IMiS®/wScan Manual

Version 1.9.2310

MARGIS, inzeniring d.o.o.
Brnciceva ulica 41 G
1231 Ljubljana – Crnuce
Slovenia

TABLE OF CONTENTS

1 INTRODUCTION 11
1.1 About manual 11
1.2 Target audience 11
2 GENERAL 11
2.1 Architecture 13
2.1.1 Modular design 13
2.1.2 Multi-level architecture 14
2.2 Security 17
2.3 Functionalities 19
2.3.1 Module for the capture of content from an optical scanner 19
2.3.2 Module for saving content 20
2.3.3 Module for barcode recognition 20
2.4 Application integration 21
2.4.1 Integration of IMiS®/Capture Service 21
2.4.2 Integration of imis.scan.js library 21
2.4.3 Integration of imis.scan.ui.js library 23
2.5 Versions 24
2.6 New functionalities in this version 25
3 SCANNING DOCUMENTS 27
3.1 Methods of Scanning Documents 27
3.2 Connecting a Scanner 27
3.3 Scanning Resolution and Quality 28
3.4 File Compression and Size 28
3.5 Recording of Scanned Documents 30
4 SYSTEM REQUIREMENTS 33
4.1 Hardware 33
4.1.1 Minimum requirements 33
4.1.2 Recommended requirements 33
4.2 Software 34
5 PRODUCT MANAGEMENT 35
5.1 Installation 35
5.1.1 Installation with the Wizard 37
5.1.2 Silent installation 43
5.2 Startup and closing 45
5.3 Additional settings 47
5.3.1 Product activation 49

2023 © MARGIS Inc., All rights reserved.

5.3.1.1 Activating the license online 49
5.3.1.2 Alternative license activation 50
5.3.1.3 Range of features 51
5.3.2 Additional profile settings 52
5.3.3 Security settings 54
5.3.4 Additional administrator settings 56
5.4 Uninstallation and modification 56
5.4.1 Uninstall 57
5.4.2 Installation modifications and repairs 60
5.4.2.1 Installation modifications 60
5.4.2.2 Installation repairs 62
5.5 Upgrade 62
6 TECHNICAL DOCUMENTATION 63
6.1 imis.scan.js 63
6.1.1 imis.scan.Scan 63
6.1.2 imis.scan.Profile 68
6.1.3 imis.scan.Job 71
6.1.4 imis.scan.Document 76
6.1.5 imis.scan.Page 79
6.1.6 imis.scan.Barcode 83
6.1.7 imis.scan.Redaction 84
6.1.8 imis.scan.Module 84
6.1.9 imis.scan.ScannerModule 85
6.1.10 imis.scan.FolderTargetModule 86
6.1.11 imis.scan.BarcodeExtractorModule 87
6.1.12 imis.scan.BlankPageDetectorModule 88
6.1.13 imis.scan.PageCountSeparatorModule 88
6.1.14 imis.scan.BarcodeSeparatorModule 89
6.1.15 imis.scan.BlankPageSeparatorModule 89
6.1.16 imis.scan.ScannerValue 89
6.1.17 imis.scan.ColorFormat 90
6.1.18 imis.scan.BarcodePattern 90
6.1.19 imis.scan.model.AttributeDefinition 91
6.1.20 imis.scan.model.DocumentAttribute 92
6.1.21 imis.scan.model.Info 92
6.1.22 imis.scan.Highlight 93
6.2 imis.scan.ui.js 93
6.2.1 imis.scan.ui.Scan 93

2023 © MARGIS Inc., All rights reserved.

6.2.1.1 ScanOptions 96
6.2.1.2 ScanButtonsOptions 99
6.2.2 imis.scan.ui.Button 100
6.2.2.1 ButtonOptions 101
6.2.3 imis.scan.ui.SplitButton 101
6.2.3.1 SplitButtonOptions 102
6.2.4 imis.scan.ui.ColorDropdownButton 103
6.2.4.1 ColorDropdownOptions 103
6.2.5 imis.scan.ui.ProfilesButton 104
6.2.5.1 ProfilesButtonOptions 104
6.2.6 imis.scan.ui.ImageDetails 105
6.2.6.1 ImageDetailsOptions 106
6.2.7 imis.scan.ui.ImageView 106
6.2.7.1 ImageViewOptions 107
6.2.8 imis.scan.ui.ImageScroll 108
6.2.8.1 ImageScrollOptions 108
6.2.9 imis.scan.ui.Progress 110
6.2.9.1 ProgressOptions 110
6.2.10 imis.scan.ui.Status 110
6.2.10.1 StatusOptions 111
6.2.11 imis.scan.ui.Thumbnails 111
6.2.11.1 ThumbnailsOptions 112
6.2.12 imis.scan.ui.Settings 114
6.2.12.1 SettingsOptions 115
6.2.13 imis.scan.ui.AlertDialog 115
6.2.13.1 AlertDialogOptions 116
6.2.14 imis.scan.ui.TargetColor 116
6.2.14.1 TargetColorOptions 116
6.2.15 imis.scan.ui.TargetFormat 117
6.2.15.1 TargetFormatOptions 117
6.2.16 imis.scan.ui.TotalDocuments 117
6.2.16.1 TotalDocumentsOptions 118
6.2.17 imis.scan.ui.TotalPages 118
6.2.17.1 TotalPagesOptions 118
6.2.18 imis.scan.ui.CursorMode 119
6.2.18.1 CursorModeOptions 119
6.2.19 imis.scan.ui.ResolutionDropdownButton 119
6.2.19.1 ResolutionDropdownButtonOptions 120

2023 © MARGIS Inc., All rights reserved.

6.3 Examples of use imis.scan.js 120
6.3.1 Reading profiles 120
6.3.2 Changing a profile 121
6.3.3 Starting a job 123
6.3.4 Deleting a profile 126
6.4 Examples of use imis.scan.ui.js 127
6.4.1 Classic sample 127
6.4.1.1 classic.html 129
6.4.1.2 classic.css 131
6.4.2 Modern sample 132
6.4.2.1 modern.html 134
6.4.2.2 modern.css 136
6.4.3 Classic dark sample 137
6.4.3.1 classic_dark.html 139
6.4.3.2 classic.dark.css 141
6.4.4 Gallery sample 142
6.4.4.1 gallery.html 143
6.4.4.2 gallery.css 145
7 USER DOCUMENTATION 146
7.1 Profile settings 147
7.1.1 Profile creation 148
7.1.2 Source 149
7.1.3 Target 151
7.1.4 Separator 152
7.1.4.1 Disabled 152
7.1.4.2 Number of pages 153
7.1.4.3 Barcode 153
7.1.4.4 Blank page 157
7.1.5 Metadata 158
7.2 Application features 160
7.2.1 Document capture 161
7.2.2 Document thumbnails 162
7.2.3 Document pages 165
7.2.4 Document data 166
7.2.5 Adding document pages 168
7.2.6 Overwriting document pages 169
7.2.7 Moving document page 169
7.2.8 Splitting a document 170

2023 © MARGIS Inc., All rights reserved.

7.2.9 Joining documents 171
7.2.10 Cropping a page area 172
7.2.11 Hiding page areas 173
7.2.12 Highlighting page areas 175
7.2.13 Defining the metadata region 177
7.2.14 Access to service log files 179
7.3 Batch scanning features 179
7.3.1 Separating documents by page count 183
7.3.2 Separating documents by barcode 183
7.3.2.1 Separator editing 184
7.3.3 Blank page separation 186
7.3.3.1 Setting the threshold 188
8 TROUBLESHOOTING 190
8.1 Problems using IMiS®/wScan 190
8.1.1 Error “Forbiden” 190
8.1.2 Error “Error in establishing connection” 190
8.1.3 Error “Socket connection error” 191
8.1.4 Error “Error in establishing connection” or “Socket connection error” 191
8.1.5 Error “No scanner is connected” 192
8.1.6 Error “Scanner: ‘{scanner name}' cannot be loaded”. 192
8.1.7 Scanning cannot be continued after successful scanning 193
8.1.8 During scanning empty pages are not being removed 193
8.1.9 Error “Your browser does not support Javascript ES6. Update browser.” 193
8.1.10 Error “Your browser does not support WebSockets. Update browser.” 194

2023 © MARGIS Inc., All rights reserved.

TABLE OF IMAGES

Table of images appearing in the manual

Image 1: Scheme of integrating the product into a multi-level web application 12
Image 2: Scheme of the IMiS®/Capture Service core service 13
Image 3: Scheme of JavaScript program levels of IMiS®/wScan application 15
Image 4: Example of using imis.scan.js library to read profiles 22
Image 5: Example of using imis.scan.ui.js library to set the appearance of the Thumbnails component 23
Image 6: Notification of prohibited simultaneous installation of the 64-bit and 32-bit version of IMiS®/wScan 35
Image 7: Notification of prohibited simultaneous installation of the 32-bit and 64-bit version of IMiS®/wScan 36
Image 8: Notification of required installation of .NET Framework 4.5 36
Image 9: Viewing and accepting the license terms 37
Image 10: Entering data on application user 38
Image 11: Selecting between typical, full or user-customized installation 38
Image 12: Selecting which shortcuts will be created and which options will be activated during installation 39
Image 13: Displaying the result of selecting “Start menu” 40
Image 14: Starting the installation procedure 41
Image 15: Displaying the progress bar during the installation procedure 41
Image 16: Notification of completing the installation procedure 42
Image 17: Selecting the features of application installation 42
Image 18: Example of a command bar for silent installation in previous version 43
Image 19: Displaying the current status of IMiS®/Capture Service: stopped 45
Image 20: Selecting the option to start IMiS®/Capture Service 46
Image 21: Displaying the status of IMiS®/Capture Service: running 46
Image 22: Selecting the option to stop IMiS®/Capture Service 46
Image 23: Selecting the option to restart IMiS®/Capture Service 47
Image 24: Selecting the option to show additional settings 47
Image 25: Dialog box for setting additional settings 48
Image 26: Dialog box for the online activation of the IMiS®/Capture Service license 49
Image 27: Dialog box for the offline activation of the IMiS®/Capture Service license 50
Image 28: Example of an activated IMiS®/Capture Service license 52
Image 29: Dialog box for profile settings and security settings 53
Image 30: Configuration dialog box of Fujitsu driver 53
Image 31: Dialog box for security settings 54

2023 © MARGIS Inc., All rights reserved.

Image 32: Example of an inbound rule in Windows Firewall settings 55
Image 33: Selecting between installation modification and uninstallation of application 57
Image 34: Selecting the uninstallation of application 57
Image 35: Displaying the progress bar of the configuration review 57
Image 36: Selecting application removal 58
Image 37: Confirming application removal 58
Image 38: Displaying the progress bar during the uninstallation procedure 59
Image 39: Notification of finishing the procedure of uninstalling the installation package 59
Image 40: Selecting between installation modifications and repairs and removing the installed application 60
Image 41: Starting the procedure of implementing installation modifications and repairs 60
Image 42: Selecting installation modification 61
Image 43: Selecting features when modifying installation 61
Image 44: Selecting installation repairs 62
Image 45: Button component “Scan” 100
Image 46: Button component 101
Image 47: Component for selecting the color of scanning 103
Image 48: Component for selecting the profile and changing the settings of the scanning profile 104
Image 49: Component for displaying information about the currently selected page 105
Image 50: Component for displaying the currently selected page 106
Image 51: Component for displaying a collection of pages 108
Image 52: Component for showing the progress of the current job 110
Image 53: Component for displaying the status 110
Image 54: Component for showing documents 111
Image 55: Component for setting profiles 114
Image 56: Component for displaying a dialog 115
Image 57: Component for selecting the scanning resolution 119
Image 58: Example of using the classic sample of a user interface display 128
Image 59: Example of using the modern sample of a user interface display 133
Image 60: Example of using the classic (dark) sample of a user interface display 138
Image 61: Example of using the gallery sample of a user interface display 142
Image 62: Homepage of the IMiS®/wScan application 146
Image 63: Profile settings 148
Image 64: Selecting the default profile 148
Image 65: Profile creation 149
Image 66: Profile source settings 150
Image 67: Popup menu of the “Scanner” setting 150

2023 © MARGIS Inc., All rights reserved.

Image 68: Setting the profile target 151
Image 69: Setting the separator to ”Disabled” 152
Image 70: Setting the separator to “Page count” 153
Image 71: Barcode separator settings 155
Image 72: Popup menu of the setting "Barcode" 156
Image 73: Setting the separator to “Blank page” 157
Image 74: Profile metadata settings 158
Image 75: Defining the metadata parameters 159
Image 76: Popup menu in the metadata settings 159
Image 77: User interface in the classic mode 160
Image 78: A command bar with document information in the classic mode 162
Image 79: Popup menu in the case of a single document in the left view 163
Image 80: Popup menu in the case of multiple documents in the left view 163
Image 81: Popup menu on the document page in the classic mode 165
Image 82: Toolbar on the document page in the classic mode 166
Image 83: Document data 167
Image 84: Document details with the “Attributes” set 167
Image 85: Adding document pages from the device or file system 168
Image 86: Overwriting document pages from the device or file system 169
Image 87: Moving a document page 170
Image 88: Separating document pages 170
Image 89: Example of joining two documents 171
Image 90: Selecting the action “Crop” 172
Image 91: Highlighting the crop area 172
Image 92: Cropping a page area 173
Image 93: Selecting the action “Redaction” 174
Image 94: Highlighting the redaction areas 174
Image 95: Hiding page areas with redactions 175
Image 96: Selecting the action “Highlight” 176
Image 97: Selecting one or multiple areas on the document page 176
Image 98: Area of highlighted text on the document page 177
Image 99: Selecting the metadata for defining the region 178
Image 100: Selecting the region for metadata capture 178
Image 101: View of the enlarged region for metadata capture 178
Image 102: Access to IMiS®/Capture Service log files 179
Image 103: User interface in the “Gallery” mode 180
Image 104: A command bar with document information in the “Gallery” mode 181
Image 105: Thumbnails of document pages in the “Gallery” mode 181

2023 © MARGIS Inc., All rights reserved.

Image 106: Popup menu on document in the “Gallery” mode 182
Image 107: View of document data in the “Gallery” mode 182
Image 108: Example of separation by page count (N = 3) 183
Image 109: Example of separation by barcode 184
Image 110: Selecting the action “Edit separator” 185
Image 111: Changing the value or adding a new separator next to the document label 185
Image 112: Changing the separator value in document properties 186
Image 113: Example of failed blank page separation 187
Image 114: Example of successful blank page separation 187
Image 115: Example of the value of the threshold for non-blank document page detection 188
Image 116: Example of the value of the threshold for blank document page detection 188
Image 117: Example of setting the “Threshold” value for successful document separation 189
Image 118: Shows the entry in the MS Edge title bar 191

2023 © MARGIS Inc., All rights reserved.

1 INTRODUCTION

1.1 About manual

This IMiS®/wScan manual describes the functionalities of and working with the IMiS®/wScan application.

1.2 Target audience

It is intended for administrators and application developers with prior technical knowledge who need information about the installation and configuration of the IMiS®/wScan application, its IMiS®/Capture Service core, and integration with third-party applications.
Below is the technical documentation with a detailed description of IMiS®/Capture Service, provided for application developers. Similarly, the user documentation with a detailed description of the settings of the IMiS®/wScan application is provided below for users.

2 GENERAL

IMiS®/wScan is up to date with all contemporary technological, functional and design standards in the field of software for the capture of physical documents. Its design enables fully functional use in a multi-level architecture, with a web browser as the integration point. Owing to its modular and level design it is highly customizable and usable in various implementation scenarios, either with or without user interaction.

Despite the fact that the capture of physical content can hardly be done without using physical computer components (optical scanner), the use of which has practically been prevented by web browser producer, IMiS®/wScan nevertheless enables the digitization of physical documents by using inherently safe technologies in a purely online solution without the use of plug-ins or similar components.

It enables users to capture contents and digitize them in pure online solutions. It has been made in compliance with the ECMAScript 2016 specification. Despite the rather new JavaScript language specification, browser support is guaranteed sufficiently.

2023 © MARGIS Inc., All rights reserved.

The web service meets the following criteria, conceptually:

- It has been designed with pure JavaScript technology without any additional requirements for e.g. plug-ins or access to the "native" protocols NPAPI, COM, etc. Therefore, it is not based on the technologies that browser makers consider unsuitable (dangerous).
- This simple, intuitive and customizable user interface enables application developers' full flexibility when integrating it into any web applications.
- Integration at a sufficiently low level enables application developers' customization even in the event of technological conflicts with an internally used framework should the application developer be unable to use e.g. constructs of the View level of the application.

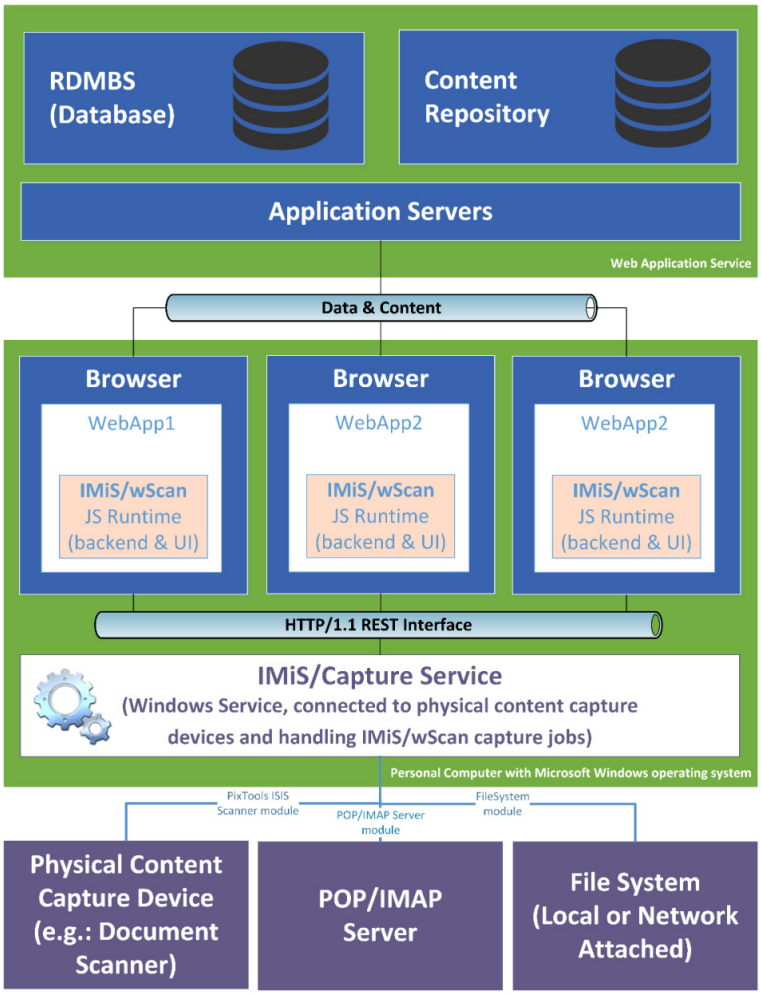

Image 1: Scheme of integrating the product into a multi-level web application

2023 © MARGIS Inc., All rights reserved.

2.1 Architecture

2.1.1 Modular design

IMiS®/Capture Service has a module chain design in which different monolithic building blocks (modules) can be arranged in sequence, depending on the needs of the job. Each module provides a certain functionality (e.g. module for communicating with an optical scanner, module for barcode recognition, module for separating documents, module for merging documents, module for document conversion, module for saving to the archive system, etc.). The modular design enables capture from a different and heterogeneous selection of sources.

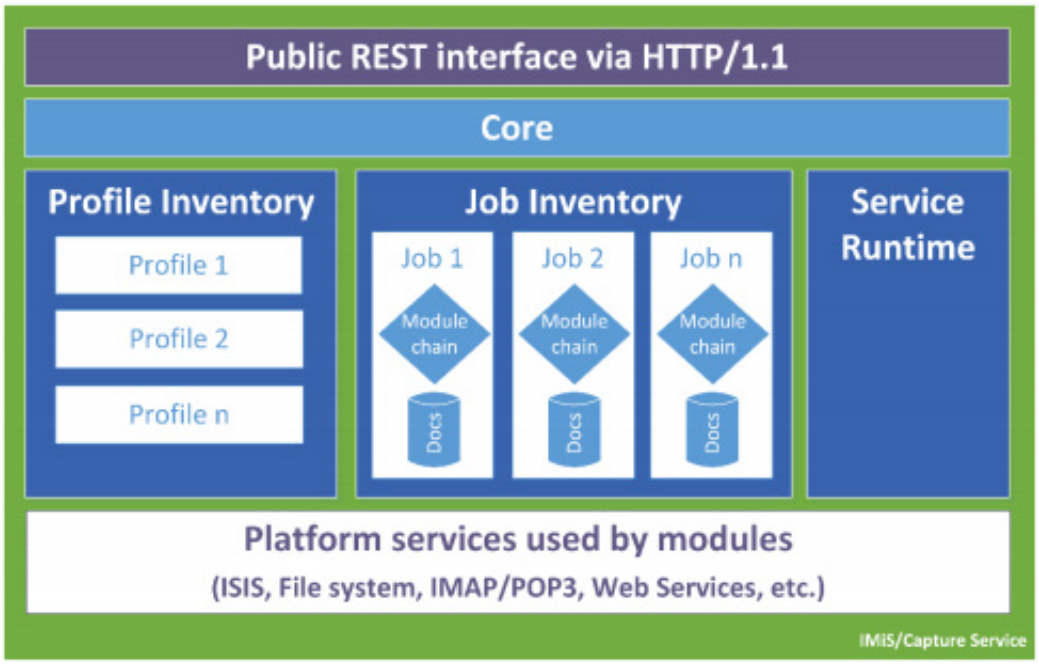

Image 2: Scheme of the IMiS®/Capture Service core service

Note: Further development anticipates a gradual introduction of modules for the capture of documents from inboxes via IMAP/POP3 technology, a file system, external sources through various web services, etc.

The main thing is that this will be a single point of capturing and processing (separating logical documents by barcodes, etc.) for all input documents regardless of the channel used for the capture.

2023 © MARGIS Inc., All rights reserved.

2.1.2 Multi-level architecture

IMiS®/Capture Service is a Windows-compatible backend service for the document capture and control of connected ISIS-compatible optical scanners. These functions make use of the Captiva PixTools technology (http://documentum.opentext.com/captiva-oem/software/pixtools-toolkit/) in its Microsoft .NET implementation. In order to provide the widest possible range of options for integration into various technologies, this service has been designed without a user interface.
Its functionality is fully utilized via its REST interface. The latter is accessible via .NET web server, build-into the service and is based on Hypertext Transfer Protocol -- HTTP/1.1 technology (https://www.w3.org/Protocols/rfc2616/rfc2616.html).

It functionally provides the service of lifecycle management of:

- Capture settings (the so-called Profile Lifecycle), which can be used for capture (a profile is the saved settings of a document capture job).
- Capture jobs (the so-called Job Lifecycle).
- An individual captured document (the so-called Document Lifecycle).

The program interface of the IMiS®/wScan product is simple, intuitive and highly customizable. It enables application developers to fully adapt the functionalities to arbitrary web applications. It has been designed as a multi-level interface with pure JavaScript technology that does not require external JavaScript frameworks (e.g. plug-ins or access to the “native” protocols NPAPI, COM, etc.).
Through integration developers will be able to descend to a sufficiently low level that will enable the necessary customizations. This will come especially in handy when the user will be unable to use e.g. constructs of the View (UI) level of the application due to technological conflicts with an internally used framework.

2023 © MARGIS Inc., All rights reserved.

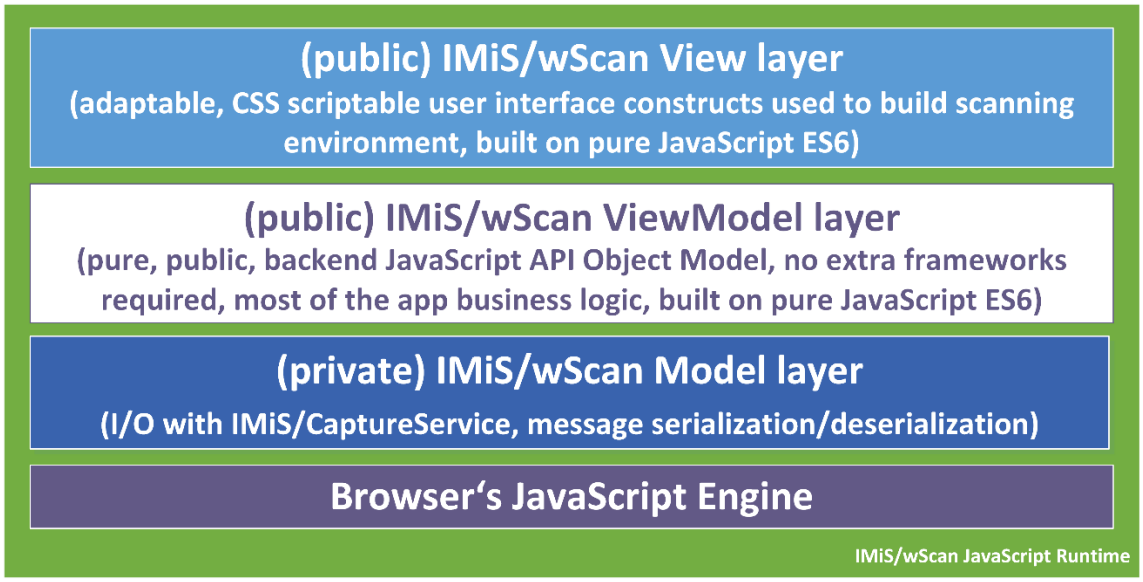

Image 3: Scheme of JavaScript program levels of IMiS®/wScan application

Browser's JavaScript Engine

The Browser's JavaScript Engine is the first and most basic level for communication with physical system components.

Model level

The Model level of the application is a private part of the application and is not intended for points of integration with applications. It connects it with IMiS®/Capture Service via an HTTP link (usually local but can be remote). It provides support to the ViewModel level. It establishes externally an internal (private) object model, in which all service messages have been deserialized and made available to higher levels in the form of JavaScript objects. It also manages the asynchronous triggering of events originating from events on the service level, which higher levels would not be able to detect and respond to. Through this level all data and commands for operations are exchanged with the service part. Even though this level has been declared private, it has an open source code that can be viewed by external developers, especially during the development of the application, when they can track errors to this level as well.

2023 © MARGIS Inc., All rights reserved.

ViewModel level

The ViewModel level of the application is the heart of the application. To higher levels (and optionally to application developers) it exposes a rich and intuitive JavaScript object model with all of the business logic that ensures the consistency of JavaScript objects and the status of the service it is communicating with.

It is an entirely backend level without any user interface constructs. Its objects enable applications to manage IMiS®/Capture Service, manage the lifecycle of captured documents, etc. Its events model enables the coordination and synchronization of events that originate from the IMiS®/wScan application or from IMiS®/Capture Service.
For capture scenarios in which a user interface is not wanted or needed, the level has been designed to be used without a user interface in a way that does not limit its range of functionalities. These are the more exceptional events enabled by the architecture.

View level

The View level of the application, as the last of the set, comprises a selection of user interface constructs that round off the application for the capture and digitization of physical documents.
It is connected with the ViewModel level. It enables application developers' easy and customizable integration of constructs into the application without possessing exact knowledge of the events and objects that enable the construct its function. The basic appearance of constructs can be customized via their object model (properties) or via CSS styles, with which the developer can customize the details of constructs to the requests and requirements of applications.
Similarly, to the other levels, this level has been built without requiring any additional JavaScript frameworks (e.g. AngularJS, etc.), which is why its integration does not cause conflicts with applications.
Constructs of the View model are conceptually and functionally independent of one another; however, for consistency purposes, the displayed information is interconnected via a network of events, which are forwarded from/to the ViewModel level. These make sure they are consistent as regards their content and status (example of displaying scanning progress).

2023 © MARGIS Inc., All rights reserved.

The events of creating new pages in a document originate from the document capture service. The creation of each page must be propagated as an event to the visual controls. The latter are connected to the ViewModel level via the events model; the ViewModel level is connected to the Model level and the latter is asynchronously connected to the Service level using WebSocket technology (otherwise it would have to execute service requests in intervals to inquire about status). Such an event is propagated from the Service level to the client's Model level; it in turn forwards it to the ViewModel level, which then refreshes all the visual constructs that are subscribed to such an event. These constructs are programmed to call such a new page from the service and add it to the displayed list of pages.

2.2 Security

The basic installation does not foresee encryption of the web traffic of the REST interface, because traffic takes place locally via the local network interface "localhost", which means that protection is generally not required. Customization of the settings is possible but requires in-depth knowledge and correct user rights settings. By default the content capture service IMiS®/Capture Service listens on the network interface "localhost" (127.0.0.1 or ::1) on port 5000/tcp, which enables local communication with the IMiS®/wScan application.
Additional user authentication is therefore not required.

Every access to the IMiS®/Capture Service must contain a special character string (a security key), which must be recorded in the IMiS®/Capture Service settings. A user with administrator authorization can enter the security key or create it via the IMiS®/wScan administrator module.

Access to the IMiS®/Capture Service is protected with the C.O.R.S. standard (https://en.wikipedia.org/wiki/Cross-origin_resource_sharing), which prevents Web browsers from accessing web domain services that are not specified in the IMiS®/Capture Service.
The Web browser will prevent access to the local IMiS®/Capture Service to any application on a different web domain, unless this domain is allowed in the IMiS®/Capture Service.
The user with administrative authorization can enter all the allowed web domains via the IMiS®/wScan administrator module.

2023 © MARGIS Inc., All rights reserved.

A user logged in a Windows operating system can access services supported by IMiS®/Capture Service via the IMiS®/wScan application.
IMiS®/Capture Service runs in the system user context (SYSTEM account), which enables the user greater access to the operating system resources than a regular user with limited rights (restricted user). Certain resources are also available to a regular user only via IMiS®/Capture Service and the IMiS®/wScan application.

All IMiS®/Capture Service settings are stored in the Windows Registry or in a file system, which cannot be accessed by a user without administrator rights. The same applies to user-defined scanning profiles. Profile settings can be modified only with the IMiS®/wScan application or with the administrator module of IMiS®/Capture Service.

A user with administrator authorization can modify them outside of these two products but has to have proper knowledge of modifying the Windows Registry. For more information see chapter Additional administrator settings.

2023 © MARGIS Inc., All rights reserved.

2.3 Functionalities

- Capture of contents and control of connected ISIS-compatible optical scanners with the option of separating contents by page count and barcode-based.
- Capture of contents via various web browsers (e.g. Google Chrome, Mozilla Firefox, Microsoft Edge). A dedicated scanning application is therefore not required.
- The IMiS®/wScan application can be integrated into existing web applications.
- The entire capture of content and its processing is executed in the IMiS®/Capture Service backend service, which is based on Microsoft .NET technology.
- IMiS®/Capture Service contains modules for the capture and processing of content (barcode recognition, capture of metadata, etc.).
- IMiS®/Capture Service has been designed modularly, with each module responsible for its own phase of the capture or processing of content. This enables easy and quick upgrading with additional modules. By updating the profile, you can create your own module execution sequence.
- The IMiS®/wScan application has been designed in a JavaScript language without the use of additional technologies that browser makers consider unsuitable or dangerous (e.g. ActiveX plug-ins or access to “native” protocols NPAPI, COM, etc.).
- IMiS®/wScan libraries enable flexibility, customization, and an easy and quick development of your own web solutions using a JavaScript language.

2.3.1 Module for the capture of content from an optical scanner

IMiS®/Capture Service can capture content from all scanners supporting the ISIS industrial standard. This standard enables a wide range of functionalities and is supported by most scanner makers.

Standard scanner settings can be modified through the IMiS®/wScan application:

- Scanner selection.
- Scanning mode.
- Scanning resolution.
- Size of scanned page.

2023 © MARGIS Inc., All rights reserved.

Additional settings that are specific to an individual scanner cannot be set through the IMiS®/wScan application, but only through the administrator module of IMiS®/Capture Service. For more information see chapter Additional administrator settings.

2.3.2 Module for saving content

IMiS®/Capture Service enables the saving of content to a file system.

Various file formats are available:

- BMP
- GIF
- TIFF
- JPEG
- PCX
- PDF/A
- PNG.

For each save format you can set the color and compression supported by the selected file format.

2.3.3 Module for barcode recognition

IMiS®/Capture Service enables the recognition of the following barcodes:

- 1D barcodes:
  Addon 2, Addon5, Australian Post, BDC Matrix, Codabar, Code-25 Datalogic, Code-25 IATA, Code-25 Industrial, Code-25 Interleaved, Code-25 Invert, Code-25 Matrix, Code-32, Code-39, Code-39 ASCII, Code-93, Code-93 ASCII, EAN-13, EAN-8, Intelligent Mail, GS1 Databar Omnidirectional, GS1 Databar Omnidirectional Stacked, GS1 Databar Expanded, GS1 Databar Expanded Stacked, GS1 Databar Limited, Postnet, Royal Post, Type-128, UCC-128, UCC-128, UPC-A, UPC-E.
- 2D barcodes:
  AZTEC, Data Matrix, PDF-417, QR Code.

2023 © MARGIS Inc., All rights reserved.

2.4 Application integration

The IMiS®/wScan application consists of three modules:

- **IMiS®/Capture Service**: a backend Windows service that executes the capture and processing of various contents.
- **imis.scan.js**: Javascript library that enables communication with IMiS®/Capture Service.
- **imis.scan.ui.js**: utility JavaScript library for displaying the already created visual components.

Each of the above-mentioned modules can be used for integration with another application.

2.4.1 Integration of IMiS®/Capture Service

Direct integration with IMiS®/Capture Service based on RESTful technology is not covered by this manual.

2.4.2 Integration of imis.scan.js library

The "imis.scan.js" library manages the exchange of data in JSON format with IMiS®/Capture Service using RESTful technology.

Through it, a developer of applications in the JavaScript language can configure profiles or capture content (e.g. scan). It enables the detection of events during scanning, the reading of jobs, documents and pages. This library also serves as a basis for making your own web solutions. It does not need any other JavaScript libraries to work.
The developer must obtain a unique security key that is recorded in the IMiS®/Capture Service. For obtaining a security key see chapter Security settings.

2023 © MARGIS Inc., All rights reserved.

```
<!DOCTYPE html>
<html>
<head>
  <title>imis.scan.js</title>
  <link rel="stylesheet" href="https://fonts.googleapis.com/css?family=Roboto" />
  <link rel="stylesheet" href="sample.css" />
</head>
<body class="sample">
  <h1>Sample</h1>
  <p>This example demonstrates reading scan profiles. Profiles are displayed in list.</p>

  <div>Profiles:</div>
  <ol id="profiles"></ol>
  <div id="error"></div>

  <script src="../imis.scan.js"></script>
  <script>
    window.addEventListener('load', function () {
      try {
        // Profiles ordered list
        var ol = document.getElementById("profiles");

        // Create a scan object
        var scan = new imis.scan.Scan();

        // Read profiles
        scan.getProfiles({
          success: function (profiles) {
            for (var i = 0; i < profiles.length; i++) {
              // Add profile to ordered list
              var li = document.createElement("li");
              li.innerHTML = profiles[i].name;
              ol.appendChild(li);
            }
          },
          error: function (error) {
            // Show error
            document.getElementById("error").innerHTML = error;
          }
        });
      } catch (e) {
        // Show error
        document.getElementById("error").innerHTML = e;
      }
    });
  </script>
</body>
</html>
```

Image 4: Example of using imis.scan.js library to read profiles

2023 © MARGIS Inc., All rights reserved.

2.4.3 Integration of imis.scan.ui.js library

The “imis.scan.ui.js” library is intended for quicker and simpler development of your own solutions. It contains some of the most commonly used visual components (for executing, stopping and continuing jobs, selecting and setting profile properties, displaying job progress, displaying the selected page and its properties, and displaying all captured pages). A web application developer can easily build these components into the application and create a useful user interface without advanced knowledge of the HTML or CSS languages. All components come with specific settings, which can be used to change their appearance. All they need to operate is the imis.scan.js library and a unique security key, which is recorded in the IMiS®/Capture Service.

For obtaining a security key see chapter Security settings.

```
<div id="imis-progress"></div>
<div class="main" id="main">
  <div id="thumbnails">Thumbnails</div>
</div>
<script src="imis.scan.js"></script>
<script src="imis.scan.ui.js"></script>
<script>
window.addEventListener('load', function () {

  // Set scan version to title attribute
  document.getElementById("title").setAttribute("title", imis.scan.ui.version);

  try {
    const scan = new imis.scan.ui.Scan({
      //url: "http://example.com",
      thumbnails: new imis.scan.ui.Thumbnails({
        id: "thumbnails",
        //darkMode: false,
        orientation: "horizontal",
        thumbnail: {
          height: 200, // thumbnail height
          title: false
        },
        gallery: true,
        contextMenu: {
          enabled: false
        }
      }),
```

Image 5: Example of using imis.scan.ui.js library to set the appearance of the Thumbnails component

2023 © MARGIS Inc., All rights reserved.

2.5 Versions

Product version labeling is based on a scheme that includes the following:

- Three separate numerical identifiers (MAJOR, MINOR, RELEASE).
- An identifier of the 32-bit or 64-bit installation platform (PLATFORM).

The following is an example of a scheme:

IMiS.wScan.MAJOR.MINOR.RELEASE.PLATFORM.msi

The example of IMiS®/wScan installation package name:

IMiS.wScan.1.9.2310.x64.msi

The scheme consists of the name of the IMiS®/wScan module and the following elements:

- MAJOR: The identifier in the MAJOR position signifies the main/major version of the product. It is arbitrary and changes with regards to the volume of the changes and functionalities introduced. The identifier in this position changes the least.
  In the event that it is changed, it signifies a major difference in the product compared to the previously issued version (with a lower MAJOR version).
  This identifier has a range of values from 1-n; it is continuous and can only increase.
- MINOR: This identifier indicates a minor version of the product. It changes more frequently than the main version in terms of changes to the system, features and fixes. A change in the minor version represents smaller changes and fixes in the framework of the same product generation (indicated by the main or major version). The range of values is from 1 to n. This number is not continuous. It resets to its base value (1) with each new MAJOR version.
- RELEASE: This identifier represents the time component of the product release in accordance with the YYMM scheme.
  MM indicates the month of the release (range of values from 01 to 12), and YY indicates the last two digits of the year.

  Example: The RELEASE identifier for a product released in October of 2023 will read 2310.

2023 © MARGIS Inc., All rights reserved.

- PLATFORM: Indicates the operating systems on which this application can be used. The 32-bit version can run on a 32-bit Windows operating system, and on a 64-bit one. The 64-bit version can run only on a 64-bit Windows operating system.

2.6 New functionalities in this version

We have implemented the following new functionalities and improvements to the previous certified version 1.7.2110 of the IMiS®/wScan module:

New functionalities – IMiS®/wScan:

- Option of adding highlights.
- Job printing component.
- Component for changing the scanning resolution.
- "FitToWidth" option for defining the default image width in the "imis.scan.ui.ImageScroll" component.
- "ZoomFactor" option for increasing the zoom factor in the "imis.scan.ui.ImageScroll" and "imis.scan.ui.ImageView" components.

New functionalities – IMiS®/Capture Service:

- Supports multiple web applications with different API keys.
- Supports highlight annotations.

2023 © MARGIS Inc., All rights reserved.

Improvements - IMiS®/wScan:

- Improved component for cropping pages and adding redactions.
- Enhanced display of actions in the popup window for changing the temporary profile settings.
- Enhanced display of icons in the SVG format.

Improvements – IMiS®/Capture Service:

- Checking for changes to the scanner settings in the administration module.
- Optimized driver upload for a scanner using Kofax VRS.
- An individual rotating log file increased to 10 MB.

2023 © MARGIS Inc., All rights reserved.

3 SCANNING DOCUMENTS

3.1 Methods of Scanning Documents

Scanning is the converting of documents in paper form into a digital data file for the purpose of viewing and saving to storage media. Paper documents can be captured using specialized scanners, multifunction devices or digital cameras.
A scanner is still the most suitable for capturing larger quantities of paper documents.
It can be connected locally or via a computer network.
Based on our experience in scanning, it is recommended that you connect the scanner locally.

3.2 Connecting a Scanner

In order to successfully convert paper documents into digital form, you need software in addition to hardware (scanner and computer).
In order for various scanners and the IMiS®/Scan software product to successfully scan paper documents, you must take into account the preset protocols or standards for connecting and transferring digital contents. The best-known standards are ISIS, TWAIN and WIA.

ISIS (**I**mage and **S**canner **I**nterface **S**pecification)

ISIS is an industrial scanner interface. It was developed by Pixel Translations in 1990. Today this standard is owned by the OpenText Corporation, which sees to its development and use. Any scanner manufacturer may develop its own ISIS drivers, but must obtain a certificate of compatibility with the ISIS standard from OpenText Corp. In doing so, it must pay a license fee to OpenText. In general, all production scanners use ISIS drivers.

TWAIN (**T**ool **W**ithout **A**n **I**mportant **N**ame)

TWAIN is a free scanning application programming interface. It was the first standard to enable the connecting of software to various scanners. The development of this standard is overseen by a consortium of large scanner manufacturers called TWAIN Working Group.

2023 © MARGIS Inc., All rights reserved.

A downside to this standard is that the user interface is integrated into the TWAIN driver, which makes it harder to implement the driver. Another problem is the long waiting periods for innovations. Generally speaking, low-cost scanners use only the TWAIN driver.

WIA (**W**indows **I**maging **A**rchitecture)

WIA is a driver and API, developed by Microsoft. It only works on the Windows operating system. The entire development of the driver is tied to Microsoft's objectives and does not really take into the account the demands of scanner developers.

3.3 Scanning Resolution and Quality

Scanning works on the principle of detecting light reflected off paper or picture.
The entire surface of the sheet is divided into small pixels. Each pixel is determined by its position, brightness level and color (color scanning). Resolution is the number of pixels per unit of length.
It is usually written down as number of dots per inch (**dpi**).
A document which is divided into more pixels has better resolution and consequently its conversion into digital form is of higher quality. The scanning resolution is conditioned by the scanner technology. Scanners support a resolution from 100 to 1600 dpi. For the needs of archiving a non-image document, a resolution of **300 dpi** is sufficient.
Scanning with a resolution of under 200 dpi may result in unreadable conversion of archived documents. Scanning with more than 300 dpi is sensible if the original text on paper is of poorer quality or when you wish to convert an image of a document into digital form with very high resolution.

3.4 File Compression and Size

When scanning, a dot on paper is being copied to a digital memory unit.
In the case of black-and-white copying this unit is 1 bit, while in the case of gray or color copying it can be a byte or bytes. The quantity of color or level of brightness which is captured with a scanner is called color depth.

2023 © MARGIS Inc., All rights reserved.

Greater color depth gives a broader range of various colors and consequently means greater memory usage. The first row in the table below gives an example of how color scanning can use up to 15 MB of memory.

Compression/Color	Black&White (1 bit)	Grayscale (8 bit)	Color (24 bit)
without	475 KB	5 MB	15 MB
CCITT G3	85 KB	x	x
CCITT G4 T6	45 KB	x	x
JBIG	36 KB	x	x
JBIG 2bit	x	84 KB	x
JBIG 3bit	x	165 KB	x
JBIG 4bit	x	420 KB	x
Packed bits	109 KB	5 MB	15 MB
LZW	75 KB	3,2 MB	x
ZIP	56 KB	3 MB	9 MB
Wang JPEG	x	315 KB	363 KB
Sequential JPEG	x	315 KB	360 KB
Progressive JPEG	x	310 KB	334 KB

Such quantities of obtained digital data may cause problems with the data transfer speed from the scanner to a computer or a lack of space on the storage medium.
For this reason, digital data is compressed. The above table shows which compression methods are suitable for each color depth. Generally speaking, compression methods are divided into "lossless" and "lossy".

Lossless Data Compression

The algorithms of methods in this group look for repeated strings in the digital document and shorten them without losing data. The quality of the scanned document prior to compression remains the same after the data is decompressed. The characteristic methods of this group are G4 T6, LZW and ZIP compressions.

2023 © MARGIS Inc., All rights reserved.

Lossy Data Compression

A characteristic of the algorithm in this group is the removal of unimportant data from scanned documents and assigning similar pixels in the document to a common denominator (e.g.: it changes various shades of a blue sky into a single color for the entire range of blue shades).

Thus, the original digital image is modified and the display quality is reduced.
The decompressed image no longer matches the original. A representative of this group is JPEG compression. Repeated compressing and decompressing of an image may significantly affect the quality of a scanned document.
Black-and-white scanning is sufficient for the majority of documents, which is why we do not recommend scanning in gray or color. The majority of scanners intended for document capture uses advanced methods and filters for graphic processing, which ensures the optimum quality of scanned documents.
We recommend that you use lossless compressions for scanning text documents, and compressions from the JPEG family for image documents. For image documents which must preserve their source file you can likewise use the lossless compression method.

3.5 Recording of Scanned Documents

The recording of scanned documents to a storage medium is executed in preset file formats. There are different types of file formats; however, not all are suitable for a specific type of document.
In general, file formats are divided into single-page and multi-page. In addition to a digital record of the document, all formats store additional information about the document in their structure.

2023 © MARGIS Inc., All rights reserved.

Single-page formats

are capable of recording only one page to a file. They are used for saving images and are suitable for the further processing of scanned images. The best-known formats in this group are BMP, PNG and JPEG.

BMP (**B**it**m**a**p** File Format)

This file format enables the saving of digital images of random height and width in various resolutions and color depths. This format was developed by Microsoft to be used in its applications and the Windows operating system. The fact is that this format is well documented and patent-free, which is why it is present in all image processing programs. A flaw of this format is that it does not support data compression and therefore the files take up a lot of space on the storage medium. This format is not recommended for saving scanned documents.

JPEG (**J**oint **P**hotographic **E**xpert **G**roup)

This format was created in 1992 as an ISO standard for describing the procedure of compressing an image into a byte stream. A characteristic of this format is that the compression of the digital image affects the image display quality. The greater the compression level, the smaller the file and, consequently, poorer image quality when decompressed. The JPEG format gave rise to other versions of file formats for image recording: JBIG, JPEG 2000, sequential JPEG and progressive JPEG.

This format is recommended for saving digitized photographs and images which contain many shades of color.
Scanned text documents which contain sharp contrasts between adjacent pixels are not suitable for being saved in this format.

2023 © MARGIS Inc., All rights reserved.

Multi-page formats

Multi-page formats enable the recording of multiple pages into a single file. The best-known representatives are the TIFF and PDF formats.

TIFF (**T**agged **I**mage **F**ile **F**ormat)

This format began to be developed in 1986 for the purpose of standardizing the saving of scanned documents from various scanners. It aimed to provide high functionality of various scanners and simplicity in exchanging scanned documents among various applications. It was designed for use in "desktop publishing". The format's structure enables the recording of various metadata about the document, in addition to the compressed digital document.
The structure of metadata can be preset and known to many applications or merely to those which enable the displaying of said information. The entire file is dissected into identifiers, with each identifier being either a digitally converted document or information about the document (compression method, document size, resolution, color depth etc.). This format is present in all applications that are used for capture.
Since 2009 the format has been owned by Adobe System, which sees to its upgrades and modifications.

PDF (**P**ortable **D**ocument **F**ormat)

This format began to be developed in 1993 by Adobe System. Its purpose was to create a file format that would be portable between all file systems and applications. It was to represent a universal format for saving all kinds of computer-generated content. It was to be used for saving text products, images in vector or raster graphics and the other audio-video content.
The PDF format enables the saving of a document in its entirety.
Since its beginnings, the format has seen many upgrades and modifications. One version of the PDF format has become an ISO standard for the long-term archiving of computer-generated content (PDF/A). The PDF/A format guarantees the same content and appearance in various operating systems and applications.

2023 © MARGIS Inc., All rights reserved.

4 SYSTEM REQUIREMENTS

For successful installation and execution, the IMiS®/wScan application has the following hardware and software requirements.

4.1 Hardware

Practically all computers currently available on the market meet the hardware requirements for running IMiS®/wScan.
Minimum and recommended requirements are listed below.

4.1.1 Minimum requirements

Minimum requirements for IMiS®/wScan:

- Intel Core i3 6th Gen processor
- 2 GB RAM
- 200 MB of unused hard disk space
- TCP/IP network access (IPv4 or IPv6).

4.1.2 Recommended requirements

Recommended requirements for IMiS®/wScan:

- Intel Core i5 6th Gen processor or faster
- 4 GB RAM or more
- 1 GB of unused hard disk space
- TCP/IP network access (IPv4 or IPv6).

2023 © MARGIS Inc., All rights reserved.

4.2 Software

Requirements for IMiS®/wScan:

- .NET Framework 4.6.2 or later;
- Javascript ECMAScript 6;
- Browsers with enabled WebSocket technology and ECMAScript6 standard support:
  - Google Chrome: minimum version 110 or later;
  - Mozilla Firefox: minimum version 115 or later;
  - Microsoft Edge: minimum version 110 or later.
- Supported operating systems:
  - Windows 11 22H2 or later;
  - Windows 10 22H2 or later.

2023 © MARGIS Inc., All rights reserved.

5 PRODUCT MANAGEMENT

The IMiS®/wScan application is managed by administrators and/or application developers. Management includes installation, startup, closing, upgrading and uninstallation.

5.1 Installation

The installation can be performed in an environment that meets at least the minimum installation requirements. IMiS®/wScan installation can be performed with a setup wizard as administrative installation or "silent" installation. In any case, the displayed notifications and dialog boxes are in English.

Warning:

Prior to installation stop the IMiS®/Scan application, because IMiS®/wScan does not work properly if the IMiS®/Scan application is running.

Warning:

Installation of a 64-bit version of IMiS®/wScan will be unsuccessful if the 32-bit version is already installed on the workstation.

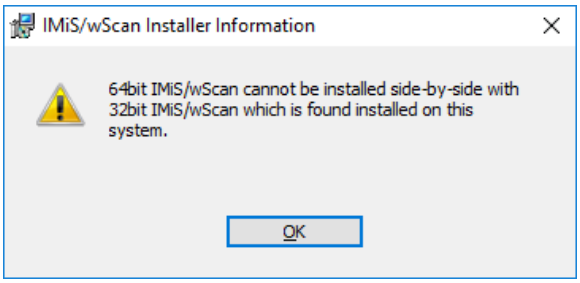

Image 6: Notification of prohibited simultaneous installation of the 64-bit and 32-bit version of IMiS®/wScan

The same applies vice-versa. If the 64-bit version of IMiS®/wScan is already installed on the workstation, installation of a 32-bit version will be unsuccessful.

2023 © MARGIS Inc., All rights reserved.

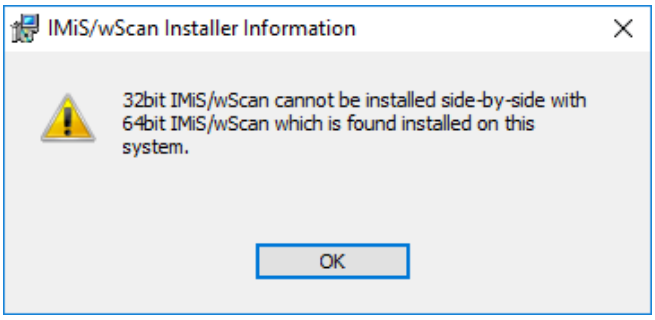

Image 7: Notification of prohibited simultaneous installation of the 32-bit and 64-bit version of IMiS®/wScan

Warning:

Installation of the IMiS®/wScan application on a workstation is not possible if .NET Framework 4.5 has not been installed.

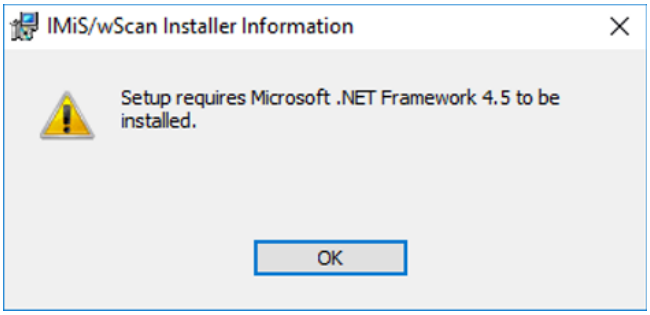

Image 8: Notification of required installation of .NET Framework 4.5

2023 © MARGIS Inc., All rights reserved.

5.1.1 Installation with the Wizard

The user interface of the installation package guides the administrator through the installation procedure.

The administrator installs the IMiS®/wScan application on a workstation in a Windows environment with one or several optical scanners physically connected. This application includes the IMiS®/Capture Service web service and the relevant libraries.

Example of the name of an installation package:

IMiS.wScan.1.9.2310.x64.msi

Installation begins with launching the installation package from the file system. A dialog box appears, informing the administrator that the installation package is preparing for installation.

In the next step read the terms of the license agreement carefully. If you agree with them, select

"I accept the terms in the license agreement" and thus fully accept the license terms.

If you do not agree with the license terms, select "I do not accept terms in the license agreement" and by clicking on the "Cancel" button cancel the installation procedure.

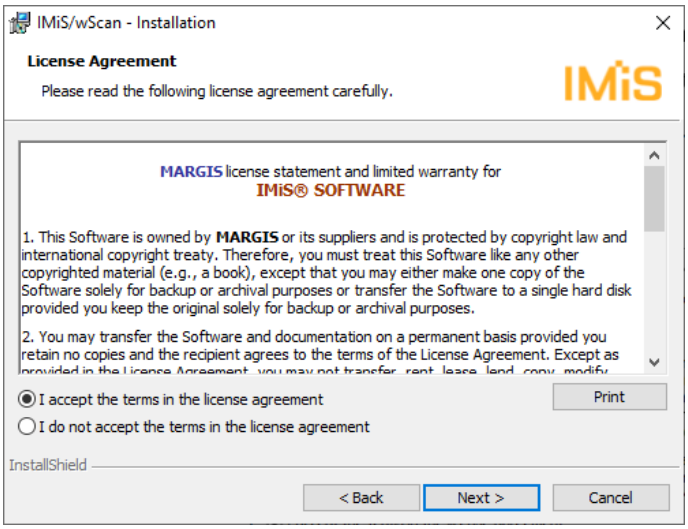

Image 9: Viewing and accepting the license terms

The installation procedure is continued by entering the username into the input field “User Name” and the organization into the input field “Organization”. Select whether the application will be installed only for the current user ("Only for me") or for all users on this computer ("Anyone who uses this computer").

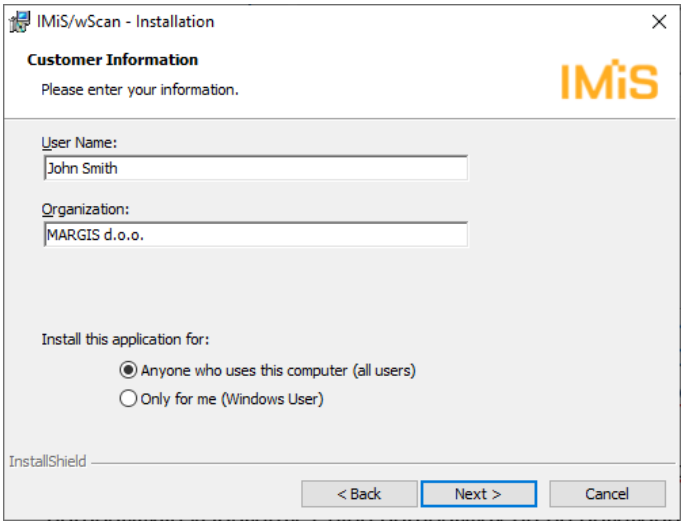

Image 10: Entering data on application user

In the next step select between typical installation ("Typical"), full installation ("Complete") or user-customized installation ("Custom").

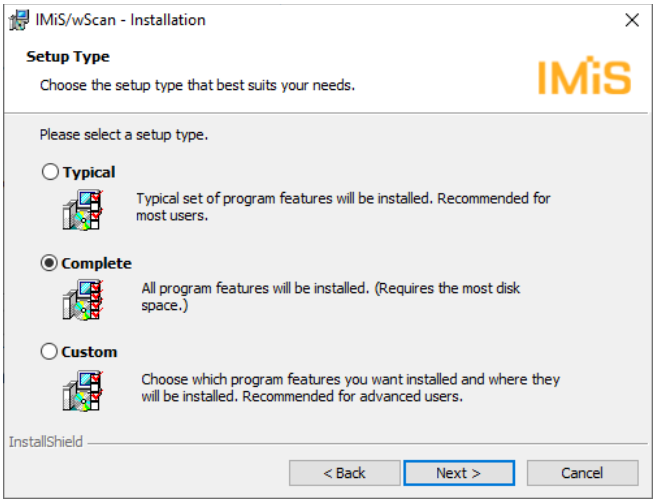

Image 11: Selecting between typical, full or user-customized installation

For all installation types the administrator specifies which shortcuts will be created and which options will be activated during the installation procedure.

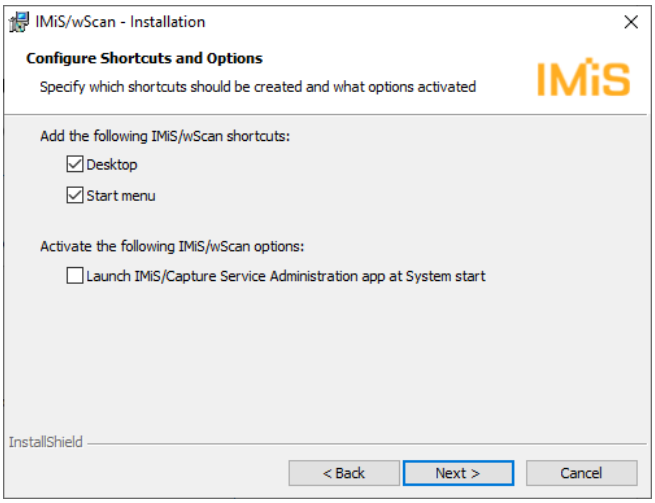

Image 12: Selecting which shortcuts will be created and which options will be activated during installation

If the administrator ticks the choice “Launch IMiS®/Capture Service Administration app at System start”, the administration module will launch when the workstation starts.

If you tick the choice “Desktop”, shortcuts to the IMiS®/wScan start page and the IMiS®/Capture Service administration module will be installed on the desktop.

2023 © MARGIS Inc., All rights reserved.

If you tick the choice “Start menu”, a shortcut to the IMiS®/wScan start page and the IMiS®/Capture Service administration module will be added to the Start menu.

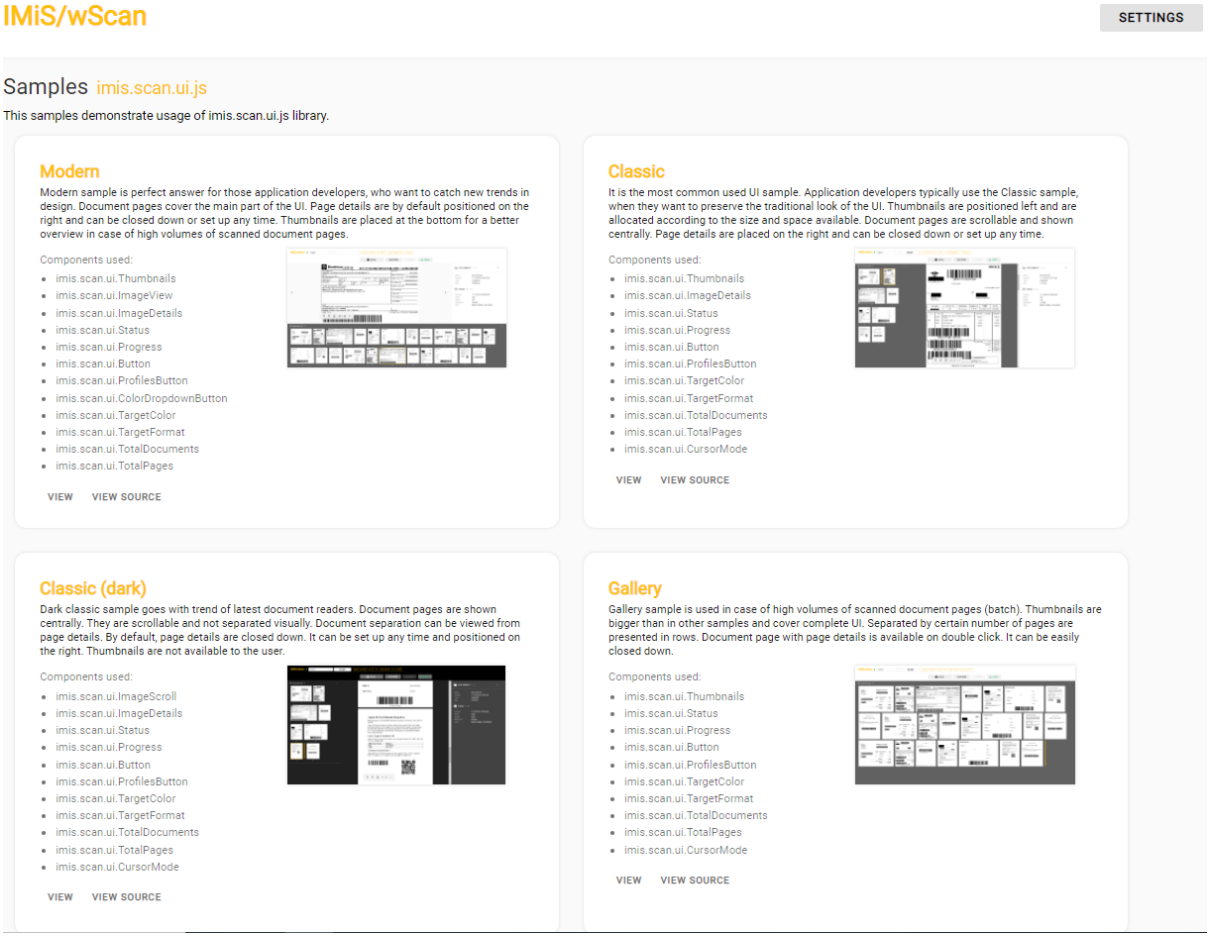

Image 13: Displaying the result of selecting “Start menu”

Typical installation, which is recommended for most users, begins by transferring predefined files to the file system. The administrator confirms the selected installation setting and starts the installation procedure by clicking on the “Install” button.

2023 © MARGIS Inc., All rights reserved.

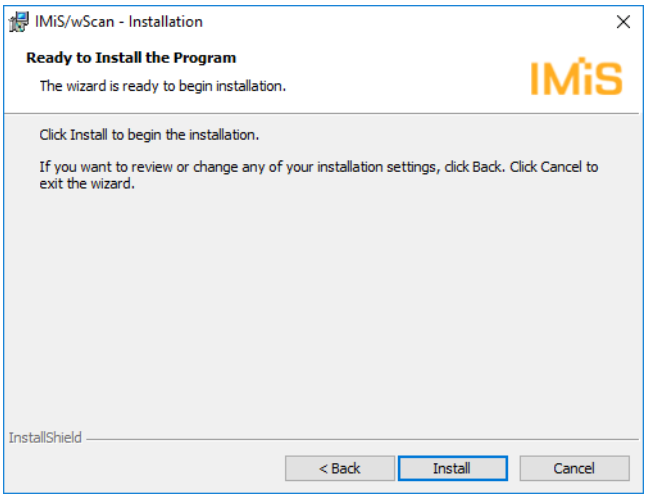

Image 14: Starting the installation procedure

The installation procedure for the IMiS®/wScan software product begins; the progress bar shows information about the transfer of files to the appropriate locations. Installation may take a few tens of seconds, depending on the version of the installation package and computer speed.

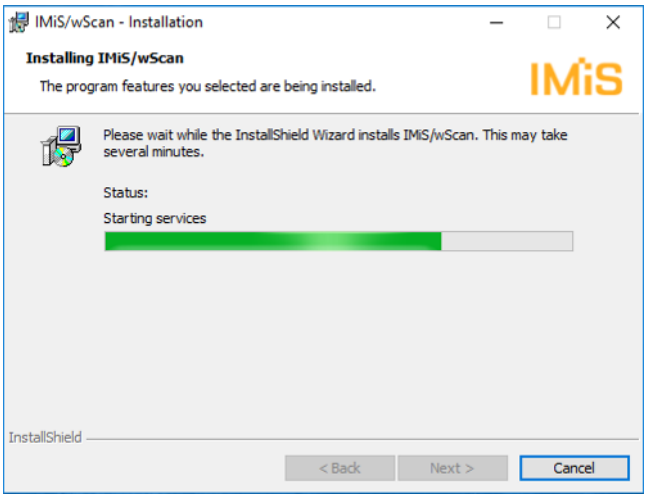

Image 15: Displaying the progress bar during the installation procedure

Installation is complete when the last display box appears, which you close by clicking on the "Finish" button.

2023 © MARGIS Inc., All rights reserved.

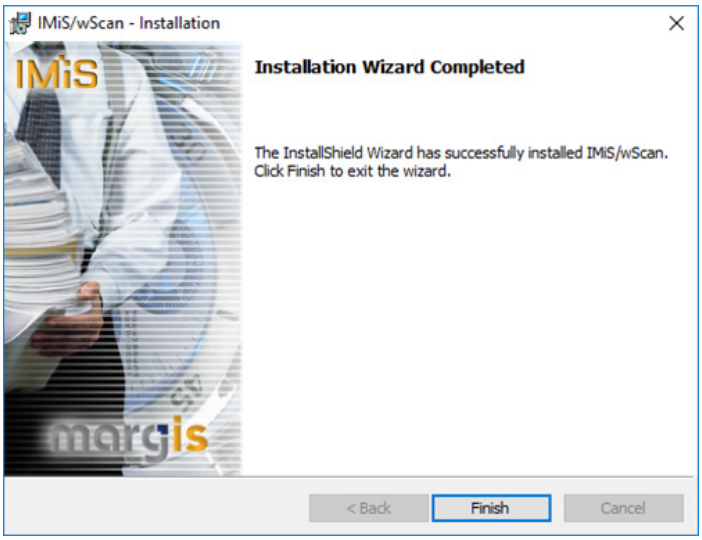

Image 16: Notification of completing the installation procedure

The same procedure as for typical installation is carried out during full installation.
Full installation will install all of the features from the installation package in the file system, which is why it requires the most disk space.

User-customized installation ("Custom") will install only specific features in the file system.
It is intended for advanced users.

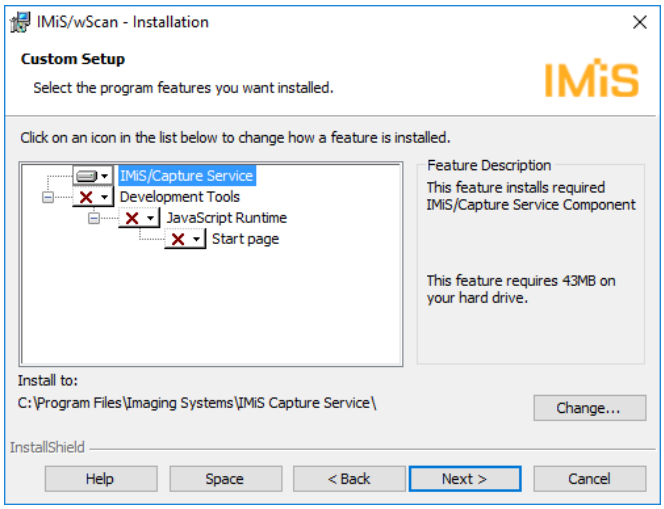

Image 17: Selecting the features of application installation

The administrator confirms the selected installation setting and starts the installation procedure by clicking on the "Install" button. Further steps are the same as for typical and full installation.

5.1.2 Silent installation

The IMiS®/wScan application can also be installed without user control. Installation is performed silently, without displaying the user interface. The installation is executed via the *msiexec.exe* utility program. This tool is a part of the Microsoft installation product and is used for performing various types of maintenance in applications that are installed in the Windows operating system.

For a full list of the supported functions of the *msiexec.exe* program, turn to the Microsoft article collection:
http://msdn.microsoft.com/en-us/library/windows/desktop/aa367449(v=vs.85).aspx
The utility program is executed from the command bar.

For a list of all parameters, turn to the Microsoft website:
http://msdn.microsoft.com/en-us/library/windows/desktop/aa367988(v=vs.85).aspx.
Installation may take a few tens of seconds, depending on computer speed.

Example of a command bar for silent installation:
C:\Windows\system32\msiexec.exe /i IMiS.wScan.1.9.2310.x64.msi /qn

Below is a command bar for the silent installation of IMiS®/wScan:

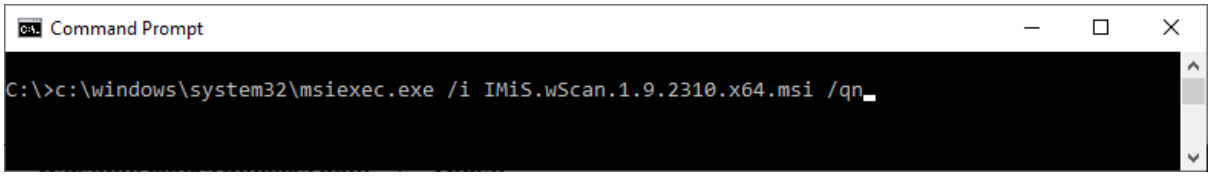

Image 18: Example of a command bar for silent installation in previous version

The table below lists the various silent installation modes:

Command bar parameters	Description
/q, /qn	No user interface.
/qn+	No user interface with a modal window at the end of installation.
/qb	Basic user interface with a simple progress bar. The "/gb!" parameter is used to hide the "Cancel" button.
/qr	Simplified user interface without a modal window at the end of installation.
/qf	A full user interface with all the display boxes, a progress bar and error message at the end of installation.

Table 1: Silent installation settings

Before launching the installation of the IMiS®/wScan application, you can specify various parameters that are specific to this installation. Add them to the end of the command bar using the syntax:

c:\windows\system32\msiexec.exe /i IMiS.wScan.1.9.2310.x64.msi /qn PARAMETER=VALUE

The table below describes the supported command bar parameters:

Parameter	Valid values	Description
INSTALLDIR	<directory name>	This parameter contains the default destination folder for installation files (default value = "%PROGRAMFILES%\Margis \IMiS Capture Service\").
USERNAME	<username>	This parameter contains the username of the user performing the installation (default value is taken from system settings).
COMPANYNAME	<company name>	This parameter adds the company name to the installation (default value is taken from system settings).

2023 © MARGIS Inc., All rights reserved.

Parameter	Valid values	Description
SHORTCUT_START	1 / 0	This parameter is used to inform the installation process to create a shortcut in the "Programs" menu (Default value is 1).
SHORTCUT_DESKTOP	1 / 0	This parameter is used to inform the installation process to create a desktop shortcut (Default value is 1).
LAUNCH_ADMIN_ON_START	1 / 0	This parameter is used to inform the installation process to create the necessary registry records and enable automatic launch of the IMiS®/Capture Service administration module when the workstation starts. (Default value is 1)
ADDLOCAL	ALL	Enables the silent installation of all components of the installation package, which is equivalent to selecting “Complete” in the installation with the Wizard.

Table 2: Supported command bar parameters

5.2 Startup and closing

IMiS®/Capture Service launches automatically when the workstation connected to the optical scanner starts.

Starting and stopping IMiS®/Capture Service is also possible manually by double-clicking on the IMiS®/wScan administration module. After startup the IMiS® icon appears at the bottom of the desktop.

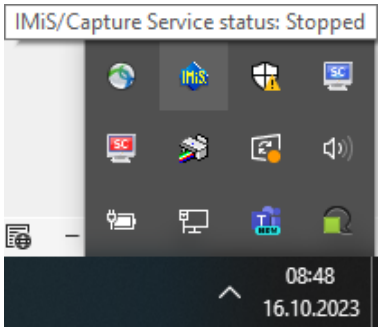

Image 19: Displaying the current status of IMiS®/Capture Service: stopped

By right-clicking on the IMiS® icon, the menu appears. By selecting the option “Start service”, the administrator starts IMiS®/Capture Service.

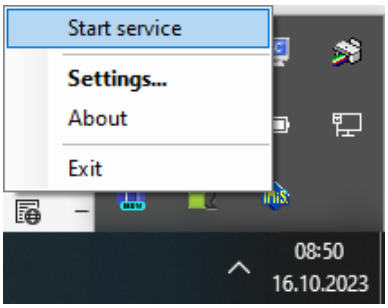

Image 20: Selecting the option to start IMiS®/Capture Service

It takes a few seconds for the service to start, as it initializes properly and checks the suitability of the scanner driver. The current status of the service is visible if you move the mouse over the IMiS® icon.

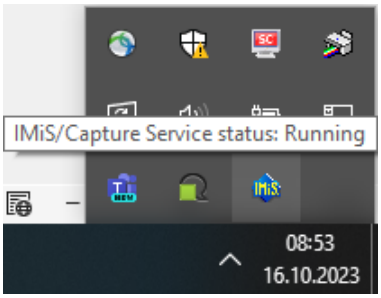

Image 21: Displaying the status of IMiS®/Capture Service: running

The service is stopped by selecting the option “Stop service” in the menu of the IMiS® icon.

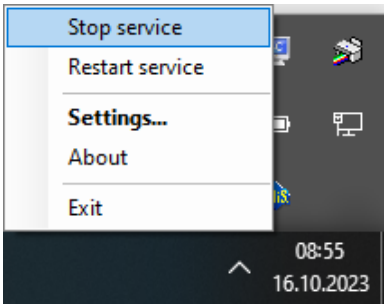

Image 22: Selecting the option to stop IMiS®/Capture Service

2023 © MARGIS Inc., All rights reserved.

If you (the administrator) wish to restart the service, select the option “Restart service” in the menu of the IMiS® icon.

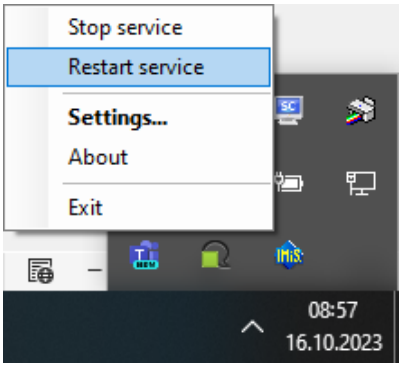

Image 23: Selecting the option to restart IMiS®/Capture Service

<u>Warning:</u> The administrator must refresh IMiS®/wScan application in browser (MS Edge, Mozilla Firefox, Google Chrome, ...), after restarting IMiS®/Capture Service.

5.3 Additional settings

Not all IMiS®/wScan application settings can be implemented via the “imis.scan.js” JavaScript library. The administrator can access additional settings by double-clicking on the IMiS®/wScan administration module. After startup the IMiS® icon appears at the bottom of the desktop. By right-clicking on the IMiS® icon, the menu appears.

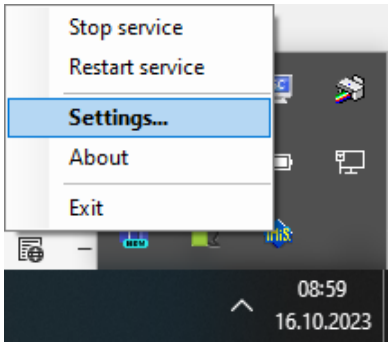

Image 24: Selecting the option to show additional settings

By selecting the option “Settings”, the configuration dialog box appears.

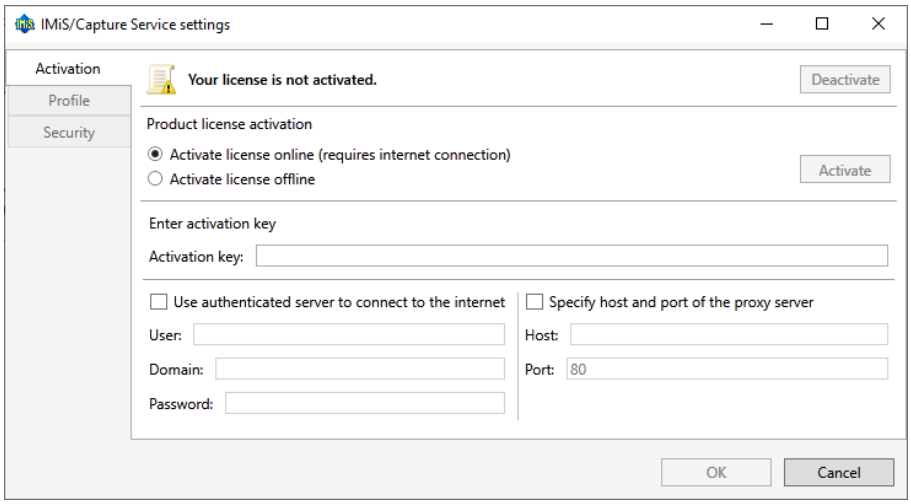

Image 25: Dialog box for setting additional settings

Tabs for product activation, additional settings for profiles and security are available. After choosing each of the tabs, additional settings are displayed. After the settings are set, the user with administrator authorization clicks the “OK” button.
If the profile has been modified and IMiS®/Capture Service has not been started, the additional settings will be saved. They will be used at the next startup of IMiS®/Capture Service.
If IMiS®/Capture Service has been started, a restart of IMiS®/Capture Service is performed.

If the administrator selects the “Cancel” button in the configuration dialog box, the box closes without saving the changes made to the profile.

<u>Warning:</u> The administrator must refresh IMiS®/wScan application in browser (MS Edge, Mozilla Firefox, Google Chrome, ...), after restarting IMiS®/Capture Service.

2023 © MARGIS Inc., All rights reserved.

5.3.1 Product activation

From version 1.6.2010 onward, the IMiS®/Capture Service product requires license activation. When starting the IMiS®/Capture Service Administration application for the first time, the "Activation" tab opens in the "IMiS®/Capture Service settings" dialog, which provides two license activation modes:

- Online license activation: activation takes place online.
- Offline license activation: activation takes place offline via an alternative path.

5.3.1.1 Activating the license online

When activating the license online, the administrator selects the option "Activate license online". Online activation requires an internet connection. The manufacturer provides the administrator with a license activation key, which the administrator enters in the "Activation key" field. By clicking on the "Activate" button, the application connects to the web service. In the event of successful activation, the application saves the activation data on the workstation and enables using the product without further license verification.

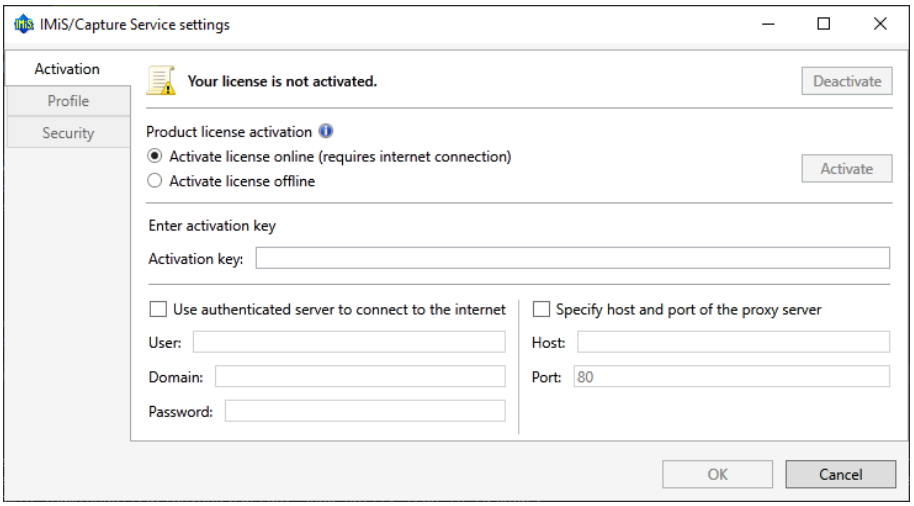

Image 26: Dialog box for the online activation of the IMiS®/Capture Service license

If the online activation requires the use of a proxy server, the administrator can enter the credentials for an authenticated web connection and the data on the proxy server's host.

2023 © MARGIS Inc., All rights reserved.

Activating the option “Use authenticated server to connect to the internet” enables entering the following:

- User: name of the authenticated user.
- Domain: domain of the authenticated server.
- Password: password of the authenticated user.

Activating the option “Specify host and port of the proxy server” enables entering the following:

- Host: host of the proxy server.
- Port: port of the proxy server.

5.3.1.2 Alternative license activation

If connecting to the internet is not possible, IMiS®/Capture Service can be activated via an alternative path. In that case, the administrator selects the option “Activate license offline”. The administrator must provide the manufacturer with the computer identifier, which the administrator obtains from the “Computer identifier” field. Based on that, the manufacturer activates the license on behalf of the user and provides the user with the computer key, which the administrator then enters in the “Computer key” field. By clicking on the “Activate” button, the application saves the activation data on the computer and activates the license.

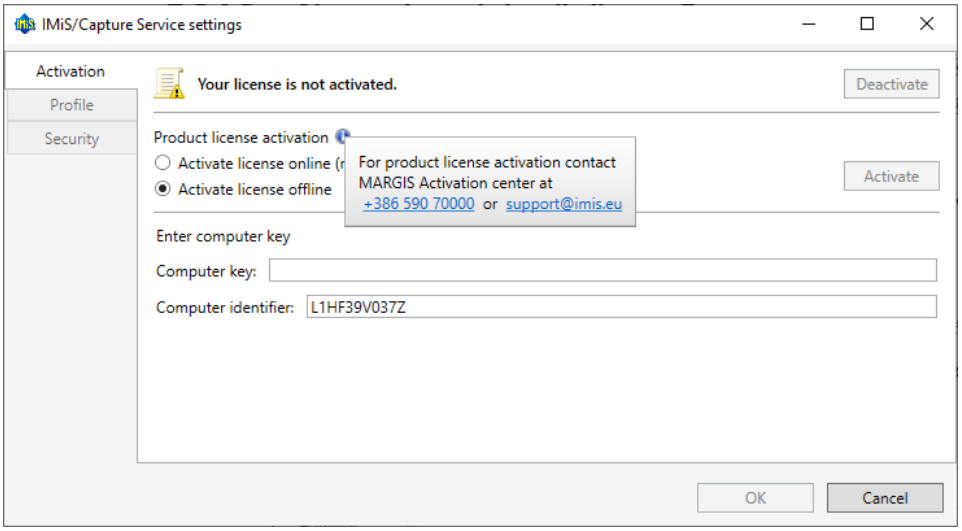

Image 27: Dialog box for the offline activation of the IMiS®/Capture Service license

Note: The manufacturer's contact data is accessible via the information icon to the right of the caption "Product license activation". By clicking on the displayed telephone number, the default call program opens with the preset number; by clicking on the displayed email address, the default email application opens with the preset email address of the recipient.

After successful activation, the top of the "Activation" tab shows the activated license record. License information is accessible via the information icon to the right of the record.
It shows the properties of license validity ("Validity") and license features ("Features").

The license validity record can be one of the following:

- Never expires: the activated license is permanent.
- Expires on <date>: valid trial license.
- Expired!: invalid trial license.

5.3.1.3 Range of features

Besides the basic document scanning feature, the features of the license in IMiS®/Capture Service from version 1.6.2010 onward also enable batch scanning features (see chapter Batch scanning features).

The IMiS/Capture Service product license has the following possible features:

- BASIC: document scanning.
- DOCSEP: document separation.
- BAR1DPIX ali BAR1DST: barcode 1D recognition.
- BAR2DPIX ali BAR2DST: barcode 2D recognition.

Note: The features BAR1DPIX and BAR2DPIX, and BAR1DST and BAR2DST are linked to two different barcode detection modules that cannot be used simultaneously.
For more information on obtaining a license key for expanding the set of features with batch scanning, send an email to info@imis.si.

2023 © MARGIS Inc., All rights reserved.

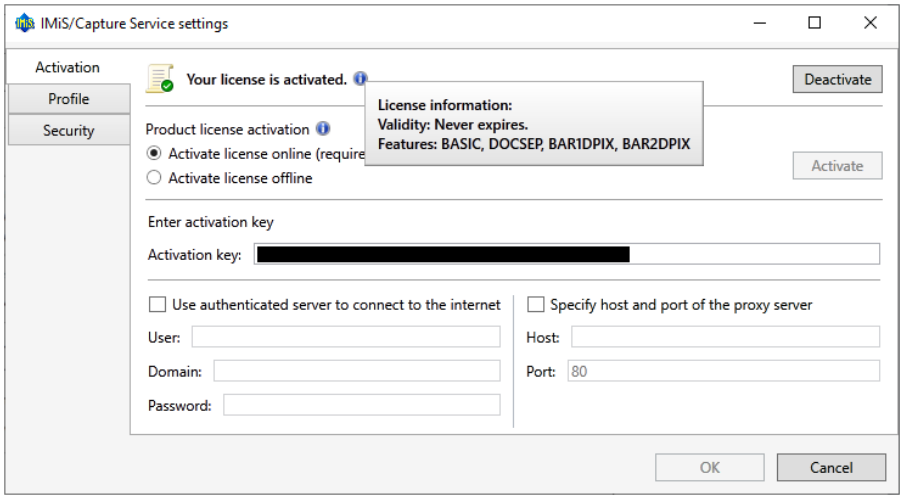

Image 28: Example of an activated IMiS®/Capture Service license

The IMiS®/Capture Service Administration application also enables product deactivation, which is necessary if the hardware is modified, if the computer is replaced, or if the activation key is replaced with one that enables a greater number of product features.

The administrator deactivates the product by clicking on the “Deactivate” button. If an online activation was performed, the application connects to the web service during deactivation, which checks for the existence of an activation key and a computer key, and removes the activation entry; afterwards the application also removes the activation data stored on the computer. If an offline activation was performed, the manufacturer must be given the computer key in order to remove the activation entry, as the deactivation removes only the activation data on the computer.

5.3.2 Additional profile settings

For additional profile settings, the user with administrator authorization selects the “Profiles” tab. Settings are displayed on the right side. In the “Profiles” dropdown menu all the specified profiles are available. Besides the name of the profile, the scanner model is also displayed.

2023 © MARGIS Inc., All rights reserved.

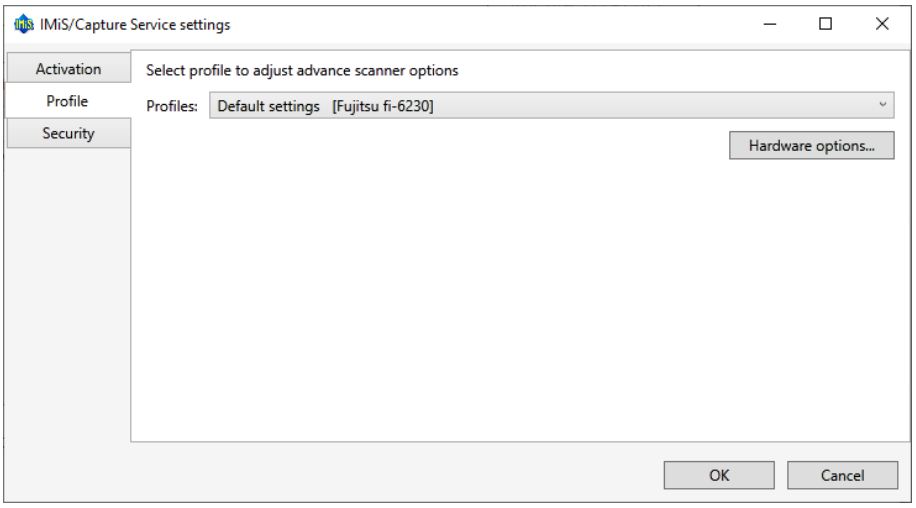

Image 29: Dialog box for profile settings and security settings

The administrator selects the profile in which to set additional scanner settings (e.g. remove empty pages). After selecting the profile, click on the button “Hardware options...”. The scanner’s configuration dialog box appears.

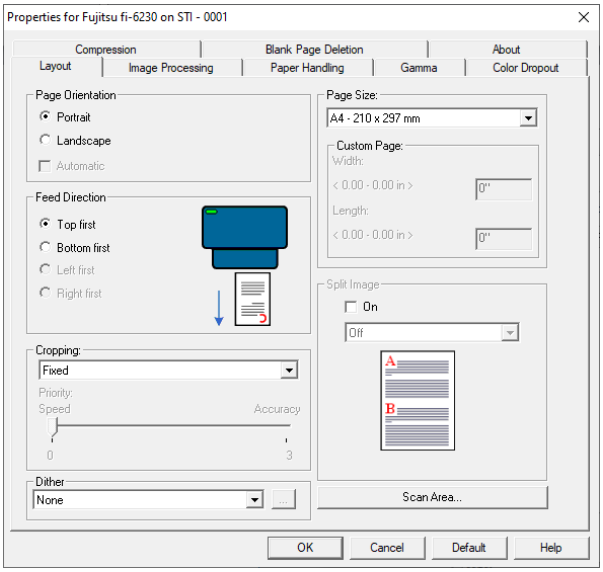

Image 30: Configuration dialog box of Fujitsu driver

Note: Scanner makers use different configuration dialog boxes.

After finishing the settings, click on the “OK” button.

2023 © MARGIS Inc., All rights reserved.

5.3.3 Security settings

A user with administrator authorization selects the “Security” tab.

Security settings that include a field for entering or obtaining a security key and entering the allowed indirect web domains is displayed.

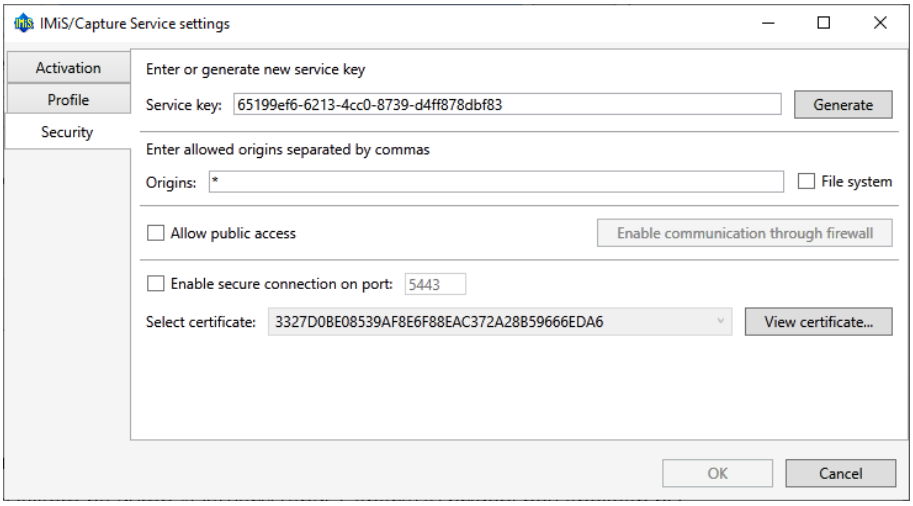

Image 31: Dialog box for security settings

A user with administrator authorization can enter any character string in the “Service key” field or click the “Generate” button to obtain a new unique character string.

The user can enter multiple security keys, thus enabling multiple web applications to use the service. The security keys must be separated by commas.

The developer of the web application enters the security key in his application by forwarding this character string to the **imis.scan.js** javascript library**.**

For more information see chapter Integration of imis.scan.js library.

The IMiS®/Capture Service will reject all REST requests by web applications that don’t contain the same security key as entered in the »Service key« field.

A user with administrator authorization enters all web domains from which the web application can execute REST requests to the IMiS®/Capture Service in the “Origins” field.

Individually entered web domains are separated by commas. If the “Origins” field is empty, only access via the “localhost” local network interface is allowed.

2023 © MARGIS Inc., All rights reserved.

For access from all web domains, the administrator must enter the " * " sign (asterisk).
If the web application is executed directly from the file system, a user with administrator authorization must tick the "File system" field.

<u>Warning:</u> The administrator must refresh IMiS®/wScan application in browser (MS Edge, Mozilla Firefox, Google Chrome, ...), after restarting IMiS®/Capture Service.

The user with administrator authorization enables public access to IMiS®/Capture Service by activating the option "Allow public access" and creating the "Inbound Rule" for IMiS®/Capture Service in the Windows Firewall via the "Enable communication through firewall" button.

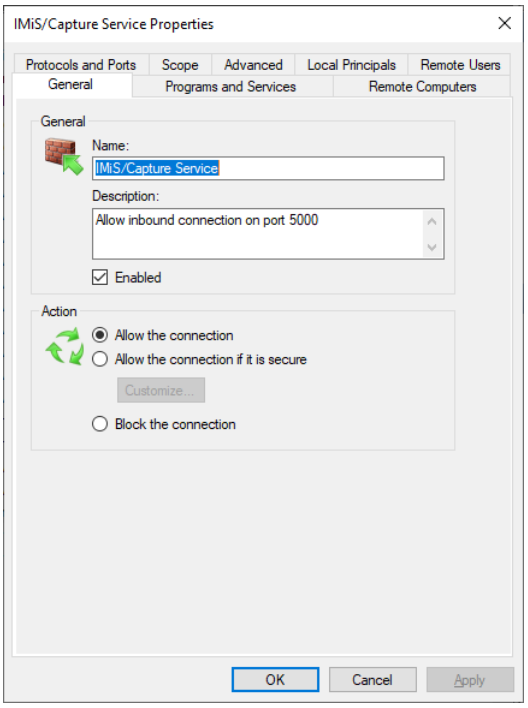

Image 32: Example of an inbound rule in Windows Firewall settings

In the event that the "Inbound Rule" in the Windows Firewall is created successfully, the button will change to "Disable communication through firewall", which enables the deletion of the existing "Inbound Rule" for IMiS®/Capture Service in the Windows Firewall settings.

2023 © MARGIS Inc., All rights reserved.

The user with administrator authorization enables a secure connection via the HTTPS protocol by activating the option “Enable secure connection on port” and entering the port of the secure connection. In the checkbox “Select certificate”, the user selects the digital certificate that will enable a secure connection.

In addition to selecting among the existing valid digital certificates from the “Trusted Root Certification Authorities” storage assigned to the computer, the user with administrator authorization can specify the digital certificate suitable for a secure connection in two other ways:

- Select from file system: uses a digital certificate selected via a dialog for selecting a digital certificate in the local file system.
- Create new self-signed certificate uses a newly created self-signed digital certificate.

5.3.4 Additional administrator settings

Profiles can also be configured outside of the IMiS®/wScan application. This can be done by a user with administrator rights and knowledge of updating Windows Registry.
All profile settings are written in Windows Registry under the key *HKEY_LOCAL_MACHINE\SOFTWARE\Margis\IMiS Capture Service* in the *profiles* field.
If IMiS®/Capture Service does not have access rights for this key, the settings are saved to the file system in the *profiles.json* file in the *%PROGRAMDATA%\Margis\IMiS Capture Service* folder.
The profile settings are written in the JSON file format. They can therefore be copied from one computer to another.

5.4 Uninstallation and modification

Installation modification or uninstallation of the IMiS®/wScan application is performed by the administrator on the workstation using the standard Windows application “Add or Remove Programs”. You access the application by clicking on the “Start” button, locating the “Settings” icon and starting “Add or Remove Programs”. From the Apps & features list the administrator selects the IMiS®/wScan application.

2023 © MARGIS Inc., All rights reserved.

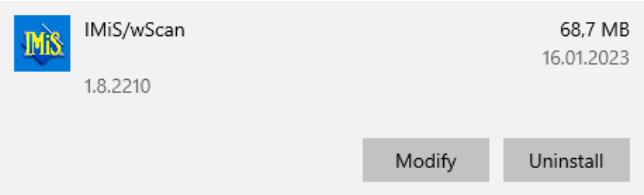

Image 33: Selecting between installation modification and uninstallation of application

5.4.1 Uninstall

By selecting the "Uninstall" option, the administrator begins the procedure of uninstalling the IMiS®/wScan application.

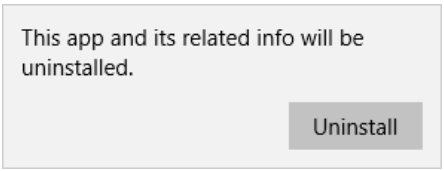

Image 34: Selecting the uninstallation of application

During the uninstallation procedure all files and application settings created by the installation package are removed. The administrator can monitor the progress of the configuration review via a dialog box. By clicking on the “Cancel” button, the review procedure is canceled.

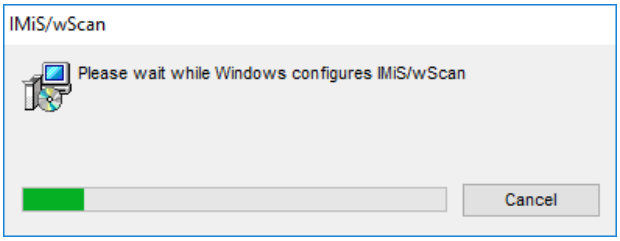

Image 35: Displaying the progress bar of the configuration review

Afterwards the administrator is shown a dialog box for selecting the options: “Modify”, “Repair” or “Remove”. To remove the installation package, select “Remove”. Confirm the selection with the “Next” button.

2023 © MARGIS Inc., All rights reserved.

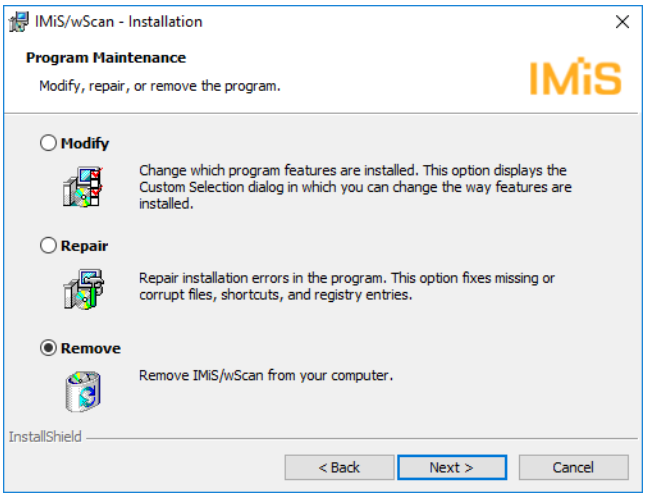

Image 36: Selecting application removal

In the next step confirm the removal by clicking on the “Remove” button.

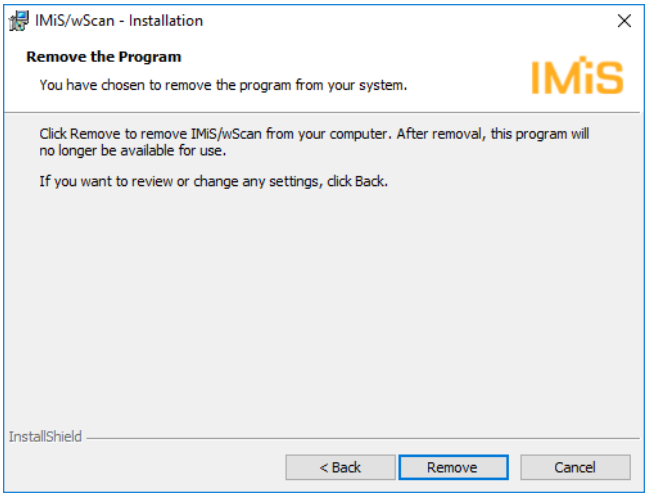

Image 37: Confirming application removal

Uninstallation may take from a few seconds to a few tens of seconds, depending on the version of the installation package and computer speed.

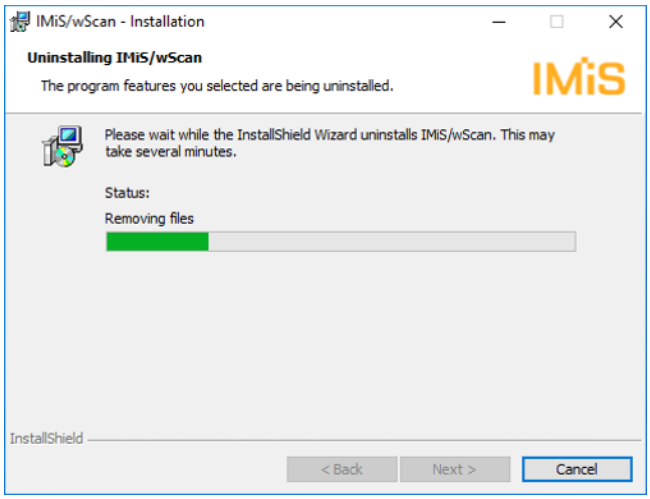

Image 38: Displaying the progress bar during the uninstallation procedure

After finishing uninstalling the application, a dialog box appears, which the administrator closes by clicking on the “Finish” button.

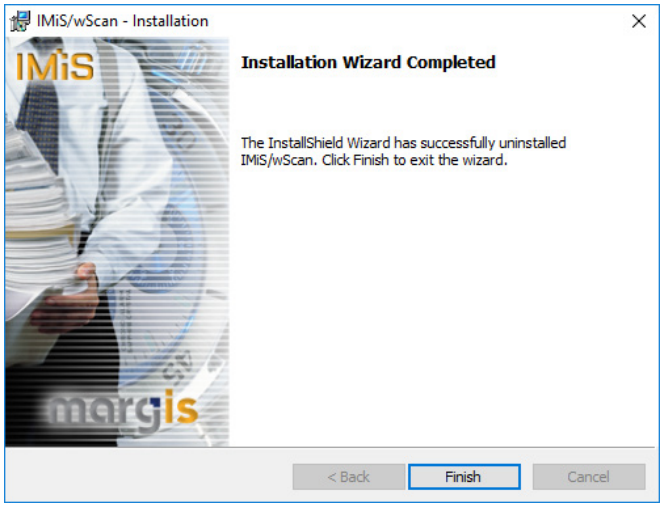

Image 39: Notification of finishing the procedure of uninstalling the installation package

2023 © MARGIS Inc., All rights reserved.

5.4.2 Installation modifications and repairs

The administrator makes modifications and repairs to the installation of the IMiS®/wScan application in a Windows environment via the “Start” button, the “Settings” icon, “Add or Remove Programs”, and an application selected from the Apps & features list.

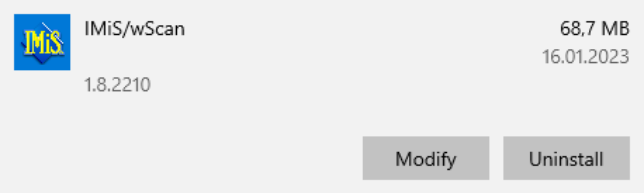

Image 40: Selecting between installation modifications and repairs and removing the installed application

The administrator confirms the selection of installation modifications or repairs by clicking on the “Next” button.

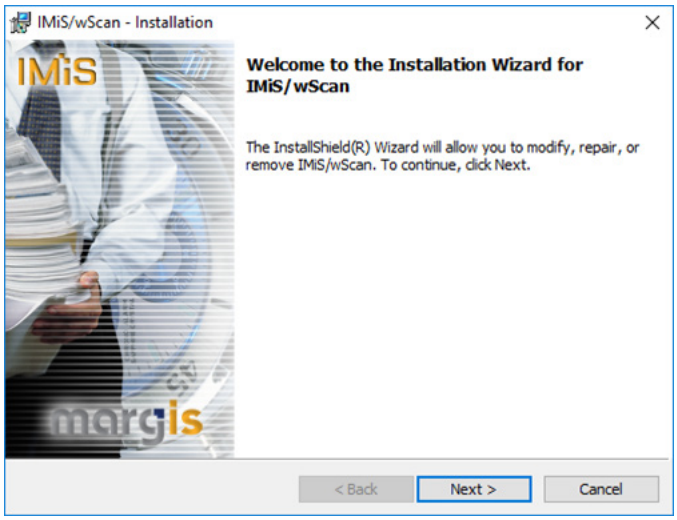

Image 41: Starting the procedure of implementing installation modifications and repairs

5.4.2.1 Installation modifications

The administrator is shown a dialog box in which the option “Modify” has been ticked. Confirm the selection with the “Next” button.

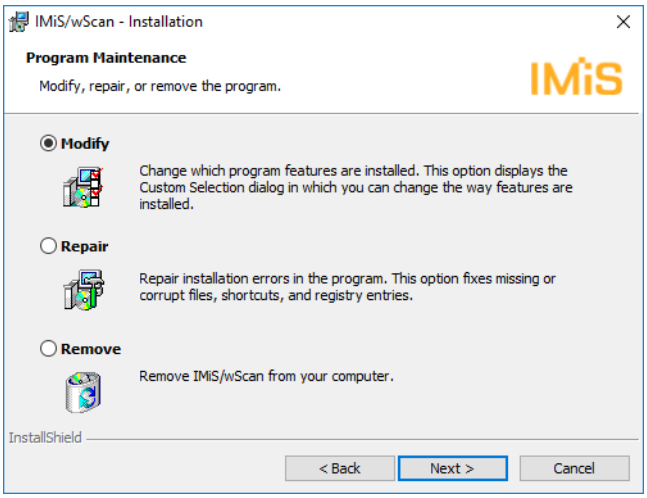

Image 42: Selecting installation modification

By clicking on the icon, the administrator ticks the application features to install.

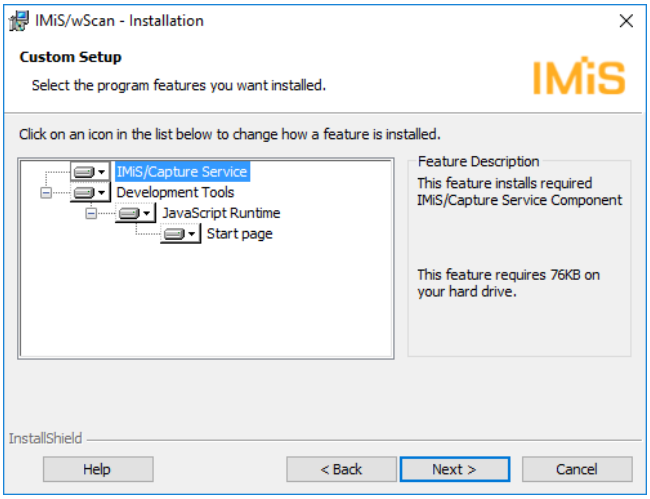

Image 43: Selecting features when modifying installation

By confirming the selection, the administrator starts the installation procedure. Further steps are the same as for typical, full and custom installation.

The procedure finishes with the installation of all required application features.

For more information see chapter Installation with the Wizard.

5.4.2.2 Installation repairs

If installation files, shortcuts or register entries were damaged during the installation of the IMiS®/wScan application or later, the administrator can repair them.

After starting the procedure of making modifications and repairs to the installation, the administrator is shown a dialog box in which the option “Repair” has been ticked.

Confirm the selection with the “Next” button.

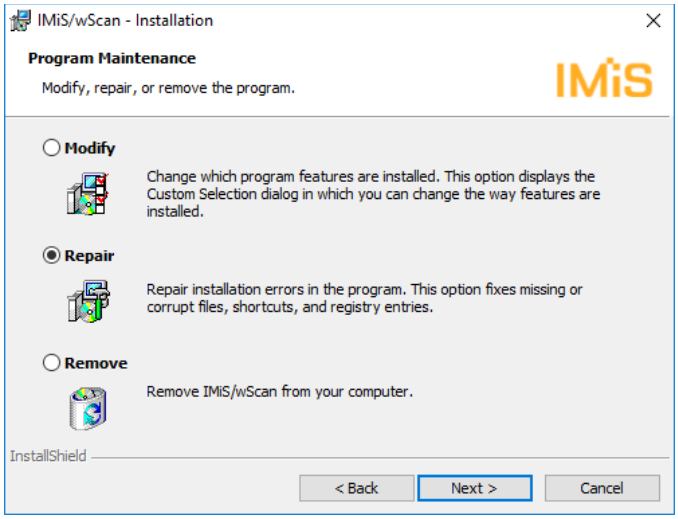

Image 44: Selecting installation repairs

Installation repairs are executed in the next few steps. The procedure finishes with the installation of all required application features and does not require administrator intervention.

For more information see chapter Installation with the Wizard.

5.5 Upgrade

When a new IMiS®/wScan version is issued, the new version must be installed on each workstation. This procedure is carried out with the Installation Wizard and matches the procedure for product installation.

During the upgrade procedure the previous version of the product is automatically uninstalled. All user settings are preserved. This is followed by the procedure of installing the new version.

For more information see chapter Installation with the Wizard.

2023 © MARGIS Inc., All rights reserved.

6 TECHNICAL DOCUMENTATION

This documentation has been prepared for developers with knowledge of the JavaScript programming language and who are familiar with the concepts of object-oriented programming.
It is divided into imis.scan.js and imis.scan.ui.js, and examples of using both libraries.

6.1 imis.scan.js

The **imis.scan.js** JavaScript library manages data exchange with IMiS®/Capture Service.
This library is built on the ECMAScript 6 standard.

This library enables different functionalities:

- Adding, reading, modifying, deleting profiles.
- Adding, reading, executing, stopping jobs.
- Reading and deleting documents.
- Reading, deleting, moving, changing orientation and cropping document pages.
- Reading barcodes on page.
- Reading and deleting redactions on page.
- Reading modules.

6.1.1 imis.scan.Scan

This object represents methods for exchanging data with the IMiS®/Capture Service server.
It enables reading, creating, updating and deleting profiles, and reading and creating jobs.
It enables the detection of creating and deleting profiles and of creating jobs.
Before starting the library, the developer must obtain a security key to access the IMiS®/Capture Service. For obtaining a security key see chapter Security settings.

2023 © MARGIS Inc., All rights reserved.

Constructor

<table>
<tr><td>imis.scan.Scan(options)</td><td>Creates a new object for data exchange and establishes a connection to the IMiS®/Capture Service server.

Options object:
<table>
<tr><td>url</td><td>string</td><td>Address to IMiS®/Capture Service, the default value is http://localhost:5000 (optional).</td></tr>
<tr><td>apiKey</td><td>string</td><td>Key to access the IMiS®/Capture Service.</td></tr>
<tr><td>reconnect</td><td>boolean</td><td>Specifies whether it attempts to automatically reconnect on unsuccessful connection to IMiS®/Capture Service (optional, default value false).</td></tr>
<tr><td>onConnect</td><td>callback</td><td>Callback on successful connection to IMiS®/Capture Service (optional).

callback: function()</td></tr>
<tr><td>onConnectError</td><td>callback</td><td>Callback on unsuccessful connection to IMiS®/Capture Service (optional).

callback: function(error: string)</td></tr>
<tr><td>onDisconnect</td><td>callback</td><td>Callback on disconnection to IMiS®/Capture Service (optional).</td></tr>
<tr><td>onError</td><td>callback</td><td>Callback on error (optional).

callback: function(error: string)</td></tr>
<tr><td>onReconnect</td><td>callback</td><td>Callback on successful connection (optional).</td></tr>
<tr><td>onReconnecting</td><td>callback</td><td>Callback on successful reconnection (optional).</td></tr>
<tr><td>onModulesChange</td><td>callback</td><td>Callback on changes to modules (optional).</td></tr>
</table></td></tr>
</table>

Properties

canGetLogs	boolean	Returns a value that tells whether log files can be retrieved.
canReconnect	boolean	Returns or sets a value that defines whether an automatic reconnection to the IMiS®/Capture Service server is possible.
canUpload	boolean	Returns a value that tells whether files can be uploaded to the job.

Methods

connect()	Establishes a connection to the IMiS®/Capture Service server.
onCreateProfile(callback)	Callback on creating a new profile. Callback is triggered if IMiS®/Capture Service detects that a new profile has been created. Parameters: - callback: function(profile: imis.scan.Profile)
onDeleteProfile(callback)	Callback on deleting a profile. Callback is triggered if IMiS®/Capture Service detects that a profile has been deleted. Parameters: - callback: function(id: String)
onCreateJob(callback)	Callback on creating a job. Callback is triggered if IMiS®/Capture Service detects that a job has been created. Parameters: - callback: function(job: imis.scan.Job)
onError(callback)	Callback on error. Parameters: - callback: function(error: string)
onModulesChange(callback)	Callback on changes to modules: Parameters: - callback: function(modules: imis.scan.Module[]

2023 © MARGIS Inc., All rights reserved.

Methods (cont.)

<table>
<tr><td>getProfile(options)</td><td>Returns a profile.

Options object:<table><tr><td>id</td><td>string</td><td>Unique profile identifier</td></tr><tr><td>success</td><td>callback</td><td>Callback on successful reading of a profile.
callback: function(profile: imis.scan.Profile)</td></tr><tr><td>error</td><td>callback</td><td>Callback on unsuccessful reading of a profile.
callback: function(error: string)</td></tr></table></td></tr>
<tr><td>getProfiles(options)</td><td>Returns a collection of profiles.

Options object:<table><tr><td>reload</td><td>boolean</td><td>Defines whether profiles are reloaded to IMiS®/Capture Service on callback.</td></tr><tr><td>success</td><td>callback</td><td>Callback on successful reading of a collection of profiles.
callback: function(profiles: imis.scan.Profile[])</td></tr><tr><td>error</td><td>callback</td><td>Callback on unsuccessful reading of a collection of profiles.
callback: function(error: string)</td></tr></table></td></tr>
<tr><td>newProfile()</td><td>Ustvari nov profil, ki ni shranjen.</td></tr>
<tr><td>createProfile(options)</td><td>Creates a profile.

Options object:<table><tr><td>profile</td><td>imis.scan.Profile</td><td>New profile.</td></tr><tr><td>success</td><td>callback</td><td>Callback on successful creation of a profile.
callback: function(profile: imis.scan.Profile)</td></tr><tr><td>error</td><td>callback</td><td>Callback on unsuccessful creation of a profile.
callback: function(error: string)</td></tr></table></td></tr>
</table>

2023 © MARGIS Inc., All rights reserved.

Methods (cont.)

<table>
<tr><td>getJob(options)</td><td>Returns a job.

Options object:<table><tr><td>id</td><td>string</td><td>Unique job identifier.</td></tr><tr><td>success</td><td>callback</td><td>Callback on successful reading of a job.
callback: function(job: imis.scan.Job)</td></tr><tr><td>error</td><td>callback</td><td>Callback on unsuccessful reading of a job.
callback: function(error: string)</td></tr></table></td></tr>
<tr><td>getJobs(options)</td><td>Returns jobs.

Options object:<table><tr><td>success</td><td>callback</td><td>Callback on successful reading of jobs.
callback: function(job: imis.scan.Job[])</td></tr><tr><td>error</td><td>callback</td><td>Callback on unsuccessful reading of jobs.
callback: function(error: string)</td></tr></table></td></tr>
<tr><td>createJob(options)</td><td>Creates a job; after successful creation destroys the last job created.

Options object:<table><tr><td>profile</td><td>string or imis.scan.Profile</td><td>Unique profile identifier or profile.</td></tr><tr><td>success</td><td>callback</td><td>Callback on successful creation of a job.
callback: function(job: imis.scan.Job)</td></tr><tr><td>error</td><td>callback</td><td>Callback on unsuccessful creation of a job.
callback: function(error: string)</td></tr></table></td></tr>
<tr><td>getModules(options)</td><td>Returns a collection of modules.

Options object:<table><tr><td>success</td><td>callback</td><td>Callback on successful reading of a collection of modules.
callback: function(job: imis.scan.Module[])</td></tr><tr><td>error</td><td>callback</td><td>Callback on unsuccessful reading of a collection of modules.
callback: function(error: string)</td></tr></table></td></tr>
</table>

2023 © MARGIS Inc., All rights reserved.

Methods (cont.)

<table>
<tr><td>getInfo(options)</td><td>Returns information on the service version.

Options object:<table><tr><td>success</td><td>callback</td><td>Callback on successful reading of information on the service version.
callback: function(data: imis.scan.Info)</td></tr><tr><td>error</td><td>callback</td><td>Callback on unsuccessful reading of the service version.
callback: function(error: string)</td></tr></table></td></tr>
<tr><td>getLogs(options)</td><td>Returns a ZIP file with log files.

Options object:<table><tr><td>success</td><td>callback</td><td>Callback on successful reading of log files.
callback: function(data:Blob)</td></tr><tr><td>error</td><td>callback</td><td>Callback on unsuccessful reading of log files.
callback: function(error: string)</td></tr></table></td></tr>
<tr><td>close()</td><td>Closes the scan and terminates the connection to the IMiS®/Capture Service server.
After calling this method, the scan object is no longer usable.</td></tr>
</table>

6.1.2 imis.scan.Profile

This object represents a profile. It enables the detection of profile modification.

Methods

setModule(module: imis.scan.Module)	Sets the module in the profile. “Save” must be called in order to save.
setModules(modules: imis.scan.Module [])	Sets a collection of modules in the profile. A module is removed if the remove property is set to *true*, if not, it is modified or added. “Save” must be called in order to save.
addModule(module: imis.scan.Module)	Adds a new module to the profile. “Save” must be called in order to save.
removeModule(module: imis.scan.Module)	Removes a module from the profile. “Save” must be called in order to save.

2023 © MARGIS Inc., All rights reserved.

Methods (cont.)

onChange(callback)	Callback on profile modification. Callback is triggered if IMiS®/Capture Service detects that the profile has been modified. Parameters: - callback: function(profile: imis.scan.Profile)
save(options)	Saves a profile. Parameters: - options.commit: boolean: Defines whether changes are committed to the service. - options.success: function(profile: imis.scan.Profile) - options.error: function(error: string)
delete(options)	Deletes a profile. Parameters: - options.success: function() - options.error: function(error: string)
clone()	Returns a copy.
equals(profile: imis.scan.Profile)	Returns *true* if the profiles are equal, if not, it returns *false*.
destroy()	Removes linked profile references.
getAttribute(id: string)	Returns an attribute.
addAttribute(attribute)	Adds an attribute.
moveAttribute(index, newIndex)	Changes the position of an attribute in the attributes collection.
removeAttribute(attribute, index)	Removes an attribute.

Properties

Id	string	Returns a unique profile identifier.
Name	string	Returns or sets the profile name.
Disabled	Boolean	Returns whether the profile has been disabled.
disabledMessage	string	Returns the reason for the disabled profile.
Source	string	Returns or sets the module identifier at the source.

2023 © MARGIS Inc., All rights reserved.

Properties (cont.)

Target	string	Returns or sets the module identifier at the target.
scannerSource	imis.scan.ScannerModule	Returns the scanner module.
folderTarget	imis.scan.FolderTargetModule	Returns the target module in charge of saving.
barcodeExtractor	imis.scan.BarcodeExtractorModule	Returns the module that recognizes barcodes.
pageCountSeparator	imis.scan.PageCountSeparatorModule	Returns a module that separates documents by page count.
barcodeSeparator	imis.scan.BarcodeSeparatorModule	Returns a module that separates documents by barcode settings.
blankPageSeparator	imis.scan.BlankPageSeparatorModule	Returns a module that separates documents by blank pages.
blankPageDetector	imis.scan.BlankPageDetectorModule	Returns a module that detects blank pages.
changed	Boolean	Returns whether an object has been changed.
modules	imis.scan.Module[]	Returns a collection of modules.
readOnly	boolean	Returns or sets the read-only profile.
canManageAttributes	boolean	Defines whether attributes are manageable.
isNew	boolean	Defines whether the profile is new.

2023 © MARGIS Inc., All rights reserved.

6.1.3 imis.scan.Job

This object represents a job, enables startup, canceling, detection of modifications to properties and detection of document creation.

Methods

<table>
<tr><td>start(options)</td><td>Starts a job. The offset defines the identifier in front of which the inserting or overwriting of pages begins.

Options object:<table>
<tr><td>offset</td><td>string</td><td>The identifier in front of which the continuation of a job begins (optional).</td></tr>
<tr><td>overwrite</td><td>boolean</td><td>Overwriting of new pages, starting from the offset (optional).</td></tr>
<tr><td>insert</td><td>boolean</td><td>Inserting new pages in front of the offset (optional).</td></tr>
<tr><td>success</td><td>callback</td><td>Callback on successfully starting a job.

callback: function(job: imis.scan.Job)</td></tr>
<tr><td>error</td><td>callback</td><td>Callback on unsuccessfully starting a job.

callback: function(error: string)</td></tr>
</table></td></tr>
<tr><td>cancel(options)</td><td>Cancels a job.

Options object:<table>
<tr><td>success</td><td>callback</td><td>Callback on successfully canceling a job.

callback: function(job: imis.scan.Job)</td></tr>
<tr><td>error</td><td>callback</td><td>Callback on unsuccessfully canceling a job.

callback: function(error: string)</td></tr>
</table></td></tr>
</table>

2023 © MARGIS Inc., All rights reserved.

Methods (cont.)

<table>
<tr><td>onChange(callback)</td><td>Callback on changing a job.

Parameters:
- callback: function(job: imis.scan.Job)</td></tr>
<tr><td>onCreateDocument(callback)</td><td>Callback on creating a document.

Parameters:
- callback: function(document: imis.scan.Document)</td></tr>
<tr><td>getJob(options)</td><td>Returns a job.

Options object:
<table>
<tr><td>success</td><td>Callback</td><td>Callback on successful reading of a job.

callback: function(job: imis.scan.Job)</td></tr>
<tr><td>error</td><td>callback</td><td>Callback on unsuccessful reading of a job.

callback: function(error: string)</td></tr>
</table></td></tr>
<tr><td>getDocuments(options)</td><td>Returns a collection of documents.

Options object:
<table>
<tr><td>success</td><td>callback</td><td>Callback on successful reading of documents.

callback: function(documents: imis.scan.Document[])</td></tr>
<tr><td>error</td><td>callback</td><td>Callback on unsuccessful reading of documents.

callback: function(error: string)</td></tr>
</table></td></tr>
</table>

2023 © MARGIS Inc., All rights reserved.

Methods (cont.)

<table>
<tr><td>getDocument(options)</td><td>Returns a document.

Options object:<table><tr><td>success</td><td>callback</td><td>Callback on successful reading of a document.

callback: function(document: imis.scan.Document)</td></tr><tr><td>error</td><td>callback</td><td>Callback on unsuccessful reading of a document.

callback: function(error: string)</td></tr></table></td></tr>
<tr><td>getNextDocument(document: imis.scan.Document)</td><td>Returns the next document imis.scan.Document if it exists.</td></tr>
<tr><td>getPrevDocument(document: imis.scan.Document)</td><td>Returns the previous document imis.scan.Document if it exists.</td></tr>
<tr><td>onError(callback)</td><td>Callback on job error.

Parameters:
- callback: function(error: string)</td></tr>
<tr><td>destroy()</td><td>Destroys a job and all documents.</td></tr>
<tr><td>download(callback, errorCallback)</td><td>Returns the URL to transfer all documents in a task.

Parameters:
- callback: function(uri: string)
- errorCallback: function()</td></tr>
<tr><td>isCompleted()</td><td>Returns whether the task has been completed.</td></tr>
<tr><td>isCancelled()</td><td>Returns whether the task has been cancelled.</td></tr>
<tr><td>isCreated()</td><td>Returns whether the task has been created.</td></tr>
<tr><td>isInProgress()</td><td>Returns whether the task is in progress.</td></tr>
<tr><td>isPending()</td><td>Returns whether the task is pending.</td></tr>
<tr><td>isError()</td><td>Returns whether an error occurred.</td></tr>
</table>

2023 © MARGIS Inc., All rights reserved.

Methods (cont.)

<table>
<tr><td>addRedactions(options)</td><td>Adds redactions on the page.

Options object:<table><tr><td>success</td><td>callback</td><td>Callback on successful adding of redactions.</td></tr><tr><td>error</td><td>callback</td><td>Callback on unsuccessful adding of redactions.</td></tr><tr><td>data.regions</td><td>AddRedactionElement []</td><td>Collection of redactions.</td></tr></table>
AddRedactionElement object:<table><tr><td>id</td><td>string</td><td>Page identifier.</td></tr><tr><td>region.left</td><td>number</td><td>Left offset on page.</td></tr><tr><td>region.top</td><td>number</td><td>Top offset on page.</td></tr><tr><td>region.height</td><td>number</td><td>Height of redaction.</td></tr><tr><td>region.width</td><td>number</td><td>Width of redaction.</td></tr></table></td></tr>
<tr><td>addHighlights(options)</td><td>Adds highlights on the page.

Options object:<table><tr><td>success</td><td>callback</td><td>Callback on successfully adding highlights.</td></tr><tr><td>error</td><td>callback</td><td>Callback on unsuccessfully adding highlights.</td></tr><tr><td>data.regions</td><td>AddHighlightElement []</td><td>Collection of highlights.</td></tr></table>
AddHighlightElement object:<table><tr><td>id</td><td>string</td><td>Page identifier.</td></tr><tr><td>color</td><td>string</td><td>Highlight color.</td></tr><tr><td>region.left</td><td>number</td><td>Left offset on page.</td></tr><tr><td>region.top</td><td>number</td><td>Top offset on page.</td></tr><tr><td>region.height</td><td>number</td><td>Redaction height.</td></tr><tr><td>region.width</td><td>number</td><td>Redaction width.</td></tr></table></td></tr>
</table>

2023 © MARGIS Inc., All rights reserved.

Methods (cont.)

<table>
<tr><td>upload(options)</td><td>Uploads files to the job.

Options object:<table>
<tr><td>file</td><td>File</td><td>A file.</td></tr>
<tr><td>success</td><td>callback</td><td>Callback on successfully adding a file.</td></tr>
<tr><td>error</td><td>callback</td><td>Callback on unsuccessfully adding a file.</td></tr>
</table></td></tr>
<tr><td>canContinue(options)</td><td>Returns whether the job can be continued.

Options object:<table>
<tr><td>profile</td><td>Imis.scan.Profile</td><td>Profile (optional).</td></tr>
<tr><td>profileId</td><td>string</td><td>Profile identifier (optional).</td></tr>
<tr><td>success</td><td>callback(continue: boolean)</td><td>Callback on successfully checking the continuation.</td></tr>
<tr><td>error</td><td>callback()</td><td>Callback on unsuccessfully checking the continuation.</td></tr>
</table></td></tr>
</table>

Properties

id	string	Returns a unique job identifier.
index	number	Returns a job index.
name	string	Returns a job name.
error	string	Returns a job error message.
created	string	Returns the date and time of job creation. Format 2018-07-03T09:58:15.9225533+02:00.
documentCount	number	Returns the number of documents in a job.
pageCount	number	Returns the number of pages in all documents in a job.
fileName	string	Returns the name of the task file.
documents	imis.scan.Document[]	Returns the collection of documents in a job.
owner	imis.scan.Scan	Returns the owner of the job instance.

2023 © MARGIS Inc., All rights reserved.

statusChanged	Boolean	Defines whether the job status has changed.
canAddHighlights	boolean	Defines whether highlights can be added.

6.1.4 imis.scan.Document

This object represents a document, enables the detection of creating pages within a document, changes to document properties, reading of pages and deleting a document.

Methods

<table>
<tr><td>getPages(options)</td><td>Returns a collection of pages.

Options object:<table><tr><td>success</td><td>callback(pages: imis.scan.Page[])</td><td>Callback on successfully reading a page.</td></tr><tr><td>error</td><td>callback(e: string)</td><td>Callback on unsuccessfully reading a page.</td></tr></table></td></tr>
<tr><td>getPage(options)</td><td>Returns a page.

Options object:<table><tr><td>pageId</td><td>string</td><td>Page identifier.</td></tr><tr><td>success</td><td>callback(pages: imis.scan.Page)</td><td>Callback on successfully reading a page.</td></tr><tr><td>error</td><td>callback(e: string)</td><td>Callback on unsuccessfully reading a page.</td></tr></table></td></tr>
<tr><td>getFirstPage()</td><td>Returns the first page imis.scan.Page in a document.</td></tr>
<tr><td>getLastPage()</td><td>Returns the last page imis.scan.Page in a document.</td></tr>
<tr><td>getNextPage(page: imis.scan.Page)</td><td>Returns the next page imis.scan.Page in a document.</td></tr>
<tr><td>getPrevPage(page: imis.scan.Page)</td><td>Returns the previous page imis.scan.Page in a document.</td></tr>
<tr><td>getNextDocument()</td><td>Returns the next document imis.scan.Document.</td></tr>
<tr><td>getPrevDocument()</td><td>Returns the previous document imis.scan.Document.</td></tr>
</table>

2023 © MARGIS Inc., All rights reserved.

Methods (cont.)

<table>
<tr><td>onChange(callback)</td><td>Callback on changing a document. Returns the same document with changed properties.
Parameters:
- callback: function(document: imis.scan.Document)</td></tr>
<tr><td>onCreatePage(callback)</td><td>Callback on creating a page within a document. Returns the new page within a document.

Parameters:
- callback: function(page: imis.scan.Page)</td></tr>
<tr><td>-onDelete(callback)</td><td>Callback on deleting a document.

Parameters:
- callback: function(document: imis.scan.Document)</td></tr>
<tr><td>onError(callback)</td><td>Callback on document error.

Parameters:
- callback: function(error: string)</td></tr>
<tr><td>destroy()</td><td>Destroys a document and document pages.</td></tr>
<tr><td>download(callback, errorCallback)</td><td>Obtains a link for document transfer.

Parameters:
- callback: function(uri: string)
- errorCallback: function()</td></tr>
<tr><td>delete(options)</td><td>Deletes a document.

Options object:
<table><tr><td>success</td><td>callback</td><td>Callback on successful deletion.</td></tr><tr><td>error</td><td>callback</td><td>Callback on unsuccessful deletion.</td></tr></table></td></tr>
<tr><td>join(options)</td><td>Joins documents.

Options object:
<table><tr><td>Ids</td><td>string []</td><td>Collection of document identifiers.</td></tr><tr><td>success</td><td>callback</td><td>Callback on successfully joining documents.</td></tr><tr><td>error</td><td>callback</td><td>Callback on unsuccessfully joining documents.</td></tr></table></td></tr>
<tr><td>getAttribute(id: string)</td><td>Returns attribute imis.scan.model.DocumentAttribute.</td></tr>
</table>

2023 © MARGIS Inc., All rights reserved.

Methods (cont.)

<table>
<tr><td>setAttribute(options)</td><td>Sets the attribute value.

Options object:<table><tr><td>id</td><td>string</td><td>Attribute identifier.</td></tr><tr><td>value</td><td>any</td><td>Attribute value.</td></tr><tr><td>success</td><td>callback</td><td>Callback on successfully setting the attribute value.</td></tr><tr><td>error</td><td>callback</td><td>Callback on unsuccessfully setting the attribute value.</td></tr></table></td></tr>
<tr><td>setSeparator(options)</td><td>Sets the document separator value.

Options object:<table><tr><td>value</td><td>string</td><td>Document separator value.</td></tr><tr><td>success</td><td>callback</td><td>Callback on successfully setting the document separator value.</td></tr><tr><td>error</td><td>callback</td><td>Callback on unsuccessfully setting the document separator value.</td></tr></table></td></tr>
<tr><td>equals(document: imis.scan.Document)</td><td>Returns “true” if the documents are equal; if not, it returns “false”.</td></tr>
</table>

Properties

id	String	Returns a unique document identifier.
name	String	Returns a document name.
mime	String	Returns the type of document content.
pageCount	Number	Returns the number of pages in a document.
created	String	Returns the date and time of creation. Format: 2018-07-03T09:56:26.4618227+02:00
length	Number	Returns the size of the document in bytes.
filename	String	Returns the name of the task file.
previousId	string	Returns the identifier of the previous document.

2023 © MARGIS Inc., All rights reserved.

Properties (cont.)

pages	imis.scan.Page[]	Returns the collection of pages on a document.
canManageSeparator	boolean	Defines whether the document separator can be changed.
barcodeSeparator	string	Returns the value of the barcode-type separator.
attributeDefinitions	imis.scan.model.AttributeDefinition []	Returns a collection of document attribute definitions.
attributes	imis.scan.model.DocumentAttribute []	Returns a collection of document attributes.
barcodes	imis.scan.Barcode []	Returns a collection of document barcodes.
index	number	Document index in the collection of job documents.
job	Imis.scan.Job	Job document.

6.1.5 imis.scan.Page

This object represents a page, enables the detection of changes to properties, reading a page preview, reading a page in basic size, deleting, moving, changing orientation, cropping, reading and deleting redactions.

2023 © MARGIS Inc., All rights reserved.

Methods

<table>
<tr><td>getThumbnailUri(options)</td><td>Obtains the link to the page preview URI in the image/png format.

Options object:<table>
<tr><td>height</td><td>Number</td><td>Height of page preview.</td></tr>
<tr><td>width</td><td>Number</td><td>Width of page preview.</td></tr>
<tr><td>mime</td><td>string</td><td>Page format type (optional). Valid values:
- image/jpeg,
- image/gif,
- image/bmp,
- image/png.</td></tr>
<tr><td>success</td><td>Callback</td><td>Call when connection was successful.

Parameters:
- callback: function(uri: string)</td></tr>
<tr><td>error</td><td>Callback</td><td>Call when connection failed.

Parameters:
- callback: function()</td></tr>
</table></td></tr>
<tr><td>getImage(callback, errorCallback, mime)</td><td>Obtains the link to the image/png format.

Parameters:
- mime: string: Page format type (optional).
Valid values: image/jpeg, image/gif, image/bmp, image/png.
- callback: function(uri: string)
- errorCallback: function()</td></tr>
<tr><td>delete(options)</td><td>Page deletion.

Options object:<table>
<tr><td>success</td><td>callback</td><td>Callback on successfully deleting a page.</td></tr>
<tr><td>error</td><td>callback</td><td>Callback on unsuccessfully deleting a page.</td></tr>
</table></td></tr>
<tr><td>onChange(callback)</td><td>Callback on changing a page. Returns the page with changed properties.

Parameters:
- callback: function(page: imis.scan.Page)</td></tr>
</table>

Methods (cont.)

<table>
<tr><td>onMove(callback)</td><td>Callback on moving a page. Returns the new document in which the page is located.

Parameters:
- callback: function(document: imis.scan.Document)</td></tr>
<tr><td>onDelete(callback)</td><td>Callback on deleting a page.

Parameters:
- callback: function()</td></tr>
<tr><td>removeRedaction(options)</td><td>Deletes a redaction.

Options object:
<table>
<tr><td>redaction</td><td>imis.scan.Redaction</td><td>Redaction.</td></tr>
<tr><td>success</td><td>callback</td><td>Callback on successful deletion of a redaction.</td></tr>
<tr><td>error</td><td>callback(error: string)</td><td>Callback on unsuccessful deletion of a redaction.</td></tr>
</table></td></tr>
<tr><td>removeHighlight (options)</td><td>Deletes a highlight.

Options object:
<table>
<tr><td>highlight</td><td>imis.scan.Highlight</td><td>A highlight.</td></tr>
<tr><td>success</td><td>callback</td><td>Callback on successfully deleting a highlight.</td></tr>
<tr><td>error</td><td>callback(error: string)</td><td>Callback on unsuccessfully deleting a highlight.</td></tr>
</table></td></tr>
<tr><td>rotate(options)</td><td>Changes the page orientation.

Options object:
<table>
<tr><td>orientation</td><td>number</td><td>Sets a new orientation.
The set of values: -270, -180, -90, 0, 90, 180, 270.</td></tr>
<tr><td>success</td><td>callback</td><td>Callback on successful change of orientation.</td></tr>
<tr><td>error</td><td>callback(error: string)</td><td>Callback on unsuccessful change of orientation.</td></tr>
</table></td></tr>
</table>

2023 © MARGIS Inc., All rights reserved.

Methods (cont.)

<table>
<tr><td>move(options)</td><td>Moves a page. The offset defines the identifier of the page in front of which the page is being moved or the identifier of the document if you are inserting a page at the end of the document.

Options object:<table><tr><td>offset</td><td>string</td><td>Defines the identifier of the page in front of which the page is being moved or the identifier of the document if you are inserting a page at the end of the document.</td></tr><tr><td>success</td><td>callback</td><td>Callback on successful move.</td></tr><tr><td>error</td><td>callback</td><td>Callback on unsuccessful move.</td></tr></table></td></tr>
<tr><td>crop(options)</td><td>Crops a page.

Options object:<table><tr><td>left</td><td>number</td><td>Left offset on page.</td></tr><tr><td>top</td><td>number</td><td>Top offset on page.</td></tr><tr><td>width</td><td>number</td><td>Width of page.</td></tr><tr><td>height</td><td>number</td><td>Height of page.</td></tr><tr><td>success</td><td>callback</td><td>Callback on successful cropping of a page.</td></tr><tr><td>error</td><td>callback(error: string)</td><td>Callback on unsuccessful cropping of a page.</td></tr></table></td></tr>
<tr><td>split(options)</td><td>Creates a new document from the page.

Options object:<table><tr><td>success</td><td>callback</td><td>Callback on successfully creating a document from the page.</td></tr><tr><td>error</td><td>callback(error: string)</td><td>Callback on unsuccessfully creating a document from the page.</td></tr></table></td></tr>
<tr><td>isLandscape()</td><td>Returns true if the page is landscape, if not, it returns false.</td></tr>
<tr><td>isPortrait()</td><td>Returns true if the page is portrait, if not, it returns false.</td></tr>
<tr><td>destroy()</td><td>Destroys the page and all the obtained links to the page.</td></tr>
<tr><td>equals(page: imis.scan.Page)</td><td>Returns true if the pages are equal; if not, it returns false.</td></tr>
</table>

2023 © MARGIS Inc., All rights reserved.

Properties

id	string	Returns a unique page identifier.
width	number	Returns the page width.
height	number	Returns the page height.
xresolution	number	Returns the horizontal page resolution in DPI.
yresolution	number	Returns the vertical page resolution in DPI.
barcodes	imis.scan.Barcode[]	Returns the collection of barcodes on the page.
colorFormat	imis.scan.ColorFormat	Returns the page color format.
previousId	string	Returns the identifier of the previous page.
created	string - DateTime	Returns the date and time of page creation.
highlights	imis.scan.Highlight[]	Returns the collection of highlights on the page.
redactions	imis.scan.Redaction[]	Returns the collection of redactions on page.
document	imis.scan.Document	Returns the document in which the page is located.
canDelete	Boolean	Defines whether page deletion is enabled.
canCrop	Boolean	Defines whether page cropping is enabled.
canRemoveRedaction	boolean	Defines whether redaction deletion is enabled.
canRemoveRedaction	Boolean	Defines whether redaction deletion is enabled.
canRotate	Boolean	Defines whether page rotation can be changed.
canMove	Boolean	Defines whether page movement is enabled.
canSplit	Boolean	Defines whether a document can be created from the page.
index	Number	Returns the page index in the collection of document pages.

6.1.6 imis.scan.Barcode

This object represents the properties of the barcode on a page.

Properties

id	string	Barcode identifier.
Height	number	Returns the barcode height.
Width	number	Returns the barcode width.
Text	string	Returns the recognized barcode content.

2023 © MARGIS Inc., All rights reserved.

Properties (cont.)

posX	number	Returns the horizontal offset of the barcode on the page.
posY	number	Returns the vertical offset of the barcode on the page.
Type	string	Returns the barcode type.

6.1.7 imis.scan.Redaction

This object represents the properties of the redaction on a page. It is used to hide specific parts of a document. The redaction is drawn over the area which the user wants to hide. After saving, the redacted content is no longer visible. The redaction can be deleted only before saving the job or document.

Properties

id	string	Redaction identifier.
width	number	Returns the redaction width.
height	number	Returns the redaction height.
left	number	Returns the left offset on page.
top	number	Returns the top offset on page.

6.1.8 imis.scan.Module

This object represents the basis of the module, from which different modules have been derived.

Methods

clone()	Returns a copy.

Properties

Id	string	Returns a unique module identifier.
sendTo	string[]	Returns or sets a collection of module identifiers, to which the module sends data.

2023 © MARGIS Inc., All rights reserved.

Properties (cont.)

Type	string	Returns the module type. Set of values: - Scanner_source - Barcode_extractor - Folder_target.
Remove	boolean	Returns or sets the value that determines whether the module will be removed; applies only when calling the setModules method on imis.scan.Profile.

6.1.9 imis.scan.ScannerModule

This object represents the scanner module, which enables the reading and changing of scanner settings.

This class has been derived from imis.scan.Module.

Properties

driverName	string	Returns or sets the driver name.
scannerModel	string	Returns the scanner model.
scannerValues	imis.scan.ScannerValue[]	Returns a collection of all connected scanners.
paperSize	string	Returns or sets the paper size.
paperSizes	string[]	Returns the collection of paper sizes for the selected driver (driverName).
defaultPaperSize	string	Returns the default paper size value.
resolution	number	Returns or sets the scanning resolution.
resolutions	number[]	Returns a collection of scanning resolutions for the selected driver (driverName).
defaultResolution	number	Returns the default scanning resolution value.
colorFormat	imis.scan.ColorFormat	Returns or sets the scanning color.
colorFormats	imis.scan.ColorFormat[]	Returns a collection of scanning colors for the selected driver (driverName).
defaultColorFormat	imis.scan.ColorFormat	Returns the default scanning color value.
duplex	boolean	Returns or sets whether duplex scanning is enabled.
defaultDuplex	boolean	Returns the default value which defines whether duplex scanning is enabled.
duplexSupport	boolean	Defines whether duplex scanning is supported.

2023 © MARGIS Inc., All rights reserved.

6.1.10 imis.scan.FolderTargetModule

This object represents the target module, which enables the setting of properties of saving files to the file system, with the option of setting the directory, file format, color and compression.

This class has been derived from imis.scan.Module.

Properties

folder	string	Returns or sets the path to the directory.
fileRoot	string	Returns or sets the file name.
fileFormat	string	Returns or sets the file format.
fileFormats	string[]	Returns a set of file formats. Set of values: - BMP - GIF - TIFF - JPEG - PCX - PDF/A - PNG.
colorFormat	imis.scan.ColorFormat	Returns or sets the color format.
colorFormats	imis.scan.ColorFormat[]	Returns a collection of color formats; this collection is connected to the file format (fileFormat).
compression	string	Returns or sets the compression.
compressions	string[]	Returns a collection of compressions; this collection is connected to the color format (colorFormat) and file format.

2023 © MARGIS Inc., All rights reserved.

6.1.11 imis.scan.BarcodeExtractorModule

This object represents the module which enables the detection of barcodes on an individual page. This class has been derived from imis.scan.Module.

Properties

types	string[]	Returns or sets a collection of barcode types for recognition.
typesValues	string[]	Returns a collection of barcode types. Set of values: - addon2 - addon5 - australianpost - aztec - bcdmatrix - codabar - code25_datalogic - code25_iata - code25_industrial - code25_interleaved - code25_invert - code25_matrix - code32 - code39 - code93 - datamatrix - ean13 - ean8 - intelligentmail - pdf417 - postnet - qrcode - royalpost - type128 - ucc128 - upc_a - upc_e.

2023 © MARGIS Inc., All rights reserved.

Properties (cont.)

orientation	string	Returns or sets the option of barcode orientation.
orientationValues	string[]	Returns a collection of barcode orientations. Set of values: - horizontal: Detection of horizontal barcodes. - vertical: Detection of vertical barcodes. - both: Detection of horizontal or vertical barcodes. - horizontalverticaldiagonal: Detection of horizontal, vertical or 45° barcodes.
mode	string	Returns or sets the barcode detection mode.
modeValues	string[]	Returns a collection of barcode detection modes. Set of values: - normal: Normal mode, faster than enhanced. - enhanced: Improved mode, enables better detection; detection is slower.
expression	string	Returns or sets the barcode detection expression.

6.1.12 imis.scan.BlankPageDetectorModule

This object represents the module which enables the detection of blank pages.

This class has been derived from imis.scan.Module.

Properties

threshold	number	Returns or sets the number that defines the blank page detection threshold.

6.1.13 imis.scan.PageCountSeparatorModule

This object represents the module which enables separating documents by page count.

This class has been derived from imis.scan.Module.

Properties

pageCount	number	Returns or sets the number of pages for separating documents by page count.

2023 © MARGIS Inc., All rights reserved.

6.1.14 imis.scan.BarcodeSeparatorModule

This object represents the module which enables barcode-based separation of documents.

This class has been derived from imis.scan.Module.

Properties

barcodes	imis.scan.BarcodePattern []	Returns or sets a collection of imis.scan.BarcodePattern objects.
actionValues	string[]	Returns a collection of actions in the barcode-based separation of documents.

6.1.15 imis.scan.BlankPageSeparatorModule

This object represents the module which enables blank page separation of documents.

This class has been derived from imis.scan.Module.

6.1.16 imis.scan.ScannerValue

This object represents scanner properties.

Properties

driverName	string	Returns the driver name.
scannerModel	string	Returns the scanner model.
colorFormats	imis.scan.ColorFormat[]	Returns a collection of colors.
paperSizes	string[]	Returns a collection of paper sizes.
resolution	number	Returns the resolution.
resolutions	number[]	Returns a collection of resolutions.
duplex	boolean	Returns whether duplex scanning is set.
duplexSupport	boolean	Defines whether duplex scanning is enabled.

2023 © MARGIS Inc., All rights reserved.

6.1.17 imis.scan.ColorFormat

This object represents the properties of the color format.

Properties

colorMode	string	Returns the image color type. Set of values: - blackwhite: Black-and-white image; - grayscale: A grayscale image; - color: A color image.
colorDepth	number	Returns the image color depth.
photometric	string	Returns the mode of reading image data.
compressions	string[]	Returns a collection of compressions; this collection exists only when reading the properties of colorFormats on imis.scan.FolderTargetModule.

6.1.18 imis.scan.BarcodePattern

This object represents the properties of the barcode separator.

Properties

value	String	Returns or sets the barcode value in the barcode-based separation. Set of values: - empty value: the separator is any recognized barcode. - non-empty value: "Regular expression" for searching for the barcode which represents the separator.
action	String	Returns an action in the barcode-based separation of documents. Set of values: - separate: the page with the recognized barcode begins a new document. - separateanddeletepage: the page with the recognized barcode is deleted and the next page begins a new document.
separate	boolean	Returns or sets whether the action equals "separate".
separateAndDelete	boolean	Returns or sets whether the action equals "separateanddeletepage".

2023 © MARGIS Inc., All rights reserved.

6.1.19 imis.scan.model.AttributeDefinition

This object represents the attribute definition properties.

Properties

id	string	Returns or sets the attribute identifier.
name	string	Returns or sets the attribute name.
description	string	Returns or sets the attribute description.
displayName	string	Returns the name of the attribute that contains the identifier.
type	string	Returns or sets the attribute type.
region	imis.scan.model.Region	Returns or sets the attribute region.
regionText	string	Returns or sets the attribute region in the form of text.
required	boolean	Returns or sets whether the attribute is required.
prefix	string	Returns or sets the fixed value at the beginning of the attribute value.
suffix	string	Returns or sets the fixed value at the end of the attribute value.
values	string[]	Returns or sets a collection of values for selecting the attribute value.
validation	string	Returns or sets the expression for validating the attribute value.
filter	string	Returns or sets the filter for the attribute value.
barcodeTypes	string []	Returns or sets a collection of barcode types.
barcodeOrientation	string	Returns or sets the barcode orientation.
barcodeMode	string	Returns or sets the barcode mode.
trueValue	string	Returns or sets the true value description.
falseValue	string	Returns or sets the false value description.

2023 © MARGIS Inc., All rights reserved.

6.1.20 imis.scan.model.DocumentAttribute

This object represents the document attribute properties.

Properties

id	string	Returns the attribute identifier.
name	string	Returns the attribute name.
type	string	Returns or sets the attribute type.
value	string|Date|boolean|number	Returns the attribute value.
definition	imis.scan.model.AttributeDefinition	Returns the attribute definition.
barcodes	imis.scan.Barcode	Returns the collection of barcodes.
isValid	boolean	Defines whether the attribute value is valid.

6.1.21 imis.scan.model.Info

This object represents service information properties.

Properties

serviceVersion	string	Returns the service version.
apiMinVersion	number	Returns the minimum interface version.
apiMaxVersion	number	Returns the maximum interface version.
isScan	boolean	Returns whether the scan module is available.
isBatchScan	boolean	Returns whether the batch scan module is available.

2023 © MARGIS Inc., All rights reserved.

6.1.22 imis.scan.Highlight

This object represents the properties of the highlight on the page. It is used to conceal specific parts of a document. The highlight is drawn over the area which the user wants to hide. After saving, the redacted content is partially visible. The highlight can be deleted only before saving the job or document.

Properties

Id	string	Highlight identifier.
color	string	Highlight color.
width	number	Returns the highlight width.
height	number	Returns the highlight height.
left	number	Returns the left offset on page.
top	number	Returns the top offset on page.

6.2 imis.scan.ui.js

This library enables easy use of components, which can be used for displaying scanning.
The displaying of components is managed by the main component imis.scan.ui.Scan, where we specify all of the components we will be using.

The imis.scan.js library is required for it to work.

6.2.1 imis.scan.ui.Scan

This object represents the main component, which manages the displaying of various components. When creating this component, the components are initialized. Enables reading profiles and setting the selected profile.

Constructor

imis.scan.ui.Scan(options: ScanOptions)	Creates a new object and components are initialized.

2023 © MARGIS Inc., All rights reserved.

Methods

<table>
<tr><td>show(callback)</td><td>Establishes a connection to the IMiS®/Capture Service server and data is loaded into the components.

Parameters:
- callback: Callback on successful establishment of a connection to the server (optional).</td></tr>
<tr><td>getProfiles(callback, error, reload)</td><td>Returns all profiles.

Parameters:
- callback: The callback on successful reading of profiles returns imis.scan.Profile[]
- error: Callback on unsuccessful reading of profiles.
- reload: Defines whether profiles are reread on IMiS®/Capture Service.</td></tr>
<tr><td>getSelectedProfile()</td><td>Returns the selected profile imis.scan.Profile.</td></tr>
<tr><td>setSelectedProfile(profile)</td><td>Sets the selected profile imis.scan.Profile.</td></tr>
<tr><td>canContinue(options)</td><td>Returns whether the current job can be continued.

Options object:<table><tr><td>profile</td><td>imis.scan.Profile</td><td>Profile (optional).</td></tr><tr><td>success</td><td>callback(value: boolean)</td><td>Callback on successfully checking if the job can be continued.</td></tr><tr><td>error</td><td>callback(e: string)</td><td>Callback on unsuccessfully checking if the job can be continued.</td></tr></table></td></tr>
<tr><td>getInsertMode(): boolean</td><td>Returns whether the selected mode is page insertion (true) or page rewrite (false).</td></tr>
<tr><td>setInsertMode(value: boolean)</td><td>Sets the value that defines whether the selected mode is page insertion (true) or page rewrite (false).</td></tr>
<tr><td>showNotification(body: string)</td><td>Shows a notification.</td></tr>
<tr><td>startJob()</td><td>Creates and starts a new job.</td></tr>
<tr><td>startJobUpload()</td><td>Creates and starts a new job where the job source is a file.</td></tr>
</table>

2023 © MARGIS Inc., All rights reserved.

Methods (cont.)

<table>
<tr><td>continueJob(offset: string, type: string)</td><td>Continues the current job with a specific offset, which is an optional attribute, and with a specific type, which is an optional attribute.

Value type:
- fileUpload: Defines whether the source for continuing the job is a file.</td></tr>
<tr><td>continueJobUpload()</td><td>Continues the current job where the job source is a file.</td></tr>
<tr><td>cancelJob()</td><td>Cancels the execution of the current job.</td></tr>
<tr><td>selectNextDocument()</td><td>Selects the first page of the next document.</td></tr>
<tr><td>selectPreviousDocument()</td><td>Selects the last page of the previous document.</td></tr>
<tr><td>selectNextPage()</td><td>Selects the next page.</td></tr>
<tr><td>selectPreviousPage()</td><td>Selects the previous page.</td></tr>
<tr><td>selectPage(page: imis.scan.Page, callback)</td><td>Selects a specific page and callback is performed on selection.</td></tr>
<tr><td>selectRegion(document: imis.scan.Document, attribute: imis.scan.model.DocumentAttribute, callback)</td><td>Selects the attribute region where the page zoom will be changed. Callback is performed after selection.</td></tr>
<tr><td>getLang()</td><td>Returns the selected browser language.</td></tr>
<tr><td>getInfo(options)</td><td>Returns information on the service version.

Options object:<table><tr><td>success</td><td>callback(value: imis.scan.Info)</td><td>Callback on successful reading of the service version.</td></tr><tr><td>error</td><td>callback(e: string)</td><td>Callback on unsuccessful reading of the service version.</td></tr></table></td></tr>
</table>

Methods (cont.)

<table>
<tr><td>downloadLogs(options)</td><td>Downloads log files.

Options object:<table><tr><td>success</td><td>callback(value: Blob)</td><td>Callback on successfully downloading the log file.</td></tr><tr><td>error</td><td>callback(e: string)</td><td>Callback on unsuccessfully downloading the log file.</td></tr></table></td></tr>
<tr><td>close()</td><td>Destroys the job and closes all scanning sources.</td></tr>
<tr><td>print()</td><td>Prints the job if the job contains only one document in the PDF format.</td></tr>
</table>

Properties

Job	imis.scan.Job	Current job.
canReconnect	boolean	Returns or sets the value that defines whether reconnection to the service is enabled.
Loading	boolean	Returns whether the current job is running.
selectedPage	imis.scan.Page	Returns the currently selected page.
canDownloadLogs	boolean	Returns whether log files can be downloaded.
canPrint	boolean	Returns whether the current job can be printed.

6.2.1.1 ScanOptions

This object represents the imis.scan.ui.Scan settings options.

Properties

url	string	Address to IMiS®/Capture Service. The default value is http://localhost:5000 (optional).
apiKey	String	Key to access the IMiS®/Capture Service.
notifications	Boolean	Displaying notifications in a browser; the default value is true (optional).

2023 © MARGIS Inc., All rights reserved.

Properties (cont.)

<table>
<tr><td>notification</td><td>Object</td><td>Displaying notifications in a browser; the default value is true (optional).

<table><tr><td>enabled</td><td>boolean</td><td>Defines whether notifications can be displayed.</td></tr><tr><td>img</td><td>string</td><td>Link to the notification image.</td></tr><tr><td>onCreate</td><td>callback()</td><td>Callback on creating a notification.</td></tr></table></td></tr>
<tr><td>language</td><td>String</td><td>Language settings (optional). If the language has not been defined, the language specified in the browser is set as default.

Set of values:<table><tr><td>en</td><td>English</td></tr><tr><td>sl</td><td>Slovene</td></tr></table></td></tr>
<tr><td>thumbnails</td><td>imis.scan.ui.Thumbnails
or
ThumbnailsOptions</td><td>Displaying documents and pages (optional).</td></tr>
<tr><td>settings</td><td>imis.scan.ui.Settings
or
SettingOptions</td><td>The settings of all profiles (optional).</td></tr>
<tr><td>imageView</td><td>imis.scan.ui.ImageView
or
ImageviewOptions</td><td>Displaying a selected page (optional).</td></tr>
<tr><td>images</td><td>imis.scan.ui.ImageScroll
or
ImageScrollOptions</td><td>Displaying all pages (optional).</td></tr>
<tr><td>status</td><td>imis.scan.ui.Status
or
StatusOptions</td><td>Displaying status (optional).</td></tr>
</table>

2023 © MARGIS Inc., All rights reserved.

Properties (cont.)

imageDetails	imis.scan.ui.ImageDetails or ImageDetailsOptions	Details of a selected page (optional).
progress	imis.scan.ui.Progress or ProgressOptions	Displaying job status (optional).
buttons	UIScanButtonsOptions	Buttons settings.
useLocalStorage	Boolean	Specifies the use of saving settings (currently selected profile) to the browser; if the value is set to false, the saved settings are deleted; the default value is true (optional).
reconnect	Boolean	Specifies whether reconnection is attempted on termination of the connection to the IMiS®/Capture Service; if the connection is successful, the page reloads (optional, default value false).
onReconnect	callback()	Callback on successfully reconnecting.
targetColor	imis.scan.ui.TargetColor ali TargetColorOptions	Shows the color for saving (optional).
targetFormat	imis.scan.ui.TargetFormat ali TargetFormatOptions	Shows the format for saving (optional).
totalDocuments	imis.scan.ui.TotalDocuments ali TotalDocumentsOptions	Shows the number of documents in a job (optional).
totalPages	imis.scan.ui.TotalPages ali TotalPagesOptions	Shows the number of pages in a job (optional).
onError	callback(e: string)	Callback on error.
onPageSelect	callback(page: imis.scan.Page)	Callback on selecting a page.
onJobChange	callback(job: imis.scan.Job)	Callback on changing the job.

2023 © MARGIS Inc., All rights reserved.

6.2.1.2 ScanButtonsOptions

This object represents the settings options for buttons.

Properties

scan	imis.scan.ui.Button or ButtonOptions	The button for starting a scan. If the profile has been disabled and an error exists, it is shown in the popup notification that appears when placing the cursor on the element.
continue	imis.scan.ui.Button or ButtonOptions	The button for continuing the scan (optional).
cancel	imis.scan.ui.Button or ButtonOptions	The button for canceling the scan (optional).
download	imis.scan.ui.Button or ButtonOptions	The button for downloading all scanned documents (optional).
color	imis.scan.ui.ColorDropdownButton or ColorDropdownOptions	The list of possible colors if a scanner is available; changes are saved only temporarily for each job started (optional).
profiles	imis.scan.ui.ProfilesButton or ProfilesButtonOptions	A collection of profiles and the option of temporarily editing the selected profile; changes are saved only for each job started (optional).
cursorMode	imis.scan.ui.CursorMode or CursorModeOptions	Button for selecting the insert or add mode (optional). The default mode is adding to the end.
print	imis.scan.ui.Button or ButtonOptions	Button for printing the current job (optional).

2023 © MARGIS Inc., All rights reserved.

6.2.2 imis.scan.ui.Button

This object represents the button component, which represents the basic component that can be used to control the start, continuation, cancellation and download of a job.

Image 45: Button component “Scan”

Constructor

imis.scan.ui.Button(options)	Creates a new button. Parameters: - options: ButtonOptions

Methods

disable()	Disable button.
enable()	Enable button.
showProgress()	Shows progress within the button.
hideProgress()	Hides progress within the button.
click()	Triggers a click on the component.
destroy()	Destroys all component sources.

Properties

app	imis.scan.ui.Scan	Returns the main component.
tooltip	string	Returns or sets the tooltip.
progressEnabled	boolean	Returns or sets whether the progress indicator is enabled.
enabled	boolean	Returns or sets whether the component is enabled.

2023 © MARGIS Inc., All rights reserved.

6.2.2.1 ButtonOptions

This object defines the properties that can be set on the button.

id	string	Unique identifier of an element in an HTML document as the id attribute.
element	HTMLElement	Element in HTML document (optional).
text	string	Button text (optional).
tooltip	string	Contents of the popup notification below the button (optional).
darkMode	boolean	Dark display mode (optional, default value *false*).
color	string	Color of button text (optional).
backgroundColor	string	Color of button background (optional).
fontSize	string	Size of button text (optional).
width	string	Minimum button width (optional).
height	string	Button height (optional).
onClick	callback	Callback on clicking on the button (optional). Parameters: - callback: function()
icon	string	The icon displayed on the left side of the button; the icon is of the Material Icons type (optional).
customStyle	boolean	Specifies whether the external appearance of the button has been defined. Only the functionalities connected to the job work (optional, default value *false*).

6.2.3 imis.scan.ui.SplitButton

This object represents the button component, which represents the basic component used to control the start and continuation of a job.

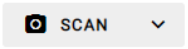

Image 46: Button component

Constructor

imis.scan.ui.SplitButton(options)	Creates a new button. Parameters: - options: ButtonOptions

Method

disable()	Disables the button.
enable()	Enables the button.
showProgress()	Shows progress within the button.
hideProgress()	Hides progress within the button.
click()	Triggers a click on the component.
destroy()	Destroys all component sources.

Properties

app	imis.scan.ui.Scan	Returns the main component.
tooltip	string	Returns or sets the tooltip.
progressEnabled	boolean	Returns or sets whether the progress indicator is enabled.
enabled	boolean	Returns or sets whether the component is enabled.

6.2.3.1 SplitButtonOptions

This object defines the properties that can be set on the button.

id	string	Unique identifier of an element in the HTML document as the “id” attribute.
element	HTMLElement	The element in the HTML document (optional).
text	string	Button text (optional).
tooltip	string	Contents of the tooltip under the button (optional).
darkMode	boolean	Dark display mode (optional, default value *false*).
color	string	Color of button text (optional).
backgroundColor	string	Color of button background (optional).
fontSize	string	Size of button text (optional).

2023 © MARGIS Inc., All rights reserved.

width	string	Minimum button width (optional).
height	string	Button height (optional).
onClick	callback	Callback on clicking the button (optional). Parameters: - callback: function()
icon	string	The icon displayed on the left side of the button; the icon is of the Material Icons type (optional).
customStyle	boolean	Specifies whether the external appearance of the button has been defined. Only the functionalities connected to the job work (optional, default value *false*).

6.2.4 imis.scan.ui.ColorDropdownButton

This object represents the component for selecting the color of scanning. This value only affects the job which will be started by pressing the scan button. This component will display values if a scanner module exists in the selected profile and if it has color selection values.

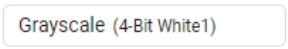

Image 47: Component for selecting the color of scanning

Constructor

imis.scan.ui.ColorDropdownButton(options)	Creates a new dropdown menu for selecting the color of scanning. Parameters: - options: ColorDropdownOptions

6.2.4.1 ColorDropdownOptions

This object defines the properties that can be set on the component for selecting the scanning color.

id	string	Unique identifier of an element in an HTML document.
element	HTMLElement	Element in HTML document (optional).

2023 © MARGIS Inc., All rights reserved.

6.2.5 imis.scan.ui.ProfilesButton

This object represents the component for selecting the profile and changing the settings of the scanning profile. Changes to the profile will be applied only at the start of each new job.
The selected profile will be saved to the browser (Local Storage), if saving has been enabled.

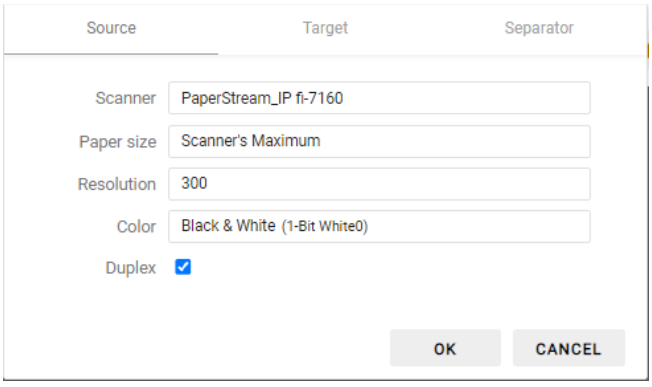

Image 48: Component for selecting the profile and changing the settings of the scanning profile

Constructor

imis.scan.ui.ProfilesButton(options)	Creates a new dropdown menu for selecting the profile with the option of changing the settings for a new job. Parameters: - options: ProfilesButtonOptions

6.2.5.1 ProfilesButtonOptions

This object defines the properties that can be set on the component for selecting the profile and temporarily changing the settings.

id	string	The unique identifier of an element in an HTML document.
element	HTMLElement	Element in HTML document (optional).
color	string	Color of the titles of profile modules.
darkMode	boolean	Dark display mode (optional, default value false).

2023 © MARGIS Inc., All rights reserved.

6.2.6 imis.scan.ui.ImageDetails

This object represents the component for displaying information about the currently selected page. It shows the recognized barcodes if a barcode recognition module exists. By placing the cursor on the barcode, the latter is tagged in the component which shows the full-size page. It also shows the redactions, which can be removed by selecting the Remove action.

For it to work, the component imis.scan.ui.ImageView or imis.scan.ui.ImageScroll in imis.scan.ui.Scan must be used.

Image 49: Component for displaying information about the currently selected page

Constructor

imis.scan.ui.ImageDetails(options)	Creates a component for displaying details of the selected page. Parameters: - options: ImageDetailsOptions

6.2.6.1 ImageDetailsOptions

This object defines the properties that can be set on the component for showing information on the currently selected page.

id	string	The unique identifier of an element in an HTML document.
element	HTMLElement	Element in HTML document (optional).
color	string	Color of the page property.
closed	boolean	Specifies whether the component is hidden (optional, default value false).
close	boolean	Specifies whether the close button is shown (optional, default value true).
darkMode	boolean	Dark display mode (optional, default value false).
onClose	callback	Callback on clicking on the close button (optional).
background	string	Background color (optional).

6.2.7 imis.scan.ui.ImageView

This object represents the component for displaying the currently selected page.

This component enables changing the size (zooming in/zooming out or showing the whole page), moving to the next or previous page, changing the orientation, deleting a page, adding redactions, cropping a page.

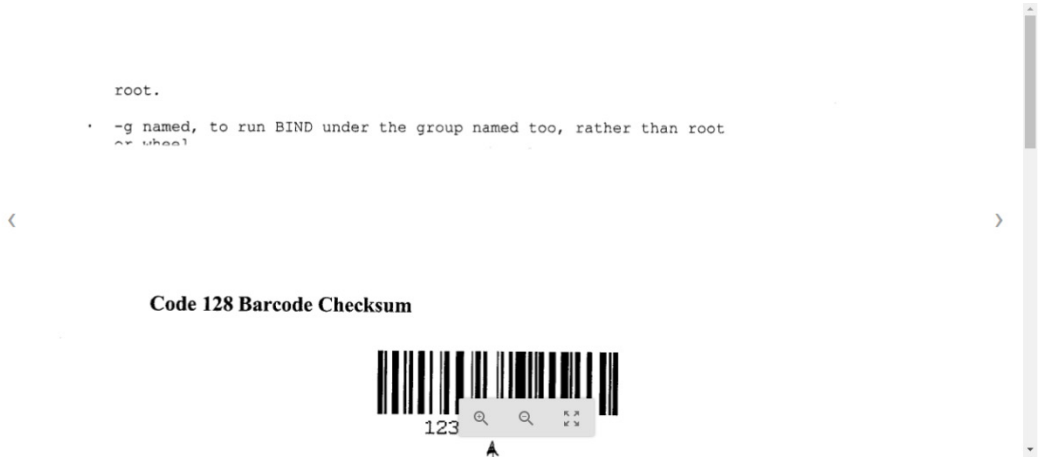

Image 50: Component for displaying the currently selected page

Constructor

imis.scan.ui.ImageView(options)	Creates a component for displaying the selected page. Parameters: - options: ImageViewOptions

6.2.7.1 ImageViewOptions

action.crop	boolean	Specifies whether cropping is enabled (optional, default value *true*).
action.delete	boolean	Specifies whether deleting is enabled (optional, default value *true*).
action.redaction	boolean	Specifies whether the adding of redactions is enabled (optional, default value *true*).
action.rotate	boolean	Specifies whether the changing of orientations is enabled (optional, default value *true*).
action.region	boolean	Specifies whether the region can be set.
background	string	Background color (optional).
fitToSize	boolean	Specifies whether each newly selected image is automatically adjusted to the size of the element (optional, default value *false*).
highlightColor	string|undefined	The selected highlight color.
highlightColors	string[]|undefined	Collection of highlight colors.
id	string	Unique identifier of an element in an HTML document as the id attribute.
element	HTMLElement	Element in HTML document (optional).
onPropertiesSelected	callback	Callback on selecting properties in the context menu (optional). callback: function()
zoomFactor	number|undefined	The zoom factor, the default value is 0.3.

2023 © MARGIS Inc., All rights reserved.

6.2.8 imis.scan.ui.ImageScroll

This object represents the component for displaying a collection of pages. This component enables changing the size (zooming in/zooming out or showing the whole page), moving to the next or previous page, changing the orientation, deleting a page, adding redactions, cropping a page.

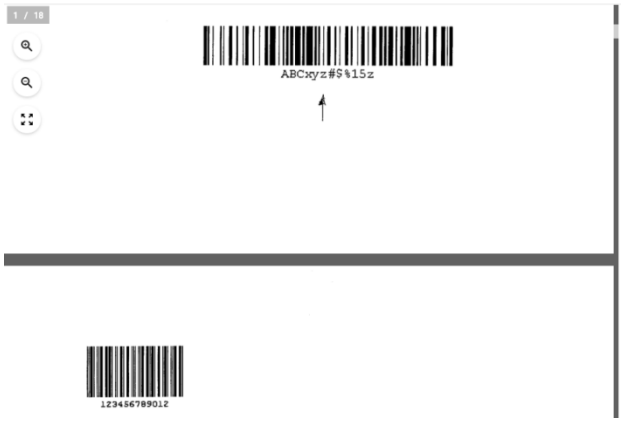

Image 51: Component for displaying a collection of pages

Constructor

imis.scan.ui.ImageScroll(options)	Creates a component for displaying a collection of pages. Parameters: - options: ImageScrollOptions

6.2.8.1 ImageScrollOptions

action.crop	boolean	Specifies whether cropping is enabled (optional, default value *true*).
action.delete	boolean	Specifies whether deleting is enabled (optional, default value *true*).
action.redaction	boolean	Specifies whether the adding of redactions is enabled (optional, default value *true*).
action.rotate	boolean	Specifies whether the changing of orientations is enabled (optional, default value *true*).

2023 © MARGIS Inc., All rights reserved.

action.region	boolean	Specifies whether the region can be set.
background	string	Background color (optional).
contextMenu.enabled	boolean	Specifies whether a context menu for an individual page has been enabled (optional).
contextMenu.onPropertiesSelected	callback	Callback on selecting properties in the context menu (optional). callback: function()
controls	boolean	Shows controls for zooming in, zooming out, adjusting the image to the screen, the add redactions mode and the crop mode (optional, default value *true*).
darkMode	boolean	Dark display mode (optional, default value *false*).
element	HTMLElement	Element in HTML document (optional).
fitToWidth	boolean|undefined	Defines whether by default the page takes up the entire component width.
focusNewPage	boolean	When adding a new page, it is shown in focus (optional, default value *true*).
highlightColor	string|undefined	The selected highlight color.
highlightColors	string[]|undefined	Collection of highlight colors.
id	string	Unique identifier of an element in an HTML document as the id attribute.
zoomFactor	number|undefined	Defines the zoom factor, the default value is 0.3.

2023 © MARGIS Inc., All rights reserved.

6.2.9 imis.scan.ui.Progress

This object represents the component for showing the progress of the current job.

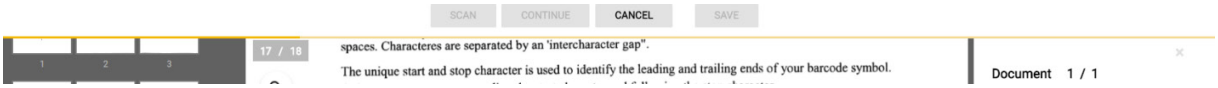

Image 52: Component for showing the progress of the current job

Constructor

imis.scan.ui.Progress(options)	Creates a component for showing the progress of the current job. Parameters: - options: ProgressOptions

6.2.9.1 ProgressOptions

This object defines the properties that can be set on the component for showing the progress of the job.

id	string	The unique identifier of an element in an HTML document.
element	HTMLElement	Element in HTML document (optional).
darkMode	boolean	Dark display mode (optional, default value *false*).
color	string	Component color while executing the job (optional).

6.2.10 imis.scan.ui.Status

This object represents the component for displaying the status, which specifies whether a connection with IMiS®/Capture Service has been established.

Image 53: Component for displaying the status

Constructor

imis.scan.ui.Status(options)	Creates a component for displaying the status. Parameters: - options: StatusOptions

6.2.10.1 StatusOptions

This object defines the properties that can be set on the component for showing the status.

id	string	The unique identifier of an element in an HTML document.
element	HTMLElement	Element in HTML document (optional).

6.2.11 imis.scan.ui.Thumbnails

This object represents the component for showing documents and pages. This component enables adjusting the size of individual page previews, orientation of a collection of documents, or the gallery mode, which shows the page and details in a dialog. With the context menu you can move an individual page, change the orientation of an individual page, delete an individual page, or show details. This component enables the moving of an individual page. With the cursor you can set the position of subsequent scanning, where the add or overwrite mode is defined in the component imis.scan.ui.CursorMode.

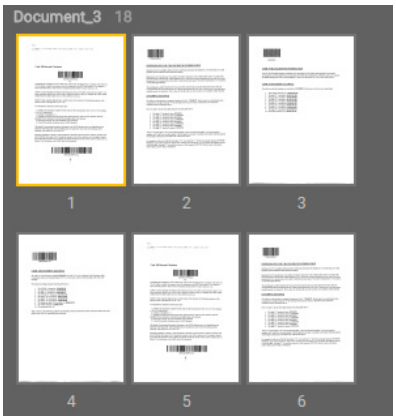

Image 54: Component for showing documents

Constructor

imis.scan.ui.Thumbnails(options)	Creates a component for showing the page as a preview. Parameters: - options: ThumbnailsOptions

6.2.11.1 ThumbnailsOptions

backgroundColor	string	Background color (optional).
cursor.enabled	boolean	Specifies whether the cursor for adding or overwriting pages is enabled (optional, default value *false*).
cursor.color	string	Color of cursor background (optional).
darkMode	boolean	Dark display mode (optional, default value *false*).
document.action.delete	boolean	Specifies whether deletion of a document is enabled (optional, default value *true*).
document.contextMenu.enabled	boolean	Specifies whether the context menu for an individual document is enabled (optional, default value *true*).
document.color	string	Color of the document title (optional).
document.icon	string	Documents icon, possible set of icons from Material Icons.
document.iconColor	string	Color of the document icon (optional).
document.separatorColor	string	Color of the document tag (optional).
element	HTMLElement	Element in HTML document (optional).
focusNewPage	boolean	When adding a new page, it is shown in focus (optional, default value *true*).
gallery.enabled	boolean	The gallery mode; enables double-clicking on an individual page, which opens a dialog with the zoomed-in page and details (optional, default value *false*).
gallery.fitToSize	boolean	Specifies whether the pages adjust to the dialog size (optional, default value *false*).

2023 © MARGIS Inc., All rights reserved.

gallery.action.crop	boolean	Specifies whether cropping is enabled in a dialog (optional, default value *true*).
gallery.action.delete	boolean	Specifies whether deleting a page is enabled in a dialog (optional, default value *true*).
gallery.action.redaction	boolean	Specifies whether the adding of redactions is enabled in a dialog (optional, default value *true*).
gallery.action.rotate	boolean	Specifies whether the changing of page orientation is enabled in a dialog (optional, default value *true*).
gallery.contextMenu.enabled	boolean	Specifies whether the context menu is enabled for an individual page in the dialog (optional, default value *true*).
id	string	Unique identifier of an element in an HTML document.
orientation	string	Orientation of a collection of documents. Set of values: - horizontal; - vertical. Default value *horizontal* (optional).
thumbnail.action.delete	boolean	Specifies whether deleting a page is enabled (optional, default value *true*).
thumbnail.action.move	boolean	Specifies whether the moving of a page is enabled (optional, default value *true*).
thumbnail.action.rotate	boolean	Specifies whether the changing of page orientation is enabled (optional, default value *true*).
thumbnail.contextMenu.enabled	boolean	Specifies whether the context menu is enabled for an individual page (optional, default value *true*).
thumbnail.contextMenu.onPropertiesSelected	callback	Callback on selecting properties in the context menu (optional). callback: function()
thumbnail.selectedColor	string	Color of the line of the selected page (optional).
thumbnail.textColor	string	Color of the text of the sequential page number (optional).
thumbnail.width	number	Page width (optional).

2023 © MARGIS Inc., All rights reserved.

thumbnail.height	number	Page height (optional, default value 150).
thumbnail.title	boolean	Specifies whether the page title is shown (optional, default value *true*).
thumbnail.titleColor	string	Color of the page title (optional).
thumbnail.mime	string	Page type. The set of values: image/jpeg, image/png, image/bmp, image/gif.

6.2.12 imis.scan.ui.Settings

This object represents the component for setting profiles. This component enables showing, adding, changing, deleting or locking profiles.

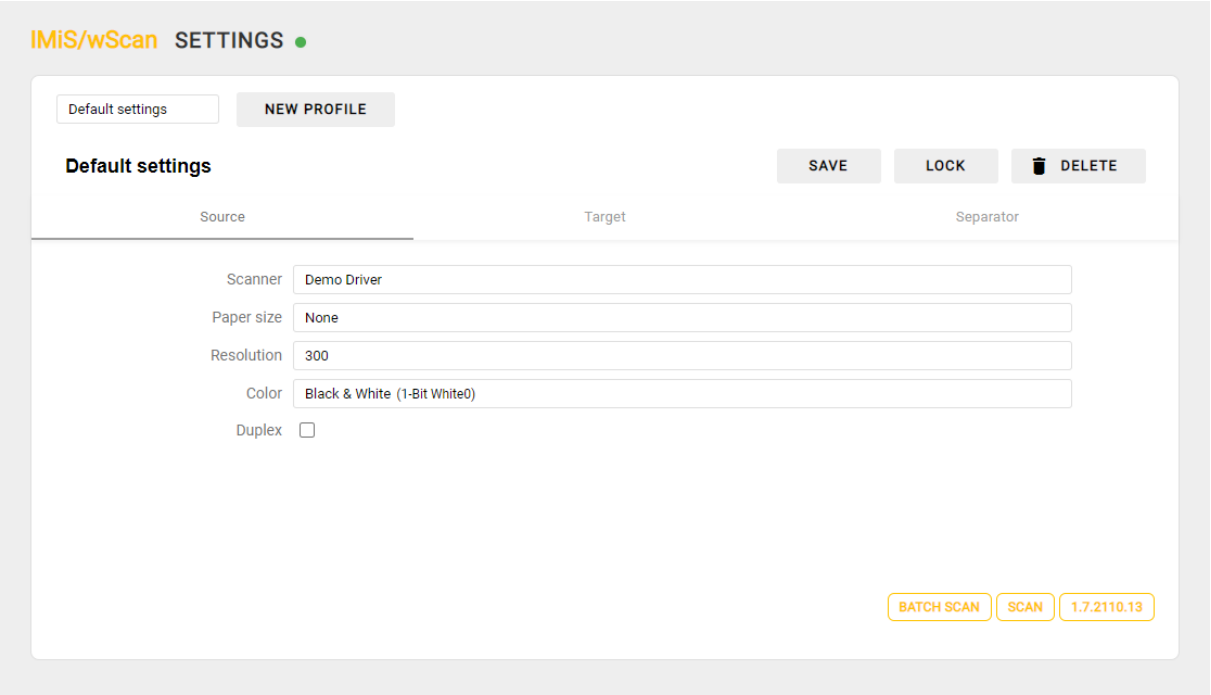

Image 55: Component for setting profiles

Constructor

imis.scan.ui.Settings(options)	Creates and displays a new component for showing the settings. Parameters: - options: SettingsOptions

6.2.12.1 SettingsOptions

This object defines the properties that can be set on the component for setting profiles.

id	string	The unique identifier of an element in an HTML document.
element	HTMLElement	Element in HTML document (optional).

6.2.13 imis.scan.ui.AlertDialog

This object represents the component for displaying a dialog. This component enables setting the title and text and detecting whether a user has confirmed or canceled a dialog.

Image 56: Component for displaying a dialog

Constructor

imis.scan.ui.AlertDialog(options)	Creates and displays a new dialog. Parameters: - options: AlertDialogOptions

6.2.13.1 AlertDialogOptions

This object defines the properties that can be set on the component for showing the dialog.

title	string	Dialog title.
text	string	Dialog content.
ok	callback	Callback on pressing the OK button. callback: function()
cancel	callback	Callback on pressing the cancel button. callback: function()

6.2.14 imis.scan.ui.TargetColor

This object represents the component for showing the currently selected color for saving the job. This component enables setting the color.

Constructor

imis.scan.ui.TargetColor(options)	Creates and displays a component if values exist. Parameters: - options: TargetColorOptions

6.2.14.1 TargetColorOptions

This object defines the properties that can be set on the component for showing the color for saving.

id	string	The unique identifier of an element in an HTML document.
element	HTMLElement	Element in HTML document (optional).
color	string	Component color (optional).

2023 © MARGIS Inc., All rights reserved.

6.2.15 imis.scan.ui.TargetFormat

This object represents the component for showing the currently selected format for saving the job. This component enables setting the color.

Constructor

imis.scan.ui.TargetFormat(options)	Creates and displays a component if values exist. Parameters: - options: TargetFormatOptions

6.2.15.1 TargetFormatOptions

This object defines the properties that can be set on the component for showing the format for saving.

Id	string	The unique identifier of an element in an HTML document.
element	HTMLElement	Element in HTML document (optional).
Color	string	Component color (optional).

6.2.16 imis.scan.ui.TotalDocuments

This object represents the component for showing the number of documents in the current job. This component enables setting the color.

Constructor

imis.scan.ui.TotalDocuments(options)	Creates and displays a component if values exist. Parameters: - options: TotalDocumentsOptions

2023 © MARGIS Inc., All rights reserved.

6.2.16.1 TotalDocumentsOptions

This object defines the properties that can be set on the component for showing the number of documents.

Id	string	The unique identifier of an element in an HTML document.
element	HTMLElement	Element in HTML document (optional).
Color	string	Component color (optional).

6.2.17 imis.scan.ui.TotalPages

This object represents the component for showing the number of pages in the current job. This component enables setting the color.

Constructor

imis.scan.ui.TotalPages(options)	Creates and displays a component if values exist. Parameters: - options: TotalPagesOptions

6.2.17.1 TotalPagesOptions

This object defines the properties that can be set on the component for showing the number of pages.

id	string	The unique identifier of an element in an HTML document.
element	HTMLElement	Element in HTML document (optional).
color	string	Component color (optional).

2023 © MARGIS Inc., All rights reserved.

6.2.18 imis.scan.ui.CursorMode

This object represents the component for selecting the mode of adding or overwriting new pages in the current job.

Constructor

imis.scan.ui.CursorMode(options)	Creates and displays a component. Parameters: - options: CursorModeOptions

6.2.18.1 CursorModeOptions

This object defines the properties that can be set on the component for display.

id	string	The unique identifier of an element in an HTML document.
element	HTMLElement	Element in HTML document (optional).
darkMode	boolean	Dark display mode (optional, default value *false*).

6.2.19 imis.scan.ui.ResolutionDropdownButton

This object represents the component for selecting the scanning resolution. This value only affects the job which will be started by pressing the scan button. This component will display values if a scanning module exists in the selected profile and if it has resolution selection values.

Image 57: Component for selecting the scanning resolution

Constructor

imis.scan.ui.ResolutionDropdownButton(options)	Creates a new dropdown menu for selecting the scanning resolution. Parameters: - options: ResolutionDropdownButtonOptions

2023 © MARGIS Inc., All rights reserved.

6.2.19.1 ResolutionDropdownButtonOptions

This object defines the properties that can be set for the component for selecting the scanning resolution.

id	String	Unique identifier of an element in an HTML document.
Element	HTMLElement	An element in an HTML document (optional).

6.3 Examples of use imis.scan.js

These examples show the use of the “imis.scan.js” library. Developers can easily learn to use the library with the help of these prepared examples. Examples of reading profiles, modifying a profile, starting a job and deleting a job are shown. These examples can be accessed from the start page if they were turned on during installation and are executable.

6.3.1 Reading profiles

An example of reading profiles shows the basic use of the library. If reading is successful, a collection of profiles will appear in the feature with the ‘profiles’ identifier; if an error occurs, it will be visible in the feature with the ‘error’ identifier. When the page has finished loading, a scan object will be created, which will be used to read all the profiles. If they were read successfully, they will appear on a list, if not, an error will appear.

```
<!DOCTYPE html>
<html>
<head>
   <title>imis.scan.js</title>
   <link rel="stylesheet" href="https://fonts.googleapis.com/css?family=Roboto" />
   <link rel="stylesheet" href="sample.css" />
</head>
<body class="sample">
  <h1>Sample</h1>
  <p>This example demonstrates reading scan profiles.</p>

  <div>Profiles:</div>
  <ol id="profiles"></ol>
  <div id="error"></div>

  <script src="../imis.scan.js"></script>
  <script>
    window.addEventListener('load', function () {
      try {
        // Profiles ordered list
        var ol = document.getElementById("profiles");

        // Create a scan object
        var scan = new imis.scan.Scan(
```

```
         apiKey = API_KEY);

        // Read profiles
        scan.getProfiles({
          success: function (profiles) {
            for (var i = 0; i < profiles.length; i++) {
              // Add profile to ordered list
              var li = document.createElement("li");
              li.innerHTML = profiles[i].name;
              ol.appendChild(li);
            }
          },
          error: function (error) {
            // Show error
            document.getElementById("error").innerHTML = error;
          }
        });
      } catch (e) {
        // Show error
        document.getElementById("error").innerHTML = e;
      }
    });
  </script>
</body>
</html>
```

6.3.2 Changing a profile

An example of changing the profile name shows the basis for changing profile properties.
A collection of all profiles is loaded to the 'select' feature with the 'profiles' identifier, where you select the current profile to be changed. Clicking on the 'button' feature with the 'btn-update' identifier initiates a profile update and updates the collection of profiles. When the page has finished loading, a scan object will be created, which will be used to read all the profiles.
If they were read successfully, they will appear on a list and the name of the selected profile will be loaded to the input field, if not, an error will appear.

When changing the profile in the drop-down menu, the input field will be updated with the profile name. When pressing the Update button, the new profile name will be saved to the profile and this change will be saved to the server; if saved successfully, the list of profiles will be updated, if not, an error will appear.

```
<!DOCTYPE html>
<html>
<head>
  <title>imis.scan.js</title>
  <link rel="stylesheet" href="https://fonts.googleapis.com/css?family=Roboto" />
  <link rel="stylesheet" href="sample.css" />
</head>
<body class="sample">
  <h1>Sample</h1>
  <p>This example demonstrates updating profile name.</p>

  <h2>Update profile</h2>
  Select profile <select id="profiles"></select>

  <h3>Edit</h3>
```

2023 © MARGIS Inc., All rights reserved.

```
  Profile name <input id="profile-name" type="text" placeholder="Profile name" />
  <button id="btn-update">Update</button>
  <div id="error"></div>

  <script src="../imis.scan.js"></script>
  <script>
    window.addEventListener('load', function () {
      var profilesList = [], // Profiles list
          selectedProfile = null, // Selected profile
          profilesSelectUI = document.getElementById("profiles"), // Profiles drop-down list
          profileNameUI = document.getElementById("profile-name"); // Selected profile name
input text

      try {
        // Create a scan object
        var scan = new imis.scan.Scan(
         apiKey = API_KEY);

        // Load profiles to drop-down list
        var load = function () {
          scan.getProfiles({
            success: function (profiles) {
              profilesList = profiles;
              // Clear options
              profilesSelectUI.innerHTML = "";

              for (var i = 0; i < profiles.length; i++) {
                // Add profile option
                const profile = profiles[i];
                var option = document.createElement("option");
                option.value = profile.id;
                option.text = profile.name;
                if (null === selectedProfile)
                  selectedProfile = profile;
                option.selected = profile.equals(selectedProfile)
                profilesSelectUI.add(option);
              }

              // Update selected profile name text input
              if (null !== selectedProfile)
                profileNameUI.value = selectedProfile.name;
            },
            error: function (error) {
              // Show error
              document.getElementById("error").innerHTML = error;
            }
          });
        };
        // Call load
        load();

        // Selected profile change listener
        profilesSelectUI.addEventListener("change", function () {
          // Update selected profile
          selectedProfile = null;
          var selectedValue = profilesSelectUI.options[profilesSelectUI.selectedIndex].value;
          for (var i = 0; i < profilesList.length; i++) {
            if (profilesList[i].id === selectedValue) {
              selectedProfile = profilesList[i];
              break;
            }
          }

          // Update selected profile name text input
          if (null !== selectedProfile)
            profileNameUI.value = selectedProfile.name;
        });

        // Update button click listener
        document.getElementById("btn-update").addEventListener("click", function () {
```

```
          if (null == selectedProfile || null === profileNameUI.value || "" ===
profileNameUI.value ||
              selectedProfile.name === profileNameUI.value)
              return;

          // Update profile name
          selectedProfile.name = profileNameUI.value;

          // Save profile
          scan.updateProfile({
            profile: selectedProfile,
            success: function (profile) {
              // Load profiles
              load();
            },
            error: function (error) {
              // Show error
              document.getElementById("error").innerHTML = error;
            }
          });
        });
      } catch (e) {
        // Show error
        document.getElementById("error").innerHTML = e;
      }
    });
  </script>
</body>
</html>
```

6.3.3 Starting a job

An example of starting a job; shows job management and displays the scanning result.

A collection of all profiles is loaded to the 'select' feature with the 'profiles' identifier, where you select the current profile to create and start a job. Clicking on the 'button' feature with the 'btn-start' identifier initiates the creation and start of a job.

Status is shown in the feature with the 'job-progress' identifier. The entire job can be downloaded by clicking on the feature with the 'job-download' identifier, after the job has been finished.

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8" />
  <title>imis.scan.js</title>
  <link rel="stylesheet" href="https://fonts.googleapis.com/css?family=Roboto" />
  <link rel="stylesheet" href="sample.css" />
</head>
<body class="sample">
  <h1>Sample</h1>
  <p>This example demonstrates starting job and displaying documents and pages.</p>

  <h2>Start Job</h2>
  <select id="profiles"></select>
  <button id="btn-start">Start</button>
  <br /><br />

  <div>Status: <span id="job-progress">None</span></div>
  <div id="error"></div>
```

2023 © MARGIS Inc., All rights reserved.

```
  <a id="job-download" href="#">Download</a>
  <br /><br />

  <div><b>Documents</b></div>
  <div id="job">None</div>

  <script src="../imis.scan.js"></script>
  <script>
    window.addEventListener('load', function () {
      var profilesList = [], // Profiles list
          profilesSelectUI = document.getElementById("profiles"), // Profiles drop-down list
          jobUI = document.getElementById("job"); // Job documents

      try {
        // Create a scan object
        var scan = new imis.scan.Scan(
         apiKey = API_KEY);
        // Get profiles
        scan.getProfiles({
          success: function (profiles) {
            profilesList = profiles;
            // Clear options
            profilesSelectUI.innerHTML = "";

            for (var i = 0; i < profiles.length; i++) {
              // Add profile option
              const profile = profiles[i];
              var option = document.createElement("option");
              option.value = profile.id;
              option.text = profile.name;
              profilesSelectUI.add(option);
            }
          },
          error: function (error) {
            console.error("getProfiles: " + error);
          }
        });

        // Start job button click listener
        document.getElementById("btn-start").addEventListener("click", function () {
          // Clear download link
          document.getElementById("job-download").setAttribute("href", "#");

          // Find selected profile
          var selectedValue = profilesSelectUI.options[profilesSelectUI.selectedIndex].value;
          var profile = null;
          for (var i = 0; i < profilesList.length; i++) {
            if (profilesList[i].id === selectedValue) {
              profile = profilesList[i];
              break;
            }
          }
          if (null == profile)
            return;

          // Create job
          scan.createJob({
            profile: profile.id,
            success: function (job) { // Job successfully created

              // Clear job documents
              jobUI.innerHTML = "";

              // Start job
              job.start({
                success: function () { // Job successfully started
                  // Update job progress
                  document.getElementById("job-progress").innerHTML =
imis.scan.JobStatus.toString[job.status];

                  // Job changed callback
```

```
                  job.onChange(function (job) {
                    // Update job progress
                    document.getElementById("job-progress").innerHTML =
imis.scan.JobStatus.toString[job.status];
                    if (imis.scan.JobStatus.COMPLETED === job.status)
                      document.getElementById("job-download").setAttribute("href",
job.download);
                  });

                  // Document created callback
                  job.onCreateDocument(function (newDocument) {
                    const documentElement = document.createElement("div");
                    documentElement.style.marginBottom = "55px";
                    // Document name
                    const documentName = document.createElement("div");
                    documentName.style.fontWeight = "bold";
                    documentName.style.fontSize = "16px";
                    documentName.innerHTML = newDocument.name + " [" + newDocument.pageCount +
"]";
                    documentElement.appendChild(documentName);
                    jobUI.appendChild(documentElement);

                    // On create page callback
                    newDocument.onCreatePage(function (page) {
                      const pageElement = document.createElement("div");
                      pageElement.style.display = "inline-block";

                      // Page image
                      const img = document.createElement("img");
                      img.setAttribute("src", page.getThumbnailUri({ height: 150 }));
                      pageElement.appendChild(img);

                      // Page index
                      const pageIndex = document.createElement("div");
                      pageIndex.innerHTML = "Page " + page.index;
                      pageElement.appendChild(pageIndex);

                      documentElement.appendChild(pageElement);
                    });

                    // Document change callback
                    newDocument.onChange(function (changedDocument) {
                      documentName.innerHTML = changedDocument.name + " [" +
changedDocument.pageCount + "]";
                    });
                  });
                },
                error: function (error) {
                  // Show error
                  document.getElementById("error").innerHTML = error;
                }
              });
            },
            error: function (error) {
              // Show error
              document.getElementById("error").innerHTML = error;
            }
          });
        });
      } catch (e) {
        // Show error
        document.getElementById("error").innerHTML = e;
      }
    });
  </script>
</body>
</html>
```

6.3.4 Deleting a profile

An example of deleting a profile shows the basic use of the library. If read successfully, the collection of profiles will appear in the drop-down menu, where you can select the profile for deletion. The profile will be deleted by clicking on the delete button and confirming the deletion dialog. This example deletes a profile saved on the server.

```
<!DOCTYPE html>
<html>
<head>
  <title>imis.scan.js</title>
  <link rel="stylesheet" href="https://fonts.googleapis.com/css?family=Roboto" />
  <link rel="stylesheet" href="sample.css" />
</head>
<body class="sample">
  <h1>Sample</h1>
  <p>This example demonstrates deleting profile.</p>

  <h2>Delete profile</h2>
  <select id="profiles"></select>
  <button id="btn-delete">Delete</button>
  <div id="error"></div>

  <script src="../imis.scan.js"></script>
  <script>
  window.addEventListener('load', function () {
    var profilesList = [], // Profiles list
        profilesSelectUI = document.getElementById("profiles");  // Profiles drop-down list

    try {
      // Create a scan object
      var scan = new imis.scan.Scan(
       apiKey = API_KEY);

      // Load profiles to drop-down list
      var load = function () {
        scan.getProfiles({
          success: function (profiles) {
            profilesList = profiles;
            // Clear options
            profilesSelectUI.innerHTML = "";

            for (var i = 0; i < profiles.length; i++) {
              // Add profile option
              const profile = profiles[i];
              var option = document.createElement("option");
              option.value = profile.id;
              option.text = profile.name;
              profilesSelectUI.add(option);
            }
          },
          error: function (error) {
            // Show error
            document.getElementById("error").innerHTML = error;
          }
        });
      }
      // Call load
      load();

      document.getElementById("btn-delete").addEventListener("click", function () {
        // Find selected profile
```

```
        var selectedValue = profilesSelectUI.options[profilesSelectUI.selectedIndex].value;
        var profile = null;
        for (var i = 0; i < profilesList.length; i++) {
          if (profilesList[i].id === selectedValue) {
            profile = profilesList[i];
            break;
          }
        }
        if (null == profile)
          return;
        // Show confirmation dialog
        if (confirm("Do you want to delete profile '" + profile.name + "'?")) {
          // Delete profile
          scan.deleteProfile({
            profile: profile,
            success: function () {
              // Load profiles
              load();
            },
            error: function (error) {
              // Show error
              document.getElementById("error").innerHTML = error;
            }
          });
        }
      });
    } catch (e) {
      // Show error
      document.getElementById("error").innerHTML = e;
    }
  });
  </script>
</body>
</html>
```

6.4 Examples of use imis.scan.ui.js

These examples show the use of the “imis.scan.ui.js” library. Developers can easily learn to use the library with the help of these prepared examples. It shows examples of classic, modern, classic dark and gallery sample. These examples use different components with a specific position on the page. The developers can create a customized page layout from individual components.

These examples can be accessed from the start page if they were turned on during installation.

6.4.1 Classic sample

Classic sample is the most commonly used user interface example. Application developers choose it when they want to preserve the traditional appearance of the user interface.

The thumbnails of document pages are located on the left side of the user interface.

They are sorted by available space and thumbnail size (portrait, landscape). Document pages are shown in the center.

2023 © MARGIS Inc., All rights reserved.

The user navigates between pages with the slider. Page details are sorted on the right and can be easily closed or reopened by selecting the menu on an individual page.

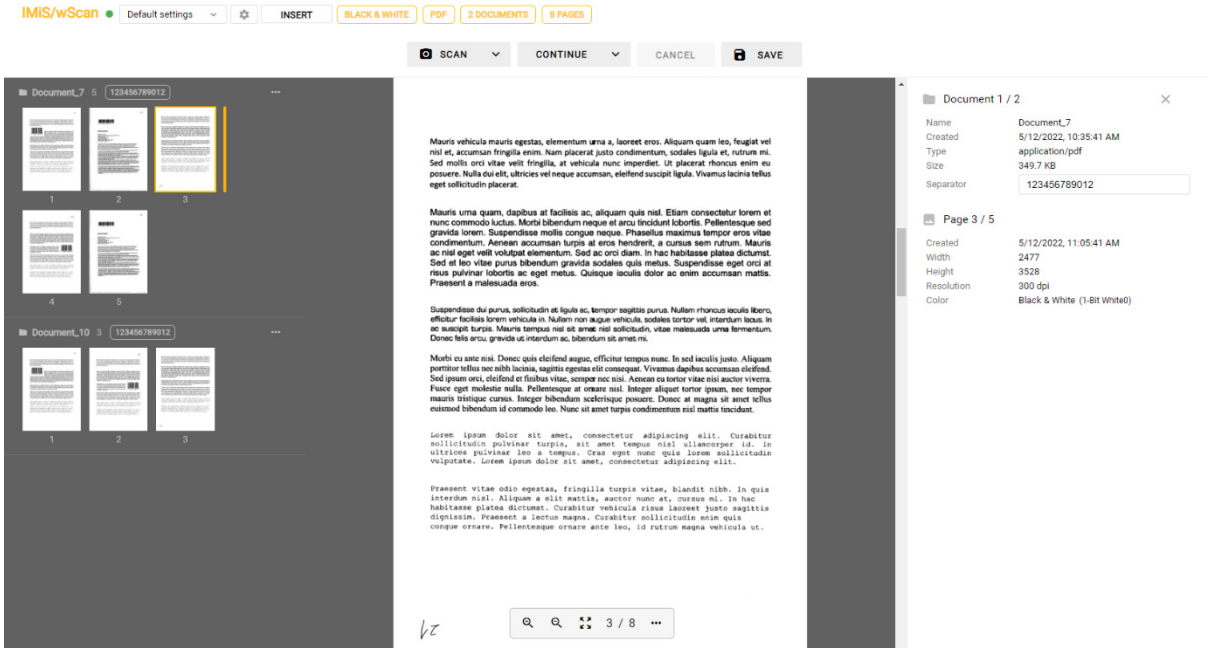

Image 58: Example of using the classic sample of a user interface display

In the classic sample, the following components are used:

- imis.scan.ui.Thumbnails
- imis.scan.ui.ImageDetails
- imis.scan.ui.Status
- imis.scan.ui.Progress
- imis.scan.ui.Button
- imis.scan.ui.ProfilesButton
- imis.scan.ui.TargetColor
- imis.scan.ui.TargetFormat
- imis.scan.ui.TotalDocuments
- imis.scan.ui.TotalPages
- imis.scan.ui.CursorMode

2023 © MARGIS Inc., All rights reserved.

6.4.1.1 classic.html

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8" />
  <title>IMiS/wScan Classic</title>
  <link rel="shortcut icon" type="image/png" href="img/favicon.png" />
  <link rel="stylesheet" href="imis.scan.ui.css" />
  <link rel="stylesheet" href="https://fonts.googleapis.com/css?family=Roboto" />
  <link rel="stylesheet" href="style/classic.css" />
</head>
<body>
  <div class="imis-scan-app">
    <div id="imis-progress"></div>
    <nav id="nav-top">
      <div id="title" class="title">
        <a href="index.html">IMiS/wScan</a>
        <div id="scan-status"></div>
      </div>
      <div id="imis-profile"></div>
      <div id="cursor-mode"></div>
      <div id="target-color"></div>
      <div id="target-format"></div>
      <div id="total-documents"></div>
      <div id="total-pages"></div>
    </nav>
    <nav id="nav">
      <div id="scan-btn"></div>
      <div id="continue-btn"></div>
      <div id="cancel-btn"></div>
      <div id="download-btn"></div>
    </nav>
    <div class="main" id="main">
      <div id="thumbnails" class="main-left"></div>
      <div id="images" class="main-center"></div>
      <div id="image-details" class="main-right"></div>
    </div>
  </div>
  <script src="apikey.js"></script>
  <script src="imis.scan.js"></script>
  <script src="imis.scan.ui.js"></script>
  <script>
    // Window load
    window.addEventListener("load", () => {
      try {
        // Create a scan object
        var scan = new imis.scan.ui.Scan({
          apiKey: API_KEY, // API_KEY is defined in apikey.js
          reconnect: true,
          thumbnails: {
            id: "thumbnails",
            orientation: "vertical",
            cursor: {
              enabled: true
            },
            document: {
              contextMenu: {
                onPropertiesSelected: () => showDetails()
              }
            },
            thumbnail: {
              height: 120,
              contextMenu: {
                onPropertiesSelected: () => showDetails()
              }
            }
          },
```

```
          imageDetails: {
            id: "image-details",
            onClose: () => hideDetails()
          },
          status: { id: "scan-status" },
          progress: { id: "imis-progress" },
          images: {
            id: "images",
            contextMenu: {
              onPropertiesSelected: () => showDetails()
            }
          },
          targetColor: { id: "target-color" },
          targetFormat: { id: "target-format" },
          totalDocuments: { id: "total-documents"},
          totalPages: { id: "total-pages" },
          buttons: {
            scan: { id: "scan-btn", icon: "camera_alt", progressEnabled: true },
            profiles: { id: "imis-profile" },
            download: { id: "download-btn", icon: "save_alt", progressEnabled: true },
            cancel: { id: "cancel-btn" },
            continue: { id: "continue-btn", progressEnabled: true },
            cursorMode: { id: "cursor-mode" }
          },
          onError: message => {
            new imis.scan.ui.AlertDialog({ title: "Error", text: message });
          }
        });
        scan.show();
        resize();
      } catch (e) {
        console.error(e);
        new imis.scan.ui.AlertDialog({ title: "Error", text: e });
      }
    });

    // Show details
    function showDetails() {
      document.getElementById("image-details").style.display = null;
      document.getElementById("images").style = null;
    }

    // Hide details
    function hideDetails() {
      document.getElementById("image-details").style.display = "none";
      document.getElementById("images").style.flex = 0.75;
    }

    // Resizes height of main element
    function resize() {
      var mainHeight = (window.innerHeight - document.getElementById("nav-top").offsetHeight
        - document.getElementById("nav").offsetHeight
        - document.getElementById("imis-progress").offsetHeight - 1);
      document.getElementById("main").style.height = mainHeight + "px";
    }

    // Resize event listener
    window.addEventListener('resize', event => resize());
  </script>
</body>
</html>
```

6.4.1.2 classic.css

```
body {
  margin: 0;
  background: #fff;
  height: 100%;
  width: 100%;
  font-family: 'Roboto', sans-serif;
}

a {
  text-decoration: none;
  color: inherit;
}

.title {
  display: inline-block;
  background: #fff;
  color: #FFC107;
  font-size: 20px;
  padding-top: 15px;
  font-weight: bold;
}

nav {
  margin: 0;
  position: relative;
  padding-bottom: 10px;
  padding-left: 25px;
  padding-right: 25px;
  background: #fff;
}

#cursor-mode {
  display: inline-block;
}

.main {
  background: #EEEEEE;
  display: flex;
  height: 500px;
  position: relative;
  box-shadow: 0 0 1px 1px #eee;
  z-index: 1000;
}
```

2023 © MARGIS Inc., All rights reserved.

```
.main-left,
.main-right {
  flex: 0.25;
}

.main-center {
  flex: 0.50;
}

#nav {
  display: table;
  padding-top: 15px;
  padding-bottom: 15px;
  width: fit-content;
  margin: auto;
}

#nav .imis-btn,
#nav .imis-split-btn {
  margin-right: 7px;
}
```

6.4.2 Modern sample

Modern sample is suitable for application developers that keep up with the more recent trends in user interface appearance. Document pages take up most of the user interface. By default, page details are located on the right. A user can easily close or reopen them by selecting the menu on an individual page. For greater transparency the thumbnails of document pages are sorted at the bottom, which is especially suitable for larger volumes of scanned documents.

2023 © MARGIS Inc., All rights reserved.

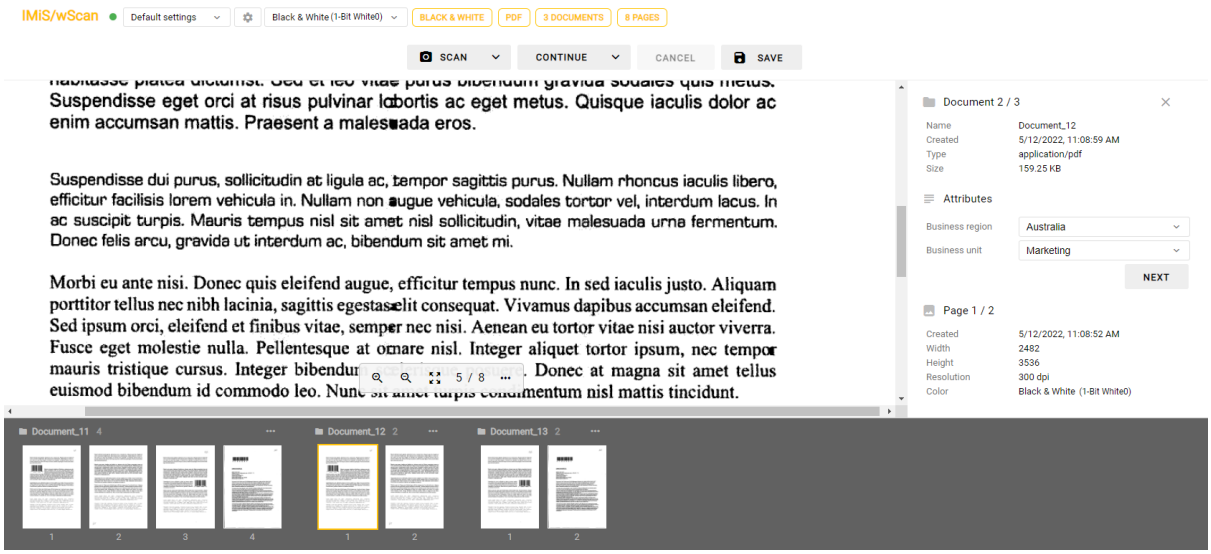

Image 59: Example of using the modern sample of a user interface display

In the modern sample, the following components are used:

- imis.scan.ui.Thumbnails
- imis.scan.ui.ImageView
- imis.scan.ui.ImageDetails
- imis.scan.ui.Status
- imis.scan.ui.Progress
- imis.scan.ui.Button
- imis.scan.ui.ProfilesButton
- imis.scan.ui.ColorDropdownButton
- imis.scan.ui.TargetColor
- imis.scan.ui.TargetFormat
- imis.scan.ui.TotalDocuments
- imis.scan.ui.TotalPages.

2023 © MARGIS Inc., All rights reserved.

6.4.2.1 modern.html

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8" />
  <title>IMiS/wScan Modern</title>
  <link rel="shortcut icon" type="image/png" href="img/favicon.png" />
  <link rel="stylesheet" href="imis.scan.ui.css" />
  <link href="https://fonts.googleapis.com/css?family=Roboto" rel="stylesheet">
  <link rel="stylesheet" href="style/modern.css" />
</head>
<body>
  <div id="imis-progress"></div>
  <nav id="nav-top">
    <div id="title" class="title">
      <a href="index.html">IMiS/wScan</a>
      <div id="scan-status"></div>
    </div>
    <div id="imis-profile"></div>
    <div id="imis-profile-color"></div>
    <div id="target-color"></div>
    <div id="target-format"></div>
    <div id="total-documents"></div>
    <div id="total-pages"></div>
  </nav>
  <nav id="nav">
    <div id="scan-btn"></div>
    <div id="continue-btn"></div>
    <div id="cancel-btn"></div>
    <div id="download-btn"></div>
  </nav>
  <div class="main" id="main">
    <div id="image-view" class="main-center"></div>
    <div id="image-details" class="main-right"></div>
  </div>
  <div id="thumbnails"></div>
  <script src="apikey.js"></script>
  <script src="imis.scan.js"></script>
  <script src="imis.scan.ui.js"></script>
  <script>
    window.addEventListener("load", () => {
      try {
        var scan = new imis.scan.ui.Scan({
          apiKey: API_KEY, // API_KEY is defined in apikey.js
          reconnect: true,
          thumbnails: {
            id: "thumbnails",
            document: {
              contextMenu: {
                onPropertiesSelected: () => showDetails()              }
            },
            thumbnail: {
              height: 120,
              contextMenu: {
                onPropertiesSelected: ( showDetails()
              }
            }
          },
          imageView: {
            id: "image-view",
            fitToSize: true,
            contextMenu: {
              onPropertiesSelected: () => showDetails()
            }
          },
          imageDetails: {
            id: "image-details",
            onClose: () => hideDetails()
```

```
          },
          status: { id: "scan-status" },
          progress: { id: "imis-progress" },
          targetColor: { id: "target-color" },
          targetFormat: { id: "target-format" },
          totalDocuments: { id: "total-documents" },
          totalPages: { id: "total-pages" },
          buttons: {
            scan: { id: "scan-btn", icon: "camera_alt" },
            profiles: { id: "imis-profile" },
            download: { id: "download-btn", icon: "save_alt" },
            cancel: { id: "cancel-btn" },
            continue: { id: "continue-btn" },
            color: { id: "imis-profile-color" }
          },
          onError: message => {
            // Display error dialog
            new imis.scan.ui.AlertDialog({ title: "Error", text: message });
          }
        });
        scan.show();
        resize();
      } catch (e) {
        console.error(e);
        // Display error dialog
        new imis.scan.ui.AlertDialog({ title: "Error", text: e });
      }
    });

    function showDetails() {
      // Show image details
      document.getElementById("image-details").style.display = null;
      document.getElementById("image-view").style.width = null;
    }

    function hideDetails() {
      // Hide image details
      document.getElementById("image-details").style.display = "none";
      document.getElementById("image-view").style.width = "100%";
    }

    // Resize main and thumbnails height
    function resize() {
      var navHeight = document.getElementById("nav-top").offsetHeight +
        document.getElementById("nav").offsetHeight + document.getElementById("imis-
progress").offsetHeight;
      var thumbnailsHeight = window.innerHeight * 0.37; // 37% of window height
      // Main height
      document.getElementById("main").style.height = (window.innerHeight - navHeight -
thumbnailsHeight) + "px";
      // Thumbnails height
      document.getElementById("thumbnails").style.height = thumbnailsHeight + "px";
    }

    // Resize event listener
    window.addEventListener('resize', event => resize());
  </script>
</body>
</html>
```

2023 © MARGIS Inc., All rights reserved.

6.4.2.2 modern.css

```
body {
  margin: 0;
  background: #fff;
  height: 100%;
  width: 100%;
  font-family: 'Roboto', sans-serif;
}

a {
  text-decoration: none;
  color: inherit;
}

.title {
  background: #fff;
  color: #FFC107;
  font-size: 20px;
  padding-top: 15px;
  font-weight: bold;
}

nav {
  margin: 0;
  position: relative;
  padding-bottom: 10px;
  padding-left: 25px;
  padding-right: 25px;
  background: #fff;
}

nav > div {
  display: inline-block;
}

.main {
  background: #EEEEEE;
  overflow: hidden;
  height: 500px;
  position: relative;
  box-shadow: 0 0 1px 1px #eee;
  z-index: 1000;
}
```

2023 © MARGIS Inc., All rights reserved.

```
.main-center {
  float: left;
  width: 75%;
}

.main-right {
  float: left;
  width: 25%;
}

.imis-status {
  display: inline-block;
  margin-left: 10px;
}

#nav {
  display: table;
  padding: 15px 0;
  width: fit-content;
  margin: auto;
}

#nav .imis-btn,
#nav .imis-split-btn {
  margin-right: 7px;
}
```

6.4.3 Classic dark sample

Classic dark sample follows the latest trends of important document viewers. Document pages are shown in the center. The user navigates between pages with the slider. Documents are not visually separated from one another. It is clear from the page details to which documents the pages belong. By default, page details are not displayed. The user opens them by selecting the menu on an individual page. They are shown on the right side of the user interface.
Thumbnails of document pages are not available.

2023 © MARGIS Inc., All rights reserved.

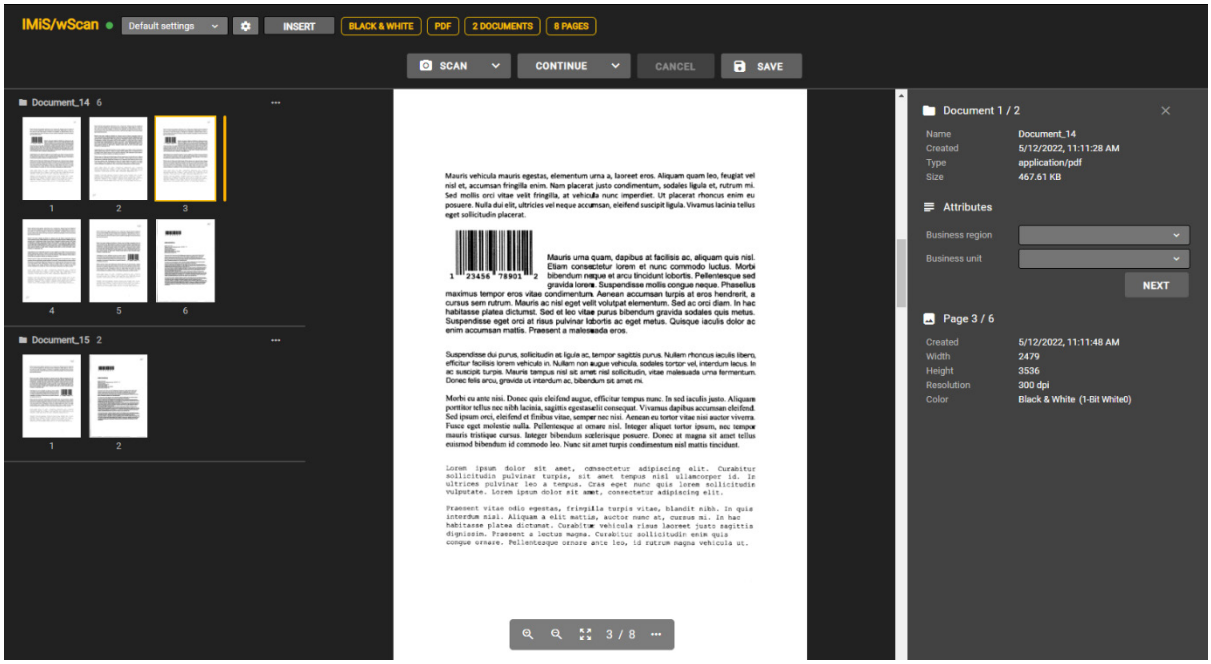

Image 60: Example of using the classic (dark) sample of a user interface display

In the classic (dark) sample, the following components are used:

- imis.scan.ui.ImageScroll
- imis.scan.ui.ImageDetails
- imis.scan.ui.Status
- imis.scan.ui.Progress
- imis.scan.ui.Button
- imis.scan.ui.ProfilesButton
- imis.scan.ui.TargetColor
- imis.scan.ui.TargetFormat
- imis.scan.ui.TotalDocuments
- imis.scan.ui.TotalPages
- imis.scan.ui.CursorMode.

2023 © MARGIS Inc., All rights reserved.

6.4.3.1 classic_dark.html

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8" />
  <title>IMiS/wScan Classic (dark)</title>
  <link rel="shortcut icon" type="image/png" href="img/favicon.png" />
  <link rel="stylesheet" href="imis.scan.ui.css" />
  <link rel="stylesheet" href="https://fonts.googleapis.com/css?family=Roboto" />
  <link rel="stylesheet" href="style/classic.dark.css" />
</head>
<body>
  <div id="imis-progress"></div>
  <nav id="nav-top">
    <div id="title" class="title">
      <a href="index.html">IMiS/wScan</a>
      <div id="scan-status"></div>
    </div>
    <div id="imis-profile"></div>
    <div id="cursor-mode"></div>
    <div id="target-color"></div>
    <div id="target-format"></div>
    <div id="total-documents"></div>
    <div id="total-pages"></div>
  </nav>
  <nav id="nav">
    <div id="scan-btn"></div>
    <div id="continue-btn"></div>
    <div id="cancel-btn"></div>
    <div id="download-btn"></div>
  </nav>
  <div class="main" id="main">
    <div id="thumbnails" class="main-left"></div>
    <div id="images" class="main-center"></div>
    <div id="image-details" class="main-right"></div>
  </div>
  <script src="apikey.js"></script>
  <script src="imis.scan.js"></script>
  <script src="imis.scan.ui.js"></script>
  <script>
    window.addEventListener("load", () => {
      try {
        var scan = new imis.scan.ui.Scan({
          apiKey: API_KEY, // API_KEY is defined in apikey.js
          reconnect: true,
          thumbnails: {
            id: "thumbnails",
            darkMode: true,
            orientation: "vertical",
            cursor: {
              enabled: true
            },
            document: {
              contextMenu: {
                onPropertiesSelected: () => showDetails()
              }
            },
            thumbnail: {
              height: 120, // Thumbnail height
              contextMenu: {
                onPropertiesSelected: () => showDetails()
              }
            }
          },
          imageDetails: {
            id: "image-details",
            darkMode: true,
            onClose: () => hideDetails()
```

```
          },
          status: { id: "scan-status" },
          progress: {
            id: "imis-progress",
            darkMode: true
          },
          images: {
            id: "images",
            darkMode: true,
            contextMenu: {
              onPropertiesSelected: () => showDetails()
            }
          },
          targetColor: { id: "target-color" },
          targetFormat: { id: "target-format" },
          totalDocuments: { id: "total-documents" },
          totalPages: { id: "total-pages" },
          buttons: {
            scan: { id: "scan-btn", darkMode: true, icon: "camera_alt" },
            profiles: { id: "imis-profile" },
            download: { id: "download-btn", darkMode: true, icon: "save_alt" },
            cancel: { id: "cancel-btn", darkMode: true },
            continue: { id: "continue-btn", darkMode: true },
            cursorMode: { id: "cursor-mode", darkMode: true }
          },
          onError: message => {
            // Show dialog with error message
            new imis.scan.ui.AlertDialog({ title: "Error", text: message });
          }
        });
        scan.show();
        resize();
      } catch (e) {
        console.error(e);
        new imis.scan.ui.AlertDialog({ title: "Error", text: e });
      }
    });

    // Show details
    function showDetails() {
      document.getElementById("image-details").style.display = null;
      document.getElementById("images").style = null;
    }

    // Hide details
    function hideDetails() {
      document.getElementById("image-details").style.display = "none";
      document.getElementById("images").style.flex = 0.75;
    }

    // Resizes height of main element
    function resize() {
      var titleHeight = document.getElementById("nav-top").offsetHeight;
      var navHeight = document.getElementById("nav").offsetHeight +
document.getElementById("imis-progress").offsetHeight;
      var mainHeight = (window.innerHeight - titleHeight - navHeight);
      document.getElementById("main").style.height = mainHeight + "px";
    }

    // Resize event listener
    window.addEventListener('resize', event => resize());
  </script>
</body>
</html>
```

6.4.3.2 classic.dark.css

```
body {
  margin: 0;
  background: #212121;
  height: 100%;
  width: 100%;
  font-family: 'Roboto', sans-serif;
}
a {
  text-decoration: none;
  color: inherit;
}

.title {
  color: #FFC107;
  font-size: 20px;
  padding-top: 15px;
  font-weight: bold;
}

nav {
  margin: 0;
  position: relative;
  padding-bottom: 10px;
  padding-left: 25px;
  padding-right: 25px;
}

nav > div {
  display: inline-block;
  margin-right: 2px;
}

.main {
  display: flex;
  height: 500px;
  position: relative;
  box-shadow: 0 0 1px 1px #757575;
  z-index: 1000;
}

.main-left,
.main-right {
  flex: 0.25;
}
```

2023 © MARGIS Inc., All rights reserved.

```
.main-center {
  flex: 0.50;
}

#nav {
  display: table;
  padding-top: 15px;
  padding-bottom: 15px;
  width: fit-content;
  margin: auto;
}

#nav .imis-btn,
#nav .imis-split-btn {
  margin-right: 7px;
}
```

6.4.4 Gallery sample

Gallery sample is suitable for displaying larger volumes of scanned documents or for batch scanning. Thumbnails of document pages are bigger and more visible than in the other samples. They are displayed in rows across the entire user interface. Documents are separated based on the number of pages specified in the settings. Page details appear to the user by double-clicking on an individual page.

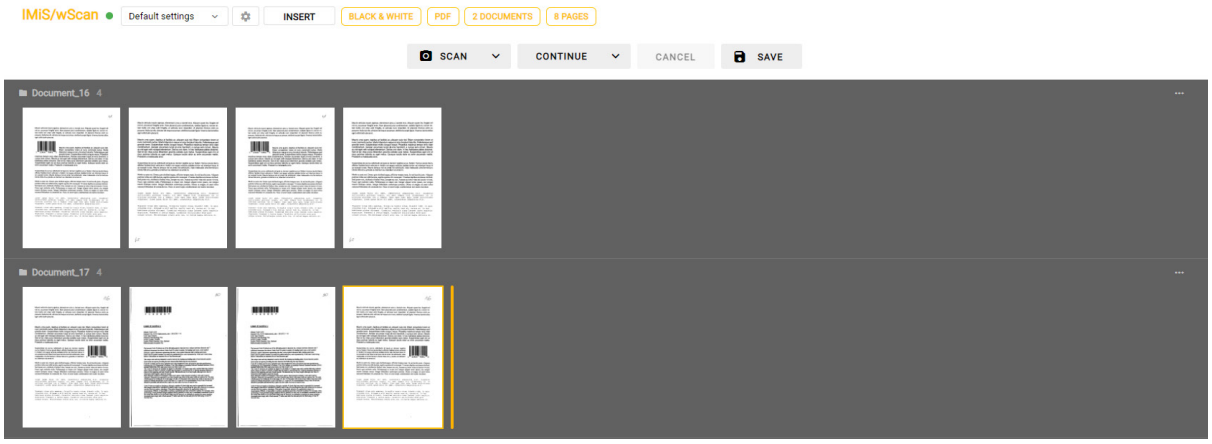

Image 61: Example of using the gallery sample of a user interface display

In the gallery sample, the following components are used:

- imis.scan.ui.Thumbnails
- imis.scan.ui.Status
- imis.scan.ui.Progress
- imis.scan.ui.Button
- imis.scan.ui.ProfilesButton
- imis.scan.ui.TargetColor
- imis.scan.ui.TargetFormat
- imis.scan.ui.TotalDocuments
- imis.scan.ui.TotalPages
- imis.scan.ui.CursorMode

6.4.4.1 gallery.html

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8" />
  <title>IMiS/wScan Gallery</title>
  <link rel="shortcut icon" type="image/png" href="img/favicon.png" />
  <link rel="stylesheet" href="imis.scan.ui.css" />
  <link rel="stylesheet" href="https://fonts.googleapis.com/css?family=Roboto" />
  <link rel="stylesheet" href="style/gallery.css" />
</head>
<body>
  <div id="imis-progress"></div>
  <nav id="nav-top">
    <div id="title" class="title">
      <a href="index.html">IMiS/wScan</a>
      <div id="scan-status"></div>
    </div>
    <div id="imis-profile"></div>
    <div id="cursor-mode"></div>
    <div id="target-color"></div>
    <div id="target-format"></div>
    <div id="total-documents"></div>
    <div id="total-pages"></div>
  </nav>
  <nav id="nav">
    <div id="scan-btn"></div>
    <div id="continue-btn"></div>
    <div id="cancel-btn"></div>
    <div id="download-btn"></div>
  </nav>
  <div class="main" id="main">
    <div id="thumbnails"></div>
  </div>
  <script src="apikey.js"></script>
  <script src="imis.scan.js"></script>
  <script src="imis.scan.ui.js"></script>
  <script>
    window.addEventListener("load", () => {
      try {
        var scan = new imis.scan.ui.Scan({
```

```
          apiKey: API_KEY, // API_KEY is defined in apikey.js
          reconnect: true,
          thumbnails: {
            id: "thumbnails",
            cursor: {
              enabled: true
            },
            orientation: "vertical",
            document: {
              contextMenu: {
                enabled: true
              }
            },
            thumbnail: {
              height: 200,
              title: false
            },
            gallery: {
              enabled: true,
              fitToSize: true
            }
          },
          status: { id: "scan-status" },
          progress: { id: "imis-progress" },
          targetColor: { id: "target-color" },
          targetFormat: { id: "target-format" },
          totalDocuments: { id: "total-documents" },
          totalPages: { id: "total-pages" },
          buttons: {
            scan: { id: "scan-btn", icon: "camera_alt" },
            profiles: { id: "imis-profile" },
            download: { id: "download-btn", icon: "save_alt" },
            cancel: { id: "cancel-btn" },
            continue: { id: "continue-btn" },
            cursorMode: { id: "cursor-mode" }
          },
          onError: message => {
            // Show dialog with error message
            new imis.scan.ui.AlertDialog({ title: "Error", text: message });
          }
        });
        scan.show();
        resize();
      } catch (e) {
        console.error(e);
        // Show dialog with error message
        new imis.scan.ui.AlertDialog({ title: "Error", text: e });
      }
    });

    // Resizes height of main element
    function resize() {
      var titleHeight = document.getElementById("nav-top").offsetHeight;
      var navHeight = document.getElementById("nav").offsetHeight +
document.getElementById("imis-progress").offsetHeight;
      document.getElementById("main").style.height = (window.innerHeight - titleHeight -
navHeight) + "px";
    }

    // Resize event listener
    window.addEventListener('resize', event => resize());
  </script>
</body>
</html>
```

2023 © MARGIS Inc., All rights reserved.

6.4.4.2 gallery.css

```
body {
  margin: 0;
  background: #fff;
  height: 100%;
  width: 100%;
  font-family: 'Roboto', sans-serif;
}

a {
  text-decoration: none;
  color: inherit;
}

.title {
  background: #fff;
  color: #FFC107;
  font-size: 20px;
  padding-top: 15px;
  font-weight: bold;
}

nav {
  margin: 0;
  position: relative;
  padding-bottom: 10px;
  padding-left: 25px;
  padding-right: 25px;
  background: #fff;
}

nav > div {
  display: inline-block;
  margin-right: 2px;
}

.main {
  position: relative;
  z-index: 1000;
}

#nav {
  display: table;
  padding-top: 15px;
  padding-bottom: 15px;
  width: fit-content;
  margin: auto;
}

#nav .imis-btn,
#nav .imis-split-btn {
  margin-right: 7px;
}
```

2023 © MARGIS Inc., All rights reserved.

7 USER DOCUMENTATION

The user documentation is intended to facilitate the understanding of the settings and individual features of the IMiS®/wScan application.

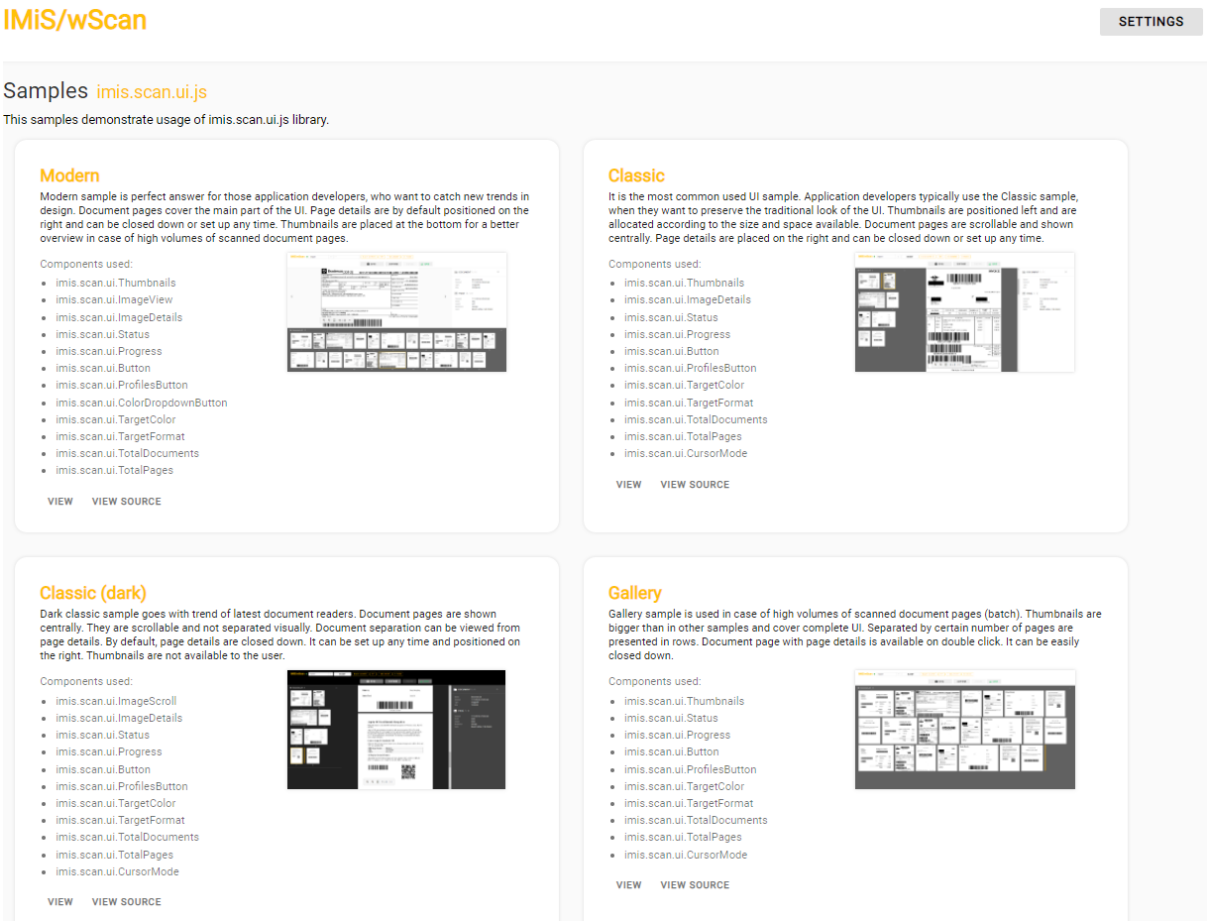

Image 62: Homepage of the IMiS®/wScan application

For greater clarity, the documentation has been divided into the following chapters:

- Profile settings.
- Scanning features.
- Batch scanning features.

2023 © MARGIS Inc., All rights reserved.

7.1 Profile settings

The profile settings enable the developer or user to create, display, modify, delete and lock a profile. By selecting the “Settings” button, the user is shown a dialog box displaying the settings.

The following settings are shown at the top:

- Profile name: name of the selected profile.
- New profile: creating a new profile.
  For more information see chapter Profile creation.
- Save: saving a profile.
- Lock: locking a profile and disabling profile modification.
- Delete: removing a profile.

If a green tag is shown next to the name “IMiS®/wScan Settings”, the connection to the scanner driver and to the IMiS®/Capture Service is established and working.

The following settings are logically sorted in tabs:

- Source: the scanner and scanning mode.
- Target: the save mode and location.
- Separator*: document separation mode.
- Metadata: user-defined data on the scanned document.

Note*: The batch scanning features are available only when the suitable license is activated.

The following information is shown at the bottom:

- type of feature (SCAN, BATCH SCAN)
- application version (e.g. 1.9.2310.15)

2023 © MARGIS Inc., All rights reserved.

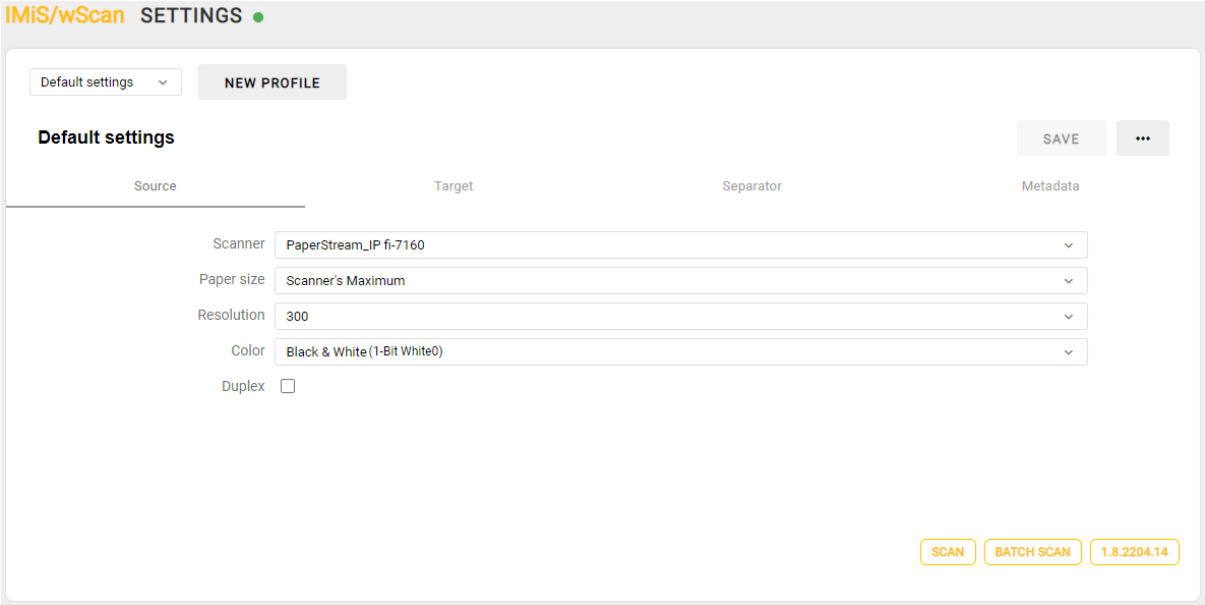

Image 63: Profile settings

7.1.1 Profile creation

By default, the user is provided the profile “Default settings”, which they can set according to their needs.

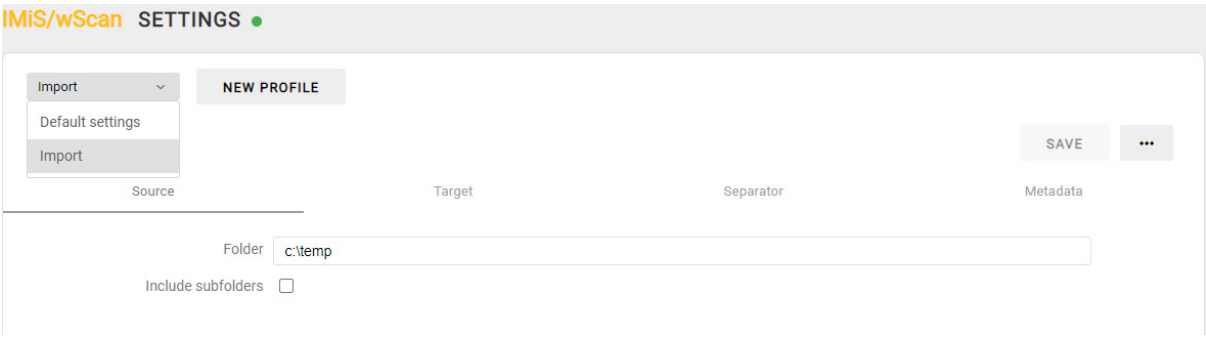

Image 64: Selecting the default profile

The user creates a new scanning profile for document scanning by selecting the action “New profile”.

A dialog box is shown for setting the profile parameters, which will be applied to the scanning of documents. Besides the profile name, which must be uniquely defined, the user also sets the remaining settings in the tabs. The user saves the new profile by selecting the "Save" button or cancels it with the "Cancel" button.

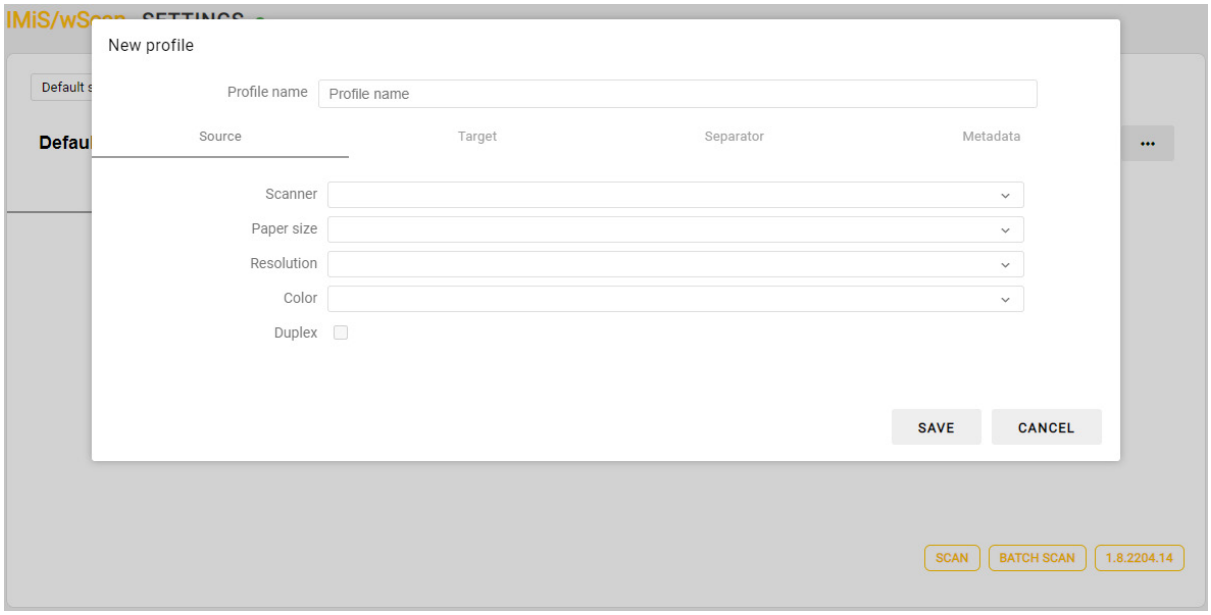

Image 65: Profile creation

7.1.2 Source

In the “Source” section the user specifies the scanner and scanning mode.

The following settings are available:

- Scanner: the source for capturing a document (scanner, a file in the file system).
- Paper size: the size and format of the scanned document.
- Resolution: scanner resolution (only resolutions supported by the selected scanner are shown).
- Color: color depth (black & write, grayscale, color).
- Duplex: scanning mode (simplex or duplex).

2023 © MARGIS Inc., All rights reserved.

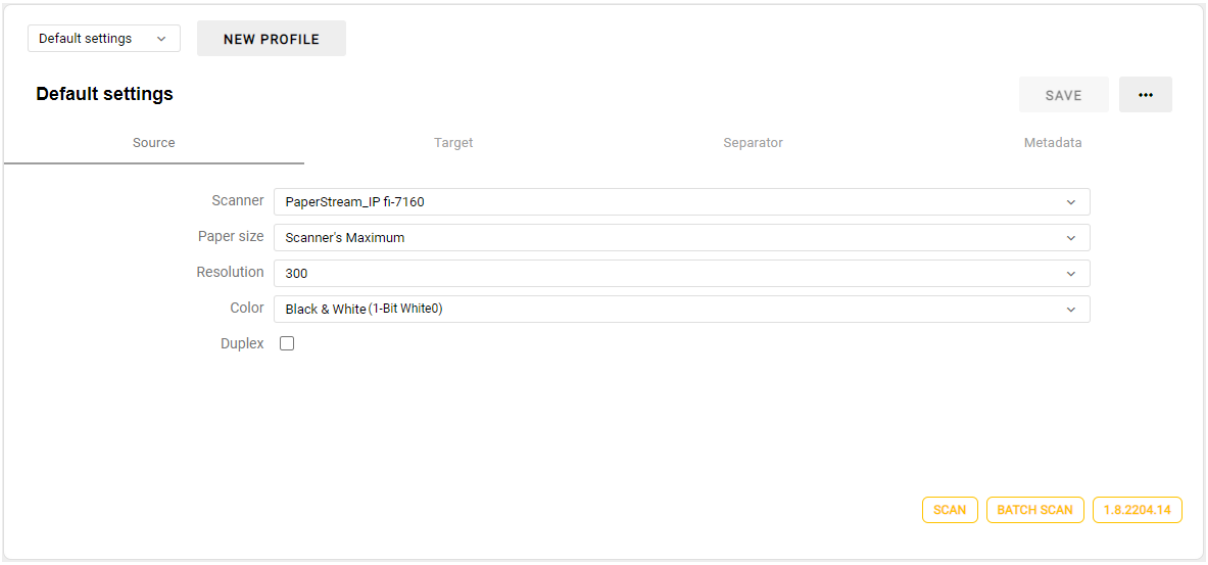

Image 66: Profile source settings

In the field “Scanner” the user is shown a popup menu where they can use the command “Reload” to reload the scanner driver.

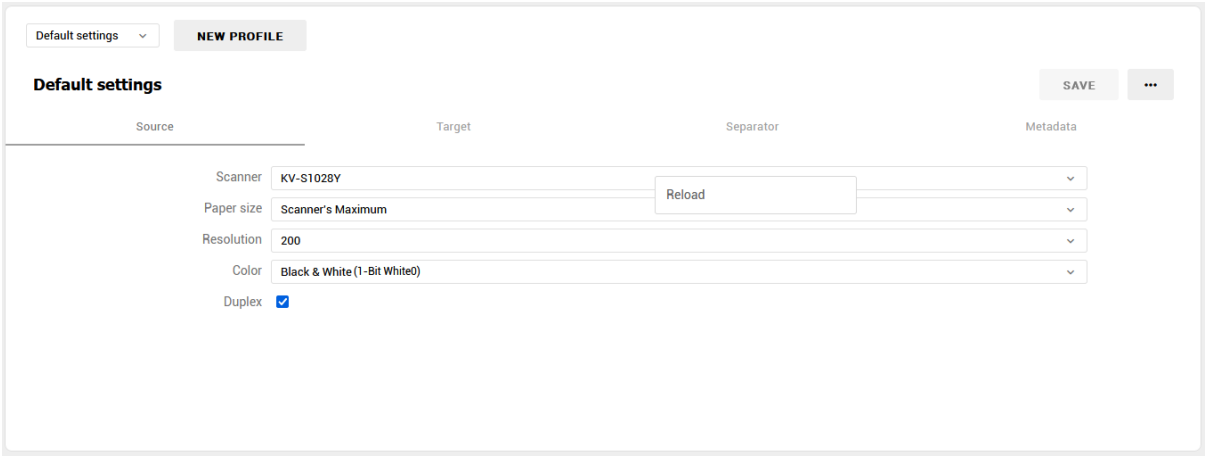

Image 67: Popup menu of the “Scanner” setting

2023 © MARGIS Inc., All rights reserved.

7.1.3 Target

In the “Target” section the user specifies the document save mode and location.

The following settings are available:

- Format: the document format (PDF, TIFF, PNG, BMP, JPEG ...).
- Color: color depth (black & write, grayscale, color).
- Compression: compression methods (CCITT G4, LZW, ZIP, JPEG, ...).
- File name: name of the saved document file.
- Folder: the folder in the file system to which documents are saved.

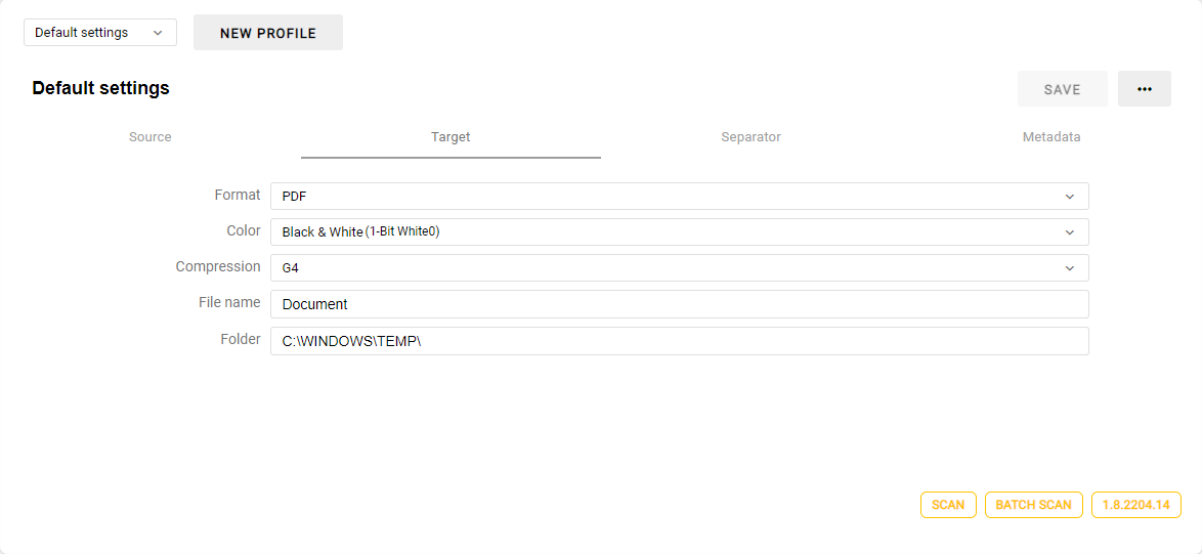

Image 68: Setting the profile target

7.1.4 Separator

In the “Separator” section the user specifies the document separation mode by selecting the separator type. This section is shown if the activated license includes the DOCSEP feature (see chapter Product activation).

The user can define the following values for each separator type:

- Disabled
- Number of pages
- Barcode
- Blank page.

7.1.4.1 Disabled

By selecting the value "Disabled", document separation is not executed.

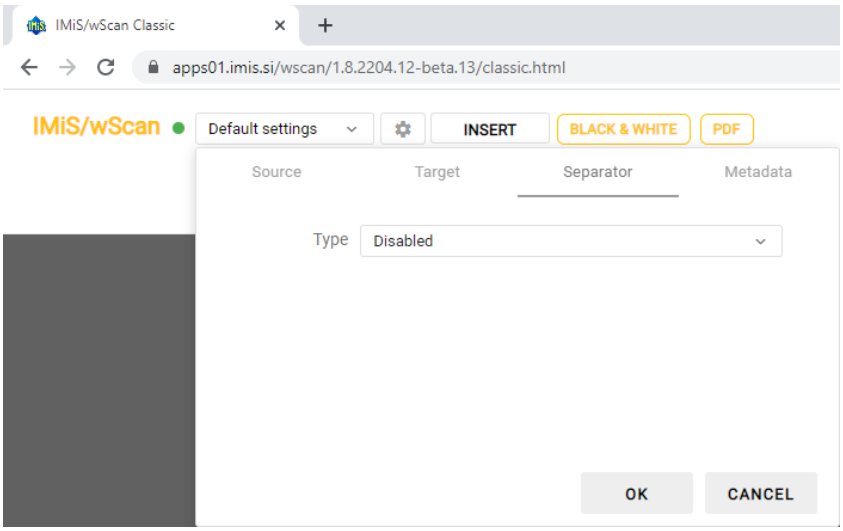

Image 69: Setting the separator to ”Disabled”

7.1.4.2 Number of pages

By selecting the value "Page count", the user enables separating documents by page count.

For more information see chapter Separation by page count.

The value “Page count” is present if the activated license includes the DOCSEP feature, which enables document separation.

For more information see chapter Product activation.

In the field “Page count” the user defines the number of scanned pages for each document.

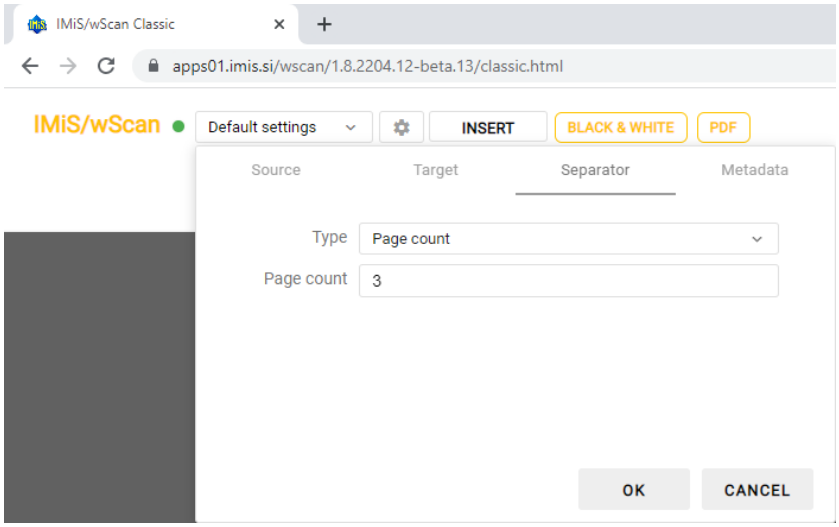

Image 70: Setting the separator to “Page count”

7.1.4.3 Barcode

By selecting the value "Barcode", the user enables separating documents by barcodes.

For more information see chapter Barcode-based separation.

The selection "Barcode" is present if the activated license includes at least one of BAR1DPIX or BAR2DPIX or BAR1DST or BAR2DST features which enables the recognition of 1D barcodes, or BAR2DPIX which enables the recognition of 1D and/or 2D barcodes. For more information see chapter Product activation.

2023 © MARGIS Inc., All rights reserved.

The user can specify the following settings for barcode recognition:

- Barcode: barcode types for recognition, which depend on the features included in the activated license:
  - 1D barcodes:
    - BAR1DPIX: Addon-2, Addon-5, Australian Post, BCD Matrix, Codabar, Code-25 Datalogic, Code-25 IATA, Code-25 Industrial, Code-25 Interleaved, Code-25 Invert, Code-25 Matrix, Code-32, Code-39, Code-93, EAN-13, EAN-8, Intelligent Mail, PostNet, Royal Post, Type-128, UCC-128, UPC-A, UPC-E
    - BAR1DST: Codabar, Code-25 Datalogic, Code-25 IATA, Code-25 Industrial, Code-25 Interleaved, Code-25 Matrix, Code-39, Code-39 ASCII, Code-93, EAN-13, EAN-8, GS1 Databar Omnidirectional, GS1 Databar Omnidirectional Stacked, GS1 Databar Expanded, GS1 Databar Expanded Stacked, GS1 Databar Limited, Type-128, UCC-128, UPC-A, UPC-E
  - 2D barcodes:
    - BAR2DPIX: AZTEC, DataMatrix, PDF-417, QR Code
    - BAR2DST: DataMatrix, PDF-417, QR Code
- Orientation: direction of barcode recognition. Set of values:
  - Horizontal
  - Vertical
  - Horizontal-Vertical
  - Horizontal-Vertical Diagonal.
- Mode: barcode recognition mode. Set of values:
  - Normal
  - Enhanced.
- Value: the barcode value in the case of barcode-based separation. Set of values:
  - Empty value: the separator is any recognized barcode.
  - Non-empty value: “Regular expression” for searching for the barcode which represents the separator.

2023 © MARGIS Inc., All rights reserved.

- Action: the action in the case of barcode-based document separation.

  The available actions are:

  – Separate: the page with the recognized barcode begins a new document.

  – Separate and delete: the page with the recognized barcode is deleted and the next page begins a new document.

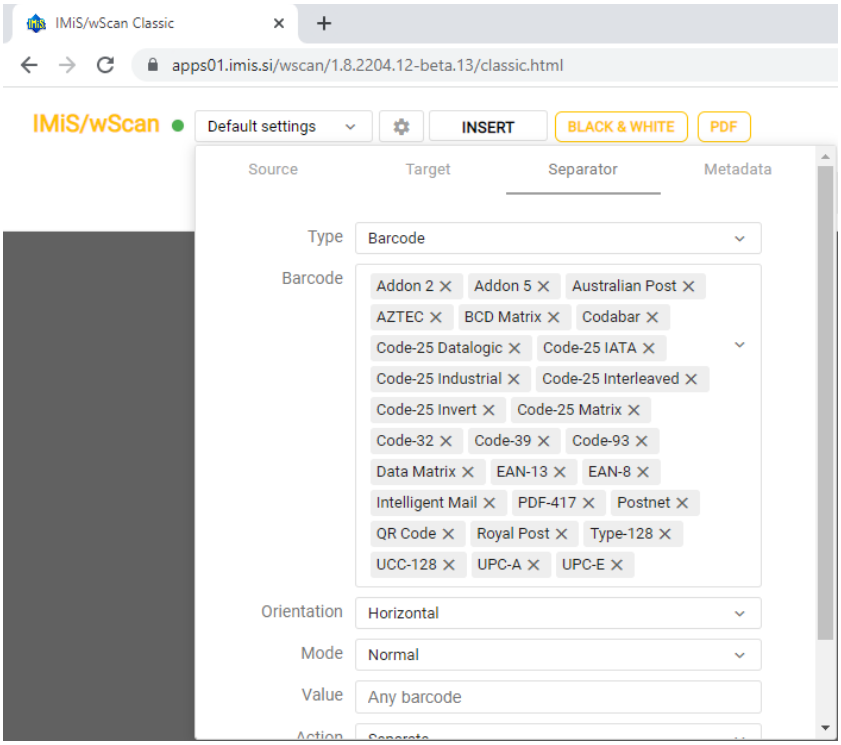

Image 71: Barcode separator settings

In the field “Barcode” the user is shown a popup menu where they can use the command “Select all” to display all types of barcodes, while the command “Clear” clears all the currently selected barcode types. The blank selection of barcode types is equivalent to selecting all barcode types.

2023 © MARGIS Inc., All rights reserved.

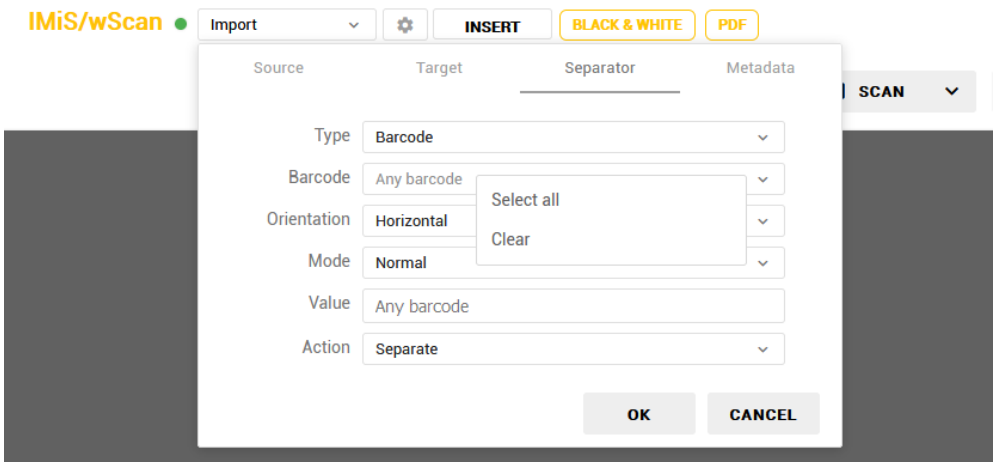

Image 72: Popup menu of the setting "Barcode"

Note: If the user wants to separate documents based on a specific barcode value, they must use special characters, e.g. for a barcode value of 1234567890 the user has to enter ^1234567890$ in the "Value" field.

The value for selecting the barcode is the "Regular expression", which in this specific example means that it finds all barcodes containing the exact string 1234567890.

The character '^' means that the detected barcode must start with the characters that follow. The character '$' means that it must end with the preceding characters.

2023 © MARGIS Inc., All rights reserved.

7.1.4.4 Blank page

By selecting the value “Blank page”, the user enables separating documents by blank pages.

For more information see chapter Blank page separation.

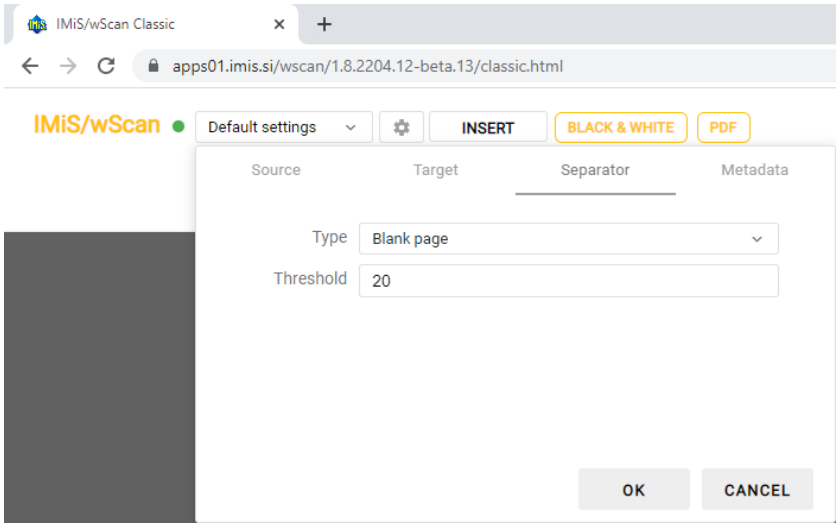

Image 73: Setting the separator to “Blank page”

By modifying the setting “Threshold”, the user changes the performance of the algorithm for blank page detection by decreasing or increasing the range of pages
that the algorithm considers blank.

For more information see chapter Setting the threshold.

2023 © MARGIS Inc., All rights reserved.

7.1.5 Metadata

In the “Metadata” section the user defines the set of metadata on the document. The user adds metadata by selecting the button “Add”.

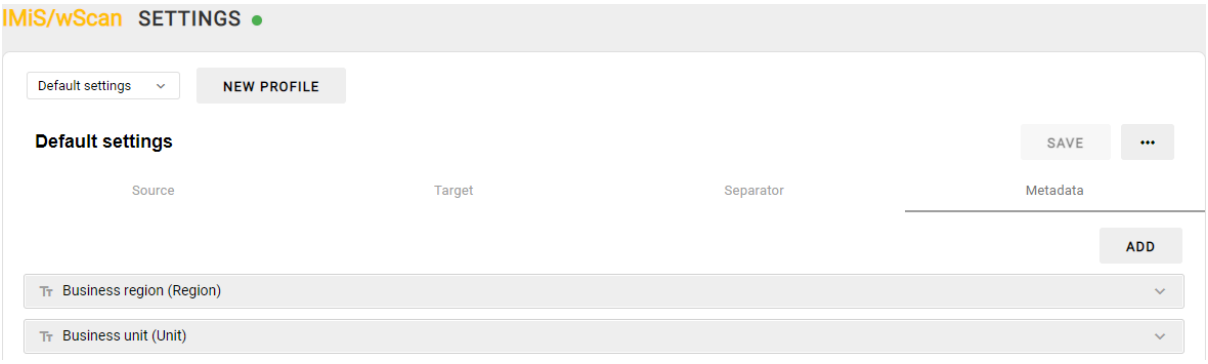

Image 74: Profile metadata settings

The following parameters can be defined for each piece of metadata:

- Identifier: the unique identifier of the metadata.
- Name: the name of the metadata.
- Description: a short description of the metadata.
- Type: the type of metadata. Possible types of metadata:
    - Integer: the valid values are integers;
    - Boolean: the valid values are the logical “True” and “False”;
    - Date: the valid values are dates;
    - DateTime: the valid values are date and time values;
    - String: the valid values are strings of alphanumeric characters;
    - Barcode: the valid values are barcodes.
- Prefix: the prefix of the metadata value. This setting is not possible for the metadata of type Boolean, Date and DateTime.
- Suffix: the suffix of the metadata value. This setting is not possible for the metadata of type Boolean, Date and DateTime.
- Predefined values: the set of values of the selected metadata type, predefined by the user. This setting is not possible for the metadata of type Boolean.

2023 © MARGIS Inc., All rights reserved.

- Validation: the value which represents a regular expression, which can be used to additionally check the validity of the metadata value. This setting is not possible for the metadata of type Boolean.
- Region: the region of the metadata view on the document.
  For more information see chapter Defining the metadata region.

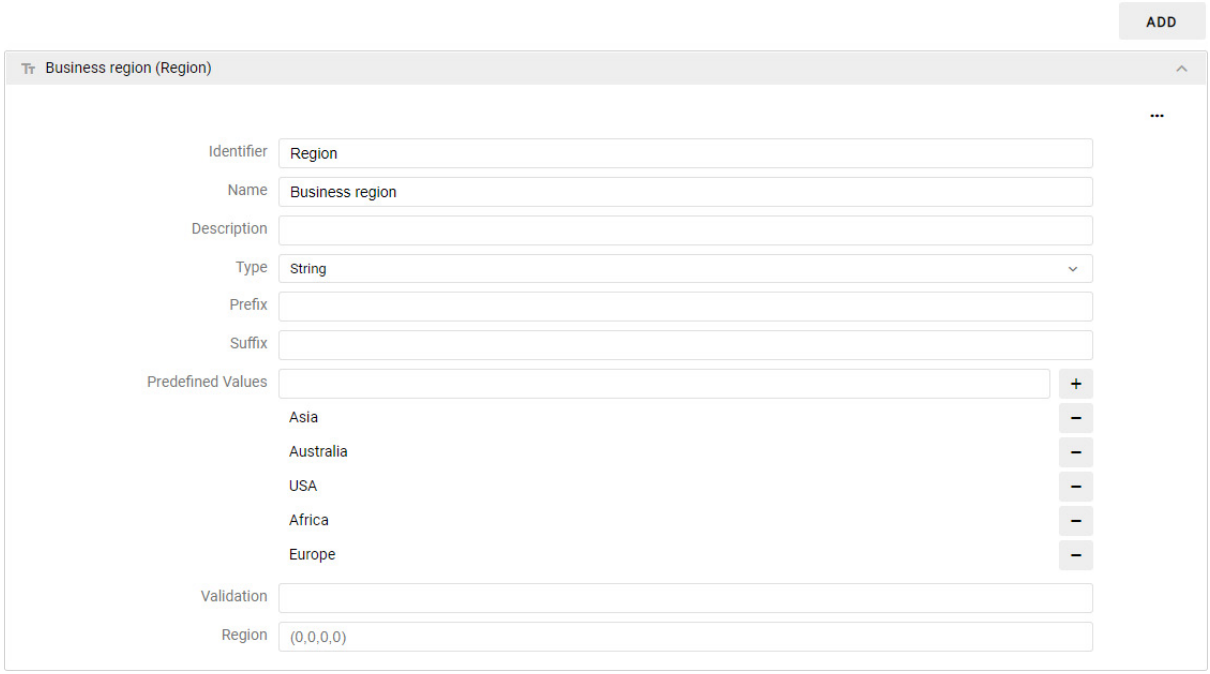

Image 75: Defining the metadata parameters

Through the popup menu on the label "..." in the metadata settings, the user moves a piece of metadata within the metadata set up or down using the commands "Move up" or "Move down", or deletes it from the metadata set using the command "Delete".

Image 76: Popup menu in the metadata settings

2023 © MARGIS Inc., All rights reserved.

7.2 Application features

The application features enable the developer or user to scan, upload, display documents, display document data, and perform basic operations on the scanned documents.

To show the set of scanning features, we have chosen the “Classic sample” user interface which is made up of the following building blocks:

- document capture;
- a view of document thumbnails;
- a view of document pages;
- a view of document data.

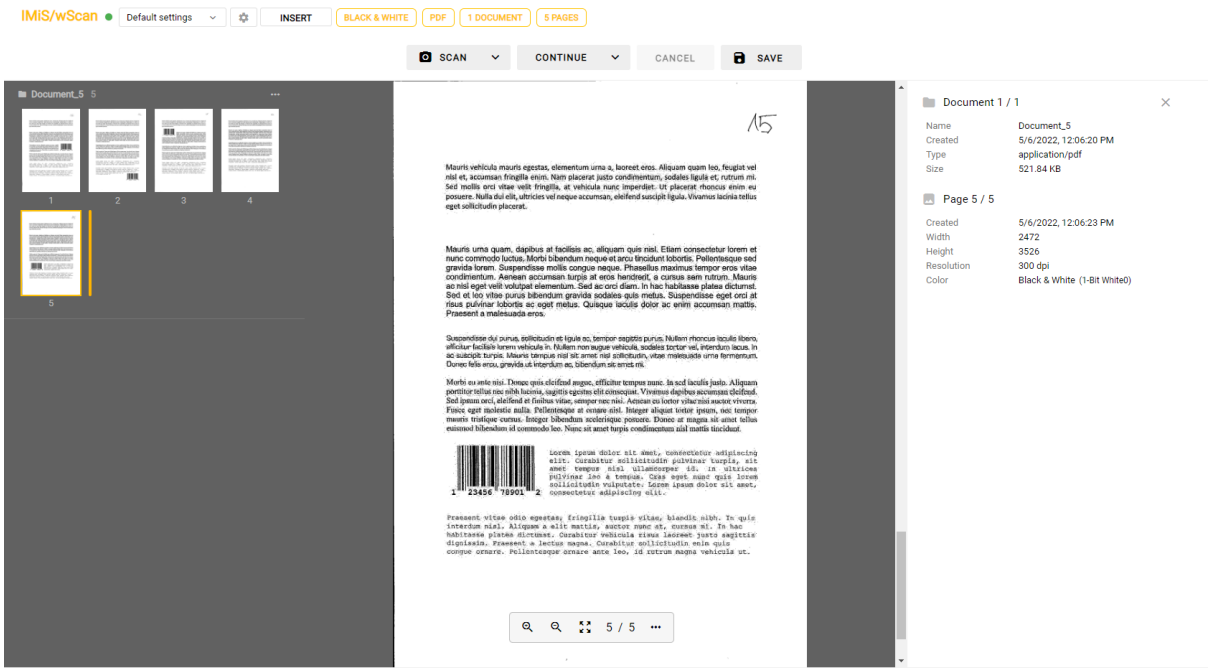

Image 77: User interface in the classic mode

7.2.1 Document capture

The basic feature of the application is the capture of a document if the user has a license that enables the basic scanning features, or the capture of multiple documents if the user has a license which enables batch scanning features.
For more information see chapter Product activation.

The document or documents are captured with the following actions:

- Scan: scanning of documents from the device or uploading of documents from the file system .
- Continue: continued scanning of documents from the device or uploading of documents from the file system .
- Cancel: canceling scanning during the scanning process.
- Save: saving scanned documents.

Each use of the “Scan” button creates a new job with the current settings of the selected profile. The selected profile is shown in the top profile bar where the user can view the following profile information and settings:

- Profile name: the name of the selected profile.
- Profile settings: the profile settings for the next job which remain valid until the profile is replaced, the view is refreshed, or IMiS®/Capture Service is restarted.
  For more information see chapter Profile settings.
- Insert or Overwrite: the mode of adding document pages (adding, overwriting).
  For more information see chapter Adding document pages and Overwriting document pages.

In the profile bar the user is provided the following document information:

- color depth (black & write, grayscale, color).
- document format (PDF, TIFF, PNG, BMP, JPEG ...).
- number of documents.
- number of pages.

2023 © MARGIS Inc., All rights reserved.

Image 78: A command bar with document information in the classic mode

7.2.2 Document thumbnails

In the left view of the user interface the user is shown the thumbnails of document pages. This provides the user with a better review of the documents and faster navigation between document pages.

By right-clicking over the selected document, the user opens a popup menu and performs the following actions:

- Download: downloading the document to a computer or device. The default save format is defined by the value of the “Format” attribute in the “Target” tab.
- Join: joining two documents, which is available when the job contains at least two documents.
  For more information see chapter Joining documents.
- Edit separator: adding a new or editing the existing barcode-type separator.
  For more information see chapter Separator editing.
- Delete: removal of the selected document.
- Properties: document data.

2023 © MARGIS Inc., All rights reserved.

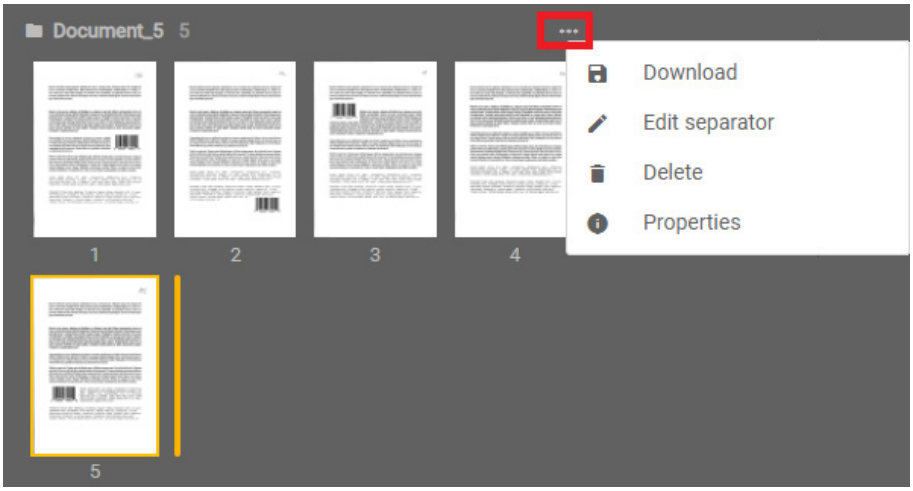

Image 79: Popup menu in the case of a single document in the left view

If there are at least two documents, the user is given the option in the left thumbnails view to join the documents by performing the following action:

- Join: joining two documents.

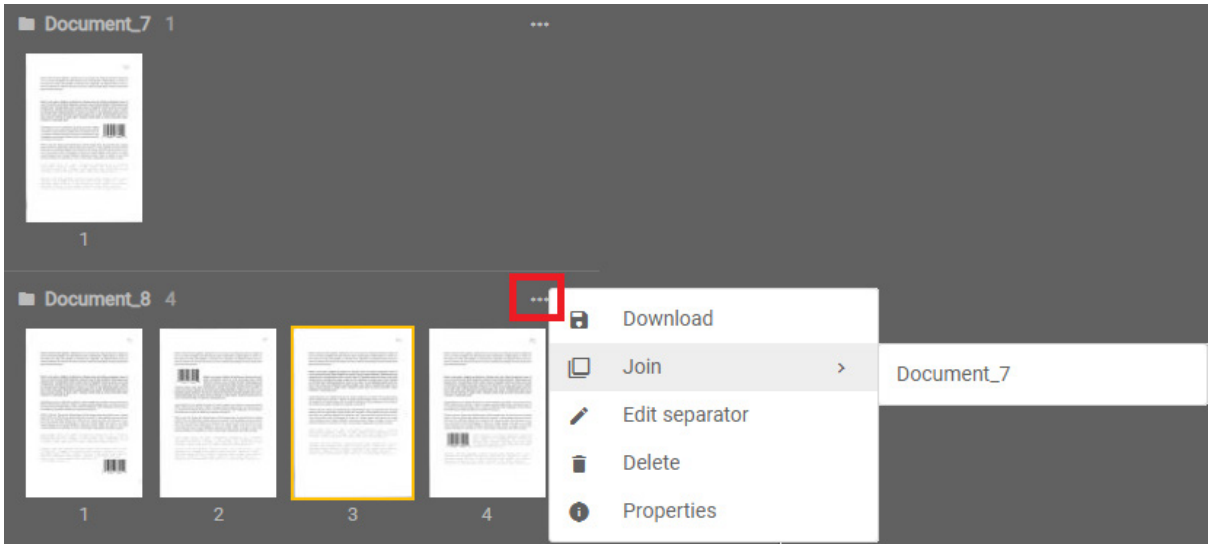

Image 80: Popup menu in the case of multiple documents in the left view

In the view, the documents are visibly separated from each other. The name and number of pages are shown for each document.

2023 © MARGIS Inc., All rights reserved.

By right-clicking over the selected thumbnail of document, the user opens a popup menu and performs the following actions:

- Download: downloading the document pages to a computer or device.
  The default save format is .PNG (black-white) and JPEG (grayscale, color).
- Insert/Overwrite: inserting new or overwriting existing document pages.
  - Insert before/Device or Insert after/Device: inserting new document pages from the device (scanner) before or after the selected page.
    For more information see chapter Adding document pages and Overwriting document pages.
  - Overwrite before/Upload or Overwrite after/Upload: overwriting existing document pages from the file system before or after the selected page. For more information see chapter Adding document pages and Overwriting document pages.
- Rotate right: rotating the document page to the right.
- Rotate left: rotating the document page to the left.
- Split: splitting the document into two parts. The next page after the selected page becomes the first page of the new document.
  For more information see chapter Splitting a document.
- Delete: removal of the selected document pages.
- Properties: document data.

2023 © MARGIS Inc., All rights reserved.

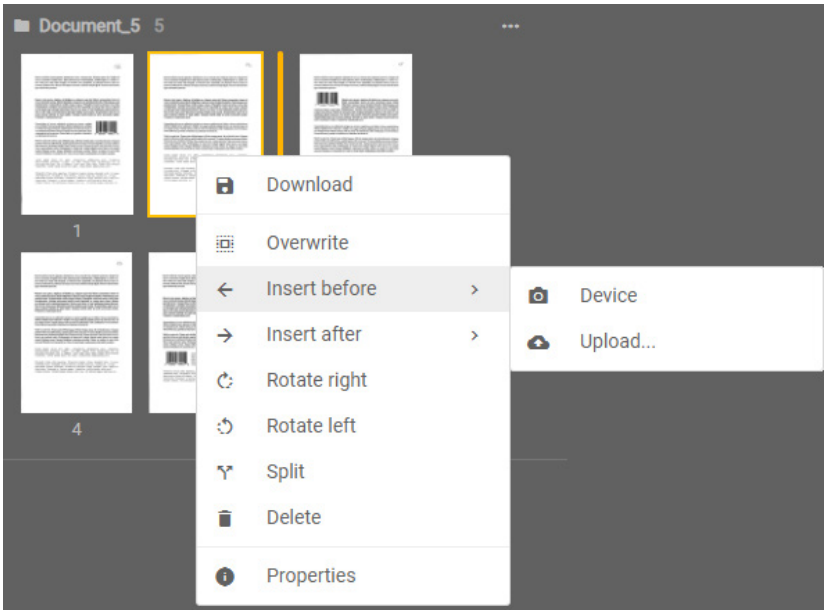

Image 81: Popup menu on the document page in the classic mode

7.2.3 Document pages

In the central view of the user interface the user is shown the document pages.

Besides displaying the sequence number of the currently displayed page and the total number of document pages, the user can also perform the following actions on the document page:

- Zooming in on the document page.
- Zooming out of the document page.
- Original document size.
- Crop: Cropping the highlighted part of the document page.
  For more information see chapter Cropping a page area.
- Redaction: hiding selected parts of the document page.
  For more information see chapter Hiding page areas.
- Highlight: highlighting selected parts of the document page.
  For more information see chapter Highlighting page areas.
- Region: defining the region for metadata capture.
  For more information see chapter Metadata and Defining the metadata region.

2023 © MARGIS Inc., All rights reserved.

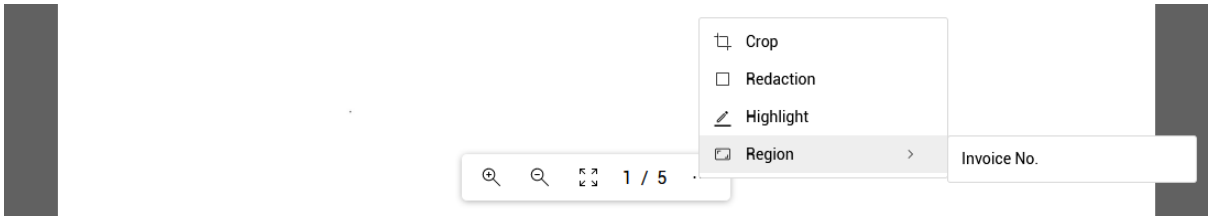

Image 82: Toolbar on the document page in the classic mode

7.2.4 Document data

In the right view of the user interface the user is shown the document data.

Besides displaying the name and sequence number of the document, the following document data is shown:

- Name: the name of the document.
- Created: the date and time of document creation.
- Type: the document type.
- Size: the document size (in KB / MB).
- Separator: the record of the barcode-type separator of the first page of the selected document. It is available to the user if the data was captured during the scan or added with the action “Edit separator” via the popup menu on the document in the left view. For more information see chapter Separator editing.

The following information is provided to the user for the selected document page:

- Created: the date and time of document page creation.
- Width: the width of the document page (in pixels).
- Height: the height of the document page (in pixels).
- Resolution: the resolution of the scanned document page (in dpi).
- Color: color depth (black & write, grayscale, color).

2023 © MARGIS Inc., All rights reserved.

The user can remove the right view by selecting the action “Properties” in the popup menu on the document.

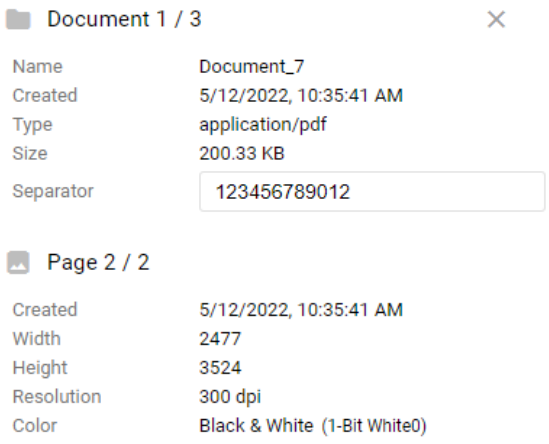

Image 83: Document data

If metadata is also defined in the settings, then the user can define their values in the “Attributes” set.

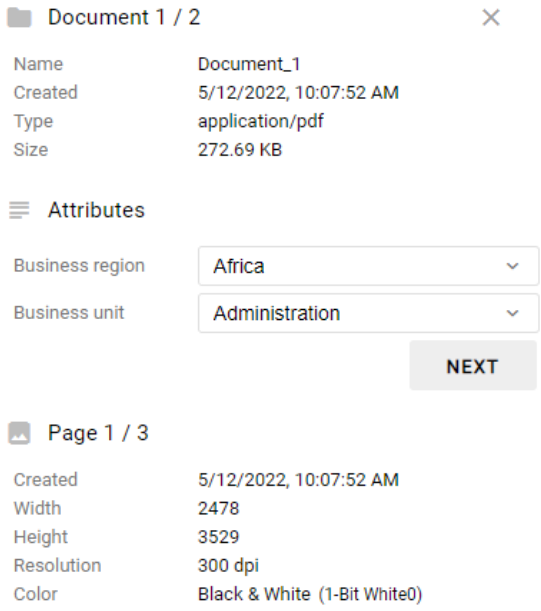

Image 84: Document details with the “Attributes” set

2023 © MARGIS Inc., All rights reserved.

The user can remove or recover the right view by selecting the action “Properties” in the popup menu on the document or document page.

7.2.5 Adding document pages

By selecting the action “Insert” in the command bar or in the popup menu on the selected document page in the left view, and the “Insert before” or “Insert after” action, the user can add new pages. By selecting the existing document page, the user specifies before which document page or after, which page the pages of the new document will be added. After adding the pages, the pages of the existing document after the selected page will be moved after the added pages, and the page position marker is located after the document page.

In the next step the user chooses whether the document pages will be added from the device (“Device”) or from the file system (“Upload”).

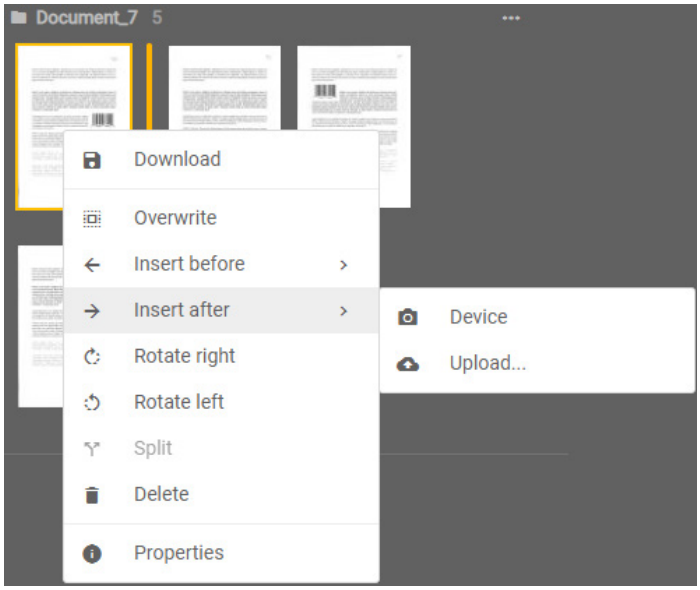

Image 85: Adding document pages from the device or file system

7.2.6 Overwriting document pages

By selecting the action “Overwrite” in the command bar or in the popup menu on the selected document page in the left view, and the “Overwrite before” or “Overwrite after” action, the user can overwrite the pages of a existing document.
By selecting the existing document page, the user specifies which pages will be overwritten with the new document.
In the “Overwite” mode the page position marker is located before the document page.

In the next step the user chooses whether the document pages will be overwritten from the device (“Device”) or from the file system (“Upload”).

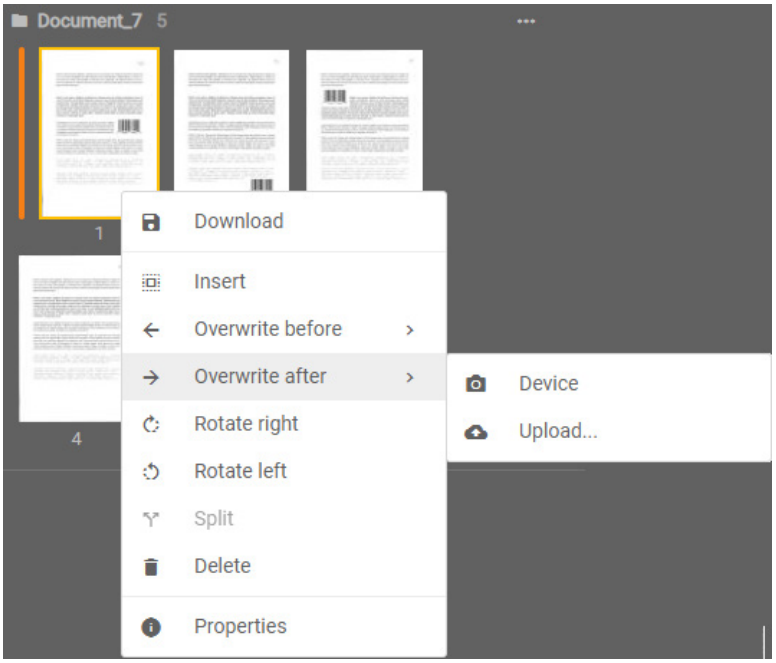

Image 86: Overwriting document pages from the device or file system

7.2.7 Moving document page

By selecting an individual document page, the user can move the page to the desired location using the “Drag n' Drop” mode. By selecting and moving the page, the marker is moved too, which specifies between which pages of the existing document a new document page will be added.

2023 © MARGIS Inc., All rights reserved.

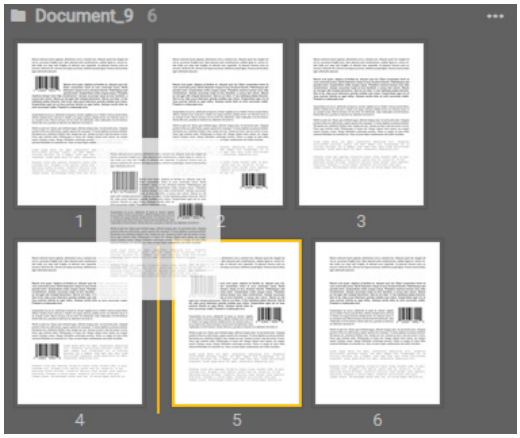

Image 87: Moving a document page

7.2.8 Splitting a document

By selecting the action “Split” in the popup menu on the selected document page in the left view, the user can split the pages of the existing document into two parts.

Note: Splitting is enabled from the second document page onward.

Image 88: Separating document pages

7.2.9 Joining documents

If the job contains at least two documents, the user has the option in the left thumbnails view to join documents. With the command “Join” the user selects one of the remaining job documents. The pages of the selected document are then moved to the current document and the former document is deleted from the job.

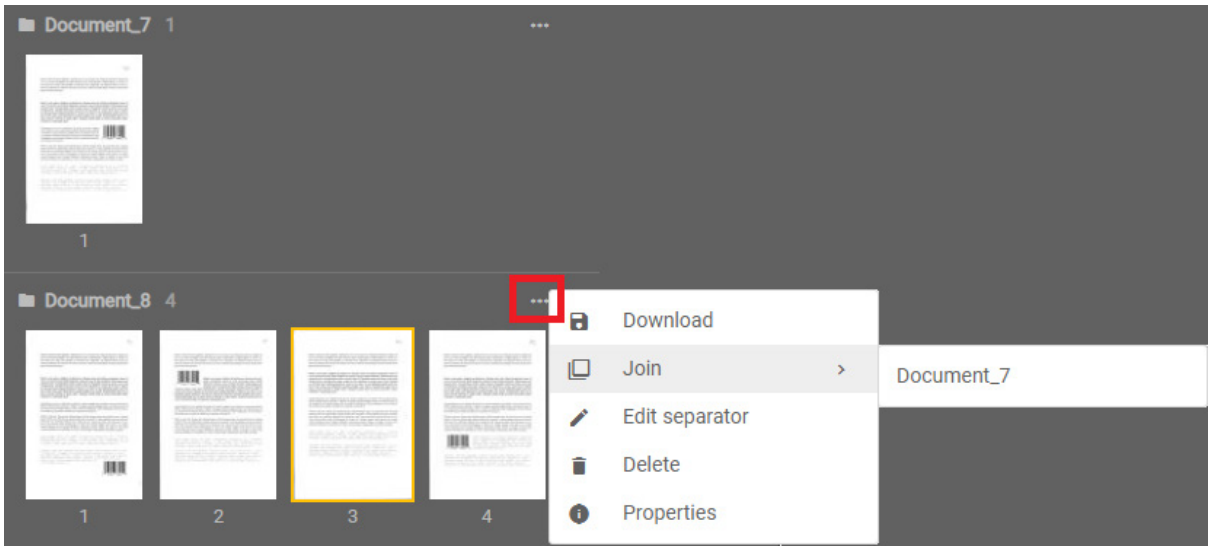

Image 89: Example of joining two documents

7.2.10 Cropping a page area

By selecting the action “Crop”, the user can crop an area on the document page. This action is used when we want to keep only a part of the page and not the entire page.

Image 90: Selecting the action “Crop”

After selecting the action “Crop” in the popup menu, the user selects the area on the document page with the mouse.

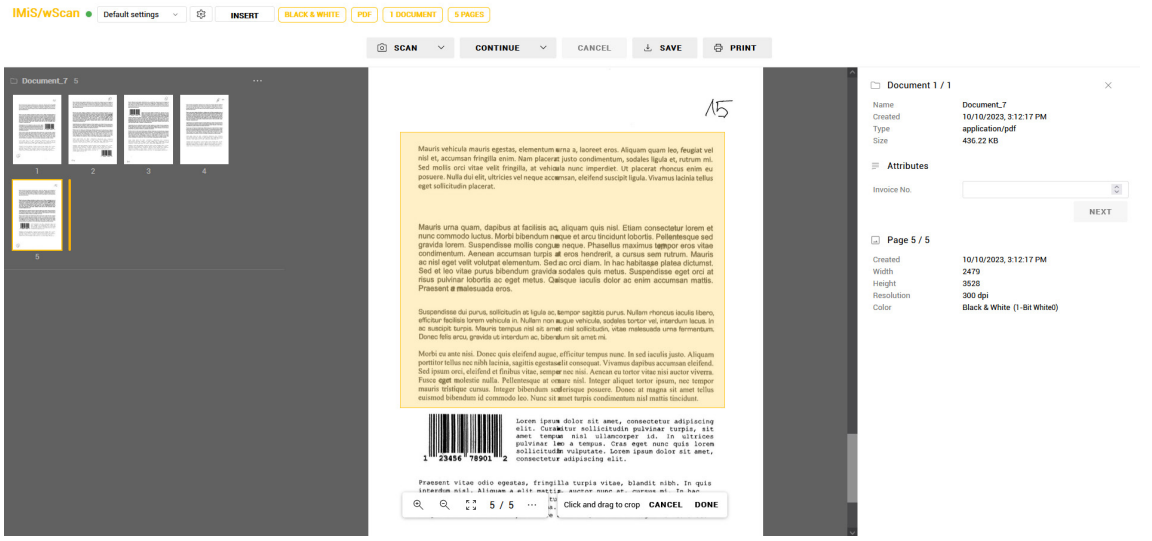

Image 91: Highlighting the crop area

The user performs the cropping by clicking “Done” or cancels it by clicking “Cancel”.

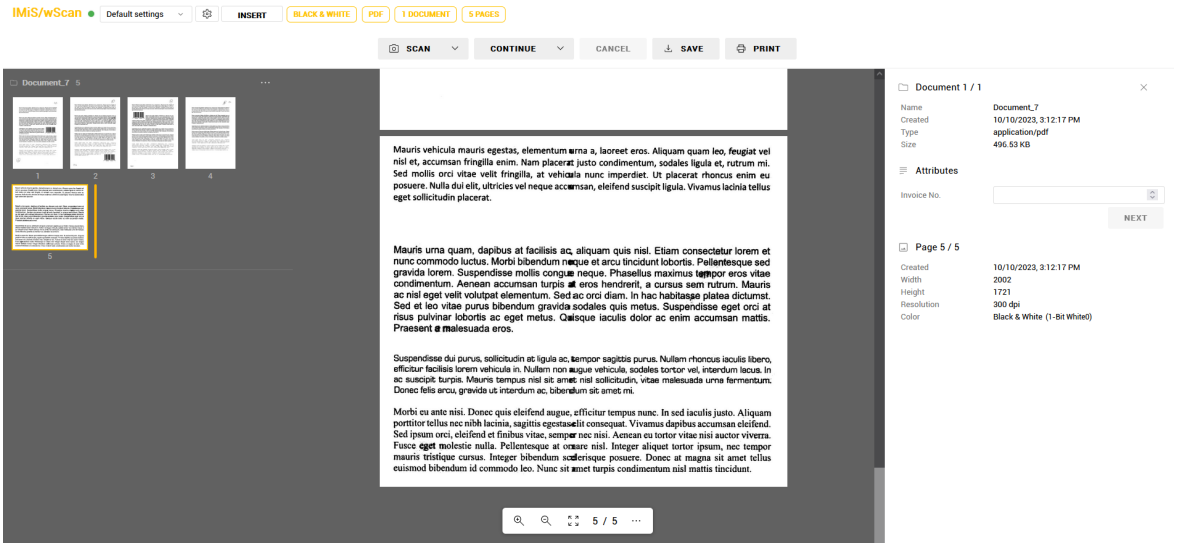

Image 92: Cropping a page area

Note: After performing the action “Crop”, the document page cannot be restored to its previous condition.

7.2.11 Hiding page areas

By selecting the action “Redaction”, the user can hide selected areas on the document page. The user uses this action when wanting to hide parts of the page that contain confidential information or for any other reason.

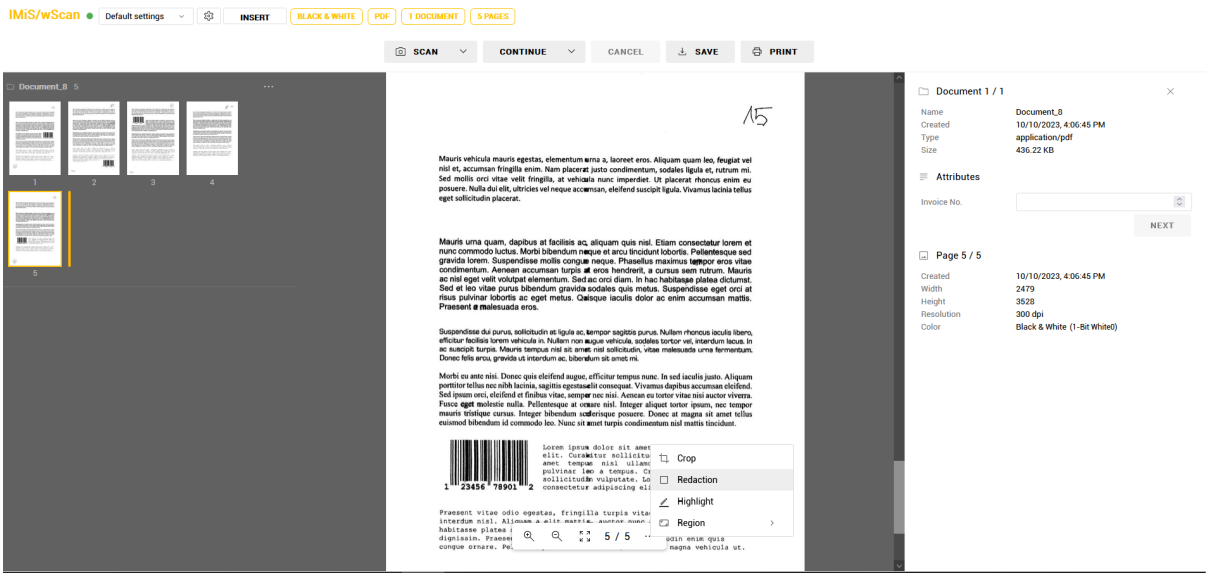

Image 93: Selecting the action “Redaction”

After selecting the action “Redaction” in the popup menu, the user selects one or multiple areas on the document page with the mouse.

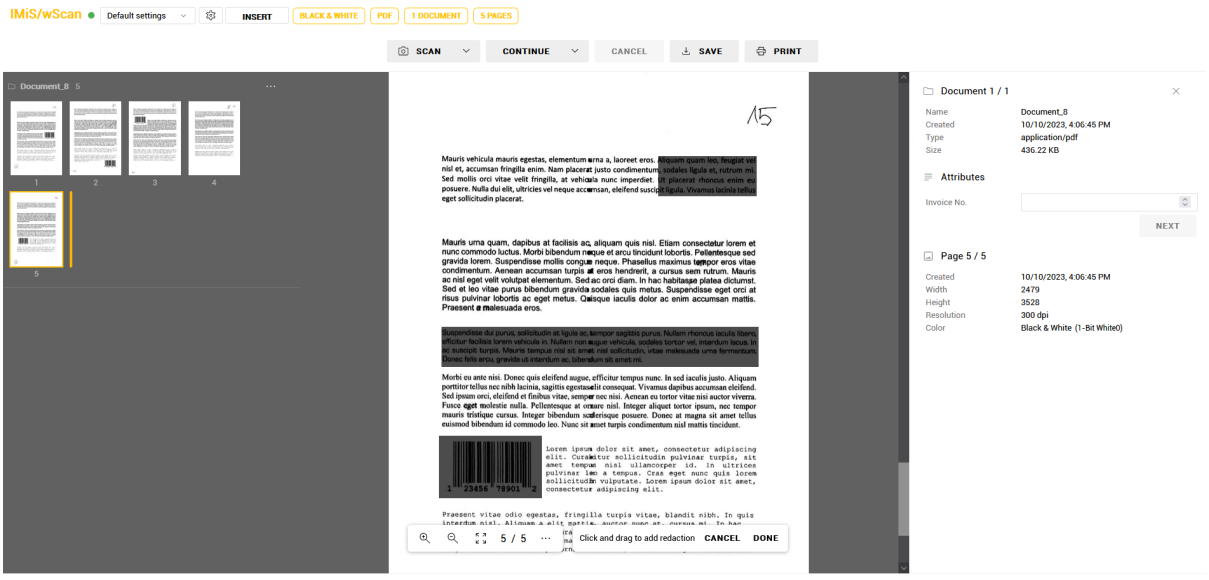

Image 94: Highlighting the redaction areas

The user finishes hiding the areas by clicking “Done” or cancels it by clicking “Cancel”.

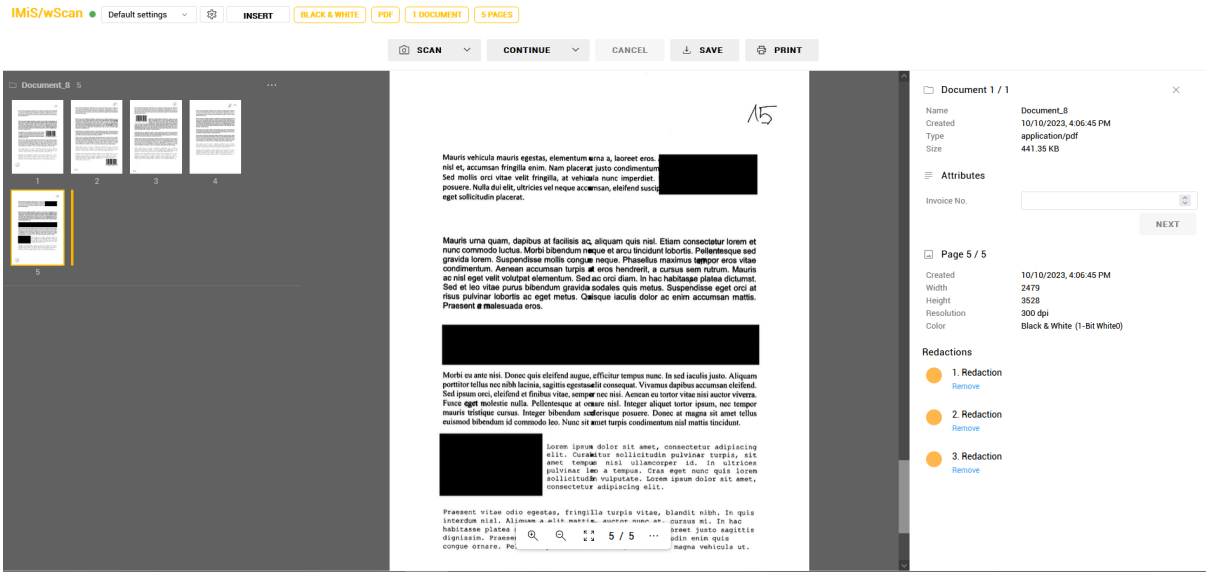

Image 95: Hiding page areas with redactions

The user can remove the redactions by selecting the action "Remove" under the selected redaction on the “Redactions” list in the document properties view.

Note: When saving the document by selecting the action “Save”, the redactions merge with the document pages, which is why they cannot be removed after saving nor can the page be restored to its previous condition.

7.2.12 Highlighting page areas

By selecting the action “Highlight”, the user can highlight selected areas on the document page using a highlight color. The user uses this action when wanting to direct attention to parts of the page.

2023 © MARGIS Inc., All rights reserved.

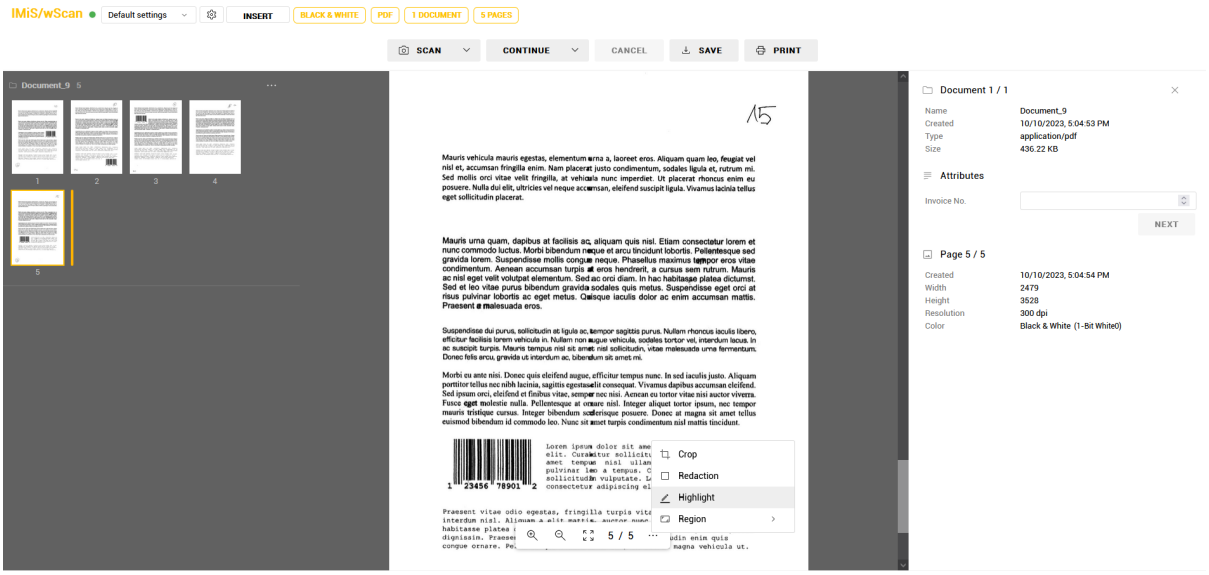

Image 96: Selecting the action “Highlight”

After selecting the action “Highlight” in the popup menu, the user selects one or multiple areas on the document page with the mouse. Each time an area is selected, the user can choose a different highlight color.

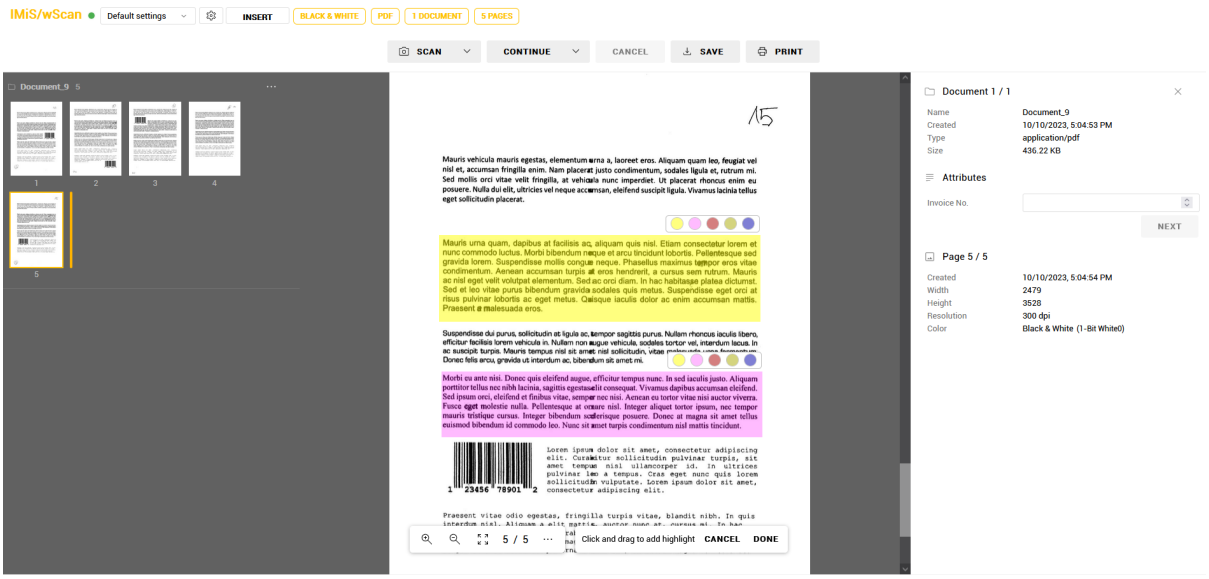

Image 97: Selecting one or multiple areas on the document page

The user finishes highlighting the areas by clicking “Done” or cancels it by clicking “Cancel”.

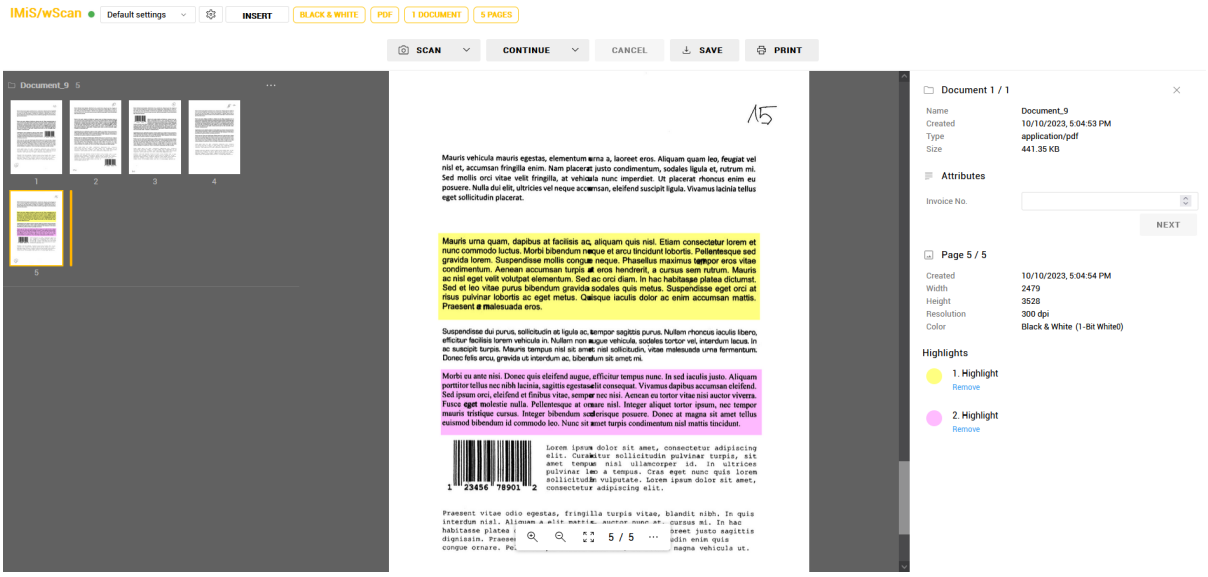

Image 98: Area of highlighted text on the document page

The user can remove the highlights by selecting the action "Remove" under the selected highlight on the “Highlights” list in the document properties view.

Note: When saving the document by selecting the action “Save”, the redactions merge with the document pages, which is why they cannot be removed after saving nor can the page be restored to its previous condition.

Note: The default highlight color is yellow; by default, the range of highlight colors consists of five different colors. Both can be customized via the “highlightColor” and “highlightColors” settings in the “imis.scan.ui.ImageView” and “imis.scan.ui.ImageScroll” objects. For more information see chapters ImageViewOptions and ImageScrollOptions.

7.2.13 Defining the metadata region

If at least one piece of metadata is defined in the settings, then the user can define the regions for metadata capture in the central view in the popup menu.

The popup menu shows the identifier of the first piece of metadata defined in the settings, while the contained menu shows all the other metadata too.

2023 © MARGIS Inc., All rights reserved.

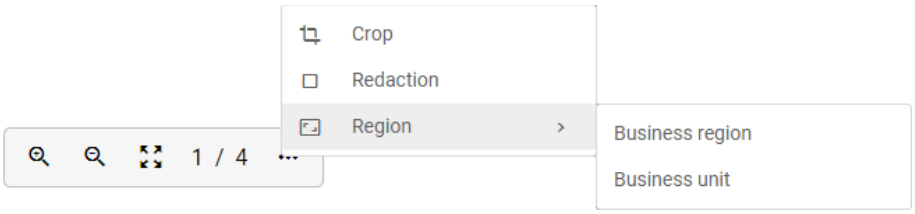

Image 99: Selecting the metadata for defining the region

The user can define the capture region for each piece of metadata. The user confirms the selection with “Done” or rejects it with “Cancel”.

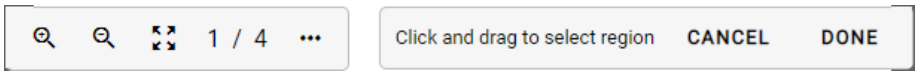

Image 100: Selecting the region for metadata capture

By selecting the metadata in the “Attributes” set in the right view, where the view details are located, the user enlarges not only the set of predefined values but also the selected metadata display region in the central view.

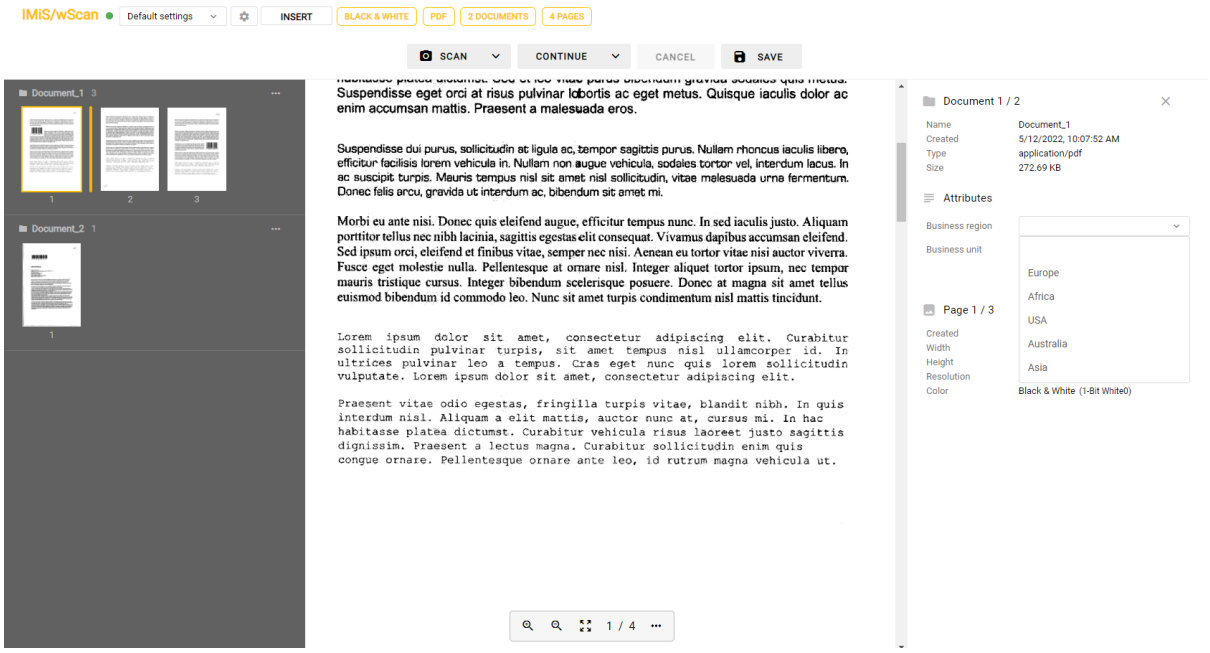

Image 101: View of the enlarged region for metadata capture

7.2.14 Access to service log files

When trying to solve problems with using the IMiS®/wScan application, it often helps to take a look at the log files of IMiS®/Capture Service, where its performance is being recorded. The log files are located in the directory *%PROGRAMDATA%\Margis\IMiS Capture Service\Logs*; access to the service log files is also enabled via the link "Download logs" on the "IMiS®/wScan Settings" page.

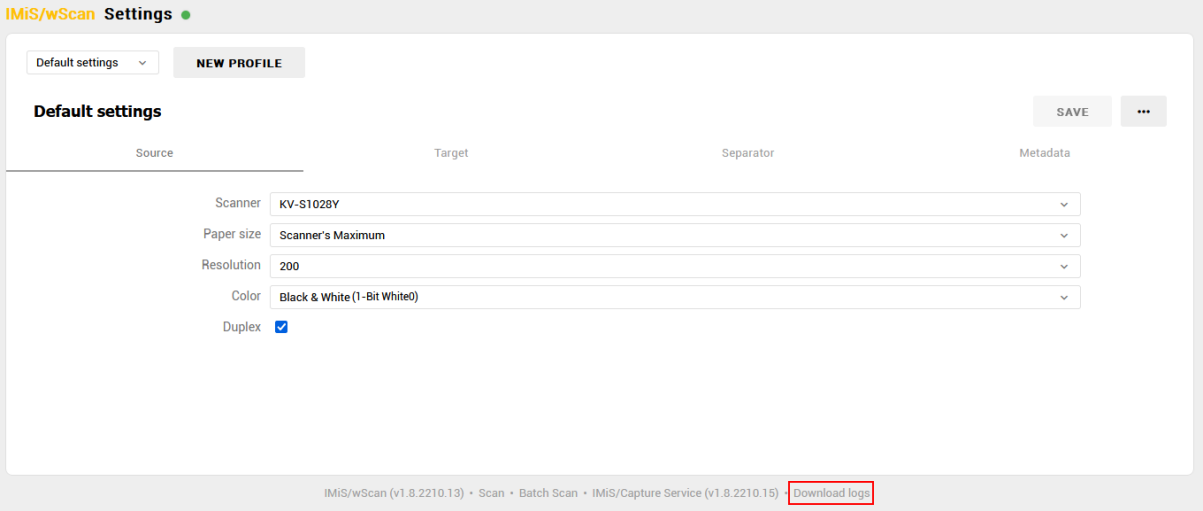

Image 102: Access to IMiS®/Capture Service log files

7.3 Batch scanning features

Important: Batch scanning features are available to the developer and to the user with the appropriate license key only.

When activating the license, the basic features of scanning, displaying documents, displaying document data and performing basic operations on the scanned documents will be expanded with additional batch scanning features which include:

- Separating documents by page count (requires the DOCSEP license feature).
- Separating documents using barcodes (requires the DOCSEP license feature and at least one feature of the BAR1DPIX or BAR2DPIX, or BAR1DST or BAR2DST license.

2023 © MARGIS Inc., All rights reserved.

For more information on activation and on the features of the license key see chapter Product activation.

To display the batch scanning features, we have chosen the “Gallery sample” mode which is suitable for displaying larger quantities of scanned documents.
The following building blocks are available to the user:

- a command bar showing document information;
- a view of document thumbnails;
- a view of document pages and document data.

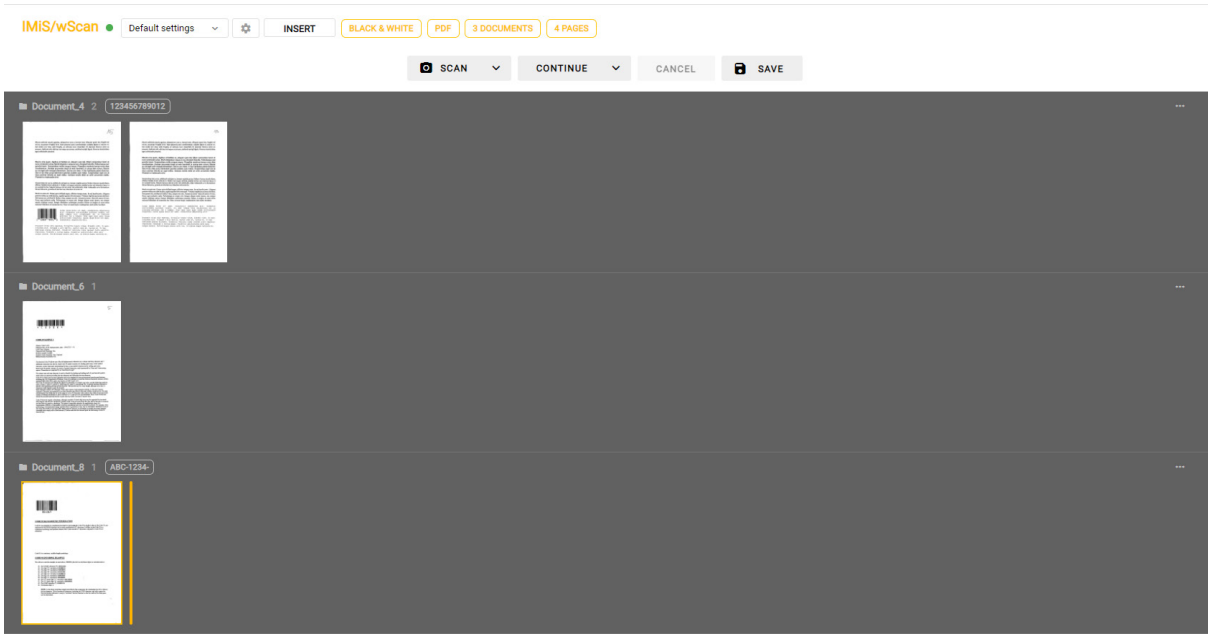

Image 103: User interface in the “Gallery” mode

In the command bar the user is provided the information on the scanned documents, the profile and job settings, and the buttons for performing actions.

For a more detailed description of individual actions see chapter Document capture.

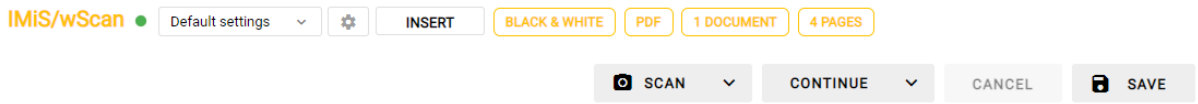

Image 104: A command bar with document information in the "Gallery" mode

In the central view of the user interface the user is shown the thumbnails of document pages and the popup menu for performing actions.

For a more detailed description of individual actions see chapter Document thumbnails.

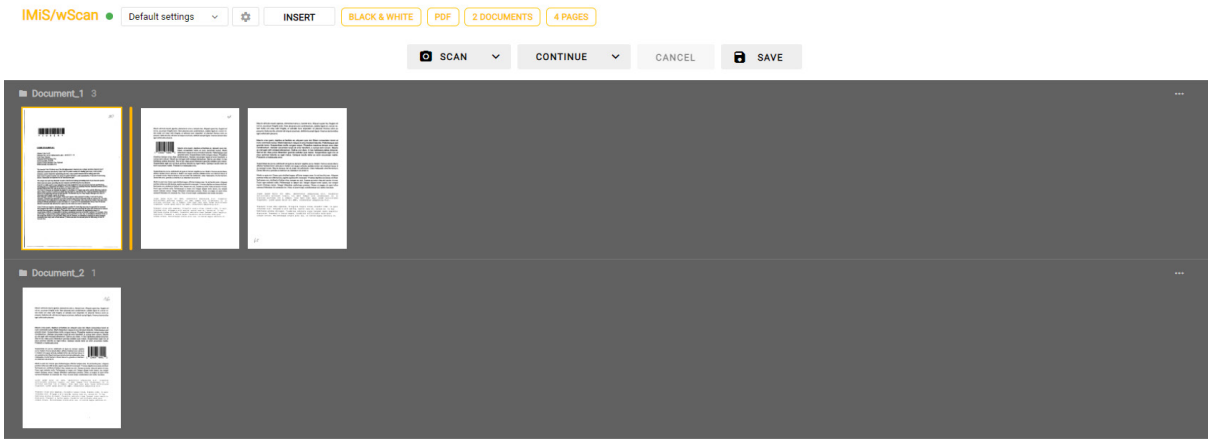

Image 105: Thumbnails of document pages in the "Gallery" mode

2023 © MARGIS Inc., All rights reserved.

Image 106: Popup menu on document in the “Gallery” mode

By selecting the action “Properties” in the popup menu on the document, the user is shown a dialog box displaying the selected document pages and document data. For more information see chapter Document pages and Document data.

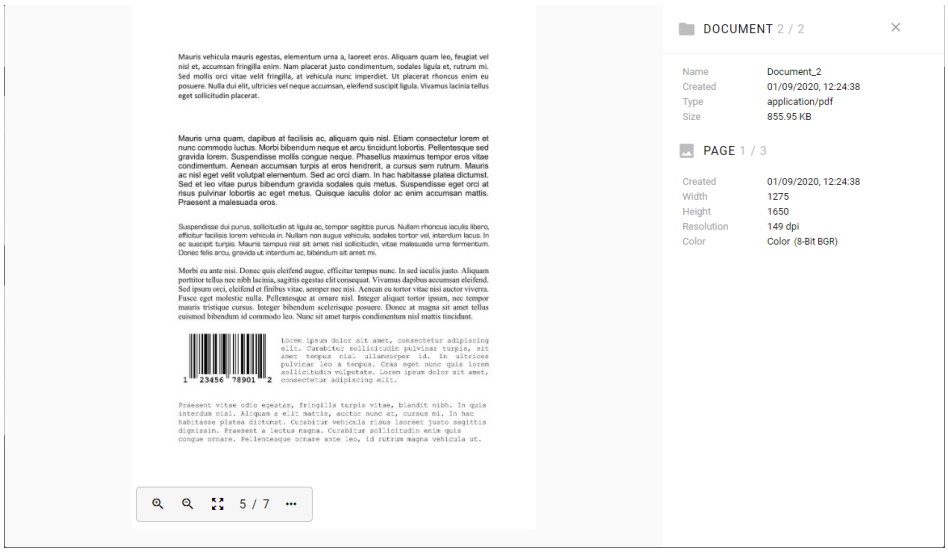

Image 107: View of document data in the “Gallery” mode

7.3.1 Separating documents by page count

Separating documents by page count requires a license with an activated DOCSEP feature. For more information see chapter Product activation.

For more information on obtaining a license key for expanding the set of features with batch scanning, send an email to info@imis.si.

When scanning with the activated separation of documents by page count, the pages are added to the document up to (and inclusive) the number of pages specified in the "Page count" field in the separator settings (see chapter Separator). Afterwards, a new document is created and the process is repeated until it runs out of pages to scan.

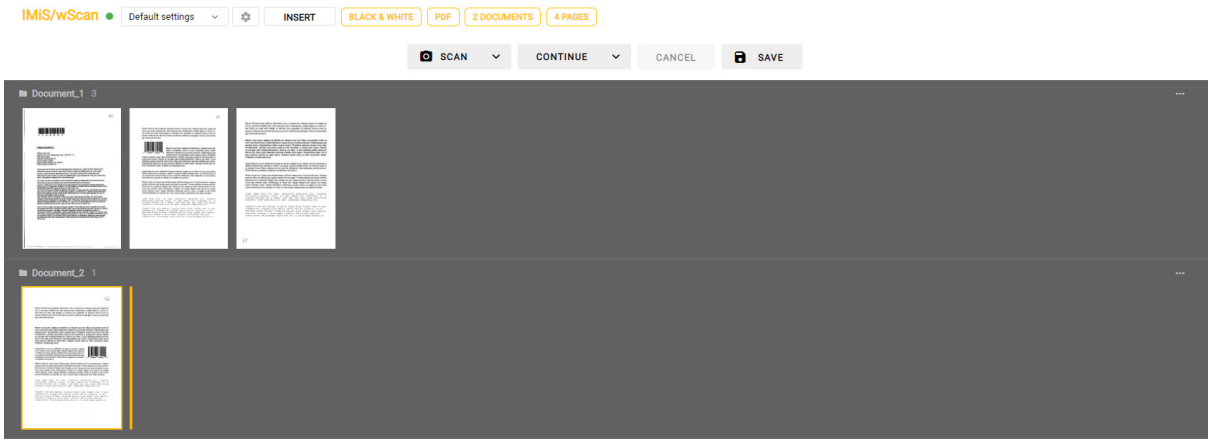

Image 108: Example of separation by page count (N = 3)

7.3.2 Separating documents by barcode

Separating documents by barcodes requires a license with an activated DOCSEP feature and at least one BAR1DPIX or BAR2DPIX, or BAR1DST or BAR2DST feature (see chapter Product activation).

For more information on obtaining a license key for expanding the set of features with batch scanning, send an email to info@imis.si.

2023 © MARGIS Inc., All rights reserved.

When scanning with the activated document separation by barcodes, the pages are added to the document up to (and exclusive) the page containing the recognized barcode that represents the separator (see chapter Separator). The page containing the barcode that represents the separator is either included in the new document if the “Separate” action has been selected, or deleted if the “Separate and delete” action has been selected, and the scanning continues until the next recognized separator.

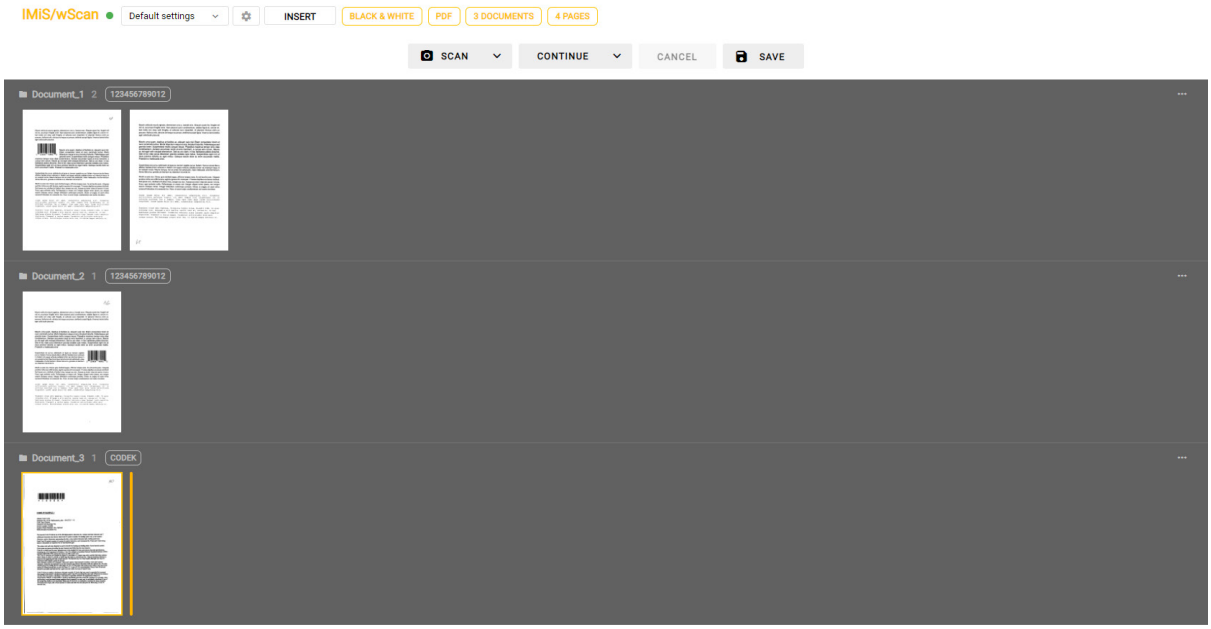

Image 109: Example of separation by barcode

7.3.2.1 Separator editing

By selecting the action “Edit separator” in the popup menu on the selected document page in the left view, the user can edit the record of the existing separator or add a new one.

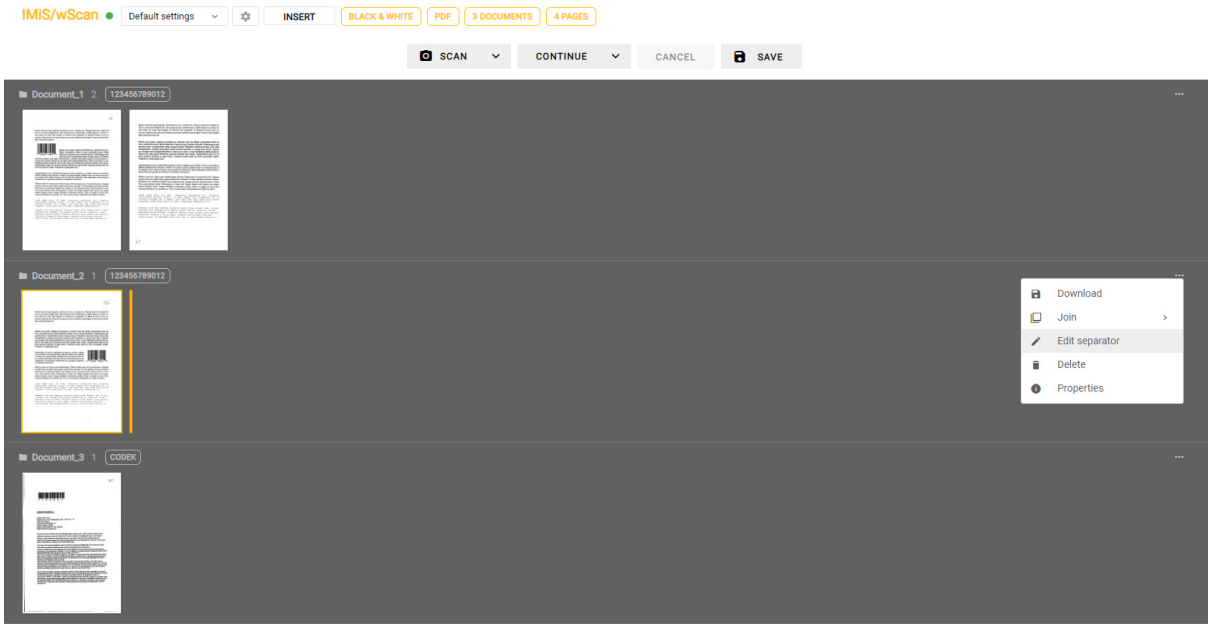

Image 110: Selecting the action “Edit separator”

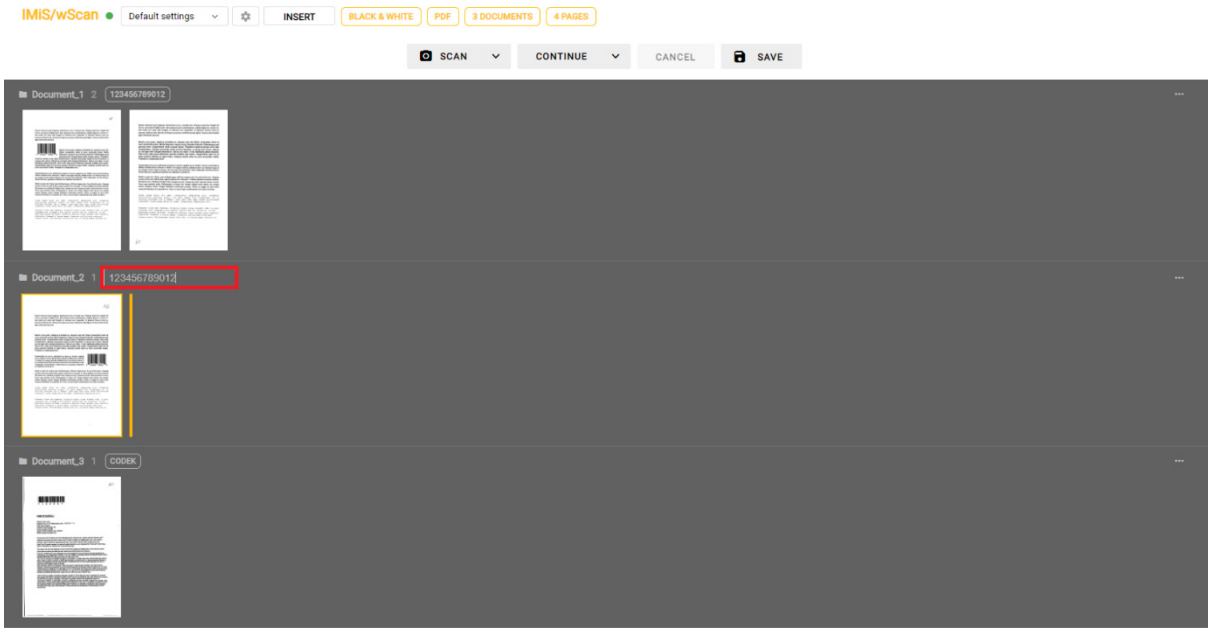

Image 111: Changing the value or adding a new separator next to the document label

2023 © MARGIS Inc., All rights reserved.

The user can also change the value of the “Separator” attribute in the document properties.

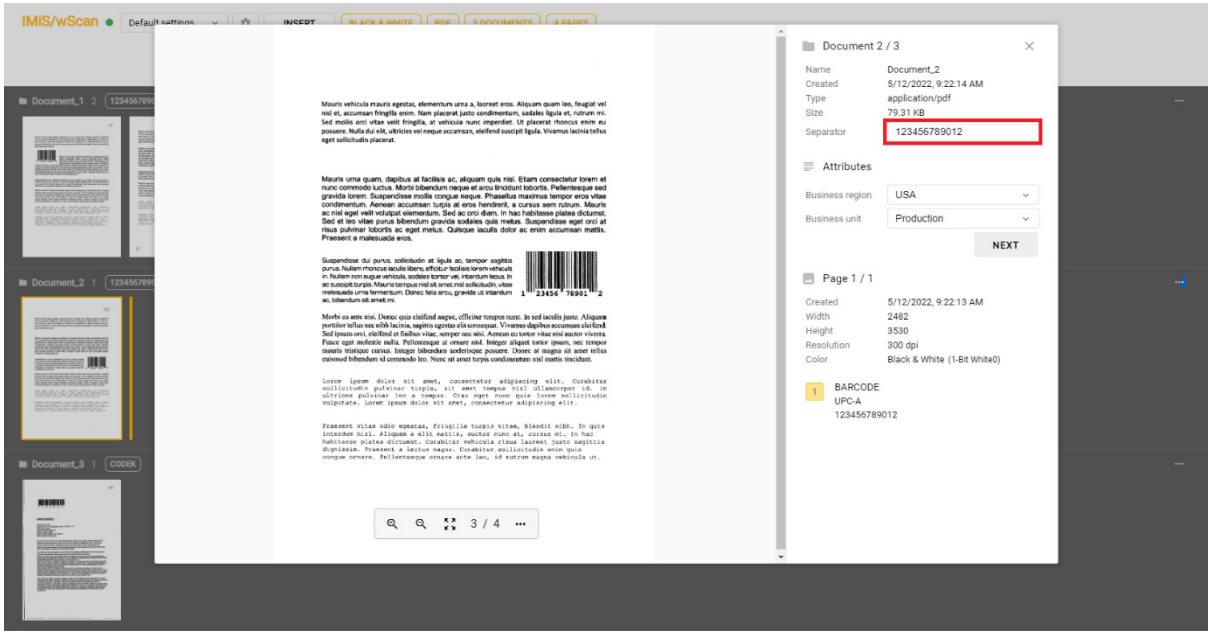

Image 112: Changing the separator value in document properties

7.3.3 Blank page separation

Blank page separation of documents requires a license key with the activated DOCSEP feature. For more information see chapter Product activation.

For more information on obtaining a license key for expanding the set of features with batch scanning, send an email to info@imis.si.

When scanning with the blank page separation of documents activated, the pages are added to the document up to (and exclusive) the blank page that represents the separator (see chapter Separator).

2023 © MARGIS Inc., All rights reserved.

The blank page is not included in the new document. The scanning continues until the next recognized separator, i.e., blank page. Blank pages at the beginning of the scan are discarded.

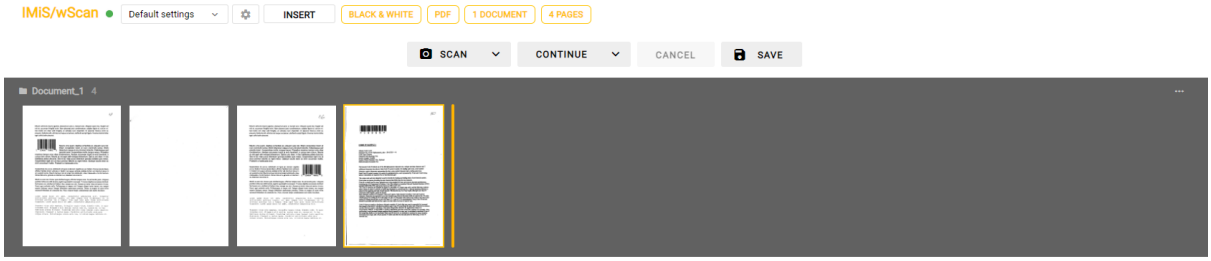

Image 113: Example of failed blank page separation

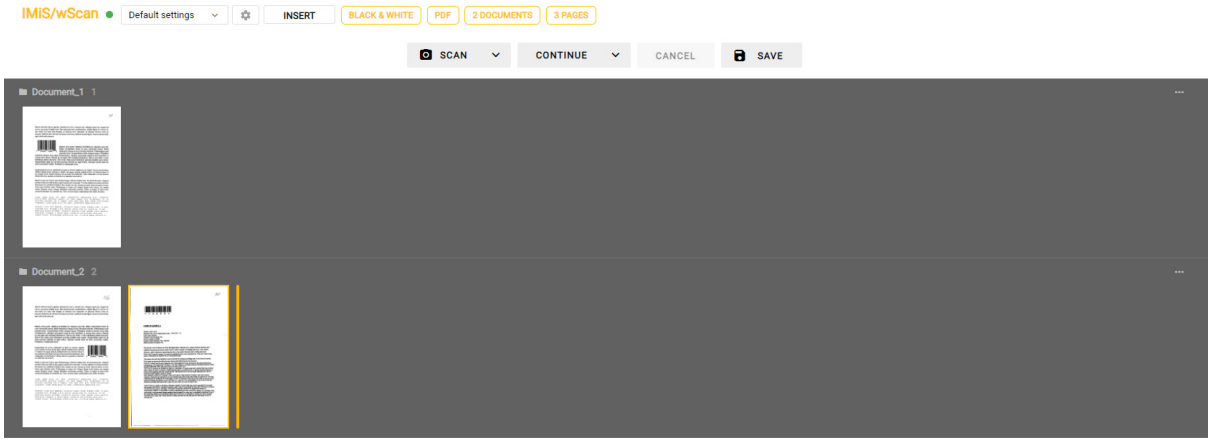

Image 114: Example of successful blank page separation

2023 © MARGIS Inc., All rights reserved.

7.3.3.1 Setting the threshold

In the case of default settings, blank page detection might not meet the user's expectations due to various factors (e.g., unsuitable scanning settings, paper show-through, the purity of the blank page or a dusty scanning device), which is reflected in incorrect document separation by blank pages.

The user can influence blank page detection through the setting "Threshold". Increasing this value captures a larger span of pages considered blank, while decreasing this value reduces the span.

The recommended values for setting the threshold for blank A4 black-and-white pages at a resolution of 300dpi with suitable scanning settings are:

- Pristine white: 0-20
- Dirty white: 20-40
- Very dirty white: 40-80.

The user can make defining the threshold setup values easier by first scanning a sample of different blank pages. In the file IMiS.Capture.Service. NET.0.log in the directory *%PROGRAMDATA%\Margis\IMiS Capture Service\Logs* the user then looks for the entries <BlankpageDetectorHandler.IsBlank> with the calculated values of the blank page detection algorithm. Based on these, the user selects the appropriate value.

```
10. 05. 2022 15:07:14.115 14492:102 imis.capture.service Verbose <BlankpageDetectorHandler.Process> Entering method with no parameters.
10. 05. 2022 15:07:14.115 14492:102 imis.capture.service Verbose <BlankpageDetectorHandler.Process> Detecting whether page is blank...
10. 05. 2022 15:07:14.152 14492:102 imis.capture.service Verbose <BlankpageDetectorHandler.IsBlank> Page image standard deviation of pixel values: 198,895909106573
```

Image 115: Example of the value of the threshold for non-blank document page detection

```
10. 05. 2022 15:07:15.024 14492:102 imis.capture.service Verbose <BlankpageDetectorHandler.Process> Entering method with no parameters.
10. 05. 2022 15:07:15.024 14492:102 imis.capture.service Verbose <BlankpageDetectorHandler.Process> Detecting whether page is blank...
10. 05. 2022 15:07:15.053 14492:102 imis.capture.service Verbose <BlankpageDetectorHandler.IsBlank> Page image standard deviation of pixel values: 77,8802452132422
```

Image 116: Example of the value of the threshold for blank document page detection

2023 © MARGIS Inc., All rights reserved.

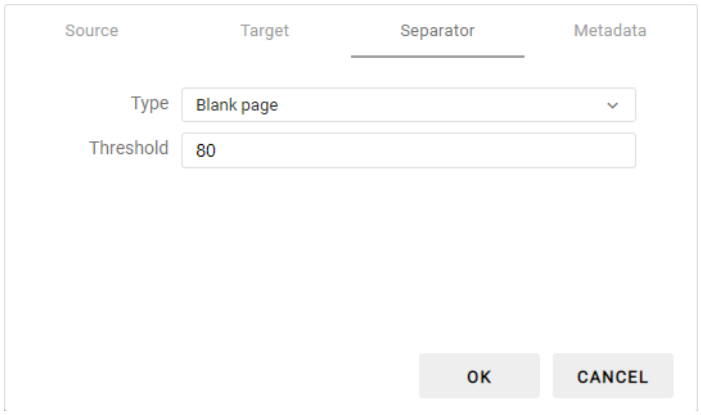

Image 117: Example of setting the "Threshold" value for successful document separation

Note: Before starting the scan and separation of documents by blank pages, the user must make sure that blank page removal is not activated on the driver. If it is, then blank page separation will not work, as the application will not get the removed blank pages.

2023 © MARGIS Inc., All rights reserved.

8 TROUBLESHOOTING

8.1 Problems using IMiS®/wScan

Below, issues frequently encountered when using IMiS®/wScan application are described and instructions for resolving them are given.

8.1.1 Error “Forbiden”

Cause of problem

The web application has not forwarded the correct security key via the “imis.scan.js” library to the IMiS®/Capture Service.

Solution for problem

The user with administrator authorization must verify that the security key is the same both in the IMiS®/Capture Service and the web application. If a new security key was entered or created in the IMiS®/Capture Service, a user with administrator authorization must restart the service. After the restart, the browser’s web page for displaying scanned documents must be refreshed.

8.1.2 Error “Error in establishing connection”

Cause of problem

The web application accesses the IMiS®/Capture Service via a domain that is not allowed in the IMiS®/Capture Service.

Solution for problem

A user with administrator authorization must enter the allowed domains from which the web application can access the IMiS®/Capture Service. After entering them in the IMiS®/Capture Service, the service must be restarted. After the restart, the browser’s web page for displaying scanned documents must be refreshed.

2023 © MARGIS Inc., All rights reserved.

8.1.3 Error “Socket connection error”

Cause of problem

The web service IMiS®/Capture Service has not been started or the client does not have access rights for the address at which IMiS®/Capture Service is located.

Solution for problem

The administrator has to restart IMiS®/Capture Service. After restarting the service refresh the web page in the browser to show the scanned documents.

8.1.4 Error “Error in establishing connection” or “Socket connection error”

Cause of problem

MS Edge is a UWP application that runs in isolation from the local computer and which does not have access to the localhost if called from another domain.

Solution for problem

The solution is running the command in the command bar as a user with administrator authorization:

CheckNetIsolation LoopbackExempt -a -n=Microsoft.Windows.Spartan_cw5n1h2txyewy

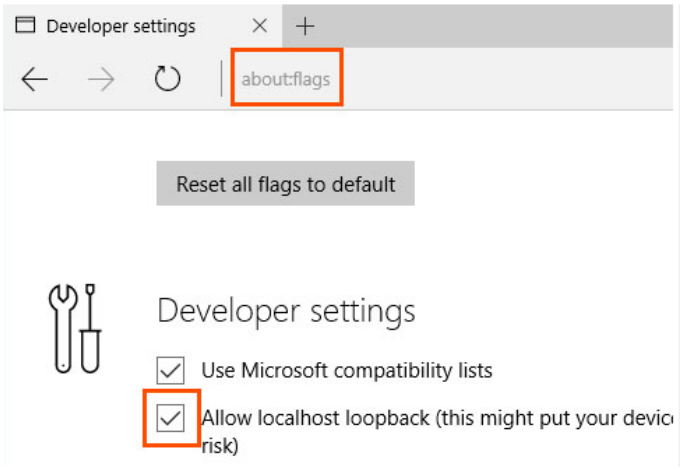

Image 118: Shows the entry in the MS Edge title bar

If executing the above command still does not enable access, the user with administrator authorization can run the following command:

CheckNetIsolation LoopbackExempt -a -p=S-1-15-2-3624051433-2125758914-1423191267-1740899205-1073925389-3782572162-737981194

Detailed information on the problem and how to resolve it is available at these links:

https://blogs.msdn.microsoft.com/msgulfcommunity/2015/07/01/how-to-debug-localhost-on-microsoft-edge/
http://solidlystated.com/software/edge-windows-10-cant-reach-localhost-sites/

8.1.5 Error “No scanner is connected”

Cause of problem

When starting IMiS®/Capture Service, the service could not locate or load a driver for the connected scanner.

Solution for problem

The administrator has to check whether the scanner is turned on and connected to the computer. Next, check if the IMiS®/Scan application is running. If it is running, you have to stop it and restart IMiS®/Capture Service. For more information see chapter Startup and closing.

8.1.6 Error “Scanner: ‘{scanner name}' cannot be loaded”.

Cause of problem

Prior to starting the scan IMiS®/Capture Service was unable to load a driver for the selected scanner.

Solution for problem

The administrator has to check whether the scanner is turned on and connected to the computer. Next, check if the IMiS®/Scan application is running. If it is running, you have to stop it and try scanning in the IMiS®/wScan application. If scanning is still not working, you have to restart IMiS®/Capture Service. For more information see chapter Startup and closing.

2023 © MARGIS Inc., All rights reserved.

8.1.7 Scanning cannot be continued after successful scanning

Cause of problem

During scanning the data connection between the scanner and computer was terminated.

Solution for problem

The administrator has to turn the scanner off and disconnect it from the workstation. Then turn the scanner back on and connect it to the workstation. Start the scanning procedure again in the IMiS®/wScan application. If scanning is still not working, open the Task Manager program. Find the process named “fjictwsw.exe” among all the processes and end it by clicking on the button “End process”.

8.1.8 During scanning empty pages are not being removed

Cause of problem

In the profile settings the scanner settings for removing empty pages have not been configured.

Solution for problem

After configuring the profile in the IMiS®/wScan application, the administrator starts the IMiS®/wScan administration module and follows the procedures described in chapter Additional settings.

8.1.9 Error “Your browser does not support Javascript ES6. Update browser.”

Cause of problem

The “imis.scan.js” library does not work if the browser does not support the ECMAScript6 JavaScript standard.

Solution for problem

The existing browser has to be updated to the latest version.

Warning: The latest version of the Internet Explorer browser does not support this standard.

2023 © MARGIS Inc., All rights reserved.

8.1.10 Error “Your browser does not support WebSockets. Update browser.”

Cause of problem

The “imis.scan.js” library does not work if the browser does not support WebSocket technology.

Solution for problem

The existing browser has to be updated to the latest version.

Warning: The latest version of the Internet Explorer browser does not support this standard.

2023 © MARGIS Inc., All rights reserved.