQL Mode section.)

Detection of records to process

The plug-in offers a choice of three detection strategies:

- Read and write process status in a dedicated field, or
- Process and delete each row, or
- Count rows.

If migrating from LABELVIEW data watch, the last method is convenient for DBF files.

The first method should be used if possible on all other database types.

Status

For this strategy, the plug-in uses a dedicated field from the table to act as a status field. The plug-in processes each row without status. Then it writes back the process result as status value.

- The field type must be numerical.

Note

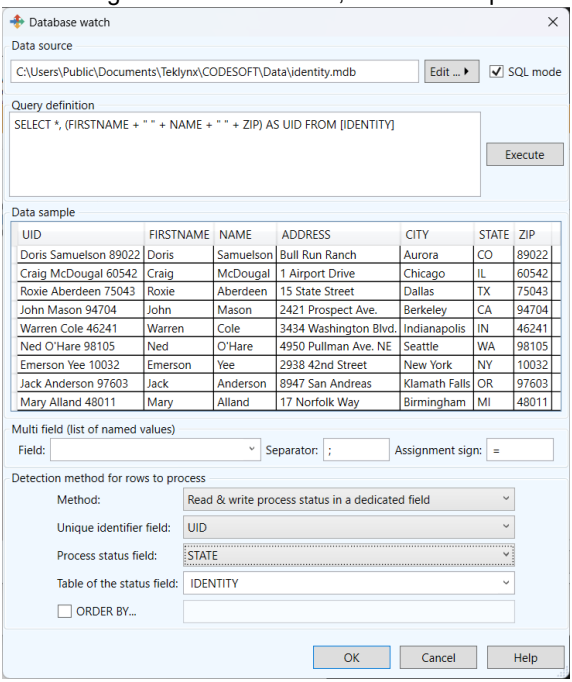
In Excel sheets, default data format is string, not numerical. To define the "Status" column as numerical, change the cell format to plain number (without decimal).

- The field must be set by your system to NULL or zero in records to process.
- At the beginning of the processing, the plug-in writes the value 1 in the field; after processing, it writes either 2 for success or 3 for failure.
- Any value (except NULL, 0, 1) can be set in a new record to postpone its processing. Value can be set to NULL or zero at a later time to release the record for normal processing.

This strategy requires also a unique identifier field to distinguish records and to allow selecting/updating records one by one.

Note

In the snapshot below, the identifier UID does not belong to the table. It is made up on the fly in the SQL query by combining the three fields First, Name and Zip.



To help you decide if this strategy is well suited for your system, here is a summary of pros and cons:

Pro:

- You can check the table to know which records succeeded or failed,
- You can reset the status to have some records processed again
- You can delete records that are not needed anymore

Con:

- The table structure may need modification to add the column acting as Status (*),
- The sentinel will be writing in the table after each process, so the database must support multiple access (your system will be adding rows while the sentinel writes the status in processed rows)

(*) Tip: By using SQL mode, you can store the status in an independent table joining the main table through a unique key. This avoids modifying the main data table.

Process and delete all rows

Each record is processed then deleted.

This strategy requires a unique identifier field (a field which value cannot appear twice through the table) to distinguish rows and allow deleting rows one by one.

When selecting this strategy, select also the field to use as unique identifier. It can be:

- A built-in counter managed by the database system (also called AUTO INCREMENT),
- A unique data managed by your data model, like a serial number, a driver license ID, an email address…etc

Pro:

- Any table can be monitored without change to its structure.

Con:

- Processed records are deleted, so the table is somewhat dedicated to the processing done by the sentinels server
- The sentinel will be updating the table after each process, so the database must support multiple access (your system will be adding rows while the sentinel reads and deletes other rows)

Count rows

The plug-in counts the number of records returned by the query. When the number has increased since the last counting, the plug-in knows how many new records were added to the table.

This method is quite straightforward but has some major drawback: new records are easily located in sequential databases (DBF, CSV and XLS) because they are guaranteed to appear at the last positions.

This natural order is not guaranteed for transactional databases. Records may be re-ordered as the result of optimization, indexing, backup restoration …etc. For transactional databases, you should favor the "Status" strategy.

Pro:

- Any table can be monitored without change to its structure.
- No writing is made by the plug-in (no shared access problem)

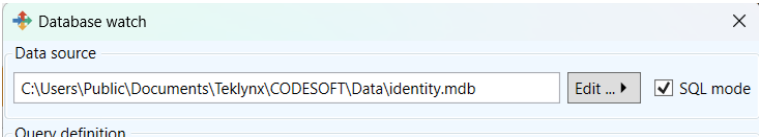
Con:

- The plug-in would miss some new records if an external process (or user) deletes some older records. So record deletion is forbidden during sentinels' activity.
- New records can only be detected if they are added while the sentinel is running. (So before adding new records, ensure the sentinel is active.)
- If the sentinel reports some process error, there is no easy way to know which records were involved.

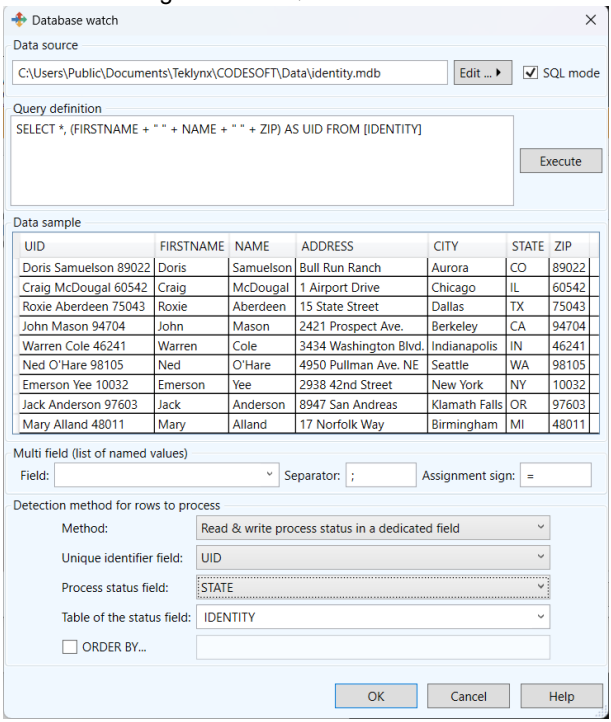
Query definition in SQL Mode

The SQL mode is required to create advanced queries (select on multiple table, conditions…etc.).

When you check the **SQL Mode** option the second section becomes "Query definition" and a button labeled **Execute** is displayed:



Input (or paste) your query and select the **Execute** button located to the right of the SQL mode switch:



If the query syntax was wrong, the error message coming from the database engine would display in the Data Sample area.

New records detection strategy and SQL mode

The field list appearing for the detection method "Delete" and "Status" are editable. If the sentinel reports syntax errors for the query, check that the selected field names are unambiguous and edit them accordingly.

In the example above, the field COUNT doubles some SQL keyword so if it were used, it should be enclosed in square braces as in [COUNT].

Automated configuration

If certain conditions are met, the plug-in configuration can be set automatically.

Conditions are:

- The database contains a table with a name starting with SENT_ .
- The table contains two fields named SENT_STATE and SENT_ID to be used as Status field and unique ID field.

For example, under SQL Server, use SQL Management Studio to create a view named SENT_PRINT.

Arrange the view to gather all the required data. Have two columns named SENT_STATE and SENT_ID.

Now in the plug-in, select your database as the data source.

The plug-in will find the table and propose to use it. If you agree, it will then select the two columns for the "Status" detection strategy.

Multi field

(Feature not appearing on the previous user interfaces)

In some situation it is convenient to bundle several pieces of data in a single field, especially when the label template

could change at each record and its variables are not known at this time.

For this type of column containing a series of named values, the section Multi field allows to define its format:

- Field : the field name.
- Separator : the character (or string) delimiting two adjacent pieces of data.
- Assignment sign : the character (or string) delimiting the name from the value.

Example:

Having a column named DynList that contains a piece of data like "Hei.:22/Wid.:40/Dep.:60/Col.:Stone" the configuration should be:
- Field DynList
- Separator /
- Assignment sign .:

For this record, the selected process plug-in will have at its disposal the four variables Hei, Wid, Dep, Col .

ORDER BY

This feature allows processing records by order. When the "ORDER BY" checkbox is set, SENTINEL will extend its "Update" query with ORDER BY, a space, and the text as exactly typed in the ORDER BY textbox.

The user interface will display a checkbox and a textbox in the Detection Method group, only when the method is "Status".

SFTP file capture

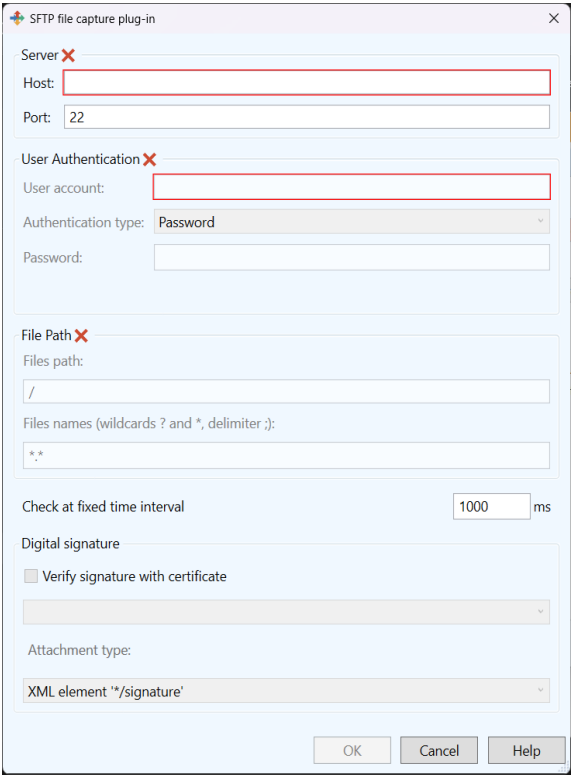
The SFTP File Capture plug-in collects files from a specific remote folder using the SFTP protocol (SSH File Transfer Protocol). The SFTP File Capture plug-in works with **SFTP version 3 protocol**, which is supported by all modern SFTP servers.

Files are processed in write-time order – the first file modified in the folder is the first file processed. Unlike the file capture plug-in, which sorts files by creation date, the SFTP file capture plug-in sorts by last modification time according to the SFTP protocol version 3. This protocol provides the last access time and last modification time of the file/ directory.

To be analyzed by the sentinel, the file must be accessible in read/write mode. Also, a user via plug-in connects to the server should have the write/read/delete permissions. In addition, the sentinel and SFTP server must have full control over the data file before it will proceed with processing. The program that is creating the data file for sentinel must have finished writing the file.

From the **Input** tab of the Sentinel Properties dialog box, select the SFTP file capture plug-in and click **Settings**.

The following dialog box appears:



Step 1 Type the hostname or IP address of the SFTP server to be connected.

Step 2 Type the port number of the SFTP server to be connected. The default SFTP port is 22.

Step 3 Enter the user account name which you use to connect to the SFTP server.

Step 4 Select the authorization type:

- **User account name and password:** A user name and password is a good method of authentication. It allows a server to authenticate a user by challenging them to submit a piece of information that (theoretically) only the user would know. That information is the user's account user name/

password combination.
If you selected the user account name and password authorization type, you should enter the password in the **password field.**

- **User account name and private key:** This option requires that you have a private/public SSH key pair, and that the public key is in place on your SFTP server.
  If you selected the user account name and private key authorization type, you should type the path of the private key file or select the file in the **File Browse** dialog. If your private key was created with a passphrase you should type this keyword in the passphrase field.

Step 5 Type the file path of the folder to be watched.

Step 6 Type the file name or file extension of the file to be captured.
You can type several file names separated by a semicolon or use wildcard characters. Asterisk (*) is used to replace zero or several characters. Question mark (?) is used to replace one and only one character in a file name.
Examples:

a*.txt	All files with extension txt starting with a (or A), like AF104.txt or a.txt.
item_n?.*	All files named item_n + one character, whatever the extension, like item_n3.dat or ITEM_NZ.txt, but not item_n24.doc.
.txt;.dat	All files with extension txt and dat.

Step 7 In the **Check at fixed time intervals** field enter the time in milliseconds in order to check the folder regularly according to the specified time interval. The minimum value is 1000 milliseconds.

REST Web Server API

Note

Depending on the product version you have purchased, this plug-in may not be available.

The plug-in allows users to send jobs for processing using the REST API server. Users can get information about job status, check processing result or update jobs that are still waiting to be processed.
This plugin supports localhost.

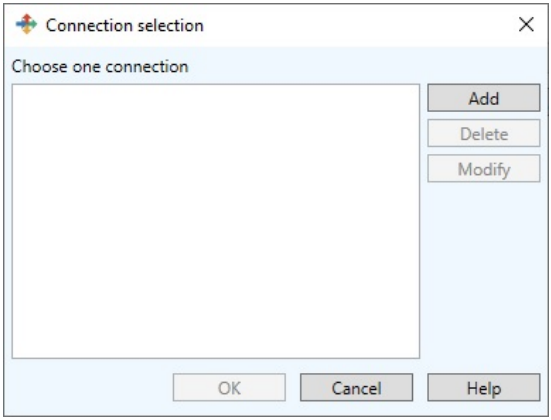


Figure 13: Connection selection window

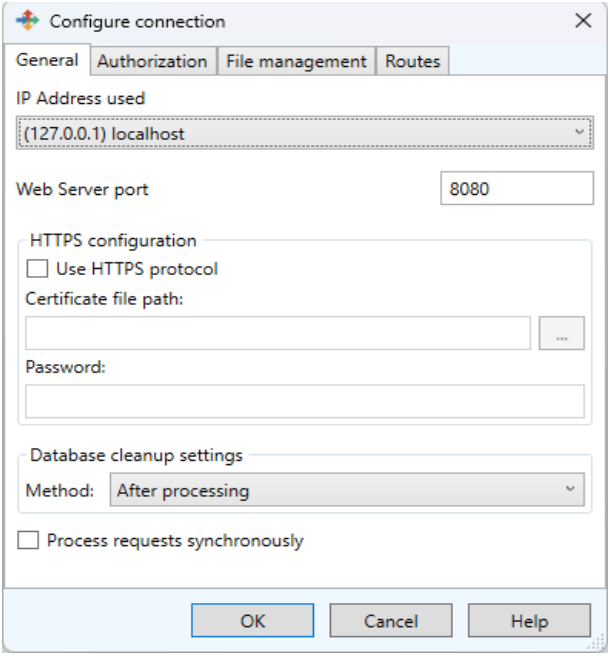
Configuration

To use the REST Web Server API, you must first create an appropriate connection.

Step 1: Open the REST Web Server API plugin settings dialog.

Step 2: Click **Add** button.

The **Configuration connection** window opens.



Step 3: Specify connection to Web server using **Connect network using** and **Port.**

Step 4: Check **Use HTTPS protocol** option if you want to use HTTPS protocol for a Web Server and specify the **Certificate file path** and **Password**.

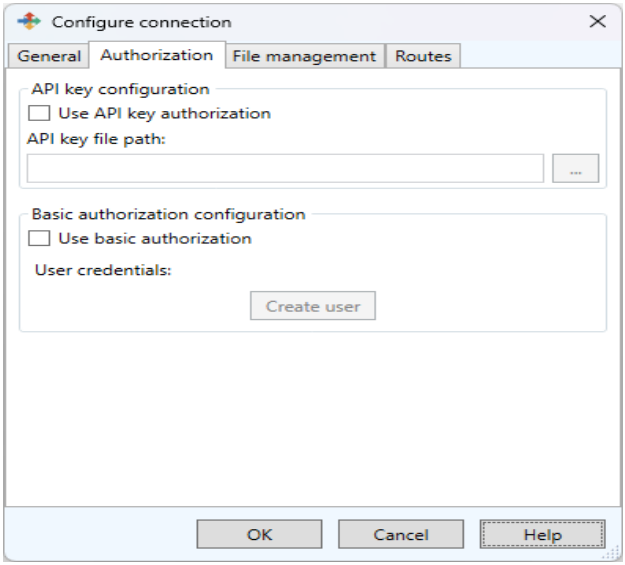
Step 5: Select method for cleaning up a database:

Regular clean up (selected by default) - clears completed job by specified period.
Do not clean - never cleans up the database.
After processing - clears job right after the processing cycle.
Note: After processing is preferred method if you do not need to get job status afterwards.

Specify **Clear completed jobs every** - period of database clearing (sets in hours) if **Regular clean up** method is selected.

Process requests synchronously: By default the server receives jobs in async mode. It means the server will not wait until job is processed before returning response. This option allows to change server behavior to wait until processing completes.

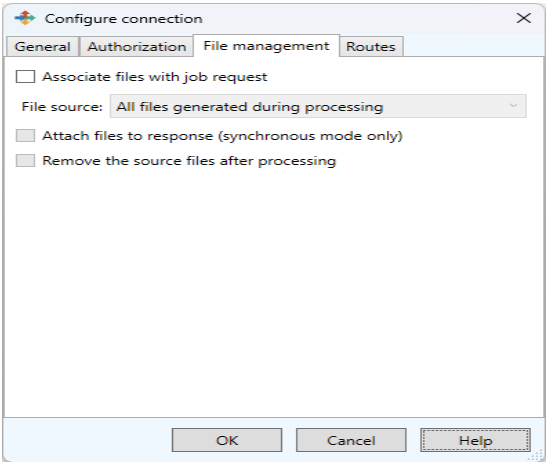
Authorization options:



Check **Use API key authorization** option if you want to identify the person making the API call and specify the API key path in the API key file path field.

Check **Use basic authorization** option if you want to use 'Authorization' header to authorize the person making the API call. Then you can define the login and password pairs. They can then be used in the header in the format "Basic <credentials>", where credentials is the Base64 encoding of the login and password separated by the single colon (:).

File management options:



- **Associate files with job request**: The plugin has the ability to store files associated with jobs and return them by corresponding endpoints. See more details below.
- **File source**: the plugin has different strategies for files searching such as:
  - All files generated during process: uses internal functionality to memorize what files were created during processing to store them.
  - A single file: a fixed absolute path to the file that should be stored.
  - Files from folder: a fixed absolute path to the folder from where files should be taken.
  - A variable value as file path: searches a variable and uses its value as an absolute path to a file.
  - A variable value as folder path: searches a variable and uses its value as an absolute path to a folder.
- To provide additional information for some File source strategies use the corresponding field: *File path* or *Folder path* or *Variable name*.
- *Block searching options* should be defined if one of variable assignment strategies was activated. The job can contain many blocks and each of them may

contain the variable, so this option allows plugin to find the needed block. The following values are supported:

- *Check every block*: the plugins gathers every compatible variable from a job.
- *First matching block from the start*: the plugin tries to find the variable from the beginning of a job.
- *First matching block from the end*: the plugin tries to find the variable from the end of a job.
- *Block number from start*: the number of block in a job from beginning (numeration starts from 1).
- *Block number from end*: the number of block from the end of the job (numeration starts from 1).

Note

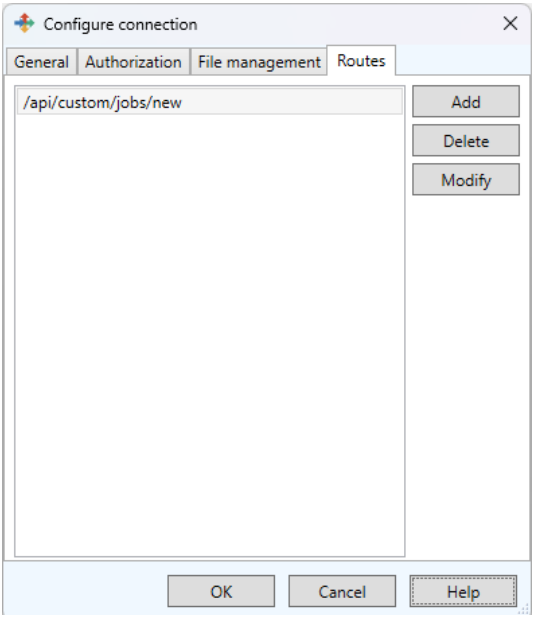
Last two options require additional information that should be defined in the corresponding field.

- **Attach files to response** (synchronous mode only): If the Process requests synchronously option was activated, this option can be activated. It allows to return generated files as a result of a job request.
- **Remove the source files after processing**: if activated, this option automatically removes assigned files from the local drive after processing (note: they still can be retrieved from the internal storage via files endpoints).

File management notes:

- The plugin will send warning message if files were not found during processing.
- The plugin has a limit of 100Mb for all files per job, files that are out of the limit will not be stored.
- Files are stored in the internal storage, they can be requested even if the original ones were removed.
- Pay attention that **files are removed with the job**, so if job clearing option *After processing* is activated, they will not be accessible through specialized endpoints.

- **Custom Endpoints**



Traditionally, the plugin operated exclusively with a set of predefined endpoints, which imposed certain limitations on its flexibility and extensibility. To address these limitations, support for user-defined custom endpoints has been introduced.

This functionality allows customers to create their own endpoints by specifying a relative route and providing a description. These endpoints are automatically documented and displayed on the Swagger page, where they appear in a dedicated section separate from the built-in API endpoints.

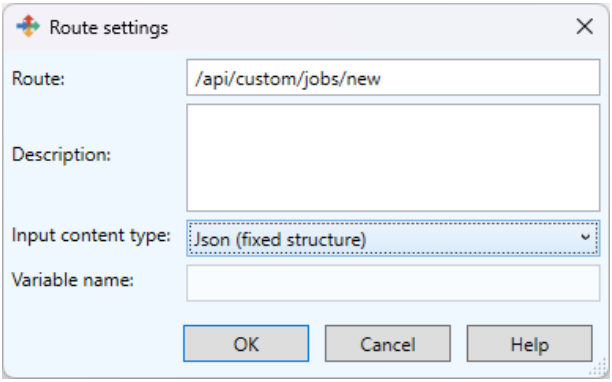
Limitations

Custom endpoints are subject to several constraints. They are limited to the HTTP POST method and operate asynchronously, meaning that requests are accepted without waiting for the processing to complete, as a result, these endpoints cannot return processed files.

Note

Customers cannot define routes that conflict with existing API paths.

Each request sent to a custom endpoint results in the creation of a job that is added to a unified processing queue. This approach enables flexible job submission strategies, allowing different methods of enqueuing tasks for the same processing units.



Custom endpoints also support configurable input data schemas. Available options include:

- Fixed JSON schema
- Fixed XML schema
- Assignment schema
- Text mapped through a mapper
- Text assigned to a variable
- Binary stream mapped through a mapper
- Binary stream assigned to a variable as Base64.

Responses from custom endpoints follow the same JSON format as standard endpoints. For schema options that rely on a mapper, a corresponding mapping configuration must be defined in the sentinel unit settings. If the required map is not configured, job processing will fail at runtime.

Note

The Sentinel Manager does not validate the mapping configuration for this plugin. This behavior reflects a current architectural limitation, leaving the responsibility for providing a valid mapping definition to the customer.

Schemas

Fixed JSON schema — this is the primary body format available for an endpoint and fully corresponds to the structure used by predefined POST endpoints. Customers can submit either a complete multi-block job or a single block.

The example of multi-block job input:

```json
[
  {
    "@LABEL_NAME": "Mailing.lab",
    "@PRINTER_NAME": "Bullzip PDF Printer,BULLZIP",
    "@JOB_NAME": "TEST JOB",
    "Address": "307 7th Ave",
    "City": "New York",
    "Code": "1346",
    "FirstName": "John",
    "Name": "Smith",
    "State": "NY",
    "Zip": "10001"
  },
  {
    "@LABEL_NAME": "Mailing.lab",
    "@PRINTER_NAME": "Bullzip PDF Printer,BULLZIP",
    "@JOB_NAME": "TEST JOB",
    "Address": "405 7th Ave",
    "City": "New York",
    "Code": "1578",
    "FirstName": "Adam",
    "Name": "Hancock",
    "State": "NY",
    "Zip": "10098"
  }
]
```

The example of single-block input:

```json
{
  "@LABEL_NAME": "Mailing.lab",
   "@PRINTER_NAME": "Bullzip PDF Printer,BULLZIP",
  "@JOB_NAME": "TEST JOB",
  "Address": "307 7th Ave",
  "City": "New York",
  "Code": "1346",
  "FirstName": "John",
  "Name": "Smith",
  "State": "NY",
  "Zip": "10001"
  }
  ```

Fixed XML schema — this option is similar to the fixed JSON schema but uses a more verbose XML format. As with the JSON variant, customers can provide either a complete multi-block job or a single block.

The example of multi-block job input:

```xml
<Job>
 <Block>
   <Variable Name="@LABEL_NAME">Mailing.lab</Variable>
   <Variable Name="@PRINTER_NAME">Bullzip PDF
Printer,BULLZIP</Variable>
<Variable Name="@JOB_NAME">TEST JOB</Variable>
   <Variable Name="Address">307 7th Ave</Variable>
   <Variable Name="City">New York</Variable>
   <Variable Name="Code">1346</Variable>
   <Variable Name="FirstName">John</Variable>
   <Variable Name="Name">Smith</Variable>
   <Variable Name="State">NY</Variable>
   <Variable Name="Zip">10001</Variable>
 </Block>
 <Block>
   <Variable Name="@LABEL_NAME">Mailing.lab</Variable>
   <Variable Name="@PRINTER_NAME">Bullzip PDF
Printer,BULLZIP</Variable>
   <Variable Name="@JOB_NAME">TEST JOB</Variable>
   <Variable Name="Address">405 7th Ave</Variable>
   <Variable Name="City">New York</Variable>
   <Variable Name="Code">1578</Variable>
```

```
    <Variable Name="FirstName">Adam</Variable>
    <Variable Name="Name">Hancock</Variable>
    <Variable Name="State">NY</Variable>
    <Variable Name="Zip">10098</Variable>
  </Block>
</Job>
```

The example of single-block input:

```xml
<Block>
  <Variable Name="@LABEL_NAME">Mailing.lab</Variable>
  <Variable Name="@PRINTER_NAME">Bullzip PDF
Printer,BULLZIP</Variable>
  <Variable Name="@JOB_NAME">TEST JOB</Variable>
  <Variable Name="Address">307 7th Ave</Variable>
  <Variable Name="City">New York</Variable>
  <Variable Name="Code">1346</Variable>
  <Variable Name="FirstName">John</Variable>
  <Variable Name="Name">Smith</Variable>
  <Variable Name="State">NY</Variable>
  <Variable Name="Zip">10001</Variable>
</Block>
```

Assignment schema — this is a fixed text-based format, similar to the one used in the TCP/IP plugin for data transmission. In this schema, variables are represented as key-value pairs separated by an equals sign `=`. Each variable starts on a new line, and blocks are separated by an empty line.

```shell
@LABEL_NAME=Mailing.lab
@PRINTER_NAME=Bullzip PDF Printer,BULLZIP
@JOB_NAME=TEST JOB
Address=307 7th Ave
City=New York
Code=1346
FirstName=John
Name=Smith
State=NY
Zip=10001

@LABEL_NAME=Mailing.lab
@PRINTER_NAME=Bullzip PDF Printer,BULLZIP
@JOB_NAME=TEST JOB
```

```
Address=405 7th Ave
City=New York
Code=1578
FirstName=Adam
Name=Hancock
State=NY
Zip=10098
```

Text mapped through a mapper — this format accepts plain text input and processes it through the mapper. <u>It requires a properly configured map file in the unit settings.

Note

The Swagger page sends line separators without carriage return characters `\r`, if the mapping logic depends on carriage returns `\r`, this may cause issues when testing via Swagger.

Text assigned to a variable — this is a minimalistic schema where the entire input text is assigned as the value of a single variable. The variable name can be configured in the custom endpoint settings.

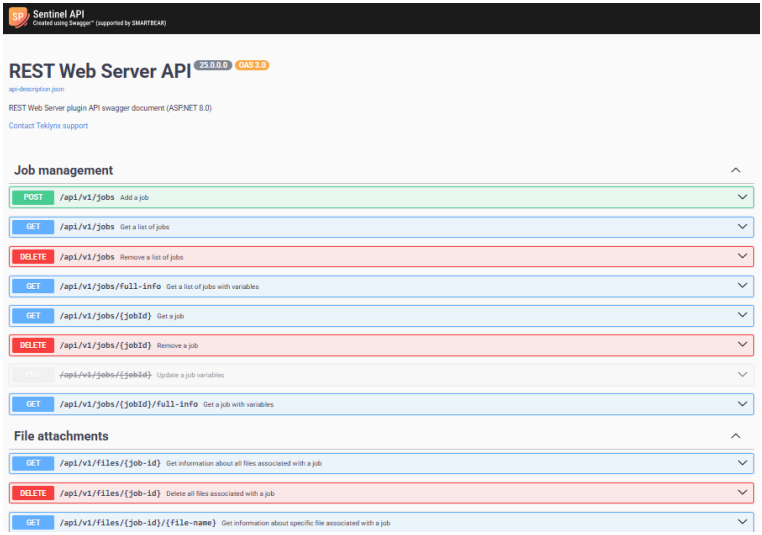
Binary stream mapped through a mapper — this format is similar to the text-based mapping approach but uses a binary data stream `application/octet-stream`. It is intended for scenarios where data MUST be passed to the mapper without any modification. <u>A map file must be configured in the unit settings.

Binary stream assigned to a variable as Base64 — in this format, data sent via `application/octet-stream` is converted into a Base64-encoded string and assigned to a variable. The variable name can be defined in the custom endpoint settings.

Testing

For demonstration purposes or to send a test job, users can use the Swagger application that is included in each server.Each endpoint in swagger contains sample data as well as details about responses that could be received.

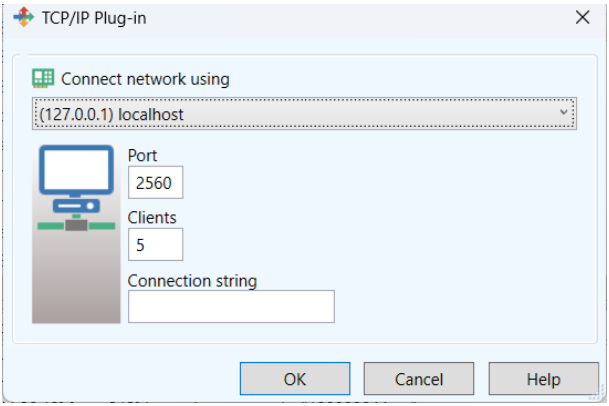
To reach the application follow the link: *http://IP_ADDRESS: PORT/Swagger.*



Modbus Server

Note

Depending on the product version you have purchased, this plug-in may not be available.



The **Modbus** server plugin allows you to exchange data between PLCs and computers. Modbus is a common means of gathering data from many different sources for viewing operations, archiving and troubleshooting from a central remote location. It is widely used and a fairly simple protocol. This plugin supports localhost.

Step 1: Specify connection to Modbus server using **Connect network using** and **Port.**

Step 2: Using **Maximum length** field specify the maximum length of the message.

Step 3: Set an appropriate error code for **Modbus error code** ("4" by default):

Code	Text	Details
1	Illegal Function	Function code received in the query is not recognized or allowed by client
2	Illegal Data Address	Data address of some or all the required entities are not allowed or do not exist in client
3	Illegal Data Value	Value is not accepted by client
4	Client Device Failure	Unrecoverable error occurred while client was attempting to perform requested action
5	Acknowledge	Client has accepted request and is processing it, but a long duration of time is required. This response is returned to prevent a timeout error from occurring in the server. Server can next issue a Poll Program Complete message to determine whether processing is completed
6	Client Device Busy	Client is engaged in processing a long-duration command. Server should retry later
7	Negative Acknowledge	Client cannot perform the programming functions. Server should request diagnostic or error information from client
8	Memory Parity Error	Client detected a parity error in memory. Server can retry the request, but service may be required on the client device
10	Gateway Path Unavailable	Specialized for Modbus gateways. Indicates a misconfigured gateway

Code	Text	Details
11	Gateway Target Device Failed to Respond	Specialized for Modbus gateways. Sent when client fails to respond

Step 4: Check **Synchronized** checkbox if you want to send the messages about Error Code.

Process plug-ins

Four process plug-ins are available:

- Label Printing
- Email
- Database
- Web Service Client

Process plug-ins are used to determine the way in which your information will be processed. Process plug-ins manage variable collections to perform the work they have been designed for. There are two types of variable:

Standard:

- The Label Printing plug-in manages variables defined in a label created with the label designer.
- The Database plug-in manages variables associated with a task that is defined with the Query Manager.
- The Web Service Client plug-in manages variables associated with the parameters of a web service function.

Control:

These are the variables that can influence the processing work of the plug-ins.

- The Label Printing plug-in uses the @LABEL_NAME variable to define the label to print, the @PRINTER_NAME to select the target printer or the @JOB_NAME to specify a name to the data stream received by the input plug-in.

Here is a list of control variables. Some may not be available with your installation.

Control Variable	Plug-in Type	Description
@ LABEL_NAME	Label Printing	Define the label to print. The name of the label can be the label's name on disk or the complete path and name of the label. ex: *@:\Desktop\ my_archivingsystem\ my_labeldesigner\ Mylabel.lab*
@PRINTER_NAME	Label Printing	Select the target printer.
@OFFSETX and @OFFSETY	Label Printing	Modify the location of the printing on the paper. The values allowed are between -32768 to 32767 in hundredths of a millimeter. These control variables may not be available with all installations.
@DEFAULT_PRINTER	Label Printing	Name of the printer to use by default when the @PRINTER_NAME control variable is not defined.
@DEFAULT_LABEL	Label Printing	Name of the printer to use by default when the @LABEL_NAME control variable is not defined.
@LABEL_QUANTITY	Label Printing	Name defined in this control variable will set the number of labels to be printed.
@DEFAULT_QUANTITY	Label Printing	Name of labels to print when the @LABEL_QUANTITY variable is not defined.
@LABEL_COPY	Label Printing	Defines the number of labels to copy – may not be available with your installation.

Control Variable	Plug-in Type	Description
@PAGE_COPY	Label Printing	Defines the number of pages to copy – may not be available with your installation.
@INTERCUT	Label Printing	Defines the number of labels to print between each page cut – may not be available with your installation.
@TASK	Database	Select the task to execute.
@WEBMETHOD	Web Client Service	Select the web service method to execute.
@JOB_NAME	Global	Gives a name to the transaction or group of transactions executed by the process plug-in. If the variable is not explicitly assigned, the file name will be used as the job name. This information is used for the archiving system's print history log to help you to quickly find and retrieve a print job.
@START_LABEL	Label Printing	Allows to specify the index number of the label to start printing from.
@PRINTER_TRAY	Label Printing	Defines the tray (paper source) used for printing. The value must be real tray name (e. g. "Cassette 1", "Manual Feed", "Auto"). The @PRINTER_TRAY control variable is ignored if the printer is set in @PRINTER_NAME by alias where tray is specified.
@PAGE_ORIENTATION	Label Printing	Defines portrait or landscape page orientation (0 - Landscape, 1 - Portrait).

Control Variable	Plug-in Type	Description
@GET_SENTINEL	Global	Internal control variable that would be set by SENTINEL itself, contains sentinel name.
@GET_GROUP	Global	Internal control variable that would be set by SENTINEL itself, contains sentinel group.
@GET_MAP	Global	Internal control variable that would be set by SENTINEL itself, contains filename without path and without extension.
@GET_INPUT_PATH	Global	Internal control variable that would be set by SENTINEL itself, contains full path to the File Capture plug-in input folder.
@GET_INPUT_SPOOL	Global	Internal control variable that would be set by SENTINEL itself, contains printer spool name selected in Print Capture plug-in.
@LABEL_ROTATION	Label Printing	Allows to set the rotation for the label. Available values: 0, 90, 180 and 270 degrees.
@DELETE_SPOOL_JOBS	Global	Deletes all the jobs in the spool of the printer on which the label must be printed.
@GENERATED_FILES	Global	Reserved variable name for automatic handling of the generated files that will be attached to emails or made available via corresponding REST API endpoints
@PDF_MODE	Label Printing	Controls PDF document generation mode. 0 - simple mode (default) 1 - advanced mode Overrides PdfMode value set in TkxCS.xml

Control Variable	Plug-in Type	Description
@PDF_QUALITY	Label Printing	Controls the quality of the images used to generate PDF in simple mode only. Recommended values - from 100 to 500 (default value is 300). Overrides PdfQuality value set in TkxCS.xml
@PRINT_PDF	Label Printing	Defines the filename of the PDF to be generated. Note: if a file with the same name already exists, it will be overwritten. If the existing file is locked by another program, PDF generation may fail.
@PDF_BORDER	Label Printing	Controls if the label borders are included into the PDF. 0 - borders are not included, 1 - borders are included (default)

Label Printing

This plug-in prints barcode label with data found by the mapper.

From the Process tab of the sentinel properties dialog box, select the process plug-in and click on **Settings**.

The Printer Selection window appears:

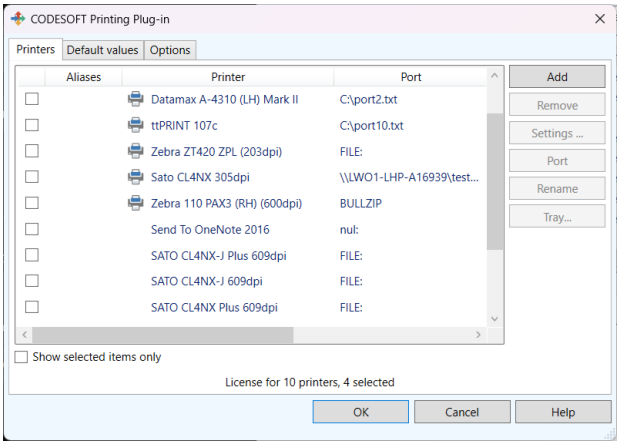


Figure 13: Printer Selection window

Step 1 From the **Printers** tab, select the printer(s) you want to use.
You can define aliases for each printer. For example if you define "Production1" as an alias for the printer "Plasmatronic TH640, \\ServerProduction1", the sentinel will select this printer if the data analysis found the data Production1 as the printer name to use for printing.
If the printer is exchanged for a different model, you will just have to assign the alias to the new printer in the plug-in configuration. No modification will need to be done on the data.

The alias is used prior to the real name of the printer. For example, if you want to redirect the printing demands from the "Plasmatronic TH640" to the "Matrix Code IV", you simply need to assign the alias "Plasmatronic TH640" to the "Matrix Code IV" printer.

Also it is possible to select the tray (paper source) for the selected printer. By clicking on "Tray..." button user is able to see the list of trays and aliases. The "Tray..." button is enabled if the selected printer has trays.

Step 2 On the Default values tab, type or select a file to be printed.

Remember that first priority is given to the control variables found in the file, then to the default value such as this one.

Note

By default, in step 2, if a label is selected from the label archive system, the selected value does not include the version number. You can add this information manually in the edit box as mentioned above.

Step 3 Select a default printer on which documents will be printed if no printer is specified in your data file.

Step 4 Fill in, if needed, the number of labels to be printed, the number of copies, etc.

Step 5 Activate the **When opening a document, clear its variables** option if you want to erase the variable values saved in the document when it is opened. If not, the values saved in the document will remain until new values are provided by the file analysis.

Step 6 In the **Options** tab, configure the available settings as per your needs. The list of available settings will change according to the label designer installed.

Step 7 If you have installed an archiving system, you will have an additional tab called Archiving. Select the Archiving tab in order to enter the user name and password information necessary to access the archiving system, to select the level of history tracking information

you would like to keep and the default category from which your labels will be extracted.
The history tracking options are as follows:

Track Printing: Allows reprinting
Track Errors: Adds one record in case of an error
Track Warnings: Adds one record in case of a warning
Track other events: Adds one record for other activities such as file opening and closing

Step 8 Click **OK** to validate.

The print management system will print the latest validated label version if you don't explicitly specify the label name and version to print. If no validated version exists for the label, then the print management system will print the latest draft of the label (requires administrator or manager rights).

To specify a given archive version, the filename must be suffixed with an underscore and the version number (for example, identity.lab_3).

- **Printing to PDF via SENTINEL**

By using SENTINEL users gain access to native printing to PDF functionality.

Note

Note that printing to PDF is subject to Electronic output licensing and requires eOutput addon.

It can be used in two different ways:

- So called "simple" mode utilizes image export functionality to create PDF. It enables PdfQuality to control the default quality of the output in TkxCS.xml config and @PDF_QUALITY control variable. All resulting PDF objects will be images.
- "Advanced" mode which is used by default. It produces high quality WYSIWYG PDF. The key feature is that resulting PDF objects might not be images only. For

instance, if there's a text object on the label, it will be a text object in the resulting PDF that can be selected/ searched. This is contrary to "simple" mode where all of the objects will be displayed as images. Note: you cannot control the quality of the resulting PDF.

Comparison of advanced (to the left) and simple (to the right) PDF printing modes output

Related control variables:

- @PDF_MODE: 0 - enables simple mode; 1 - enables advanced mod. Overrides PdfMode value set in TkxCS.xml.
- @PDF_QUALITY: controls the quality of the images used to generate PDF in simple mode only. Recommended values - from 100 to 500. Overrides PdfQuality value set in TkxCS.xml.
- @PRINT_PDF: filename with the path to generate PDF. Note: if a file with the same name already exists, it will be overwritten.
- @PDF_BORDER: controls if the label border is included into the result PDF. 0 - border is not included, 1 - border is included (default).

Note

If required, the PdfMode and the PdfQuality can be defined manually in the configuration file of the sentinel.

• Electronic Outputs Licensing Description

General information

Electronic Outputs licensing limits the number of outputs (printouts, exports, etc.) in reusable electronic formats such as PDF, XPS, image formats like JPG, BMP, and more. It is a consumable license, which is renewed automatically by time (7-day cycle for electronic output use).

For example, if a user has a license for eOutput with two (2) API Printers, the total available Electronic Output units would be 10000 per week (5000 per printer).

The following operations are subject of Electronic Outputs licensing:

- Printing via Windows drivers producing common document formats (PDF, XPS, image formats, etc.);
- Printing with any printer to ports capturing the printout electronically (FILE:, prompt for file, or fixed file ports);
- Exporting the document to PDF or image formats directly.

Note

In order to be able to perform native printing to PDF or programmatically export printer commands to a file, the eOutput addon is required.

The license defines a quota for the Electronic Outputs license restricted activities. The unit of the licensing is the number of labels generated by the operation. The quota applies for an interval of any consecutive 7-days.

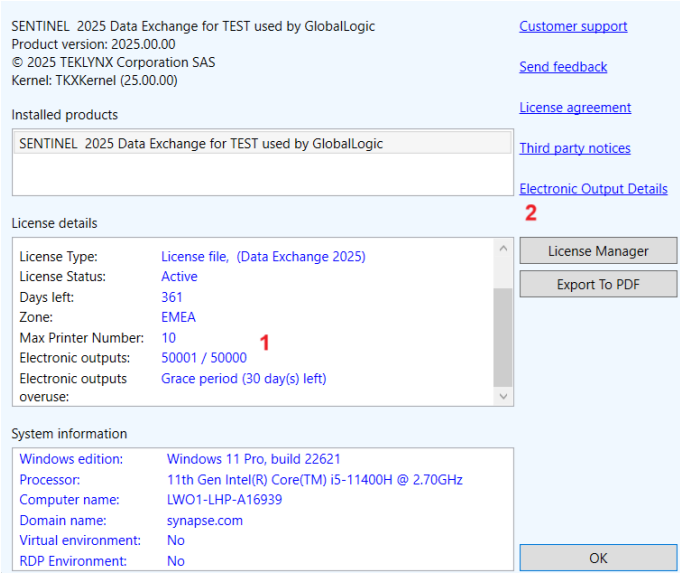
The quota is proportional to the number of printers allowed by the license - 5000 Electronic Outputs license units are provided per printer in the license.

For example, a license with Electronic Output add-on and two (2) API Printers, the total available Electronic Output units would be 10,000 per 7-day period (5,000 per printer).

A grace period is implemented for Electronic Outputs licensing, allowing a license to exceed the quota for a maximum of 30-days.

User Interface elements

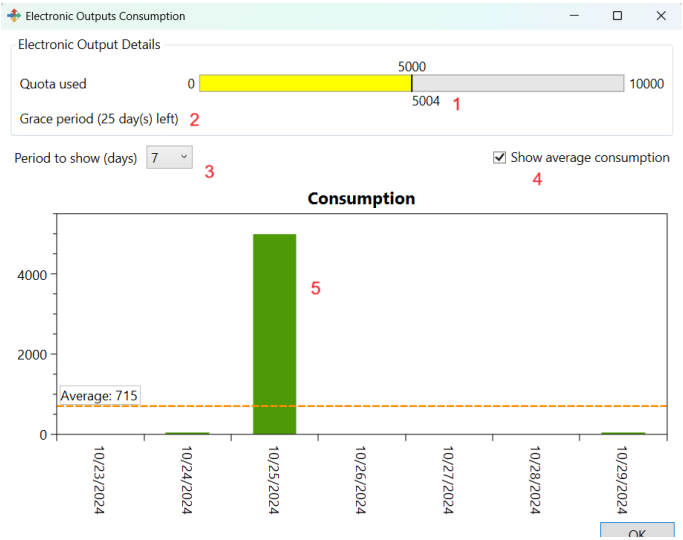
If Electronic Outputs licensing is active, the about dialog contains information about the licensing status:



1 The **About** dialog shows the consumed ‘Electronic outputs:’ quantity and the maximum available. The grace period status is also indicated by the ‘Electronic outputs overuse’.

2 The **Electronic Output Details** button opens the ‘Electronic Outputs Consumption’ dialog.

The **Electronic Outputs Consumption** dialog shows the following information:



1 The consumption bar shows the actual consumption relative to the licensed limits;

2 The grace period status;

3 The consumption time period to show (recent 7, 15, 30 or 60-days);

4 The average consumption display can be enabled;

5 The bar diagram shows the consumption for the selected number of days.

Events

The following Windows event log entries are generated related to the Electronic Outputs licensing:

Event ID	Description	Severity	Occurs	Name
1104	License is depleted. (<= 10% of licenses left)	Warning	Daily	LICENSE DEPLETED WARNING
1109	License overuse grace period is active	Warning	Always/Daily	LICENSEOVERUSE GRACEPERIOD ACTIVE
1110	License overuse grace period is expired	Warning	Always/Daily	LICENSEOVERUSE GRACEPERIOD EXPIRED
1112	License overuse grace period is reset	Info	Always	LICENSEOVERUSE GRACEPERIOD RESET
1120	Application consumed a quantity of consumable license resource	Info	Always	EVENT LICENSE CONSUME SUCCESS
1121	Application failed to consume a quantity of consumable license resource	Error	Always	EVENT LICENSE CONSUME FAILED ERROR

Email

The Email Process plugin allows to send custom emails during block processing.

Connection options

- *SMTP Server*: allows to indicate the address of the SMTP server. This address is a TCP/IP address.
- *Port*: allows to indicate the port. SMTP servers commonly use 25.
- *Use server authentication*: should be checked if using an SMTP server with authentication is needed. There are two types of it: standard or Google OAuth2.
- *Use secure connection (TLS)*: should be checked in the case when an SMTP server requires secure socket protocol.

Server authentication

- *Use standard authentication*: this option represents the basic authentication by using **User account** and **Password**.
- *Use Google authentication*: for the security reasons Google requires using OAuth2 **Access Tokens**.
  If Google authentication option is selected, the **ClientId** and **ClientSecrete** should be added. To define your own **ClientId** and **ClientSecrete** please follow the article: *https://support.google.com/cloud/answer/6158849*. You have to create it for the desktop application with the following scope defined: 'https://mail.google.com/'
  After the user fills the **ClientId** and **ClientSecrete** fields, the application has to pass the authentication process with permission granting. The process is performed in the browser (according to the Google recommendations). If everything is OK and **Access Token** for email sending is received, plugin settings can be saved.

Subject and body

The **Recipients** field allows to indicate the e-mail address where mail should be sent.

Several values can be put with ';' or/and space char as a separator.

Email subject cannot be configured and predefined, it contains:

'SENTINEL (time) / (sentinel) / Process'

Where the following placeholders are used:

- (time) is in format "1:44:57 PM"
- (sentinel) contains sentinel group and sentinel name with underscore as a separator

Note

The "Process" part is localized.

By default, email body template looks like this:

Time stamp: (date) (time)(crlf)plugin: (plug)(crlf)

The template contains placeholders like (date) that will be replaced during email body forming, there are several of them:

- (date) - in format: year month day, e.g. "2023 12 12"
- (time) - looks like this "5:58:08 PM"
- (crlf) - adds new line
- (plug) - contains identifier of a plugin: "TkxEmail"
- (num)/(event)/(info) - obsolete placeholders, will be empty

Remarks: you don't have to use (crlf) for new lines, the following example will work as well:

‘Time stamp: (date) (time) plugin: (plug)’

Available customization in runtime

@EMAIL_RECIPIENTS and @EMAIL_MESSAGE control variables provide customization possibilities during job processing:

@EMAIL_RECIPIENTS allows to change a list of recipients during runtime. Supports the same separators (semicolon and white-space) as fixed list of recipients. In case of invalid data an error message is reported.

@EMAIL_MESSAGE allows to control message template. There's no validation - any data is accepted (even empty line). Template functionality is supported.

@EMAIL_SUBJECT control variable was added. The value of the variable works as a template, similar to the email body, with the following four placeholders available:

- (date) for short date;
- (time) for the long time value;
- (sentinel) includes group and name of the current sentinel;
- (info) for a localized "Process" string.

Note

The empty value of the variable is considered valid. In case the variable is not present the default template will be used: "SENTINEL (time) / (sentinel) / (info)".

File Assignment

The plugin has the ability of sending files with emails. To activate this functionality the **Attach files** option has to be activated.

- *File source*: the plugin has different strategies for files searching such as:
  - *All files generated during process*: uses internal functionality to memorize what files were created during processing to send them.
  - *A single file*: a fixed absolute path to the file that should be sent.
  - *Files from folder*: a fixed absolute path to the folder from where files should be taken.
  - *A variable values as file path*: searches a variable and uses its value as an absolute path to a file.
  - *A variable values as folder path*: searches a variable and uses its value as an absolute path to a folder.
- *File path* or *Folder path* or *Variable name*: the field where should be defined additional information for some **File source** strategies.
- *Remove source files after this step*: if activated, this option automatically removes assigned files from the

local drive after processing (note: they still can be retrieved from the internal storage via files endpoints).

Note

The plugin will create a warning message in the log if files were not found.

To check the configuration users can click the **Test** button. The plugin will try to send an e-mail to all specified recipients. The e-mail sent uses the same format as the one specified in the plugin settings.

Note

Test functionality doesn't send files.

Database

Note

To use the Database plug-in, you require an additional license. Please contact your reseller to activate this functionality and ensure your database system is supported.

The Database process plug-in allows a sentinel to execute queries and call stored procedures over different database systems with the information received from the input plug-ins. Queries are defined at design time on the workstation using the Query Manager application.

Once the queries have been created they can be associated with stored procedures and called into a task. Task and query definitions are saved into two configuration files (task.xdb, database.xdb). These files must be used from the sentinels server when defining sentinel tasks with the information captured from the incoming data stream. It is then possible to execute a set of queries combined with stored procedure calls over different database systems for each block of data found by the sentinel.

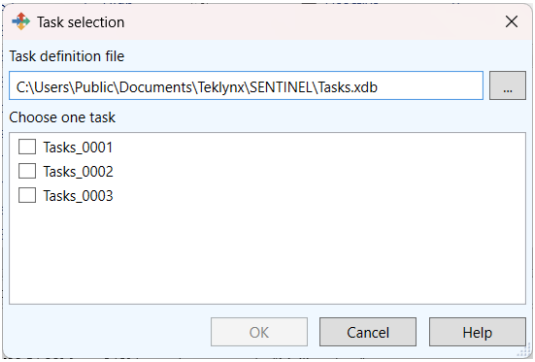


Figure 14: Task Selection window

If you want to dynamically select the task which would be called by the data stream received by a sentinel, you must assign a value to the @TASK control variable.

Step 1 Select **Database.**
Step 2 Click **Settings.**
Step 3 Enter a **Task Definition File** or click **Browse** to search for an existing file.
Step 4 Select one task from the list of configured tasks and click **OK**.

This task will be called up by default by the sentinel if the @TASK control variable is not set in the data stream.

Web Service Client

The goal of the web service client is to send data through WebService calls over the internet or intranet network.

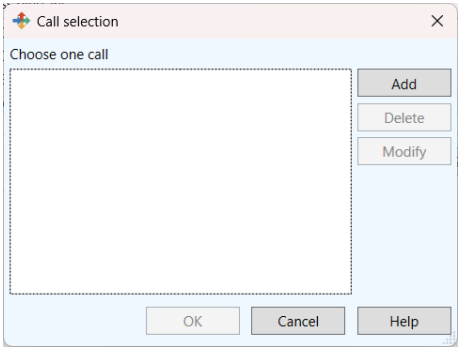


Figure 15: Task Selection window

Configuration

To use web calls, you must first ask for the web service description file (WSDL file).

Step 1 Open the WebService plug-in dialog box.

Step 2 Click the **Add** button.
The WebService Resolution window opens.

Step 3 Type the WSDL address (URL) in the edit box and click on the green arrow.
The plug-in will look for the service description. If a user login and password is required, a dialog box will be opened to enter required values.
Methods are listed in the table by method name followed by its parameters. Parameters are indented. If you click on a line, corresponding documentation will be shown on the bottom panel.
To add a method to the plug-in's usable method list, just enter an alias beside the method name. You can also give an alias name to the parameter name. When running sentinel, if parameters have an alias name, this one will be used as the variable resolution. The sentinel's variable value will be sent to the corresponding parameter. If the

parameter alias is omitted, its name will be used.
When methods are selected, you can close the Service Resolution dialog box. Then, in the configuration box, select the method to call from the plug-in as long as the @WEBMETHOD control variable of the plug-in is not set in the sentinel's incoming data stream.

Web Service Test

You can test a method through the Web Service Resolution Dialog Box. Open this dialog box, then click the **Test** button. A combo box allows you to select the method. Enter the parameter values and click on call. If a result is sent back, it will be displayed in the result panel.

Limitation

During the sentinel process, the result from a web call cannot be used or interpreted.

Output plug-ins

There are the following output plug-ins available:

- Transfer
- Email

Transfer

The transfer plug-in transfers input data to a specific directory according to the process result.

On the **Output** tab of the sentinel properties dialog box, select the output plug-in and click **Settings**.

The Transfer Plug-in window appears:

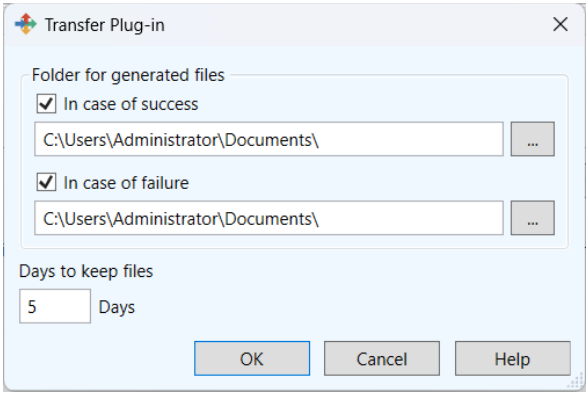


Figure 16: Transfer Plug-in window

In the **Folder for generated files** zone, select where you want to save your file in case of failure or in case of success.

In the **Days to keep file** zone, enter the number of days that the files will be kept.

The file name created by the output plug-in is identical to the name of the data set that is received by the input plug-in. If the same name is used for two different data sets, the old transferred file will be overwritten by the new one.

The **Days to keep files** option is only activated when a sentinel sends a file to the Transfer plug-in. If the plug-in is used, files remain in the transfer folder the amount of time specified in the **Days to keep files** option.

Email

The Email Output plugin allows to send custom emails after job processing.

Connection options

- *SMTP Server*: allows to indicate the address of the SMTP server. This address is a TCP/IP address.
- *Port*: allows to indicate the port. SMTP servers commonly use 25.
- *Use server authentication*: should be checked if using an SMTP server with authentication is needed. There are two types available: standard or Google OAuth2.
- *Use secure connection (TLS)*: should be checked in cases when an SMTP server requires secure socket protocol.

Server authentication

- *Use standard authentication*: this option represents the basic authentication by using **User account** and **Password**.
- *Use Google authentication*: for the security reasons

Google requires using OAuth2 **Access Tokens**.
If Google authentication option is selected, the **ClientId** and **ClientSecrete** should be added. To define your own **ClientId** and **ClientSecrete** please follow the article: *https://support.google.com/cloud/answer/6158849*. You have to create it for the desktop application with the following scope defined: 'https://mail.google.com/'
After the user fills the **ClientId** and **ClientSecrete** fields, the application has to pass the authentication process with permission granting. The process is performed in the browser (according to the Google recommendations). If everything is OK and **Access Token** for email sending is received, plugin settings can be saved.

Subject and body

The **Recipients** field allows to indicate the e-mail address where mail should be sent.

Several values can be put with ';' or/and space char as a separator.

Email subject cannot be configured and predefined, it contains:
'SENTINEL (time) / (sentinel) / (result)'

Where the following placeholders are used:

- (time) is in format "1:44:57 PM"
- (sentinel) contains sentinel group and sentinel name with underscore as a separator
- (result) localized string "Success"/"Failure" depending on result of processing

The Output Email plugin allows to configure different Success/Failure bodies, but by default both templates look like this:
'Time stamp: (date) (time)(crlf)plugin: (plug)(crlf)'

The template contains placeholders like (date) that will be replaced during email body forming, there are several of them:

- (date) - in format: year month day, e.g. "2023 12 12"
- (time) - looks like this "5:58:08 PM"
- (crlf) - adds new line
- (plug) - contains identifier of a plugin: "TkxEmail"
- (num)/(event)/(info) - obsolete placeholders, will be empty

Note

You don't have to use (crlf) for new lines, the following example will work as well: 'Time stamp: (date) (time) plugin: (plug)'

File Assignment

The plugin has the ability of sending files with emails. To activate this functionality the **Attach files** option has to be activated.

- *File source*: the plugin has different strategies for files searching such as:
  - *All files generated during process*: uses internal functionality to memorize what files were created during processing to send them.
  - *A single file*: a fixed absolute path to the file that should be sent.
  - *Files from folder*: a fixed absolute path to the folder from where files should be taken.
  - *A variable values as file path*: searches a variable and uses its value as an absolute path to a file.
  - *A variable values as folder path*: searches a variable and uses its value as an absolute path to a folder.
- *File path* or *Folder path* or *Variable name*: the field where should be defined additional information for some **Assignment type** strategies.
- *Block searching options*: should be defined if one of the variables assignment strategy was activated. The job can contain many blocks and each of them may

contain the variable, so these options allows plugin to find the needed block:

- *Check every block*: the plugin gathers every compatible variable from a job.
- *First matching block from the start*: the plugin tries to find the variable from the beginning of a job.
- *First matching block from the end*: the plugin tries to find the variable from the end of a job.
- *Block number from start*: the number of a block in a job from beginning.
- *Block number from end*: the number of a block from the end of the job.

Note

Block numeration starts from 1.

The last two options require additional information that should be defined in the corresponding field.

- *Remove source files after this step*: if activated, this option automatically removes assigned files from the local drive after processing (note: they still can be retrieved from the internal storage via files endpoints).

Note

The plugin will create a warning message in the log if files were not found.

To check the configuration users can click the **Test** button. The plugin will try to send an e-mail to all specified recipients. The e-mail sent uses the same format as the one specified in the plugin settings.

Note

Test functionality doesn't send files.

Report plug-ins

You have three **Report** plug-ins at your disposal:

- Log File
- E-mail
- TCP/IP Port Listening

Log file

The Log File plug-in logs sentinel events in a text file. On the **Report** tab of the Sentinel Properties window, select the report plug-in and click **Settings**.

The Log File Plug-in window appears:

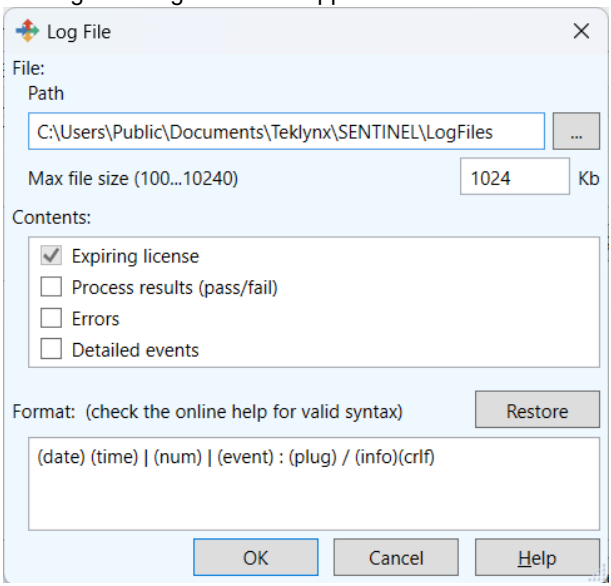


Figure 17: Log File Plug-in window

Step 1 On the file zone, enter the file path. The generated log file name is made as follows: REPORT (<sentinel name>).txt.

Step 2 Check one or several options in the **Contents**.

Step 3 In the **Format** zone, change the syntax of the file.

E-mail

The E-mail plug-in sends an e-mail containing all of the chosen events.

On the **Report** tab of the Sentinel Properties dialog box, select the input plug-in and click **Settings**.

The E-mail Plug-in window appears:

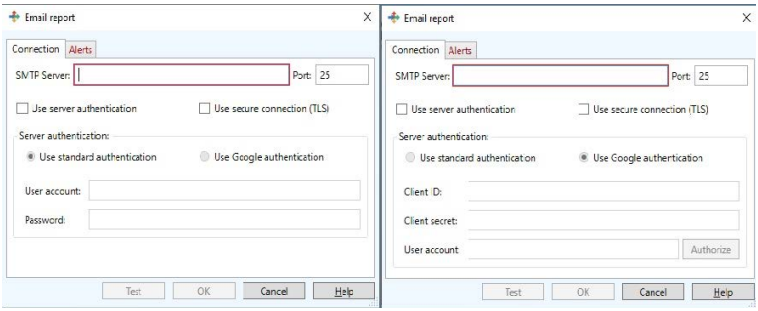


Figure 18: E-mail Plug-in window

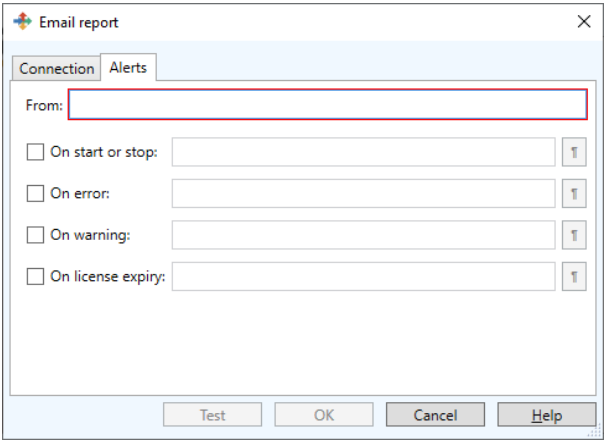
Step 1 On **Connection** tab in the **SMTP Server** zone, enter your SMTP Server address and select your port. The SMTP port number is commonly set to 25.

Step 2 Check the **Use server authentication** box if you want to use SMTP server with authentication: standard or Google. Check the **Use secure connection (TLS)** box if your SMTP server requires secure socket protocol.

Step 3 Check **Use standard authentication** if your server requires authentication and enter a **User account** and **Password**.

Step 4 Check **Use Google authentication** box if you want to use OAuth secure type, enter **Client ID** and **Client secret,** specify an appropriate **User account** and click **Authorize** button for connection verification**.**

Step 5 On **Alerts** tab in the **Form** zone, enter the address of the sender.



Step 6 In the four **Recipients** zones, enter the e-mail address of the person you want to contact when the corresponding event occurs.

Once you have entered your parameters, you can click **Cancel** or **Test** if you want to check the parameters you entered. Then the plug-in tries to send an e-mail to all of the specified recipients. The e-mail sent uses the same format as the one specified in the settings of the plug-in so that you can check it.

TCP/IP port listening

The TCP/IP Port Listening plug-in is viewed as a Socket server and receives data from client in a pre-defined format. For more information about this format, please refer to the example available on the DVD. This plug-in can be considered as an input and/or report plug-in. This plug-in can support any kind of platforms. It supports localhost. For more information about this plug-in configuration, please refer to the *Input plug-in* section of this chapter.

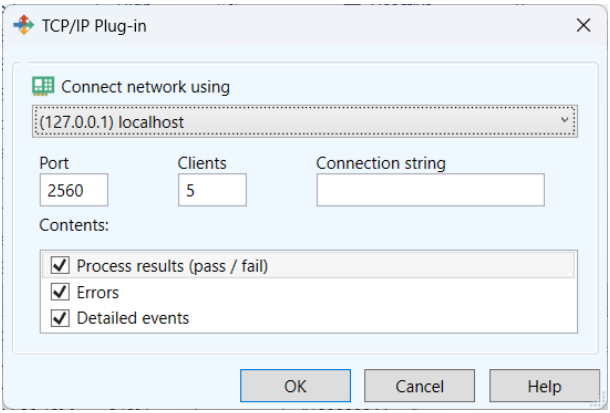


Figure 19: TCP/IP Plug-in window

Client/Server communication

1 The client tries to connect.

2 The server validates the connection.

3 The server sends the events linked to the job processing.

4 The client disconnects.

Message format

A message is sent to exchange information between the client and the Socket Server. The structure of this message is defined below (order is important):

Structure	Size and type
Message size (including these 4 bytes)	4 bytes (int)
Command number	4 bytes (int)
Parameter 1	4 bytes (int)

Parameter 2	4 bytes (int)
Name	260 bytes (array of char)
Data	variable length (array of char)

List of different commands

Login request

The connection string specified in the plugin configuration allows the client to connect to the server. Each client that is validated with this command counts towards maximum connections count. When maximum number of connections is reached, new connections will be refused.

Structure	Value
Command number	0
Parameter 1	0 - message in **ANSI (Default Windows Code Page, 1 byte)** 1 - message in **UNICODE (UTF16LE, 2 byte)**
Parameter 2	Client ID Optional parameter used to check information sent from the server to the client.
Name	Not used
Data	Connection string (if activated)

Login request response

Once the connection request has been made, the client must wait for an answer from the server to continue.

Structure	Value
Command number	1
Parameter 1	0 - OK 1 - Connection string is invalid 2 - Maximum number of clients reached
Parameter 2	Not used
Name	IP address or Client PC name + Client ID (value set during connection)
Data	Not used

Job request

Structure	Value
Command number	2
Parameter 1	0 - message in **ANSI (Default Windows Code Page, 1 byte)** 1 - message in **UNICODE (UTF16LE, 2 byte)**
Parameter 2	Not used
Name	Job name string
Data	Data string to process

Job report response

During processing the plugin reacts on the events in the Sentinel Unit and generates report responses.

Note

The TCP/IP server memorizes last encoding from login request or job request and uses it for the messages

sending.

Structure	Value
Command number	3
Parameter 1	0 - OK 1 - Cancelled 2 - Message 3 - Error
Parameter 2	**See table below**
Name	IP address or Client PC name + Client ID (value set during connection)
Data	Text message linked to the receiving of an error or a message. The module name or the name of the plugin is defined before this message.

Parameter 2 description:

Parameter	Value
On OK (0)	0 – No info
On Cancelled (1)	0 – No info
On message (2) reception and according to what has been selected in the Report plugin	0 - Sentinel stopped (client disconnected) 1 - Data received 2 - Data to be processed 3 - Data are sent to the output plugin 4 - Free message
On end of process (4)	0 - OK 1 - Process canceled 6 - Process error 7 - Output plugin error

The plugin sends the plugin name followed by {CRLF}, followed by the event message.

Logout request

This command is optional. Use it when you prefer to use plugin in "connect > process > disconnect" mode.

Structure	Value
Command number	4
Parameter 1	Not used
Parameter 2	Not used
Name	Not used
Data	Not used

Logout request response

Indicates logout result.

Structure	Value
Command number	5
Parameter 1	1 - OK
Parameter 2	Not used
Name	IP address or Client PC name + Client ID (value set during connection)
Data	Not used

Note

Communication between the client and the server can either be done in UNICODE(UTF16LE) or in ANSI (Default Windows Code Page). Pay attention: the default Windows code page (ANSI) is not the same for different locals (for US locale it is CP-1252) and

different code pages are not compatible. If the data is sent in UNICODE, the server will answer in UNICODE and vice versa.

APPENDIX B

API Capabilities

This appendix covers:

Kernel API

CODESOFT Plugin API

REST Web Server API

Definitions/Introduction

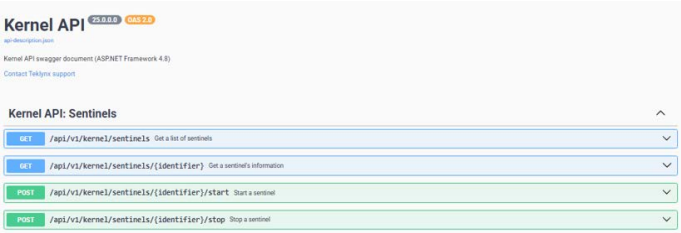
SENTINEL provides access to Kernel and CODESOFT plugin functions via REST API endpoints available over HTTP.

It follows best practices of REST API standard, uses JSON for responses and offers possibilities for the customers to automate the label printing.

On the dedicated swagger page, available by default at http://localhost:4505/swagger, all the endpoints and related information can be reviewed in great detail.

Kernel Sentinels API

Base address of the sentinels endpoints is http://localhost:4505/api/v1/kernel/sentinels. The following endpoints are available:



- **GET /api/v1/kernel/sentinels** - returns list of the defined sentinels. Supports optional query parameters offset and limit.

  Sample response model - JSON array of the sentinel models:

  ```
  [{
  "identifier": "Test_Rest",
  "isActivated": false,
  "state": "Stopped",
  "processedJobs": 0,
  "failedJobs": 0,
  "name": "Rest",
  "group": "Test",
  "comment": "",
  "isEnabled": true,
  "priority": "RealTime",
  "mapPath": "",
  "isErrorLoggingPathActivated": true,
  "errorLoggingPath": "C:\\Users\\username\\Desktop",
  "isErrorActionActivated": false,
  "errorAction": "StopSentinel"
  }]
  ```

- **GET /api/v1/kernel/sentinels/{identifier}** - retrieves information about the sentinel with specified identifier, which is a "{sentinel group}_{sentinel name}".

  Sample response JSON model:

  ```
  {
  ```

```
  "identifier": "Test_Rest",
  "isActivated": false,
  "state": "Stopped",
  "processedJobs": 0,
  "failedJobs": 0,
  "name": "Rest",
  "group": "Test",
  "comment": "",
  "isEnabled": true,
  "priority": "RealTime",
  "mapPath": "",
  "isErrorLoggingPathActivated": true,
  "errorLoggingPath": "C:\\Users\\username\\Desktop",
  "isErrorActionActivated": false,
  "errorAction": "StopSentinel"
}
```


- **POST /api/v1/kernel/sentinels/{identifier}/start** - starts the specified sentinel.

  Returns the latest sentinel state, for example: "Initialize"

- **POST /api/v1/kernel/sentinels/{identifier}/stop** - stops the specified sentinel.

  Returns the latest sentinel state, for example: "Stopped"

Sentinels models description

SentinelPriority (**enum**, used as a **string**)

Possible values:

- RealTime
- High
- Medium
- Low

SentinelState (**enum**, used as a **string**)

It is a flag, so individual values can be combined.
Possible values:

- Stopped (1) - a sentinel is deactivated

- Started (2) - a sentinel is activated
- Initialize (3) - a sentinel is starting
- Analysing (4) - a sentinel is processing
- Waiting (8) - a sentinel is started and waits for a job to process
- Construction (16) - a sentinel has invalid settings and should be configured
- Error (32) - a sentinel has at least one failed job since the start

SentinelErrorAction (**enum**, used as a **string**)
Possible values:

- StopSentinel - the application stops only this sentinel
- StopSentinelGroup - the application stops every sentinel of the same group
- StopAllSentinels - every sentinel will be stopped

SentinelInfoModel

- identifier - string, the sentinel identifier
- name - string, the sentinel name
- group - string, the sentinel group
- comment - string, the user defined information about sentinel
- isEnabled - boolean, true if the sentinel can be activated for processing
- isActivated - boolean, true if the sentinel was activate for processing
- mapPath - string, an absolute path to the map file, can be null or empty
- priority - SentinelPriority value from the enum (see above)
- state - SentinelState value from the enum (see above)
- isErrorLoggingPathActivated - boolean, true if error logging is active
- errorLoggingPath - string, contains an absolute path to the folder which should be used for error logging
- isErrorActionActivated - boolean, true if error action is

active

- state - action in case of error, SentinelErrorAction, value from the enum (see above)
- processedJobs - number, the number of successfully processed jobs
- failedJobs - number, the number of failed jobs

CODESOFT Plugin Printers API

Base address of the printers endpoints is http://localhost:4505/api/v1/cs-plugin/printers. The following endpoints are available:

- **GET /api/v1/cs-plugin/printers/identities** - retrieves the list of supported printer types. This is available for informational purposes.

  Response: ["THT","WINDOWS"]

- **GET /api/v1/cs-plugin/printers/license** - returns printers-related license information.

  Sample response model:
  ```
  {
    "enabledPrinters": 7,
    "maxPrinters": 30
  }
  ```

- **GET /api/v1/cs-plugin/printers** - retrieves list of the installed printers. Supports optional query parameters

offset and limit.

Sample response model:


```
[
    {
        "alias": "SNAP",
        "info": {
            "name": "SNAP 700",
            "port": "COM1:",
            "identity": "THT",
            "isDirect": false
        },
        "isEnabled": true,
        "rotation": 180,
        "lastUsed": "2023-07-14T14:13:29"
    }, …
]
```


- **GET /api/v1/cs-plugin/printers/{identity}** - same as the previous endpoint, but returns printers of the specified type only. Supports optional query parameters offset and limit.

  Sample response model:


```
[
    {
        "alias": "SNAP",
        "info": {
            "name": "SNAP 700",
            "port": "COM1:",
            "identity": "THT",
            "isDirect": false
        },
        "isEnabled": true,
        "rotation": 180,
        "lastUsed": "2023-07-14T14:13:29"
    }, …
]
```


- **GET /api/v1/cs-plugin/printers/{identity}/{printer-name}** - retrieves printer information based on the provided type and name.

  Sample response model:


```
[
```

```
    {
        "alias": "SNAP",
        "info": {
            "name": "SNAP 700",
            "port": "COM1:",
            "identity": "THT",
            "isDirect": false
        },
        "isEnabled": true,
        "rotation": 180,
        "lastUsed": "2023-07-14T14:13:29"
    }, …
]
```


- **GET /api/v1/cs-plugin/printers/{identity}/{printer-name}/status** - returns status of the printer based on the provided type and name. Note: it is recommended to use this endpoint with the printers that are using direct connection mode.


```
{
    "statusCode": 68353,
    "status": "Communication error."
}
```


Printers models descriptions

LimitsResponseModel

- enabledPrinters - **number** of activated printers
- maxPrinters - **number** of maximum possible activated printers

PrinterItemModel

- alias - **string**, user defined alias for printer (unique among all aliases)
- info - **PrinterInfoModel** additional printer information
- isEnabled - **boolean**, true if printer can be used for printing

- rotation - **number**, can be next values: 0, 90, 180, 270
- trays - collection of **PrinterTrayItemModel**
- lastUsed - **DateTime** string with the last saved time of printer usage (UTC)

PrinterInfoModel

- name - **string**, user defined name of printer
- port - **string**, the name of printer port
- isDirect - **boolean**, indicates port mode
- identity - **string**, THT or WINDOWS

PrinterTrayItemModel

- name - **string**, device defined name of the tray
- alias - **string**, user defined alias for tray (unique among all aliases)

StatusResponseModel

- statusCode - **number**, internal status identifier
- status - **string**, user friendly description (English only)

Status codes:

Ready = 0
Print head error = 1
Ribbon error = 2
Paper error = 3
Memory error = 4
Cutter error = 7
Generic printer error = 5
Memory card error = 9
Print head lifted = 65794
Print head heat error = 65795
Out of ribbon = 66049
Out of paper = 66305
Paper jammed = 66307

Printer paused = 66817
Printer offline = 66818
Printer not ready = 66821
Cutter jammed = 67330
Communication error = 68353

REST Web Server API

REST Web Server API Swagger page is available at http(s)://IP_ADDRESS:PORT/Swagger, depending on configuration.

Add a job

POST /api/v1/jobs

Adds a new job.

Receives a JSON array of arrays of Key Value pairs, every pair is represented by
two Strings:

```
[
  {
    "@LABEL_NAME": "Identity.lab",
    "@PRINTER_NAME": "Bullzip PDF Printer,BULLZIP",
    "@JOB_NAME": "TEST JOB"
  },
  {
    "@LABEL_NAME": "Identity.lab",
    "@PRINTER_NAME": "Bullzip PDF Printer,BULLZIP",
    "@JOB_NAME": "TEST JOB 2"
  }
]
```

200 (sync request) or **201** (async request): Returns JSON JobModel:

```
{
  "id": 335,
  "status": "Waiting",
  "creationDate": "2023-12-
```

```
28T15:11:54.8499136+02:00"
}
```

400: Error in the sent data

401: Unauthorized, if ApiKey is used and wasn't provided with request

500: Internal server error

Get job list

GET /api/v1/jobs

Returns jobs from the server database. This request has two optional query parameters as any other "list" request:

- offset (default - 0)
- limit (default - 100)

```
/api/v1/jobs?offset=1&limit=2
```

and several parameters which are made for searching purposes, the request works as "AND" clasue:

- **variables[variable-name]** (can be repeated up to three times)
- **result** (see JobResult enumeration in API Models section)
- **status** (see JobStatusenumeration in API Models section)

```
/api/v1/jobs?variables[@JOB_NAME]=TEST
```

200: Returns array of JobModels:

```
[
  {
    "id": 335,
    "status": "Completed",
    "result": "Passed",
    "creationDate": "2023-12-28T15:11:54.8499136",
    "processingDate": "2023-12-28T15:11:58.9759059",
```

```
    "resultMessage": "The current job item processed successfully."
  },
  {
    "id": 336,
    "status": "InProgress",
    "creationDate": "2023-12-28T15:15:06.9695091"
  }
]
```


401: Unauthorized, if ApiKey is used and wasn't provided with request

403: Query parameters error

500: Internal server error

Get job list with variables

GET /api/v1/jobs/full-info

Returns jobs from the database with variables included. This request has two optional query parameters:

- **offset** (default - 0)
- **limit** (default - 100)

```
/api/v1/jobs/full-info?offset=1&limit=2
```

and several parameters which are made for searching purposes, the request works as
"AND" clasue:

- **variables[variable-name]** (can be repeated up to three times)
- **result** (see JobResult enumeration in API Models section)
- **status** (see JobStatusenumeration in API Models section)

```
/api/v1/jobs/full-info?variables[@JOB_NAME]=TEST
```

200: Returns array of JobModels with variables:


```
[
  {
    "id": 335,
    "status": "Completed",
    "result": "Passed",
    "creationDate": "2023-12-28T15:11:54.8499136",
    "processingDate": "2023-12-28T15:11:58.9759059",
    "resultMessage": "The current job item processed
successfully.",
    "blocks": [
      {
        "order": 0,
        "variables": {
          "@LABEL_NAME": "Identity.lab",
          "@PRINTER_NAME": "Bullzip PDF
Printer,BULLZIP",
          "@JOB_NAME": "TEST JOB",
          "Address": "307 7th Ave",
          "City": "New York",
          "Code": "1346",
          "FirstName": "John",
          "Name": "Smith",
          "State": "NY",
          "Zip": "10001"
        }
      }
    ]
  },
  {
    "id": 336,
    "status": "InProgress",
    "creationDate": "2023-12-28T15:15:06.9695091",
    "blocks": [
      {
        "order": 0,
        "variables": {
          "@LABEL_NAME": "Identity.lab",
          "@PRINTER_NAME": "Bullzip PDF
Printer,BULLZIP",
          "@JOB_NAME": "TEST JOB",
          "Address": "307 7th Ave",
          "City": "New York",
          "Code": "1346",
```

```
        "FirstName": "John",
        "Name": "Smith",
        "State": "NY",
        "Zip": "10001"
      }
    }
  ]
 }
]
```


401: Unauthorized, if ApiKey is used and wasn't provided with request

403: Query parameters error

500: Internal server error

Get job information

GET /api/v1/jobs/{job-id}

Returns an information about a single job from the database by given identifier (jobId).

200: Returns single JobModel:


```
{
 "id": 335,
 "status": "Completed",
 "result": "Passed",
 "creationDate": "2023-12-28T15:11:54.8499136",
 "processingDate": "2023-12-28T15:11:58.9759059",
 "resultMessage": "The current job item processed
successfully."
}
```


401: Unauthorized, if ApiKey is used and wasn't provided with request

500: Internal server error

Get job information with variables

GET /api/v1/jobs/{job-id}/full-info

Returns an information about a single job with included variables from the database by given identifier (jobId).

200: Returns single JobModel:


```
{
  "id": 335,
  "status": "Completed",
  "result": "Passed",
  "creationDate": "2023-12-28T15:11:54.8499136",
  "processingDate": "2023-12-28T15:11:58.9759059",
  "resultMessage": "The current job item processed
successfully.",
  "blocks": [
    {
      "order": 0,
      "variables": {
        "@LABEL_NAME": "Identity.lab",
        "@PRINTER_NAME": "Bullzip PDF
Printer,BULLZIP",
        "@JOB_NAME": "TEST JOB",
        "Address": "307 7th Ave",
        "City": "New York",
        "Code": "1346",
        "FirstName": "John",
        "Name": "Smith",
        "State": "NY",
        "Zip": "10001"
      }
    }
  ]
}
```


401: Unauthorized, if ApiKey is used and wasn't provided with request

500: Internal server error

Update job variables

PUT /api/v1/jobs/{jobId} (deprecated)

Updates variables of existing job by identifier (job should be in the Waiting status).

Deprecated due to the impossibility of guaranteeing the current status of the job due to which the success of execution is a pure random.

Receives a block structure with variables that have to be updated:

```
[
  {
    "order": 0,
    "variables": {
      "@LABEL_NAME": "Identity.lab",
      "@JOB_NAME": "CA TEST JOB"
    }
  },
  {
    "order": 1,
    "variables": {
      "@LABEL_NAME": "Identity.lab",
      "@JOB_NAME": "CA TEST JOB",
      "FirstName": "Devid"
    }
  }
]
```

200: The job was updated successfully

400: Invalid input data

401: Unauthorized, if ApiKey is used and wasn't provided with request
403: The job status is incompatible with updating operation

500: Internal server error

Remove job

DELETE /api/v1/jobs/{job-id}

Removes a single job by the given identifier (jobId).

200: The job was removed successfully

401: Unauthorized, if ApiKey is used and wasn't provided with request

404: The job was not found

500: Internal server error

Remove jobs

DELETE /api/v1/jobs

Removes several jobs if the server finds them.

Receives a JSON array of identifiers:

```
[
  10356,
  10389,
  10390
]
```

200: Jobs were removed successfully (at least one of them), returns special structure where marks what jobs were removed and what weren't found:

```
{
  "deleted": [
    10356,
    10389,
  ],
  "notFound": [
    10390
  ]
}
```

401: Unauthorized, if ApiKey is used and wasn't provided

with request

404: All job were not found

500: Internal server error

API Models

All presented models have to be presented in the JSON format; other formats won't be recognized by a server. Any null value will not be placed in server response to reduce traffic.

JobStatus (**enum**, used as a **string**) Possible values:

- Completed - the job is processed already
- InProgress - the job is currently in process
- Waiting - the job is created but not processed yet

JobResult (**enum**, used as a **string**) Possible values:

- Passed - the job successfully processed
- Fail - the procession of the job is failed
- Canceled - the processing was aborted

BlockModel

- order - integer **number**, index number of the block (starts from zero)
- variables - **array** of **key/value** pairs of **strings**

JobModel

- id - long integer **number**, Identifier from Database
- status - **JobStatus** (see above)
- result - **JobResult** (see above, can be null)

- creationDate - **string**, date time structure
- processingDate - **string**, date time structure (can be null)
- resultMessage - **string** (can be null)

JobFullModel

- id - long integer **number**, Identifier from Database
- status - **JobStatus** (see above)
- result - **JobResult** (see above, can be null)
- creationDate - **string**, date time structure
- processingDate - **string**, date time structure (can be null)
- resultMessage - **string** (can be null)
- blocks - **array of BlockModel** (see above)

DeleteResultModel

- deleted - **array** of long integer **numbers**, identifiers of jobs that were removed
- notFound - **array** of long integer **numbers**, (can be null) identifiers of jobs that weren't found

Note

Every date time structure presented in format: yyyy-MM-dd HH:mm:ss.fffffff

Get a list of files

GET /api/v1/files/{job-id}

Enumerates all files associated with the job in the storage and returns array of file info models. This request has two optional query parameters: offset (default - 0), limit (default - 100):

/api/v1/files/{jobId}?offset=1&limit=2

200: Returns array of file info models:


```
[
  {
    "jobId": "337",
    "fileName": "Frame 1.png",
    "creationTime": "12/28/2023 4:27:49 PM",
    "lastWriteTime": "5/22/2023 4:27:13 PM",
    "size": 18150
  },
  {
    "jobId": "337",
    "fileName": "IDLgqEuM.png",
    "creationTime": "12/28/2023 4:27:49 PM",
    "lastWriteTime": "5/29/2023 2:14:43 PM",
    "size": 8900403
  },
  {
    "jobId": "337",
    "fileName": "jazz-jack-artist-7442947.jpeg",
    "creationTime": "12/28/2023 4:27:49 PM",
    "lastWriteTime": "6/15/2022 8:52:23 PM",
    "size": 98259
  }
]
```


401: Unauthorized, if ApiKey is used and wasn't provided with request

404: The job was not found

500: Internal server error

Get a single file information

GET /api/v1/files/{job-id}/{file-name}

Returns info model for a file associated with a job from the storage.

200: Returns a single info model:


```
{
```

```
  "jobId": "337",
  "fileName": "Frame 1.png",
  "creationTime": "12/28/2023 4:27:49 PM",
  "lastWriteTime": "5/22/2023 4:27:13 PM",
  "size": 18150
}
```


401: Unauthorized, if ApiKey is used and wasn't provided with request

404: The job or the file was not found

500: Internal server error

Download all files

GET /api/v1/files/{job-id}/attachment

Returns zip archive with all files associated with a job from the storage.

200: Returns a zip archive stream

401: Unauthorized, if ApiKey is used and wasn't provided with request

404: The job was not found

500: Internal server error

Download a single file

GET /api/v1/files/{job-id}/{file-name}/attachment

Returns a file associated with a job from the storage.

200: Returns a file as an application/octet-stream

401: Unauthorized, if ApiKey is used and wasn't provided with request

404: The job or the file was not found

500: Internal server error

Remove all files

DELETE /api/v1/files/{job-id}

Removes all files which are associated with a job from the storage.

200: The job files were removed successfully

401: Unauthorized, if ApiKey is used and wasn't provided with request

404: The job was not found

500: Internal server error

Remove a single file

DELETE /api/v1/files/{job-id}/{file-name}

Removes file associated with a job from the storage.

200: The job file was removed successfully

401: Unauthorized, if ApiKey is used and wasn't provided with request

404: The job or the file was not found

500: Internal server error

API Models

FileInfoModel

- jobId - long integer **number**, the job identifier from the database
- fileName - **string**, absolute path to the file
- creationTime - **string**, date time structure
- lastWriteTime - **string**, date time structure
- size - long integer **number**, the size of the file in the bytes

France
+33 (0) 562 601 080

Germany
+49 (0) 2103 2526 0

Singapore
+65 6908 0960

United States
+1 (414) 837 4800

Copyright 2026 TEKLYNX Corporation SAS. All rights reserved. LABEL MATRIX, LABELVIEW, CODESOFT, LABEL ARCHIVE, SENTINEL, PRINT MODULE, BACKTRACK, TEKLYNX CENTRAL, TEKLYNX, and Barcode Better are trademarks or registered trademarks of TEKLYNX Corporation SAS or its affiliated companies. All other brands and product names are trademarks and/or copyrights of their respective owners.

www.teklynx.com