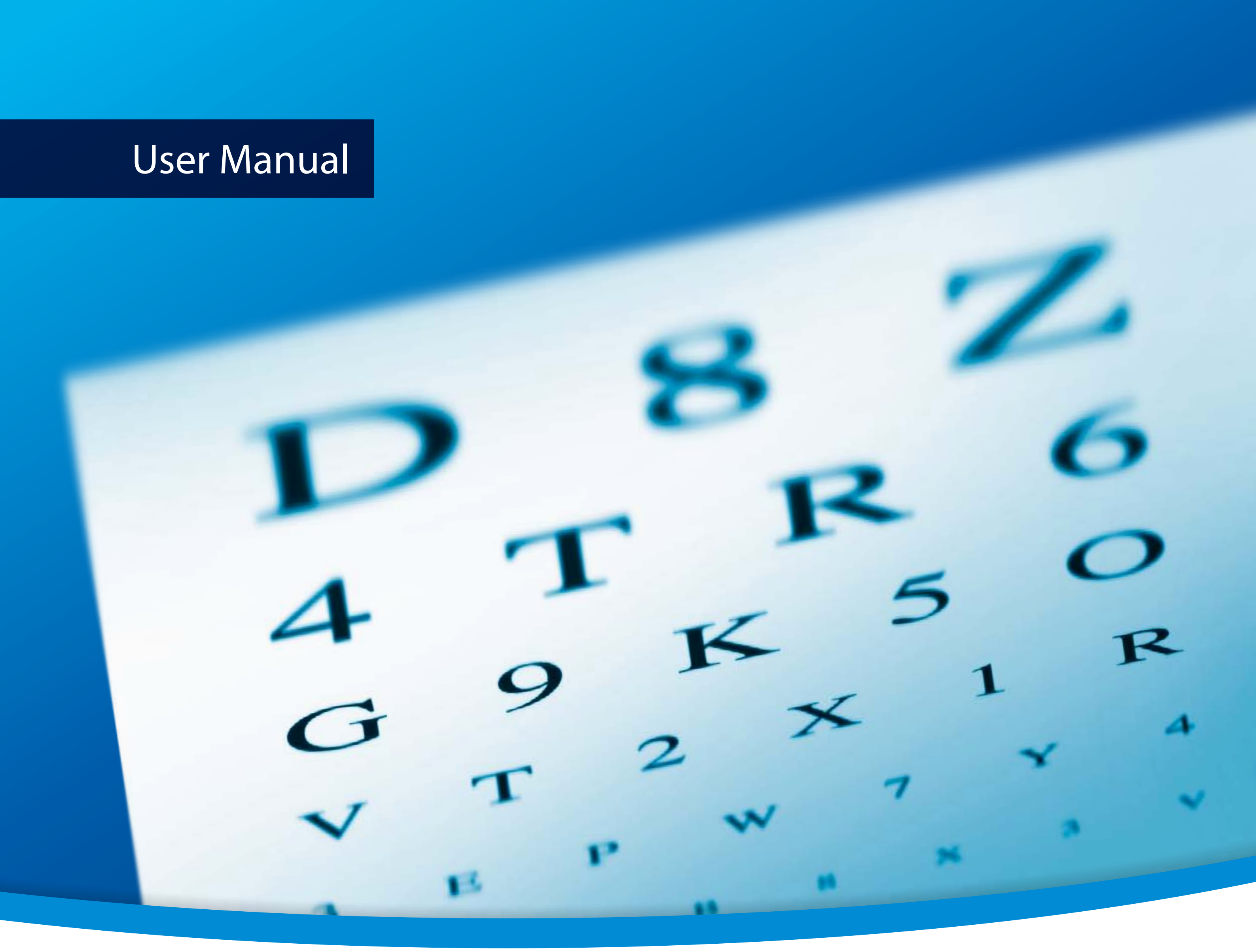

User Manual

3-Heights™ OCR Service

Version 6.15.0

PDF-TOOLS.COM
Premium PDF Technology

Contents

1 Introduction

The 3-Heights™ OCR Service provides an infrastructure for out-of-process OCR processing.

1.1 Advantages

The 3-Heights™ OCR Service has several advantages over the direct use of an OCR engine:

1.1.1 Parallel Execution

Most supported OCR engines are not thread-safe. That means that a single process can only process one page at the time.

The 3-Heights™ OCR Service solves that problem by starting multiple worker processes.

By configuring multiple service points in the client application, the load can even be distributed across multiple computers.

For multithreaded applications, using the 3-Heights™ OCR Service is mandatory. This includes all 3-Heights™ watched folder services that can use OCR:

- 3-Heights™ Image to PDF Converter Service
- 3-Heights™ PDF to PDF/A Converter Service
- 3-Heights™ Document Converter
- 3-Heights™ Scan to PDF Server

1.1.2 Performance

Initializing an OCR engine is quite an expensive operation. The 3-Heights™ OCR Service can reuse already initialized engines for following documents and pages.

This is especially useful for short-living client processes like shell tools.

1.1.3 Platform Support

The 3-Heights™ OCR Service uses a HTTP interface. This enables OCR support for platforms that are otherwise not supported by the OCR engine.

1.1.4 Robustness

The fact that the OCR processing is done in a separate process greatly increases the robustness of the client application.

If the OCR engine produces a crash, only the respective worker process is terminated. The 3-Heights™ OCR Service and the client application remain untouched.

1.1.5 Resource Usage

Image processing is a very memory intensive task.

By using the 3-Heights™ OCR Service you can greatly decrease the memory consumption of the client application.

Having a dedicated OCR process also increases the maximum image size since no memory is needed for other purposes in that process.

2 Installation

2.1 Requirements

2.1.1 Operating Systems

The 3-Heights™ OCR Service is available for the following operating systems:

- Windows Client 7+ | x86 and x64
- Windows Server 2008, 2008 R2, 2012, 2012 R2, 2016, 2019 | x86 and x64

'+' indicates the minimum supported version.

2.1.2 OCR Engine

The OCR Service needs a working Installation of an OCR engine that is supported by the 3-Heights™ product line.

Supported are:

- ABBYY FineReader Engine 10
- ABBYY FineReader Engine 11
- ABBYY FineReader Engine 12
- Barcode and QR Code Recognition Engine

2.2 Installation of the OCR Service

The 3-Heights™ OCR Service can be downloaded from your customer account at `http://www.pdf-tools.com`.

1. Download the MSI `Ocr-Service-‹version›-Windows-(‹platform›).msi` from your download account
2. Ensure the port 7982 is available on the system where you install the service.
3. Double-click the MSI file to start the installation wizard.
4. Follow the installation wizard.

The installation automatically installs and optionally starts the 3-Heights™ OCR Service. Upon un-installing, the service is stopped and un-installed.

3 License Management

The 3-Heights™ OCR Service requires a valid license in order to run correctly. If no license key is set or the license is not valid, then an error message will be printed to the service log.

More information about license management is available in the license key technote.

4 User's Guide

4.1 Architecture

The 3-Heights™ OCR Service uses a client/server architecture to perform the actual OCR recognition out of process.

The client and server are typically on the same computer, but in a high load scenario it makes sense to use a dedicated machine just for OCR processing.

4.1.1 Server

For robustness reasons, the 3-Heights™ OCR Service uses multiprocessing on the server, so that only a single recognition is performed at a time in the same process.

Dispatcher Process

The dispatcher process provides a lightweight HTTP/REST webservice that can be accessed by client applications.

The actual recognition work is delegated to worker processes that are automatically managed by the dispatcher process.

The dispatcher process is thus not affected by errors or even crashes of the actual OCR engine.

Worker Process

The worker performs the actual recognition work by loading the configured OCR plugin.

Communiction with the dispatcher process is done over a TCP channel.

4.1.2 Client

On the client side, the standard OCR plugin infrastructure is used. This infrastructure is supported by all PDF Tools products with support for OCR.

Client Application

The client application (a PDF Tools Product) uses the `"service"` OCR plugin the same way as it would use a normal OCR plugin.

OCR Plugin "service"

The `"service"` OCR plugin implements the standard PDF Tools OCR plugin interface.

Instead of performing the recognition work itself, it delegates the job to the 3-Heights™ OCR Service.

4.2 Monitoring

4.2.1 Windows event log

The OCR Service can be configured to use the **Windows Event Log** by setting the `<eventlog>` element in the Service Configuration File.

The Windows Event Log can in turn be monitored using the **Windows Task Scheduler**.

4.2.2 Minimum remaining page credits

Page credits can be automaticall monitored using the `min-credits` attribute in the Service Configuration File.

If this attribute is set, the service will check the remaining page credits once every hour and if the number is below the limit, a warning is issued to the log (File or Windows event log).

4.2.3 Service status

The service status can be queried using the HTTP interface:
`http://<host>:<port><service-root>status.xml[?version=<version>]`

For the example configuration, this would be:
`http://exampleserver:7982/status.xml?version=1`

The XML format is defined in Appendix A.

The `version` attribute specifies the maximum version that is understood by the client. The server reply can however reply with an older XML format version.

If no version is specified, it is assumed to be 1.

4.3 Asynchronous Processing

The 3-Heights™ OCR Service supports non-blocking retrieval of the OCR results. This allows the client to parallelize OCR processing without using multi-threading.

> **Note:** The client application has to be specifically designed and optimized for asynchronous processing. At the moment, the only PDF Tools product that supports asynchronous processing is the newly released **3-Heights™ PDF OCR**.

Asynchronous processing works in a limited fashion also with older versions (< 4.11) of the OCR Service. In that case, the client relies on a short timeout setting for nonblocking operations. However, performance is much worse than with true non-blocking requests.

4.4 Load Considerations

4.4.1 Load Balancing

Load balancing can be achieved by specifying multiple connections in the Client Configuration file.

- For a single recognition job, a server is selected at random.
- If multiple jobs are issued from the same client process, the client tries to distribute the jobs intelligently across the servers by taking into account the current queue size of each server.
- Servers that are not responding are skipped.

Since load balancing is done on the client side, it also works (in a limited fashion) with older versions (< 4.11) of the OCR Service or with a mix of versions: Older versions don't report their current queue size, so the distribution of the jobs across the servers will not be optimal and they will overload more quickly.

4.4.2 Backpressure

The 3-Heights™ OCR Service has only a limited internal job queue (configurable in the Service Configuration File). Once the queue is full, incoming requests are blocking until the queue size falls under the limit again.

This mechanism tries to minimize service overload by slowing down the client.

Since backpressure is done on the server side, it also works (in a limited fashion) with older client versions (< 4.11): Older versions use a shorter timeout setting for posting new jobs, which will lead to errors more quickly.

4.4.3 Overload and Timeout

Despite the Load Balancing and Backpressure mechanisms it is still possible to overload the OCR service by having too many client connections that all try to post new jobs at the same time. Backpressure can only limit the job creation rate of a single client connection, it cannot control the **number** of client connections.

If that scenario happens, the service will at some point not be able to accept jobs in time and requests will start to time out. This will lead to errors on the client side but does not otherwise affect the function of the OCR service.

The timeout values can be configured in the Client Configuration file. However, timeouts are usually just a symptom of the problem, not the problem itself. Increasing timeout settings only delays the error and does not prevent it.

The correct solution to this problem is to limit the number of client connections (See Client parallelism).

4.4.4 Dimensioning

To ensure a reliable operation, it is important that the settings for all components are coordinated.

Server parallelism

The server parallelism is configured in the Service Configuration File. The value is limited by the number of logical processors and by the license of the actual OCR engine.

If the license allows it, the setting should be roughly equal to the number of logical processors.

If the license of the OCR engine is restricted to a number of logical processors (cores), this number must not be exeeded. Doing so will inevitably lead to errors.

Client parallelism

Ideally, the client parallism should not exceed the parallelism of the server. Since OCR processing is usually the bottleneck in the process, increasing the client parallelism above that value will only result in higher latency, not higher throughput.

The absolute maximum number of clients connections that can be handled simultaneously without timeout errors is calculated as:
$N = \mathrm{r}_S * \mathrm{t}_C$ where

- r_S is the server's processing rate, i.e. the average number of jobs that can effectively be processed per second.
- t_C is the client timeout setting.

> **Note:** This value is only valid for the average case. The server's actual processing rate cannot be statically determined as it varies with the size and complexity of the images to be recognized.
> It is therefore not recommended to exhaust that limit.

Example: Assuming default settings (server parallelism setting `parallel="2"`, client timeout $t_C = 600$) and an average processing time of 3 seconds per job, the maximum number of concurrent client connections is:
$N = (2/3) * 600 = 400$

4.5 Structure Information

The 3-Heights™ OCR Service supports recognition of page structure information. That information can be used to enhance the resulting document, e.g. by generating accessibility tags.

This feature is only available if supported by the actual OCR plugin. Currently, the only plugins that provides that information are “abbyy11” (ABBYY FineReader Engine v11) and “abbyy12” (ABBYY FineReader Engine v12).

Note: While the OCR Service provides the structure information to the client, it is up to the client application to actually use that information. At the moment, the only PDF Tools product that makes use of structure information is the newly released **3-Heights™ PDF OCR**.

5 Reference Manual

5.1 Service Configuration File

The service configuration of the 3-Heights™ OCR Service is done by editing the configuration file `OcrSvr.xml`. The file must reside in the same directory as the executable `OcrSvr.exe`.

5.1.1 XML Structure

`<ocrserver>`

`port` The TCP-Port where the service is listening.

`service-root` The absolute URL-path for accessing the service.
This value must start and end with a slash (/).

`work-folder` The working directory of the service.
Default value is `C:\ProgramData\PDF Tools AG\3-Heights(TM) OCR Service`.

`parallel` The number of parallel OCR processes.

The number is automatically reduced if the license has a CPU core restriction.

`default-engine` The default OCR engine used by the service.
Possible values are:

`"abbyy10"` ABBYY FineReader Engine v10

`"abbyy11"` ABBYY FineReader Engine v11

`"abbyy12"` ABBYY FineReader Engine v12

`min-credits` The minimum required page credits. If the number of credits falls below this limit, a warning is issued. If this attribute is missing or has a negative value, page credit monitoring is disabled.

`max-queue` The maximum internal job queue size. If the queue is full, new job requests are blocked. (See Chapter 4.4.2).
The default value is the value of `parallel` + 2.

`<event-log>` Enable logging to the Windows event log.

`reporting-level`

`"none"` Logging disabled

`"error"` Only errors are logged

`"warning"` Errors and warnings are logged

`"all"` Everything is logged

`<log-file>`

`path` The path of the file (relative to the work-folder).

`unique-pattern` C format string to make the filename unique.

`reporting-level`

`"none"` Logging disabled

`"error"` Only errors are logged

`"warning"` Errors and warnings are logged "all" Everything is logged

<config> Default configuration for a specific engine.

engine The engine to be configured.

<params> The default parameters for the OCR engine. Possible values are specific to the respective OCR engine. The value can be overridden by clients.

<languages> The default languages for the OCR engine. Possible values are specific to the respective OCR engine. The value can be overridden by clients.

Example: Typical configuration

```
<?xml version="1.0" encoding="utf-8"?>
<ocrserver xmlns="http://www.pdf-tools.com/ocrsrv"
           port="7982"
           default-engine="abbyy11"
           parallel="2"
           service-root="/"
           >
  <event-log reporting-level="warning"/>
  <log-file reporting-level="warning"
            path="OcrSvr.log"
            unique-pattern=".%d"/>
</ocrserver>
```

Example: Extensive configuration

```
<?xml version="1.0" encoding="utf-8"?>
<ocrserver xmlns="http://www.pdf-tools.com/ocrsvr"
           port="7982"
           service-root="/"
           work-folder="C:\OCR Server Working Directory"
           parallel="2"
           default-engine="abbyy11"
           min-credits="100"
           max-queue="10">
    <event-log reporting-level="warning"/>
    <log-file reporting-level="all"
              path="OcrSvr.log"
              unique-pattern=".%d"/>
    <config engine="abbyy11">
      <params>PredefinedProfile=DocumentArchiving_Accuracy</params>
      <languages>English, German</languages>
    </config>
</ocrserver>
```

5.1.2 Service point URL

The URL where the 3-Heights™ OCR Service can be accessed is defined as follows:
`http://<host>:<port><service-root>`

- <host> The hostname or IP address of the server.
- <port> The port as configured in the OcrSvr.xml file.
- <service-root> The service root path as configured in the OcrSvr.xml file.

Example: The service URL for the example above running on the server `exampleserver` would be `http://exampleserver:7982/`

5.2 Client Configuration

5.2.1 Service engine proxy

The 3-Heights™ OCR Service can be used in a client application via the `"service"` OCR plugin proxy. This is a special OCR plugin that relays OCR processing to a server instead of doing it on the client.

The `"service"` plugin can be instead a local OCR engine like `"abbyy11"` or `"abbyy12"`.

The `"service"` plugin supports 3 different forms to specify the service point URL:

- Explicitly in the engine name like: `"service@http://<host>:<port><service-root>"`
- Using a Client Configuration file `ocrserver.ini` and only `"service"` als engine name.
- Without specifying the service point explicitly and without a configuration file, in which case the default service point `"http://localhost:7982/"` is used.

5.2.2 Client Configuration file

One way to tell the client application where to find the 3-Heights™ OCR Service is to deploy the configuration file `ocrserver.ini`. The INI-file must reside in the same directory as the main client application (e.g. `pdf2pdfsvr.exe`).

`[OCRServer]` INI-File Section

`Count`

Key: `Count` Type: `Integer`

Number of service points.

`<Number>`

Key: `<Number>` Type: `URL`

Service point URL, where `<Number>` is a value from `1` to `Count` and identifies a single service point.

The URL is the service point URL of the service as configured in the `OcrSvr.xml` congfiguration file on the server.

`Compression`

Key: `Compression` Default: `AUTO`

HTTP compression (deflate) for download.
Supported values are:

TRUE Compression enabled.

FALSE Compression disabled.

AUTO Compression enabled if the host of the service point is not `localhost`

`Proxy`

Key: `Proxy`

URL of a HTTP proxy server that is used for all connections.

`Timeout`

Key: `Timeout` Default: `60`

The default timeout in seconds for normal operations: This timeout applies for all requests don't use a special timeout.

`BlockingTimeout`

Key: `BlockingTimeout` Default: `600`

The timeout in seconds for explicitly blocking long running operations:

- Posting a new job to the service. This primarily relevant in case backpressure.
- Blocking retrieval of OCR results.

`NonBlockingTimeout`

Key: `NonBlockingTimeout` Default: `5`

The timeout in seconds for explicitly non-blocking short running operations:

- Non-blocking retrieval of OCR results.

This timeout improves compatibility with older versions of the OCR service that don't natively support non-blocking operations.

Example: Extensive configuration.

In a typical configuration, most values can be omitted.

```
[OCRServer]
Count = 2
1 = http://localhost:7982/
2 = http://backup:7982/
Compression = AUTO
Timeout = 20
BlockingTimeout = 200
```

```
NonBlockingTimeout = 2
```

5.3 Status XML Structure

`<ServiceStatus>`

`version` The format version of the status XML. Currently 1.

`<Version>` The version of the OCR service.

`<DefaultEngine>` The default engine of the OCR service.

`<QueueSize>` The current queue size of the OCR service.

`<RemainingPageCredits>` The remaining page credits.

6 Version History

6.1 Changes in Version 6

- **Improved** HTTP processing and job scheduling: Reimplementation in .NET/WCF to improve robustness and stability.
- **Changed** log rollover behavior: Current log file now has no added number.
- **Changed** log format: A consistent format is used for all types of messages.
- **Improved** behavior with CPU-core restricted licenses: The parallelity is now reduced automatically to avoid runtime errors.
- **Changed** default location of work folder to `C:\ProgramData\PDF Tools AG\3-Heights(TM) OCR Service`

6.2 Changes in Version 5

- **New** additional supported operating system: Windows Server 2019.
- **Improved** robustness of the service in case of internal errors in the OCR engine.
- **New** element `<TotalPageCredits>` in status XML.

6.3 Changes in Version 4.12

- **New** OCR plugin `"abbyy12"` for the ABBYY FineReader 12 engine.
- **New** HTTP proxy setting in the GUI license manager.

Client configuration file `ocrserver.ini`

- **Changed** default value of key `NonBlockingTimeout` from `1` to `5`.

6.4 Changes in Version 4.11

- **New** feature: Intelligent load balancing.
- **New** feature: Backpressure to avoid overload.
- **Improved** handling of many concurrent requests.
- **New** feature: Asynchronous processing.
- **New** feature: Support for recognition of document structure information.

Server configuration file `OcrSvr.xml`

- **New** attribute `max-queue` to specify the maximum internal job queue size.

Client configuration file `ocrserver.ini`

- **New** key `Proxy` to specify a proxy server for all connections.
- **New** key `Timeout` to specify the general response timeout.

- **New** key `BlockingTimeout` to specify the response timeout for explicitly blocking operations.
- **New** key `NonBlockingTimeout` to specify the response timeout for explicitly non-blocking operations.

6.5 Changes in Version 4.10

- **Improved** robustness with regards to network errors/delays.
- **Improved** error handling.

6.6 Changes in Version 4.9

No functional changes.

6.7 Changes in Version 4.8

No functional changes.

6.8 Changes in Version 4.4

- **Removed** Support for legacy service (V1).

6.9 Changes in Version 4.2

- Complete reimplementation (V2):
  - HTTP/ReST based protocol instead of .NET Remoting
  - All engines are supported, not just ABBYY FineReader
  - The OCR server can now have a default configuration for the selected OCR engine.
  - File exchange via file system is no longer supported. The Key `FileExchange` in the `ocrserver.ini` file is ignored.
  - The OCR service main process is no longer able to act as worker process itself.
- The old version is still available for compatibility.
- Servers running the old version are still supported by the client plugin.

Upgrade checklist V1 -> V2

- Make sure that all clients use at least version 4.2.1.0 of the 3-Heights™ products.
- Conversion from the old `OCRServer.exe.config` file to the new `OcrSvr.xml` file has to be done manually.
- Update the service point URLs in all client configuration files (`ocrserver.ini`)

Note: The new URLs use `http://` instead of `tcp://`

7 Licensing, Copyright, and Contact

PDF Tools AG is a world leader in PDF (Portable Document Format) software, delivering reliable PDF products to international customers in all market segments.

PDF Tools AG provides server-based software products designed specifically for developers, integrators, consultants, customizing specialists and IT-departments. Thousands of companies worldwide use our products directly and hundreds of thousands of users benefit from the technology indirectly via a global network of OEM partners. The tools can be easily embedded into application programs and are available for a multitude of operating system platforms.

Licensing and Copyright The 3-Heights™ OCR Service is copyrighted. This user's manual is also copyright protected; It may be copied and given away provided that it remains unchanged including the copyright notice.

Contact

PDF Tools AG
Brown-Boveri-Strasse 5
8050 Zürich
Switzerland
http://www.pdf-tools.com
pdfsales@pdf-tools.com

A Status XML Format

A.1 Versions

The XML format has evolved over time and will continue to do so. Incompatible changes are denoted by increasing the format version. The current version is 1.

If applicable, the minimum format version of attributes or child elements is specified in parentheses.

The addition of new optional attributes or elements is not considered an incompatible change. Applications that consume the XML must therefore be prepared to ignore unknown attributes and elements.

A.2 Elements

A.2.1 `<ServiceStatus>` Element

The root element of the XML.

Attributes:

`version` (**required**) The version of the XML format.

Child elements:

`<Version>` (v1, **1**) `<DefaultEngine>` (v1, **1**) `<QueueSize>` (v1, **0..1**) `<RemainingPageCredits>` (v1, **0..1**) `<TotalPageCredits>` (v1, **0..1**)

A.2.2 `<Version>` Element

The version of the OCR service.

A.2.3 `<DefaultEngine>` Element

The name of the default engine of the OCR service.

A.2.4 `<QueueSize>` Element

The current number of OCR jobs in queue.

See Chapter 4.4.2. The absence of this element means that the service does not support backpressure.

A.2.5 `<RemainingPageCredits>` Element

The remaining page credits.

The absence of this element means that the remaining credits are unknown or that the service does not support this functionality.

Note: The value 2147483647 means unlimited credits.

A.2.6 <TotalPageCredits> Element

The total page credits.

The absence of this element means that the total number is unknown or not relevant because the of unlimited credits.

A.3 Example

```xml
<?xml version="1.0" encoding="utf-8"?>
<ServiceStatus version="1">
  <Version>5.0.1.7</Version>
  <DefaultEngine>abbyy12</DefaultEngine>
  <QueueSize>0</QueueSize>
  <RemainingPageCredits>8569</RemainingPageCredits>
  <TotalPageCredits>10000</TotalPageCredits>
</ServiceStatus>
```