

# **Setup and Operations Manual**

## **BLU DELTA Invoice Capture Service**

### **(On Premise)**



## Contents

<b>CONTENTS .....</b>	<b>2</b>
<b>REVISION HISTORY .....</b>	<b>4</b>
<b>PURPOSE AND AUDIENCE .....</b>	<b>4</b>
<b>INTRODUCTION .....</b>	<b>4</b>
<b>VERSION .....</b>	<b>5</b>
<b>BLU DELTA ONPREMISE ARCHITECTURE .....</b>	<b>5</b>
<b>BLU DELTA WINDOWS SERVER INSTALLATION .....</b>	<b>6</b>
GENERAL INFOS.....	6
INSTALLATION .....	6
CONFIGURE IP OF BLU DELTA LINUX SERVER .....	6
FIREWALL.....	7
SERVICE WARM-UP .....	7
<b>BLU DELTA LINUX SERVER INSTALLATION .....</b>	<b>7</b>
INSTALL DOCKER.....	7
INSTALL TRITON 🕒 GPU(s) HIGHLY RECOMMENDED! .....	8
SETUP BLU DELTA MODEL STORE.....	8
HOW TO UPDATE A MODEL .....	9
INSTALL BLU DELTA DOCKER CONTAINERS (VIA DOCKER COMPOSE).....	9
. /BLUDELTΑ-ONPREMISE-BOOTSTRAPPER .SH -I .....	9
INTEGRATE TRITON INTO BLU DELTA DOCKER NETWORK .....	9
<b>START/STOP BLU DELTA SERVICE (WINDOWS SERVER) .....</b>	<b>9</b>
SCRIPT LOCATION (BLUDELTΑ-ONPREMISE-BOOTSTRAPPER.PS1) .....	9
SCRIPT PARAMETER: BLUDELTΑENVIRONMENTACTION.....	10
EXAMPLES/USAGE .....	10
<b>BLU DELTA WINDOWS SERVICE CONFIGURATION .....</b>	<b>11</b>
CONFIGURATION PROCEDURE .....	11
ENVIRONMENT OPTIONS & SCALING.....	11
OCR (OPTICAL CHARACTER RECOGNITION) WORKER COUNT.....	12
OCR (OPTICAL CHARACTER RECOGNITION) TIMEOUT SECONDS.....	12
OCR (OPTICAL CHARACTER RECOGNITION) LANGUAGE SETTING.....	12

RESULT-PDF CONFIGURATION .....	12
FEATURE CONFIGURATION (DYNAMIC CONFIG).....	13
<i>QR-Code detection (disabled by default)</i> .....	13
<i>Prosa detection (disabled by default)</i> .....	14
<i>Typhon header details (enabled by default)</i> .....	14
<i>Typhon contacts (enabled by default)</i> .....	15
<i>Typhon line items (enabled by default)</i> .....	15
<i>Response strategy (ForceReturnBestMatches = disabled by default)</i> .....	16
<i>RecommendedThresholds (enabled by default)</i> .....	16
PLUG-IN INSTALLATION .....	17
<b>LOAD BALANCING (RECOMMENDED) – REDUNDANCY &amp; PERFORMANCE .....</b>	<b>18</b>
<b>TYPHON ENGINE FOR MORE ACCURATE PREDICTIONS .....</b>	<b>18</b>
<b>LOGGING .....</b>	<b>18</b>
LOGPATH .....	19
LOGGING SETTINGS.....	19
LOG LEVELS .....	19
<b>MONITORING RECOMMENDATIONS .....</b>	<b>20</b>
PASSIVE MONITORING.....	20
ACTIVE MONITORING .....	21
ERROR SCHEME.....	22
WORKLOAD MONITORING.....	22
<b>UPGRADE / ROLLBACK.....</b>	<b>23</b>
UNINSTALL BLU DELTA SERVICE (WINDOWS) .....	23
<b>TROUBLE SHOOTING .....</b>	<b>23</b>
HOW TO RESTART SERVICES .....	23
HANDLE KNOWN ISSUES .....	23
PROVIDE LOGFILES .....	23
POTENTIAL FIX IF SERVICE DOES NOT START .....	23
<b>KNOWN ISSUES .....</b>	<b>24</b>
RELEASE NOTE – KNOWN ISSUES .....	24

## 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
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, Rudolf Dittrich	Updates for V1.18.12 (Architecture, Installation, Configuration, Triton,...)
14.0	2023-01-13	Bernhard Wimmer	Updates for V1.18.55 - Add dynamic config sections
15.0	2023-02-23	Bernhard Wimmer	Updates for V1.18.55 - Add post processing container
16.0	2023-03-09	Bernhard Wimmer	Updates for V1.18.55 - Add workload monitoring

## 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.

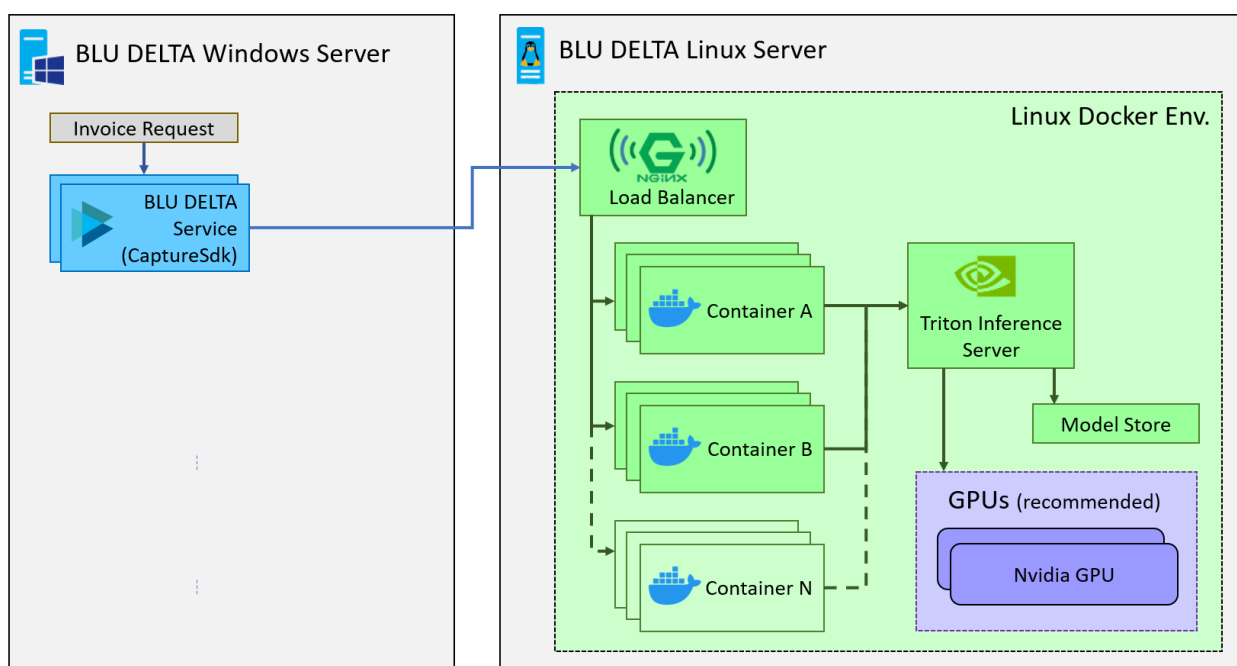
## Introduction

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



The BLU DELTA default setup consists of a Windows and a Linux server:

- Windows server (Win Server 2019): The **BLU DELTA Service** (Windows service) is listening on **port 8090** of this machine for incoming documents (by default). You can run multiple service instances in parallel (using subsequent port numbers), but in that case you are self-responsible for load-balancing. For installation details please refer to the following chapters: *BLU DELTA Windows Server Installation* and *Start/stop BLU DELTA Service (Windows Server)*
- Linux server: All docker containers and the Triton Inference Server are installed on this machine (by default). The installation of Docker and Triton are left to the customer. Communication takes place between the Docker containers and **Triton** using **gRPC via port 8001**. For docker container requests the Nginx 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	
PostProcessing	13880	
Resource Service	13999	

## BLU DELTA Windows Server Installation

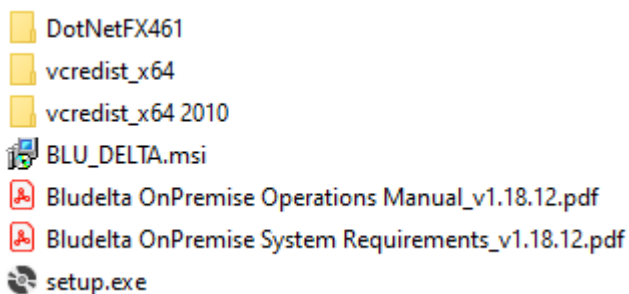
### General Infos

#### 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](https://*.blob.core.windows.net)

#### BLU DELTA Windows Setup package content



### Installation

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 have to 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

The setup does not automatically start the BLU DELTA Service after installation because the IP of the BLU DELTA Linux Server must be configured first.

### Configure IP of BLU DELTA Linux Server

After successful installation of the BLU DELTA Service, you have to manually set the correct IP (or Hostname) in the environment configuration file (bludelta-config.json) → check the sections *BLU DELTA Linux Server Installation* and *Environment options & Scaling* for more details.

```
{...  
  "bluDeltaLinuxServerIP": "192.168.100.2",    // Exemplary IP  
...}
```

Afterwards, you can start the BLU DELTA Service → Check section *Start/stop BLU DELTA Service (Windows Server)*

## Firewall

You may have to configure an inbound rule (TCP) for your local firewall for the port(s) of the BLU DELTA Service to receive incoming requests (by default: 8090).

## 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.

# BLU DELTA Linux Server Installation

---

The Bludelta OnPremise package consists of the following files:

- **bludelta-compose.yaml**: A docker-compose file which contains all bludelta services as well a nginx as a loadbalancer and optionally a Triton services which is commented out.
- **bludelta-onpremise-bootstrapper.sh**: A bash script which shall be used for installing and running the Bludelta containers.
- **.env**: Contains environment variables which can be modified by the customer.
- **nginx.conf**: A configuration file for nginx.
- **README.pdf**: Contains detailed install instructions

## Install Docker

The Bludelta containers will only run if Docker is also installed on the system. Therefore, Docker must be installed on your Linux system. In addition to Docker, Docker Compose must be installed as well.

The Bludelta containers were tested with the following Docker and Docker Compose versions:

- Docker: 20.10.7
- Docker Compose: 1.29.2

Please refer to <https://www.docker.com> for further information on Docker and Docker Compose.

## Install Triton à GPU(s) highly 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 (highly 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.





```
├── typhon_lineitems
│   └── config.pbtxt
└── 1
    └── model.onnx
```

## How to update a model

Blumatix provides new, improved models via download links provided by Release e-mails. 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.

## Install BLU DELTA Docker Containers (via docker compose)

Please install all bludelta docker images with the **bludelta-onpremise-bootstrapper.sh**

```
./bludelta-onpremise-bootstrapper.sh -i
```

After having installed all docker image you can start the Bludelta environment with:

```
./bludelta-onpremise-bootstrapper.sh -u
```

Shut down the bludelta docker containers with:

```
./bludelta-onpremise-bootstrapper.sh -d
```

## Integrate Triton into BLU DELTA Docker network

We use Triton as an Inference Server and is a mandatory requirement for the Bludelta service. You can either use your own Triton server, in which case you need to change the hostname in the .env file, or use the server we provide as a Docker image. The image is located in a commented-out section in the bludelta-compose.yaml file.

**IMPORTANT:** If your own Triton server is running as a docker container on the same host where our containers are running, then you need to make sure that you integrate both into the same docker network, otherwise the communication will not work. If you have enabled Triton in the bludelta-compose.yaml file, then the communication will work out-of-the-box.

## Start/stop BLU DELTA Service (Windows Server)

### Script location (bludelta-onpremise-bootstrapper.ps1)

The BLU DELTA environment is manageable via a single Powershell script with the file name "bludelta-onpremise-bootstrapper.ps1". 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

**Attention:** The script needs to be executed from an **Administrator** Powershell!

## Script parameter: bluDeltaEnvironmentAction

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

**start:** Will start up (or restart) the BLU DELTA Service. The following actions will be done:

- Stop BLU DELTA Service(s), if already running.
- Register BLU DELTA Service(s) as Windows service, if not already done.
- Apply the configured BLU DELTA Linux Server IP to the app configuration file of the BLU DELTA Service  
→ values of multiple keys which have Urls will be updated.
- Start the BLU DELTA Service(s) on the configured port(s) → check section *Environment options & Scaling* for more details

**stop:** Will shut down all running BLU DELTA Services. The following actions will be done:

- Stop BLU DELTA Service(s)
- Un-register BLU DELTA Windows Service(s)

## 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"
```

## BLU DELTA Windows 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**. Check section *Start/stop BLU DELTA Service (Windows Server)* for details.

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!

### Environment options & Scaling

The environment options 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:

```
{
  "bluDeltaLinuxServerIP": "localhost",
  "bluDeltaPortRangeStart": 8090,
  "bluDeltaServiceCount": 1
}
```

#### bluDeltaLinuxServerIP

This value specifies the IP (or hostname) of the Linux server which runs the BLU DELTA Docker Containers.

**Attention:** The BLU DELTA Service needs to communicate with the Docker Containers (Resource service) during start-up. It will try to retrieve resources in a loop until it gets them.

**So you have to set the correct bluDeltaLinuxServerIP before you start the BLU DELTA Service!**

You can stop/kill the BLU DELTA Service at any time with the bootstrapper script. Check section *Start/stop BLU DELTA Service (Windows Server)* for details

#### bluDeltaPortRangeStart

This value specifies the port number (default = 8090) to use for the first BLU DELTA Service Instance which will be started. All additional service instances will use the upcoming port-numbers. So, if you configure to start 2 service instances and you do not change the default port, these services will be accessible via the ports 8090 and 8091.

#### bluDeltaServiceCount

With this value you can adjust the number of parallel BLU DELTA Service instances (default = 1) individually for each installation. If you configure more than 1 instance, This can increase the performance and will improve reliability, but it will also occupy more memory and you are self-responsible for load-balancing in that case.

## OCR (Optical Character Recognition) Worker Count

To define the number of OCR Worker components of the BLU DELTA Service instances, edit the file „*Blumatix.Capture.Webservice.Client.Selfhosted.exe.config*“ which can be found in the installation directory. Then you can adjust the “value” of the key “OcrWorkerCount” to specify the number of OCR-Workers to create for each BLU DELTA Service instance.

```
<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 Service will NOT(!) abort document extraction after a specific time period.

```
<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.

```
<ocrConfig ... Languages="de,en" ... />
```

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:

<b>de</b>	German	<b>pl</b>	Polish	<b>cs</b>	Czech	<b>nb</b>	Norwegian	<b>pt</b>	Portuguese
<b>en</b>	English	<b>it</b>	Italian	<b>sk</b>	Slovakian	<b>hr</b>	Croatian	<b>sv</b>	Swedish
<b>es</b>	Spanish	<b>fr</b>	French	<b>nl</b>	Dutch	<b>tr</b>	Turkish	<b>sl</b>	Slovenian
<b>da</b>	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 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": false,
    "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 Linux Server**) and therefor disabled by default!

#### Prosa detection (disabled by default)

```
...
"DocumentPreparation": {
  "JunkDetection": {
    "Enabled": false,
    "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.

#### Typhon header details (enabled by default)

```
...
"TyphonHeaderDetails": [
  {
    "Name": "InvoiceId",
    "Enabled": true,
    ...
  },
  {
    "Name": "InvoiceDate",
    "Enabled": true,
    ...
  },
  {
    "Name": "GrandTotalAmount",
    "Enabled": true,
    ...
  }
]
```

The Typhon-engine delivers more accurate predictions than the standard models (which utilize older technologies). Typhon is not yet(!) able to predict all header-details, but the list of supported details (currently: InvoiceId, InvoiceDate, GrandTotalAmount) will be continuously extended with upcoming releases.

Performance-wise, Typhon-predictions are more expansive than those of the standard-models. So, if performance is relevant for you and you do not have GPU-support on your Linux machine, you may want to turn this feature off → Set “false” as value for each TyphonHeaderDetail in that case!

Check section *Typhon Engine for more accurate predictions* for further details .

#### Typhon contacts (enabled by default)

```
...
"Contacts": {
  "TyphonContacts": {
    "Enabled": true,
    ...
  }
},
```

The Typhon-engine delivers more accurate contact (Sender, Receiver) predictions than the standard model (which utilizes older technologies). As of yet(!), typhon will only deliver contact predictions for german documents. For all other languages the standard-model will be used as fallback mechanism. More languages will be supported with future releases.

Performance-wise, Typhon-predictions are more expansive than those of the standard-models. So, if performance is relevant for you and you do not have GPU-support on your Linux machine, you may want to turn this feature off.

Check section *Typhon Engine for more accurate predictions* for further details .

#### Typhon line items (enabled by default)

```
...
"LineItems": {
  "TyphonLineItems": {
    "Enabled": true,
    ...
  }
},
```

The Typhon-engine delivers more accurate line item predictions than the standard model (which utilizes older technologies).

Performance-wise, Typhon-predictions are more expansive than those of the standard-models. So, if performance is relevant for you and you do not have GPU-support on your Linux machine, you may want to turn this feature off.

Check section *Typhon Engine for more accurate predictions* for further details.

#### Response strategy (ForceReturnBestMatches = disabled by default)

```
...
"ResponseStrategy": {
  "ForceReturnBestMatches": {
    "Enabled": false,
    ...
  }
},
```

If set to "true", the thresholds for all (wherever possible) predictions will be ignored. Thus the best match (prediction with highest score) that we have available will be returned. By default (if set to "false"), internal thresholds will be active which validate if the score of a potential prediction is high enough to be returned.

#### RecommendedThresholds (enabled by default)

```
...
"Thresholds": {
  "RecommendedThresholds": [
    {
      "Enabled": true,
      "CustomerNames": [],
      "SupportedLanguages": [],
      "Details": {
        "InvoiceId": 0.3,
        "SenderOrderId": 0.83,
        "ReceiverOrderId": 0.8,
        "CustomerId": 0.5,
        "DeliveryNoteId": 0.65,
        "CostCenter": 0.5,
        "InvoiceDate": 0.2,
        "SenderOrderDate": 0.85,
        "ReceiverOrderDate": 0.8,
        "DeliveryDate": 0.85
      }
    }
  ]
}
```

Each element of the BLU DELTA API Response which has a "Score" property also has a "RecommendedThreshold" property on the same level. By default, the API will return empty elements instead of predictions if the score of these elements is lower than the configured threshold. You can change this behaviour in the "ResponseStrategy" section of the dynamic config.

Apart from that, the pre-configured thresholds can also be adapted to better suit your needs. Recommended-Thresholds are configurable for the following details:



### 1.) Single predictions --> [DetailName]

#### Examples:

```
"InvoiceId": 0.3,  
"InvoiceDate": 0.2,
```

### 2.) Prediction groups --> [ParentDetailName] | [ParentDetailName].[ChildDetailName]

#### Examples:

```
"VatGroup": 0.5,  
"VatGroup.VatRate": 0.55,
```

**VatGroup.VatRate** is more specific than **VatGroup**, so in this case **VatGroup.VatAmount** and **VatGroup.NetAmount** would get 0.5 and **VatGroup.VatRate** 0.55 as RecommendedThreshold.

### 3.) Contacts --> Contact | Contact.Sender | Contact.Receiver | Contact.Other

#### Examples:

```
"Contact": 0.5,  
"Contact.Sender": 0.6,
```

**Contact.Sender** is more specific than **Contact**, so in this case **Receiver** contact and **Other** contacts would get 0.5 and Sender 0.6 as RecommendedThreshold.

### 4.) Lineltems --> Lineltem | Lineltem.AllDetails | Lineltem.[DetailName]

#### Examples:

```
"LineItem": 0.4,  
"LineItem.AllDetails": 0.5,
```

Here, the RecommendedThreshold for each **Line** of the LineltemTable is set to 0.4.

The RecommendedThreshold for **each Detail** of the Line (Description, Quantity, UnitPrice,...) is set to 0.5.

```
"LineItem": 0.4,  
"LineItem.AllDetails": 0.5,  
"LineItem.TotalAmount": 0.6,
```

Here, the RecommendedThreshold for each **Line** of the LineltemTable is set to 0.4.

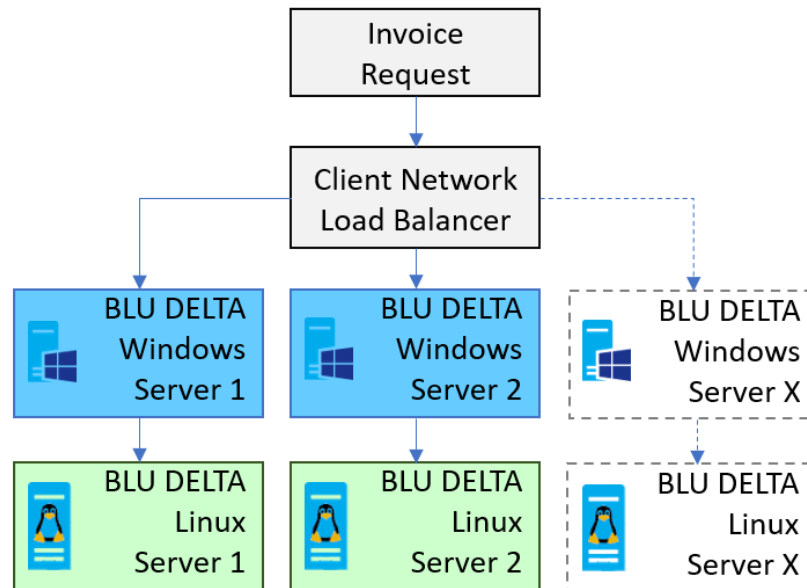
The RecommendedThreshold for **each Detail** of the Line (Description, Quantity, UnitPrice,...) is set to 0.5.

**Lineltem.TotalAmount** is more specific than **Lineltem.AllDetails**, so in this case the **TotalAmount** would get 0.6 and **all other Details** of the Line 0.5 as RecommendedThreshold.

## 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.

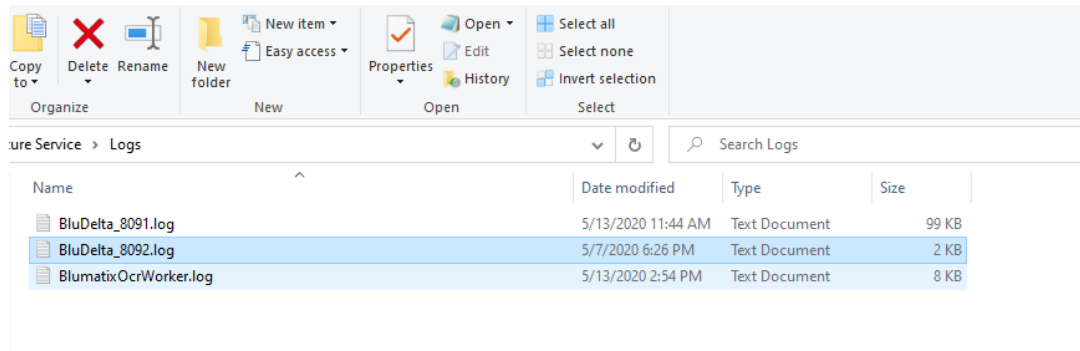
Typhon needs a lot of processing power, so it is highly recommended to install Triton on a Linux Server with compatible Graphic Card(s) → Check section *“Prepare a Linux Server with Triton”*

**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

BLUDELT 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="RollingFileAppender" 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 on the Windows server as well as of several docker containers on the Linux server. Depending on your configuration you should see the following containers on a single Linux machine:

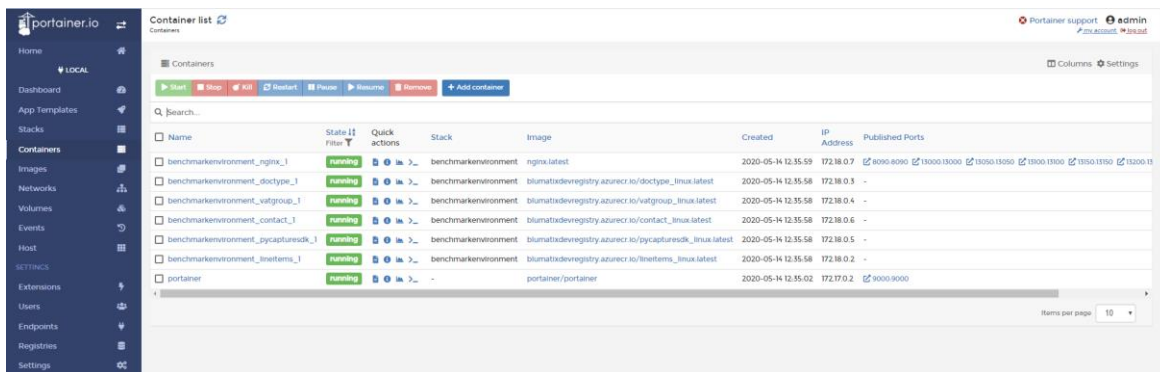
```
PS> docker ps --format "table {{.Image}}\t{{.Names}}"
IMAGE                                     NAMES
blumatixreleaseregistry.azurecr.io/nginx:1.17      environment_nginx_1
blumatixreleaseregistry.azurecr.io/invoice_ner_linux:v1.18.1  environment_ner_1
blumatixreleaseregistry.azurecr.io/typhon_contact_cpu:v1.18.1  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
blumatixreleaseregistry.azurecr.io/post_processing:1.0.0         environment_postprocessing_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.



## Active Monitoring

We suggest two methods for monitoring the overall health state of the BLU DELTA environment.

### 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://win-hostname:8090/ping>: Ping each BLU DELTA Service separately in case that more than one instance is running.
- PyInvoiceCapture Service: <http://linux-hostname:13000/ping>
- Contact Service: <http://linux-hostname:13050/ping>
- VatGroup Service: <http://linux-hostname:13100/ping>
- DocType Service: <http://linux-hostname:13150/ping>
- Lineltems Service: <http://linux-hostname:13200/ping>
- Typhon Lineltems Service: <http://linux-hostname:13201/ping>
- NER Service: <http://linux-hostname:13500/ping>
- NERDeliveryPeriod Service: <http://linux-hostname:13501/ping>
- Typhon Contact Service: <http://linux-hostname:13600/ping>
- Typhon HeaderDetails Service: <http://linux-hostname:13700/ping>
- QR Code Service: <http://linux-hostname:13800/ping>
- Post Processing Service: <http://linux-hostname:13880/ping>
- Resource Service: <http://linux-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 Apidentifier)
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.

## Workload Monitoring

It is important to understand that a single BLU DELTA server instance cannot handle an endless number of concurrent requests. The request processing-time will increase with the total number of requests to handle in parallel.

**Attention:** If you consequently send more requests into a single BLU DELTA server instance than you receive responses from it, you will eventually reach a point where the service is no more accessible or crashes.

The number of documents that a single BLU DELTA server can handle efficiently in parallel depends on many different factors. Most important are:

- Hardware Configuration
- System Configuration
- Network Configuration
- BLU DELTA Configuration
- Complexity of the documents (page count, word count, etc...)

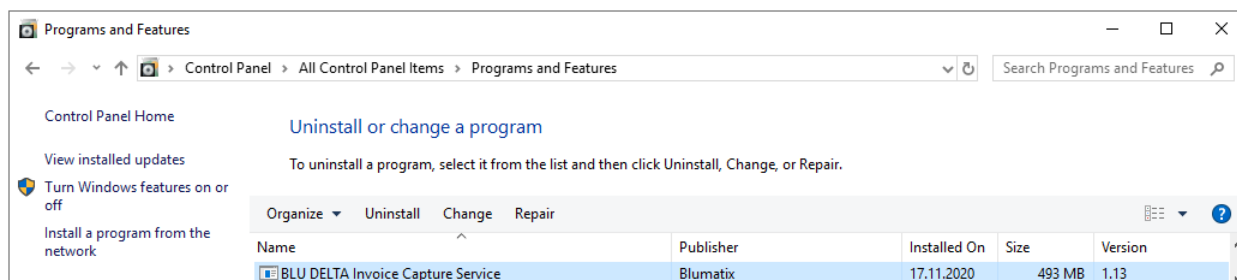
You should use additional BLU DELTA server instances and place them behind a loadbalancer, to increase the number of documents to handle concurrently. (Check section "Load Balancing")

The BLU DELTA Windows service features a ping-endpoint which can be used to check, if the service is up and running (by default: <http://localhost:8090/ping> --> check section: "Passive Monitoring"). If the BLU DELTA Windows service is under high pressure by many concurrent requests, the ping-requests will also be delayed drastically. High ping-times can be used as an indicator that the service is under pressure (horizontal scaling needed), but a scaling mechanism should not purely rely on it.

## Upgrade / Rollback

### Uninstall Blu Delta Service (Windows)

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 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.

## 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

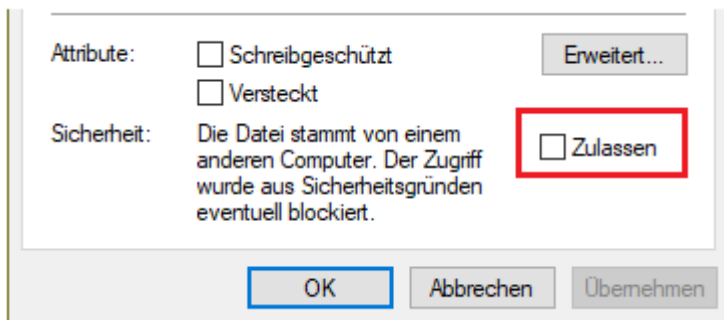
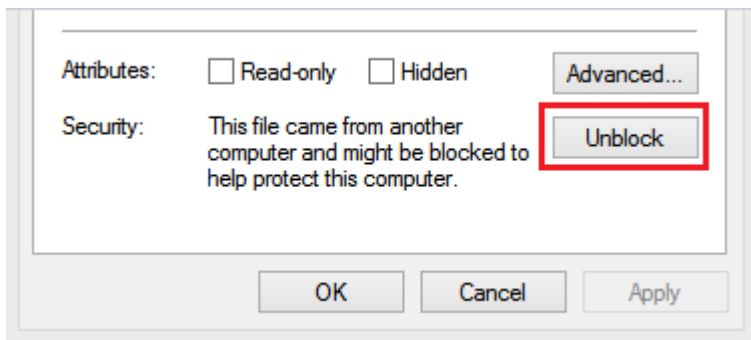
This document includes a chapter of known issues.

### 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