Setup and Operations Manual
BLU DELTA Invoice Capture Service
(On Premise)

Blumatix Intelligence GmbH, Schwarzstraße 48, A-5020 Salzburg

Contents

Revision History

Version Number	Date	Author/Owner	Description of Change
1.0	2020-05-11	Bernhard Wimmer	Baseline Version
2.0	2020-08-05	Bernhard Wimmer	Minor updates for V1.11
3.0	2020-10-27	Bernhard Wimmer	Minor updates for V1.12
4.0	2020-12-08	Bernhard Wimmer	Updates for V1.13; Additional info sections

Version Number	Date	Author/Owner	Description of Change
5.0	2021-03-15	Bernhard Wimmer	Minor Updates for V1.14; Add some details
6.0	2021-07-09	Bernhard Wimmer	Updates for V1.15
7.0	2021-08-18	Bernhard Wimmer, Rudolf Dittrich	Updates for V1.15.1 à Typhon
8.0	2021-10-04	Bernhard Wimmer	Minor updates for V1.16
9.0	2021-12-20	Bernhard Wimmer	Minor updates for V1.17
10.0	2022-04-26	Bernhard Wimmer, Rudolf Dittrich	Updates for V1.18
11.0	2022-06-24	Bernhard Wimmer	Updates for V1.18.2
12.0	2022-08-22	Rudolf Dittrich	Added Triton Chapter
13.0	2022-11-14	Bernhard Wimmer	Updates for V1.18.12 (Architecture, Installation, Configuration, Typhon)

Purpose and Audience

This document contains information for setup and maintenance of the BLU DELTA Invoice Capture Service On-Premise solution.

Target Audience: Operation teams and people involved in the maintenance process.

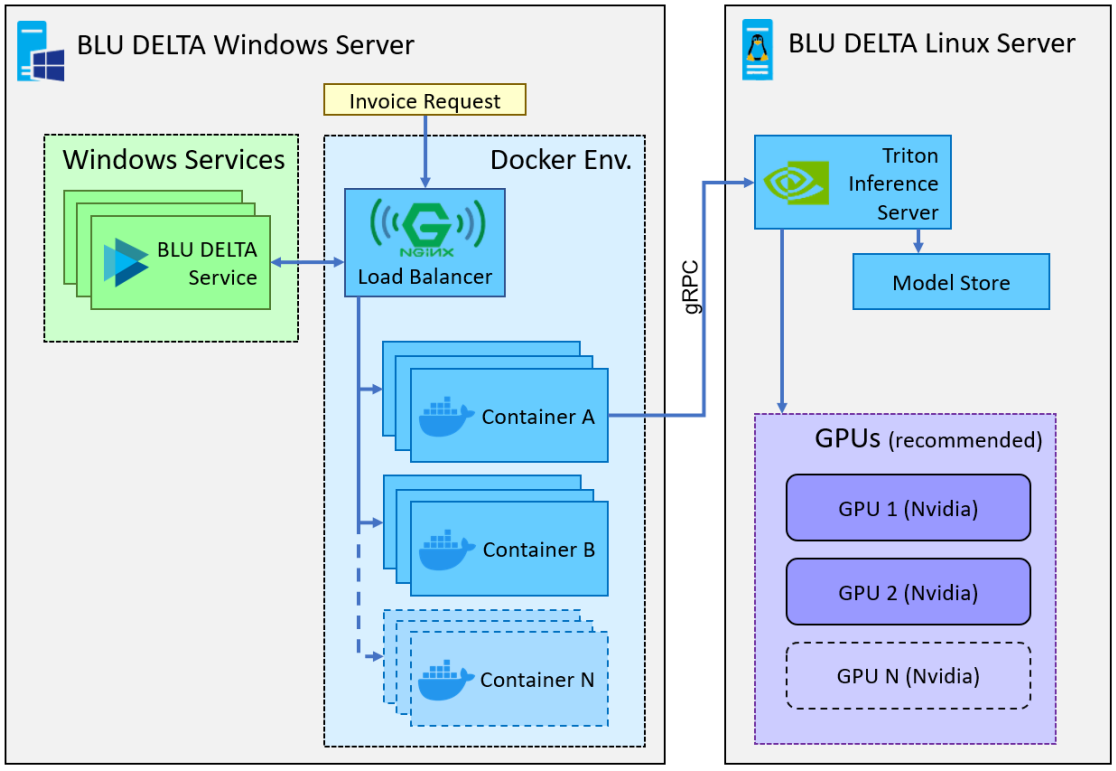
Introduction

BLU DELTA is an invoice and document recognition service that interprets data of documents and returns it in a structured format. The On-Premise solution is a specific form of the BLU DELTA product since it is running onsite.

Version

This version of the manual is valid for the BLU DELTA Invoice Capture Service V1.18.12 (and higher) in the On-Premise configuration setup.

BLU DELTA OnPremise Architecture



Bludelta's default setup consists of a Windows and a Linux machine:

- Windows Server (Win Server 2019): The BLU DELTA Service (Windows service) and all docker containers (even Typhon based containers) are installed on this machine. All docker images are started as Linux containers with Docker-Desktop. For installation details please refer to the following chapters: *BLU DELTA Installation* and *Manage BLU DELTA Windows Environment.*
  The main entry point of the BLU DELTA Environment is the internal Nginx load balancer which listens on port **8090** of the windows server for incoming documents. The BLU DELTA Service instances will use the upcoming port numbers from **8091 forward**. For docker container requests the load balancer is listening on the following ports:

Container	Port	Triton
PyFeatures	13000	
Contacts	13050	
VatGroups	13100	
DocType	13150	
LineItems	13200	
Typhon LineItems	13201	yes
NER (Named Entity Recognition)	13500	yes
NERDeliveryPeriod	13501	yes
Typhon Contact	13600	yes
Typhon Header Details	13700	Yes
QR Codes	13800	
Resource Service	13999	

- Linux machine: Triton should be installed on a linux machine - the installation of the Triton Inference Server is left to the customer. Communication takes place between the Docker containers and Triton using gRPC via port 8001. After installation, please set the correct IP-address/hostname (port must be 8001) in the bludelta-config.json file and re-start the BLU DELTA Environment again as it is described in *Manage BLU DELTA Windows Environment.*

BLU DELTA Installation

BLU DELTA Windows Setup Prerequisites

You should always install the BLU DELTA Invoice Capture Service with the same Windows user account which will be used to run the BLU DELTA environment (Docker Desktop) later on. This user account needs administrator rights on the local system.

Allow Web-Access during setup

BLU DELTA installation requires web-access to the following resources, so configure your firewall accordingly:

- To download the BLU DELTA Setup: **https://mlstoragerelease.file.core.windows.net**
- During setup / Blumatix Container Registry: **https://blumatixreleaseregistry.azurecr.io**
- During setup / Azure Container Storage CDN: ***.blob.core.windows.net**

Important info regarding Docker Desktop

Docker Desktop cannot run via the Windows SYSTEM account, so you need an admin user account instead! Therefore, during the installation of the BLU DELTA Invoice Capture Service, an autostart task for Docker Desktop is registered in the windows task scheduler which will automatically start the Docker Desktop Engine as background process when the system re-boots, as the windows user which performed the BLU DELTA installation. If you want to change the user account which runs Docker Desktop for the BLU DELTA environment, you do not have to perform a re-installation. Check the section "Switch BLU DELTA Docker User" further down in this document.

Important info regarding remote sessions

When you start a remote session with the user account which runs the BLU DELTA environment (Docker Desktop) and you end the session via "Log off", then all running processes of this user will be shut down. So, this would also end the Docker Desktop engine! To avoid that, always end a remote session with the BLU DELTA user with a "Disconnect" instead.

Important info regarding VMWare

The versions between 10.3.2 and 11.0.6 of the VMWare Tools are not compatible with the Windows Hyper-V which runs the Docker VM (for the Linux Containers).
You should definitely use VMWare Tools with version 11.0.7 or higher.

BLU DELTA Windows Setup package content

- Docker
- DotNetFX461
- vcredist_x64
- vcredist_x64 2010
- Blu Delta Invoice Capture ReleaseNotes 1.14.pdf
- BLU_DELTA.msi
- Bludelta OnPremise Operations Manual_v1.14.pdf
- setup.exe

Step 1) Install Docker Desktop

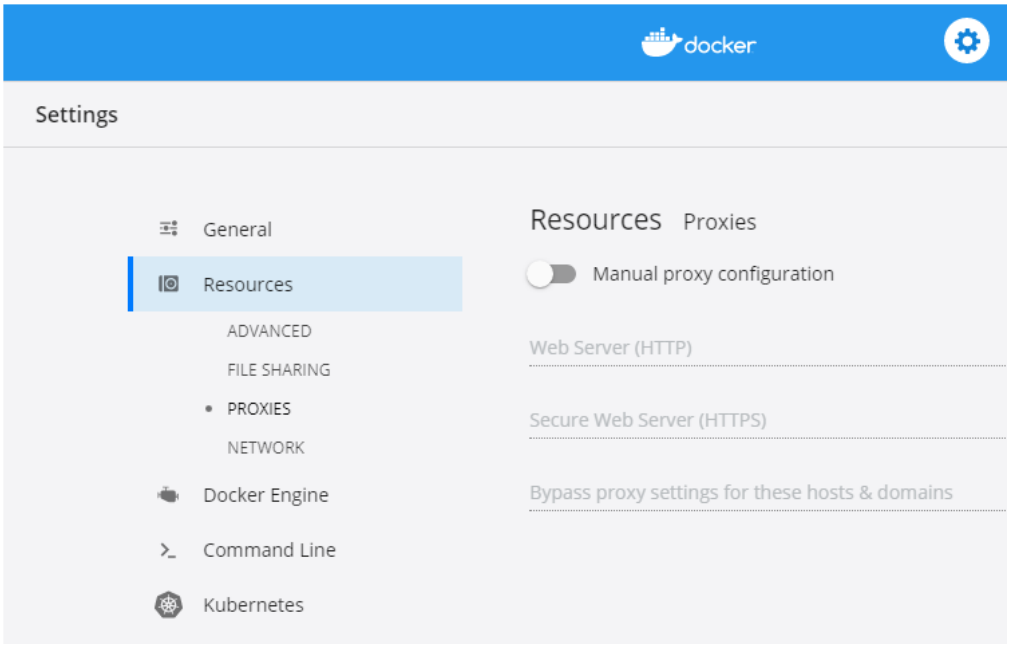
Execute the “Docker Desktop Installer.exe” which is placed inside the directory “Docker”.

Set: “Enable required Windows Features”
Do NOT set: “Use Windows containers instead of Linux containers”

After the successful installation, a restart may be required. In this case the docker installer dialog will show a “Restart and Close” button, otherwise the button will just say “Close”.

When this is done, start Docker Desktop and wait till it’s up (this may take a while). The initial start of docker will create a **settings file** which will be changed later by the BLU DELTA setup.

To install the “BLU DELTA Invoice Capture Service” docker will need to download container images from the Blumatix Container Registry. If you need to go out into the Cloud via a proxy, this is the time to configure one via: Docker Settings → Resources → Proxies → Enter as “[server]:[port]”



Afterwards “Quit Docker Desktop” via the icon in the tray.

Step 2) Install BLU DELTA Invoice Capture Service

Run the file **"setup.exe",** which is located in the setup root directory, **as Administrator**.

The setup will check some pre-requisites at the beginning. It will try to install them automatically if they are not yet existing on the system, you just must commit the dialogs. If older versions of these pre-requisites are installed, you will have to manually delete them first.

List of prerequisites:

- .NET Framework 4.6.1
- Microsoft Visual C++ 2010 x64 Redistributable – 10.0.40219
- Microsoft Visual C++ 2017 Redistributable (x64) – 14.16.27012

When the setup finished to install and unpack all required files, it will try to configure and bootstrap the whole BLU DELTA environment by running the script "bludelta-onpremise-bootstrapper.ps1". If something goes wrong during this process, you can always re-execute this script from an Admin-Powershell (check section "Manage BLU DELTA Windows Environment" for details).

The following two commands will be executed by the setup:

1.) **Apply default Docker settings and bootstrap the environment:**

```
.\bludelta-onpremise-bootstrapper.ps1 `
      -applyDefaultDockerConfig `
      -bluDeltaEnvironmentAction "start" `
      -performContainerRegistryLogin `
      -exitOnKeyPress
```

2.) **Add an autostart task for Docker Desktop to the task scheduler:**

```
.\bludelta-onpremise-bootstrapper.ps1 `
      -bluDeltaEnvironmentAction "addAutostart" `
      -exitOnKeyPress
```

Step 3) Manual actions after successful installation

Configure Triton-Url

IMPORTANT: After successful installation of Triton on a Linux server, you have to manually set the correct Triton-Url in the "bludelta-config.json" → check the sections *Prepare a Linux Server with Triton → GPU(s) recommended* and *Configuration & Scaling* for details. You have to restart the BLU DELTA environment afterwards.

```
{...
    "bludeltaTritonUrl": "[LinuxServer]:8001",
...}
```

IPv4 configuration

During the startup-phase of the BLU DELTA Environment the correct local IP address needs to be passed to the docker containers. By default, the IP is autodetected. You can find the detected value in the output of the script "bludelta-onpremise-bootstrapper.ps1".

```
Successfully found ip-address: 192.168.137.103
Successfully found ip-address to use for BLU DELTA and set environemnt variable 'bluDeltaIpv4' to value '192.168.137.103
' --> You can manually configure the IP address to use via 'bludelta-config.json'
```

If the auto-detection detects an inappropriate IP, you can manually specify the correct address via the file "bludelta-config.json" → check the section "BLU DELTA Environment Configuration & Scaling" for details.

Firewall

You may have to configure an inbound rule (TCP) for your local firewall for port 8090 for BLU DELTA to receive incoming requests.

Service warm-up

After a fresh start (or restart) of the BLU DELTA Environment, the first couple of requests will be processed significantly slower than all upcoming requests because some of the software-components need those requests to finalize their initialization process.

Prepare a Linux Server with Triton → GPU(s) recommended

NVIDIA Triton™ Inference Server, part of the NVIDIA AI platform, is an open-source inference serving software that helps standardize model deployment and execution and delivers fast and scalable AI in production.
NVIDIA Triton™ Inference Server must be installed on the Linux Machine(s) with CPUs und GPUs dedicated for AI Model execution.
NVIDIA Triton™ Inference Server can be downloaded from NVIDIA NGC Catalog (https://catalog.ngc.nvidia.com/orgs/nvidia/containers/tritonserver) - Blumatix has tested BLU DELTA with Triton docker image version 21.10-py3.
NVIDIA Triton™ Inference Server can be started in CPU-Mode only or in GPU-Mode (recommended!) if you have CUDA compliant GPUs. For further information please refer to the Triton-User-Manual (https://github.com/triton-inference-server/server/tree/main/docs).

Setup BLU DELTA Model Store

A Model Store must be provided to Triton. BLU DELTA models are provided as a compressed **bludelta-models.zip file**. The following folder structure should be available after unzipping the file. Please use this as Triton's Model-Store.

```
D:\TRITON-MODELS
├───contact_lines_classifier
│   │   config.pbtxt
│   │
│   └───1
│           model.onnx
│
├───contact_ner
│   │   config.pbtxt
│   │
│   └───1
```

```
│           model.onnx
│
├──delivery_period_bert_ner
│   │   config.pbtxt
│   │
│   └──1
│           model.onnx
│
├──payment_condition
│   │   config.pbtxt
│   │
│   └──1
│           model.onnx
│
│
├──typhon_contacts
│   │   config.pbtxt
│   │
│   └──1
│           model.onnx
│
├──typhon_header_details
│   │   config.pbtxt
│   │
│   ├──1
│   │       model.onnx
│   │
│   └──2
│           model.onnx
│
└──typhon_lineitems
    │   config.pbtxt
    │
    └──1
            model.onnx
```

How to update a model

Blumatix provides new, improved models in a download link provided to you in an Release email. The new model must then be copied into the correct folder where it replaces the old one. Triton then loads the new model. Depending on your Triton Model update strategy you might have to trigger the reload.

Manage BLU DELTA Windows Environment

bludelta-onpremise-bootstrapper.ps1

The BLU DELTA environment is manageable via a single Powershell script with the file name “bludelta-onpremise-bootstrapper.ps1”. This script is automatically executed during the setup process to bootstrap the BLU DELTA environment with the default configuration. You find it in the following location:

[Installation-Path]\CA\Environment

The default Installation-Path is: C:\Program Files\Blumatix\BLU DELTA Invoice Capture Service V1.18.12

➔ C:\Program Files\Blumatix\BLU DELTA Invoice Capture Service V1.18.12\CA\Environment\bludelta-onpremise-bootstrapper.ps1

The script needs to be executed from an **Administrator** Powershell.

Bootstrapper script parameters

applyDefaultDockerConfig

When you pass this switch-parameter to the script, then the basic Docker configuration will be re-applied.

The Docker configuration is stored in the following location: %APPDATA%\Docker\settings.json

This will set the following values:

- autoStart: false
- checkForUpdates: false
- sharedDrives: [BLU DELTA installation drive letter] → this is required!
- cpus: [Logical processor count / 2]
- memoryMiB: [Total physical memory / 2]

bluDeltaEnvironmentAction (pass one of these options: **"start", "stop", "setAutostart", "uninstall"**)

- **start:** Will start up (or restart) the whole BLU DELTA Environment. The following actions will be done:
  - Start Docker if it is not running
  - Stop BLU DELTA Environment, if it is already running
  - Perform a login to the Blumatix Container Registry (only if the *performContainerRegistryLogin* switch parameter was passed)
  - Start the BLU DELTA Environment and apply the parameters from the file "bludelta-config.json" → check section "BLU DELTA Environment Configuration" for details

- **stop:** Will shut down a running BLU DELTA Environment.

- **setAutostart**: Will add a task to the windows task scheduler which will automatically start Docker Desktop when the system is restarted. The task will run under the Windows account of the current/active user. Docker Desktop will not automatically start the linked containers until a user logs on to the system. So, this is required to automatically run the BLU DELTA Environment after a system reboot.
  The BLU DELTA setup will automatically add this task to the scheduler → also check section "BLU DELTA Installation - Instructions". If you re-execute "setAutostart", the old task will be deleted before the new one is added to the scheduler.

- **uninstall:** This action is called automatically during the uninstallation of the "BLU DELTA Invoice Capture Service". It will delete all BLU DELTA docker images from the system and it will remove the Docker Desktop autostart task from the scheduler. The BLU DELTA program files will NOT(!) be removed when you call this action manually. After calling "uninstall", you can still re-bootstrap the whole BLU DELTA environment with the "start" action.

performContainerRegistryLogin

When you pass this switch-parameter to the script together with ***-bluDeltaEnvironmentAction "start"***, then a log-in to the Blumatix Container Registry will be done before the docker images get downloaded and/or registered. A successful log-in is cached on the system by Docker. As the log-in to the container registry is part of the setup process, passing this parameter should not be required any more after a successful installation.

exitOnKeyPress

This switch parameter simply specifies if the script shall demand a keypress of the user before it exits. It is used by the setup to keep the Powershell windows open so that the user can capture the log-output.

Examples/Usage

To execute the sample-commands you need to open a Powershell window as administrator at the location of the script "bludelta-onpremise-bootstrapper.ps1", which is by default:

C:\Program Files\Blumatix\BLU DELTA Invoice Capture Service V1.18.12\CA\Environment

Start BLU DELTA Environment

```
.\bludelta-onpremise-bootstrapper.ps1 -bluDeltaEnvironmentAction "start"
```

Stop BLU DELTA Environment

```
.\bludelta-onpremise-bootstrapper.ps1 -bluDeltaEnvironmentAction "stop"
```

Register the Docker Desktop autostart task

```
.\bludelta-onpremise-bootstrapper.ps1 -bluDeltaEnvironmentAction "addAutostart"
```

Apply default Docker settings

```
.\bludelta-onpremise-bootstrapper.ps1 -applyDefaultDockerConfig
```

BLU DELTA Environment Configuration

Configuration & Scaling

For some specific components of the BLU DELTA Environment (check section “BLU DELTA Architecture”) the number of parallel instances can be scaled individually for each installation. This can increase the performance and will improve reliability, but it will also occupy more memory! Scaling values can be adjusted via the file “**bludelta-config.json**” which you find in the following location: **[Installation-Path]\CA\Environment**. The content of this file looks like this:


```
{
    "bluDeltaContainerRegistryUrl": "blumatixreleaseregistry.azurecr.io",
    "bluDeltaContainerToken": "onpremise-token-v1-18-12",
    "bluDeltaContainerTokenPwd": "[password]",
    "bluDeltaIpv4": "[autodetect]",
    "bludeltaTritonUrl": "localhost:8001",
    "bluDeltaServiceCount": 2,
    "bluDeltaPyCaptureSdkCount": 2,
    "bluDeltaContactCount": 2,
    "bluDeltaVatGroupCount": 2,
    "bluDeltaDocTypeCount": 2,
    "bluDeltaLineItemCount": 2,
    "bluDeltaNERCount": 2,
    "bluDeltaNERDeliveryPeriodCount": 2,
    "bluDeltaTyphonContactCount": 2,
    "bluDeltaTyphonHeaderDetailCount": 2,
    "bluDeltaTyphonLineItemCount": 2,
    "bluDeltaQRCodeServiceCount": 2,
    "bluDeltaResourceServiceCount": 2,
    "bluDeltaPyCaptureSdkImageTag": "v1.18",
    "bluDeltaContactImageTag": "v1.18",
    "bluDeltaVatGroupImageTag": "v1.18",
    "bluDeltaDocTypeImageTag": "v1.18.22278.1929",
    "bluDeltaLineItemImageTag": "v1.18",
    "bluDeltaNERImageTag": "v1.18.1",
    "bluDeltaNERDeliveryPeriodImageTag": "v1.18.1",
    "bluDeltaTyphonContactImageTag": "v1.18.1",
    "bluDeltaTyphonHeaderDetailImageTag": "1.1.63",
    "bluDeltaTyphonLineItemImageTag": "1.1.54",
    "bluDeltaQRCodeServiceImageTag": "1.0.88",
    "bluDeltaResourceServiceImageTag": "1.0.40",
    "bluDeltaEnvironmentOptions": ""
}
```

Key	Description
bluDeltaContainerRegistryUrl	URL from where the docker images get downloaded
bluDeltaContainerToken	Authentication token for the bluDeltaContainerRegistryUrl
bluDeltaContainerTokenPwd	Authentication password for the bluDeltaContainerRegistryUrl
bluDeltaIpv4	a) Use "[autodetect]" to let the bootstrapper script detect the IP, or b) directly specify the IP to use
bluDeltaTritonUrl	URL (name or IP address and port) of Triton inference service
bluDeltaServiceCount	Number of BLU DELTA Service instances to create
bluDeltaPyCaptureSdkCount	Number of PyFeatures container instances to create
bluDeltaContactCount	Number of Contacts container instances to create. *(The instances will only be created if bluDeltaEnvironmentOption **"typhon.contact" is NOT set**)*
bluDeltaVatGroupCount	Number of VatGroups container instances to create
bluDeltaDocTypeCount	Number of DocType container instances to create
bluDeltaLineItemCount	Number of LineItems container instances to create
bluDeltaNERCount	Number of NER container instances to create
bluDeltaNERDeliveryPeriodCount	Number of NERDeliveryPeriod container instances to create
bluDeltaTyphonContactCount	Number of Typhon Contacts container instances to create.
bluDeltaTyphonHeaderDetailCount	Number of Typhon HeaderDetails container instances to create
bluDeltaTyphonLineItemCount	Number of Typhon LineItems container instances to create
bluDeltaQRCodeServiceCount	Number of QR Code container instances to create
bluDeltaResourceServiceCount	Number of ResourceService container instances to create.
bluDeltaPyCaptureSdkImageTag	PyFeatures container image tag to use
bluDeltaContactImageTag	Contacts container image tag to use
bluDeltaVatGroupImageTag	VatGroups container image tag to use
bluDeltaDocTypeImageTag	DocType container image tag to use
bluDeltaLineItemImageTag	LineItems container image tag to use
bluDeltaNERImageTag	NER container image tag to use
bluDeltaNERDeliveryPeriodImageTag	NERDeliveryPeriod container image tag to use
bluDeltaTyphonContactImageTag	Typhon Contacts container image tag to use
bluDeltaTyphonHeaderDetailImageTag	Typhon HeaderDetails container image tag to use
bluDeltaTyphonLineItemImageTag	Typhon LineItems container image tag to use
bluDeltaQRCodeServiceImageTag	QR Code container image tag to use
bluDeltaResourceServiceImageTag	ResourceService container image tag to use
bluDeltaEnvironmentOptions	[unused / Reserved for future-use]

Attention: If you want to change any values here, you first must stop the BLU DELTA Environment via the script "bludelta-onpremise-bootstrapper.ps1" (see section "Manage BLU DELTA Windows Environment" for details). Use the same script to re-start the environment when you have saved all changes.

Switch BLU DELTA Docker User

If you want to change the user account which runs Docker Desktop for the BLU DELTA environment, you do not have to perform a re-installation.

1.) Log on as the Current BLU DELTA Docker Desktop User
2.) Stop BLU DELTA environment
    From an Admin Powershell execute these 2 commands:
    cd "C:\Program Files\Blumatix\BLU DELTA Invoice Capture Service V1.18.12\CA\Environment"

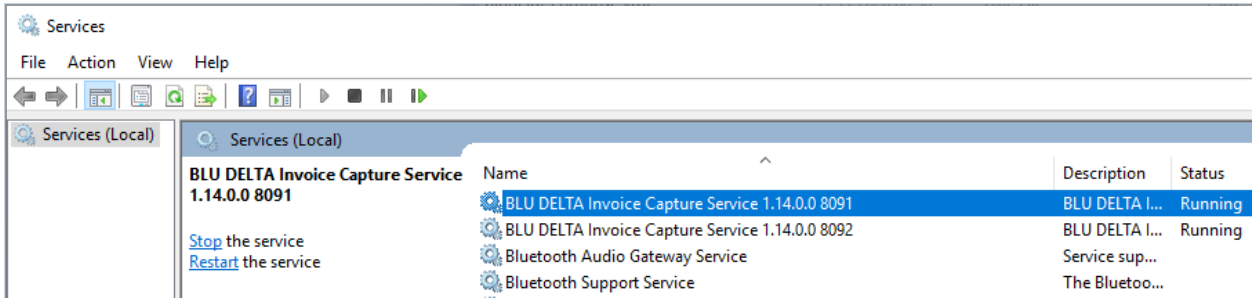
.\bludelta-onpremise-bootstrapper.ps1 -bluDeltaEnvironmentAction "stop"

3.) Quit Docker Desktop, or kill the process via the TaskMgr (if it is running as background process)
4.) Copy “settings.json” from “%APPDATA%\Docker” to a temporary directory
    For example → D:\Temp
5.) Log on as the new BLU DELTA Docker Desktop Windows User
    Must also be an Admin user
6.) Copy “settings.json” from the temp location to “%APPDATA%\Docker”
7.) Register the new Autostart action
    From an Admin Powershell:
    cd “C:\Program Files\Blumatix\BLU DELTA Invoice Capture Service V1.18.12\CA\Environment”
    .\bludelta-onpremise-bootstrapper.ps1 -bluDeltaEnvironmentAction "addAutostart"
8.) Start BLU DELTA environment
    From an Admin Powershell:
    cd “C:\Program Files\Blumatix\BLU DELTA Invoice Capture Service V1.18.12\CA\Environment”
    .\bludelta-onpremise-bootstrapper.ps1 -bluDeltaEnvironmentAction "start"

BLU DELTA Service Configuration

Configuration Procedure

Before you can edit configuration files or install plug-ins, all BLU DELTA Service instances must be stopped and they need to be restarted afterwards.



Additionally, before you change configuration files, make sure to create a backup of the original files. Do not make any changes except the ones described!

OCR (Optical Character Recognition) Worker Count

To define the number of OCR Worker components for each BLU DELTA Service instance, edit the file *„Blumatix.Capture.Webservice.Client.Selfhosted.exe.config“* which can be found in the installation directory. Add a new appSetting with key “OcrWorkerCount” and give it a value. By default (without this key) 2 OCR Workers are created per BLU DELTA Service.

```
<appSettings>
   ...
   <add key="OcrWorkerCount" value="4" />
</appSettings>
```

OCR (Optical Character Recognition) Timeout Seconds

To change the maximum time (in seconds) for the BLU DELTA Service to extract the document text (=OCR processing), edit the file *„Blumatix.Capture.Webservice.Client.Selfhosted.exe.config“* which can be found in the installation directory. Then you can change the “value” of the key “OcrTimeoutSeconds”. If you remove this key, the BLU DELTA Servie will NOT(!) abort document extraction after a specific time period, but the local nginx loadbalancer is configured to cancel requests which run longer than 600 seconds anyway. (for more infos please contact bludelta support)

```
<appSettings>
   ...
   <add key="OcrTimeoutSeconds" value="595" />
</appSettings>
```

OCR (Optical Character Recognition) Language setting

For the OCR, you should only activate those languages which are relevant for your specific use case. To configure the active OCR languages, edit the file *„Ocr.config“* which can be found in the installation directory.

```
<?xml version="1.0"?>
<ocrConfig Multithreaded="true" ReturnPageBitmaps="false" MaxPages="10" Languages="de,en" FilterPlus="" SingleZone="false" />
```

The API request also provides a “Languages” parameter to pass and override the default language configuration for a specific document. Specifying the correct document language (or a small potential set of languages) will reduce OCR-failures and can improve the quality of the predictions. *(check API metadata page → DetectInvoiceRequest)*

Supported Languages are:

de	German	**pl**	Polish	**cs**	Czech	**nb**	Norwegian	**pt**	Portuguese
en	English	**it**	Italian	**sk**	Slovakian	**hr**	Croatian	**sv**	Swedish
es	Spanish	**fr**	French	**nl**	Dutch	**tr**	Turkish	**sl**	Slovanian
da	Danish								

Result-Pdf configuration

To adapt the visual appearance of Result-Pdfs, edit the file *„ PdfBuilder.config “* which can be found in the installation directory.

```
<?xml version="1.0"?>
<pdfBuilderConfig
  isEnabled="true"                "true" | "false" --> enable/disable Result-Pdf
                                  feature
  markerColorA="40"               "0...255" --> Alpha channel value of prediction-marks
  markerColorR="18"               "0...255" --> Red color channel value of
                                  prediction-marks
  markerColorG="204"              "0...255" --> Green color channel value of
                                  prediction-marks
  markerColorB="56"               "0...255" --> Blue color channel value of
                                  prediction-marks
  markerColorLowScoreA="128"      "0...255" --> Alpha channel  value of prediction-marks
                                  with low score
```

```
  markerColorLowScoreR="255"    "0...255" --> Red  color  channel value of prediction-
                                marks with low score
  markerColorLowScoreG="152"    "0...255" --> Green color channel value of prediction-
                                marks with low score
  markerColorLowScoreB="0"      "0...255" --> Blue  color  channel value of prediction-
                                marks with low score
  markerBorderWidth="1"         border width of prediction marks in pixel
  markerDrawMode="FillArea"     "FillArea" | "OuterBorder" --> Draw mode for predic-
                                tion marks → "OuterBorder" will only draw a border-
                                rectangle (no fill color)
  namedDestinationOffsetY="10"  Vertical offset (in relation to
                                prediction-mark) for locations of
                                NamedDestinations in Result-Pdf
  isFocusLayerEnabled="false"> "true" | "false" --> enable/disable highlighting of
                                prediction  marks  when  bookmark  or  NamedDestination
                                is selected (enable => NamedDestinations are created,
                                disable => only Bookmarks are created)
</pdfBuilderConfig>
```

Feature configuration (dynamic config)

Some specific features of the BLU DELTA Invoice Capture Service can be configured even at runtime, so you do not have to stop the windows service instances before editing. But ensure to use an editor which does not lock the opened file (use “notepad.exe” for example).
The following features can be activated/deactivated via the file *„dynamic_config.json“*, which can be found in the installation directory.

Attention: Some values of the dynamic_config.json must not be changed without an explicit request! Never change any values of properties which are not explicitly documented in this section here on your own!

QR-Code detection (disabled by default)

```
…
"QRCode": {
    "QRCodeService": {
      "Enabled": true,
      "CustomerNames": []
    }
  }
```

The QRCode service can search for QR-Codes on the input-documents. It can deliver specific details for SwissQR and CodigoQR. For un-interpretable QR codes the raw (encoded) string will be delivered. For further details please have a look at http://localhost:8090/metadata on any running BLU DELTA Server instance.

Attention: With the current version, QR code detection is rather CPU intensive (on the Windows Server) and therefor disabled by default!

Prosa detection (disabled by default)

```
…
"DocumentPreparation": {
    "JunkDetection": {
      "Enabled": true,
      "CustomerNames": []
    }
  },
```

If Prosa-Detection is enabled, the BLU DELTA Service will try to find and ignore pages which have a very high density of prosa-text (like terms and conditions) before the OCR (Optical Character Recognition) starts.

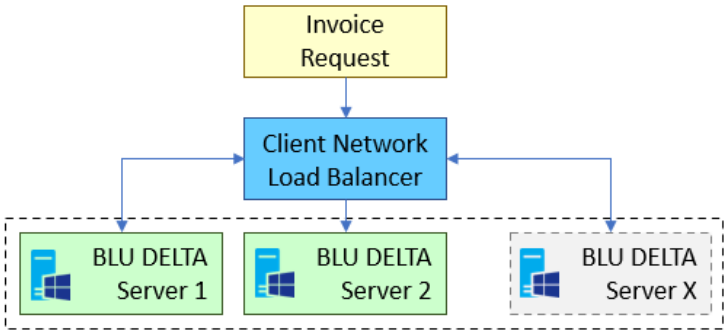
Attention: With the current version, Prosa-Detection is rather CPU intensive (on the Windows Server) and therefor disabled by default!

If you have a lot of documents with prosa texts, then it will be beneficial to turn this feature on.

Plug-in installation

If you must install plug-ins for the BLU DELTA Invoice Capture Service, place those files in the "Plugins"-folder of the BLU DELTA Installation directory.

Load Balancing (recommended) – Redundancy & Performance



You can increase the invoice processing throughput and improve the reliability of the whole system by the usage of multiple BLU DELTA Servers. This works best in combination with a load balancer.

Our system is tested with HAProxy load balancer (http://www.haproxy.org/), but any other product is possible as well.

If you need further information about setting up a distributed system with multiple instances of the BLU DELTA Invoice Capture Service in combination with a load balancer, please feel free to contact our support.

Typhon Engine for more accurate predictions

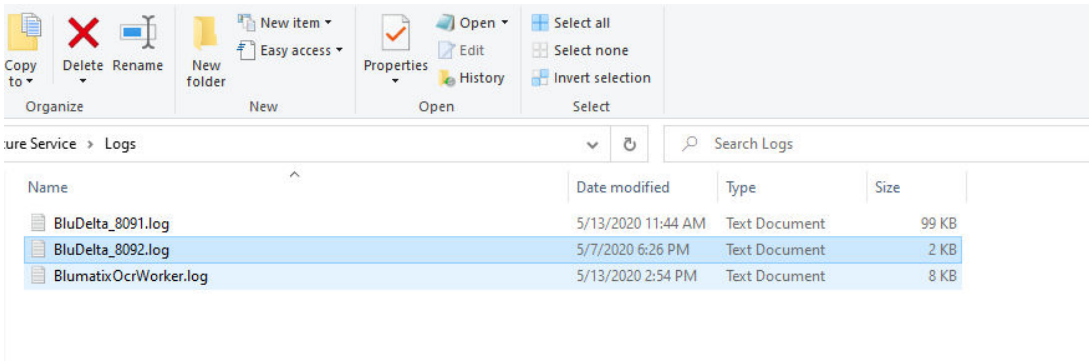
In V1.18.12, (by Default) the BLU DELTA Invoice Capture Service uses some Models/Containers based on the **"Typhon"-Engine,** which delivers more accurate predictions.

Typhon needs a lot of processing power, so it is highly recommended to install Triton on a Linux Server with compatible Graphic Card(s) → Check section *"Prepare a Linux Server with Triton"*

Attention: The Typhon-Engine does not yet support all languages. Additional languages will be supported with future updates. If the detected document language is not support by a typhon container/model, the corresponding predictions will be made by a fallback-model (with other technologies).

Logging

One log file per BLU DELTA Service instance is created. The log file name consists of the prefix "BluDelta_" + the port number on which this instance is listening for requests, e.g.: "*BluDelta_8091.log*". In addition to the BLUDELTA log files, a separate log file "*BlumatixOcrWorker.log*" is also created for all OcrWorkers. The following examples shows two BLU DELTA logfiles for two service instances listening on port 8091 and 8092, respectively as well as one logfile for all BlumatixOcrWorkers.



The image above shows log files for two BLU DELTA Service instances and one for all running BlumatixOcrWorkers

Logpath

The default log path for all BLU DELTA log files is:
".\Windows\System32\config\systemprofile\Documents\Capture Service\Logs".

Logging Settings

The log path as well as other log specific settings can be changed in the config file: ".\Program Files\Blumatix\BLU DELTA Invoice Capture Service VERSION\Blumatix.Capture.Webservice.Client.Selfhosted.exe.config" in the *log4net* section. For a detailed description of all log settings and possible adaptations please refer to log4net.

Log Levels

The following levels are defined in order of increasing priority:

- DEBUG
- INFO
- WARN
- ERROR

The default level is set to WARN. The log level can be changed at runtime. Open the Blumatix.Capture.Webservice.Client.Selfhosted.exe.config and change the threshold value in the log4net section. The log level is set to *INFO* in the example configuration below.

```
<log4net>
    <appender name="Console" type="log4net.Appender.RollingFileAppender">
        <threshold value="INFO" />
        ...
    </appender>
  ...
</log4net>
```

Monitoring Recommendations

Passive Monitoring

Log Files Monitoring

BLU DELTA uses Log4Net as logging framework. There are several monitoring tools available. (e.g. Log4View) for monitoring Log4Net log files. We recommend monitoring only “ERROR” entries in the logfile.

Process Monitoring

The following processes shall be monitored:

- Blumatix.Capture.Webservice.Client.Selfhosted
- OcrWorker.exe

Docker-based Services Monitoring

The BLU DELTA service consists of the Blumatix.Capture.Webservice.Client.Selfhosted and OcrWorker processes as well as of several docker containers. Depending on your configuration you should see the following containers on a single machine:

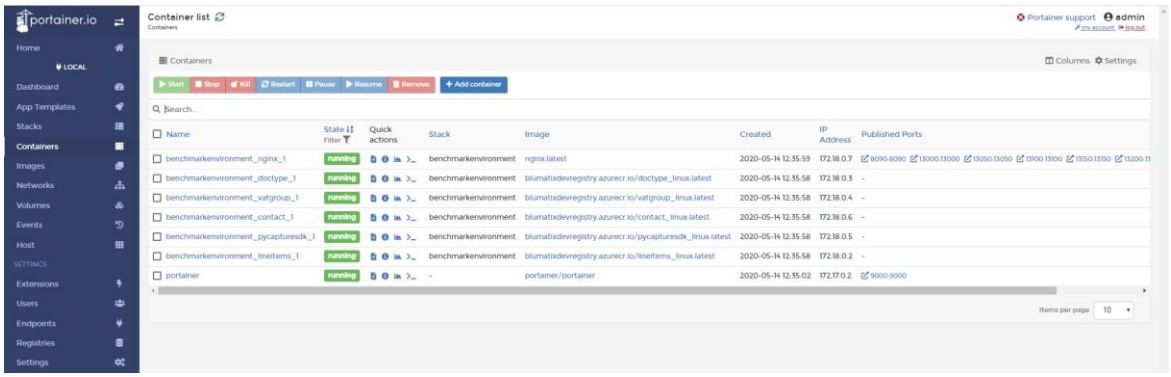
```
PS> docker ps --format "table {{.Image}}\t{{.Names}}"
IMAGE                                                                NAMES
blumatixreleaseregistry.azurecr.io/nginx:1.17                        environment_nginx_1
blumatixreleaseregistry.azurecr.io/invoice_ner_linux:v1.18.1         environment_ner_1
blumatixreleaseregistry.azurecr.io/typhon_contact_cpu:v1.18.1        environment_typhon_contact_1
blumatixreleaseregistry.azurecr.io/vatgroup_linux:v1.18              environment_vatgroup_1
blumatixreleaseregistry.azurecr.io/pycapturesdk_linux:v1.18          environment_pycapturesdk_1
blumatixreleaseregistry.azurecr.io/contact_linux:v1.18               environment_contact_1
blumatixreleaseregistry.azurecr.io/delivery_period_ner_cpu:v1.18.1   environment_nerdeliveryperiod_1
blumatixreleaseregistry.azurecr.io/doctype_linux:v1.18.22278.1929    environment_doctype_1
blumatixreleaseregistry.azurecr.io/typhon_header_details_cpu:1.1.63  environment_typhon_headerdetails_1
blumatixreleaseregistry.azurecr.io/typhon_line_item_cpu:1.1.54       environment_typhon_lineitems_1
blumatixreleaseregistry.azurecr.io/resource_service:1.0.40           environment_resourceservice_1
blumatixreleaseregistry.azurecr.io/lineitems_linux:v1.18             environment_lineitems_1
blumatixreleaseregistry.azurecr.io/qrcode_service:1.0.88             environment_qrcodeservice_1
```

The container names may slightly differ on your machine and the number of each container as this depends on your configuration.

Blumatix does not provide special tools for monitoring docker containers, but there are several commercial and free tools available. To name just of few:

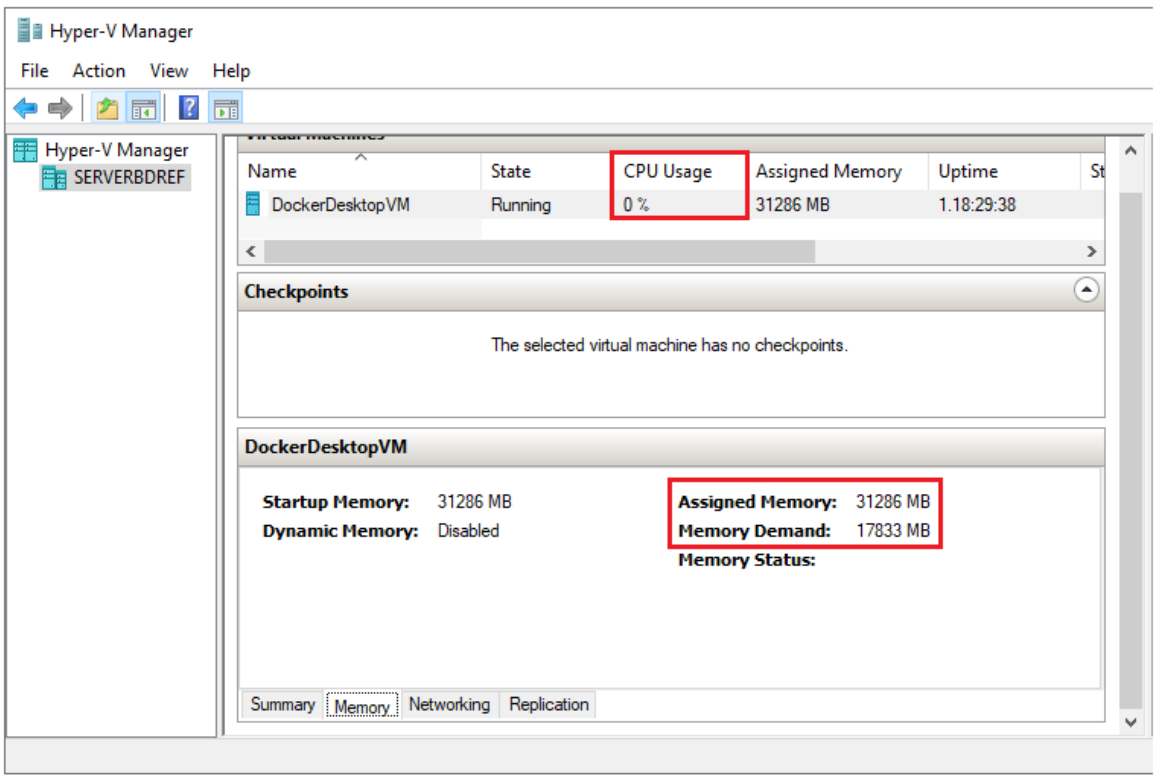
- https://www.portainer.io/
- https://kitematic.com/
- https://dockstation.io/

This is a screen shot of the Portainer tool. It comes with a new web-based ui.



Hyper-V Manager

The Hyper-V Manager is a tool to manage virtual machines. It is part of all recent Windows & Windows Server editions. Besides other valuable stuff, it can show the CPU consumption of the Docker VM (Linux containers) as well as the available & demanded memory.



Active Monitoring

We suggest two methods for monitoring the overall health state of the Bludelta system.

Test Invoice

One way to check the system's health state is to send a simple Test-Invoice periodically to your service, e.g. every 30 sec or every minute.

Http Ping

Use the integrated http ping functionality to monitor the BLU DELTA Services. Periodically send a GET request to http://hostname:service-port/ping. The response should contain a "pong" as shown in this example:

```
bash~$ curl http://localhost:8090/ping
pong
```

Ping the following services:

- Bludelta Service: http://hostname:8090/ping: Ping each Bludelta Service separately in case that more than one instance is running.
- PyInvoiceCapture Service: http://hostname:13000/ping
- Contact Service: http://hostname:13050/ping
- VatGroup Service: http://hostname:13100/ping
- DocType Service: http://hostname:13150/ping
- LineItems Service: http://hostname:13200/ping
- Typhon LineItems Service: http://hostname:13201/ping
- NER Service: http://hostname:13500/ping
- NERDeliveryPeriod Service: http://hostname:13501/ping
- Typhon Contact Service: http://hostname:13600/ping
- Typhon HeaderDetails Service: http://hostname:13700/ping
- QR Code Service: http://hostname:13800/ping
- Resource Service: http://hostname:13999/ping

NOTE: Because the Docker-based services run in a user-defined bridged network, they can only be accessed via the integrated nginx load-balancer by default. This means that you cannot reach all instances of a service at once using the http ping. Nginx uses "Round Robin" (this is the default load-balancing method) to evenly distribute requests across the instances of a service. All instances are addressed one after the other only after repeated pinging.

Error Scheme

The caller of the BLU DELTA REST service is responsible for a proper http error handling. The possible http return codes are described in the following table:

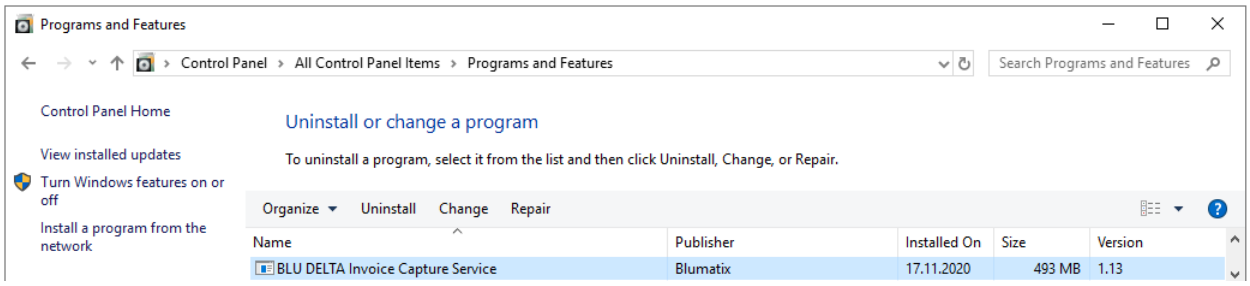
Http Code	Description
200 OK	Service call successfully finished
400 Bad Request	Invoice format error
401 Unauthorized	Unauthorized. Invalid ApiKey (or invalid ApiIdentifier)
403 Forbidden	Customer is not allowed to make any predictions
500 Internal Server Error	Oops, something broke

For further details please have a look at http://localhost:8090/metadata on any BLU DELTA Server instance.

Upgrade / Rollback

Uninstall Blu Delta Service

To uninstall the BLU DELTA Invoice Capture Service, simply remove the application from the list of installed programs:

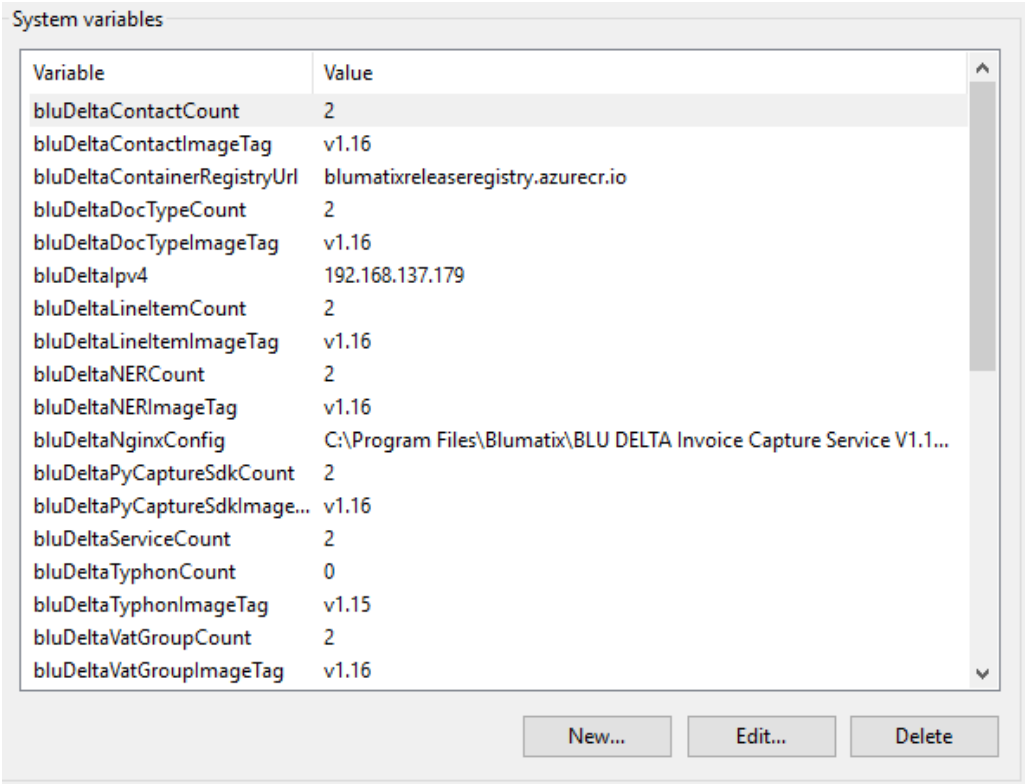


If you did not stop the BLU DELTA Environment before pressing “Uninstall”, a dialog will show up and ask if you want to stop all running BLU DELTA Service instances. Commit the dialog and continue in that case. The uninstaller will remove all BLU DELTA program files together with all the Docker images from your System.

Attention:

If you want to keep any configuration files (which you manually changed before) for future installations, please ensure to create backups of these files before uninstalling.

The uninstaller of the BLU DELTA Invoice Capture Service V1.18.2 will not delete system environment variables created for BLU DELTA. For an absolutely clean uninstallation you have to delete them manually:



(Example image from v1.16)

Trouble Shooting

How to restart services

To restart the whole BLU DELTA environment, you can use the bootstrapper Powershell script which is explained in detail in the section “Manage BLU DELTA Windows Environment”.

Handle known issues

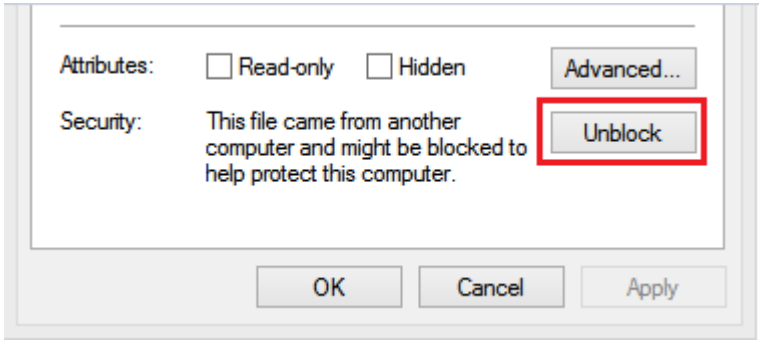
This document includes a chapter of known issues.

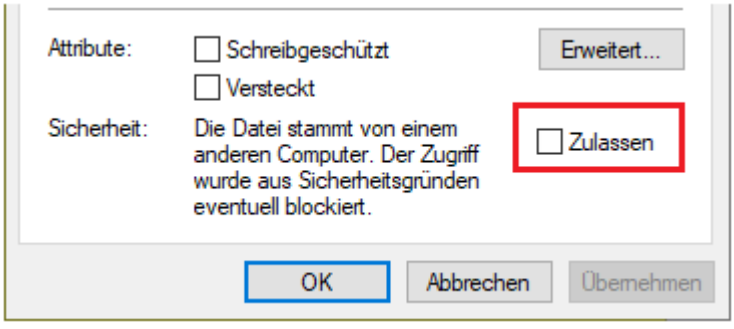
Provide Logfiles

If any errors come up, ensure to create backups of the log files to be able to provide detailed information for further investigations by the support team of Blumatix.

Potential Fix if service does not start

If you have installed custom plugins (DLL plugins), then you should check that these files are not blocked by the system. (Right-click file in windows explorer → Properties → Press “Unblock” button/checkbox (the button or checkbox will only be visible if the file is blocked)





Known Issues

Release Note – Known Issues

ResultPDF file size gets bigger for some documents (for some stays the same, for some gets smaller).