InvoiceReceiverV3 manual

Program Version:
V3.0.11
Document Version: V3.0.
Modified: 2021-12-11

Table of contents

Document changes....4
I. Preface....5
I.1 Overview of basic flow of information....6
II. InvoiceReceiver working modes....7
III. Installation....8
III.1 Distribution format....8
III.2 Prerequisites....8
III.2.1 Required Java version....8
III.3 Installation steps....9
III.3.1 Accessing NAV production system....10
IV. Application configuration....10
IV.1 Credential configuration....10
IV.2 Optional configuration parameters....10
IV.3 Java SSL configuration....11
IV.3.1 Run time Java SSL configuration variables....11
IV.3.2 External tools to handling of SSL certificates....12
IV.4 Command line parameters....12
IV.5 Configure JDE Connectivity....13
IV.6 Testing of configuration....13
V. Running of application....14
V.1 Starting the application....14
V.2 Application locking....14
V.2.1 Lock timeout feature....15
VI. Housekeeping....15
VI.1 Backup and restore....15
VI.2 Internal database maintenance....15
VI.3 Purging of obsolete data....15
VI.4 Maintenance of Java SSL certificates....16
VII. Log files....16
VII.1 Application logs....16
VIII. Known issues....17
VIII.1 Transmission: “NoSuchAlgorithmException: SHA3-512 MessageDigest not available”. 17
VIII.2 Multiple failures during connect....17
IX. Deploying upgrades....18
IX.1 V3.x.x→ V3.0.6....18
IX.2 V3.x.x→ V3.0.7....18
Appendix A.) ChangeLog of current version....19
Appendix B.) Distributed files and their descriptions....20
Appendix C.) Used OpenSource components and licensing....21
Appendix D.) Sample of InvoiceReceiver.properties config....22
Appendix E.) In case of program errors....24
Appendix F.) Oracle SQL Script for JDE interfacing....25

Document changes

Program Version	Document version	Date	Author	Summary of change
V1.0.0	V1.0.0	2020-08-24	János TŐBERLING	Initial version of manual
V1.0.3	V1.0.3	2020-10-06	János TŐBERLING	Fix error on incomplete NAV data Adding customerGroupMemberTaxNo Adding SingleInvoiceQueyMode
V3.0.0	V3.0.0	2020-12-08	János TŐBERLING	For NAV API V3 Adding connectiontimeout and sockettimeout properties parameters
V3.0.6	V3.0.6	2021-10-05	János TŐBERLING	InvoiceReceiver trunk supplierName & customerName to 100 chars customerGroupTaxNumber path change
V3.0.7	V3.0.7	2021-11-07	János TŐBERLING	Adding taxNo parameter to INVOICEQUERIES table

I. Preface

From 2018. July 1st Hungarian Tax Authority (NAV) requests that all the invoices referring to companies (none for individual persons) have to send to the tax authority immediately during invoice generation process.
NAV defined an interface – on where invoice data can be send over Internet.
The interface itself consists of 2 layers:
– Data layer is to contain invoice data in XML format (Data exchange)
– API layer is to transfer embedded Data XML over the internet (Communication layer)

InvoiceReceiver application is created to handle NAV's API layer communication for querying invoice data.

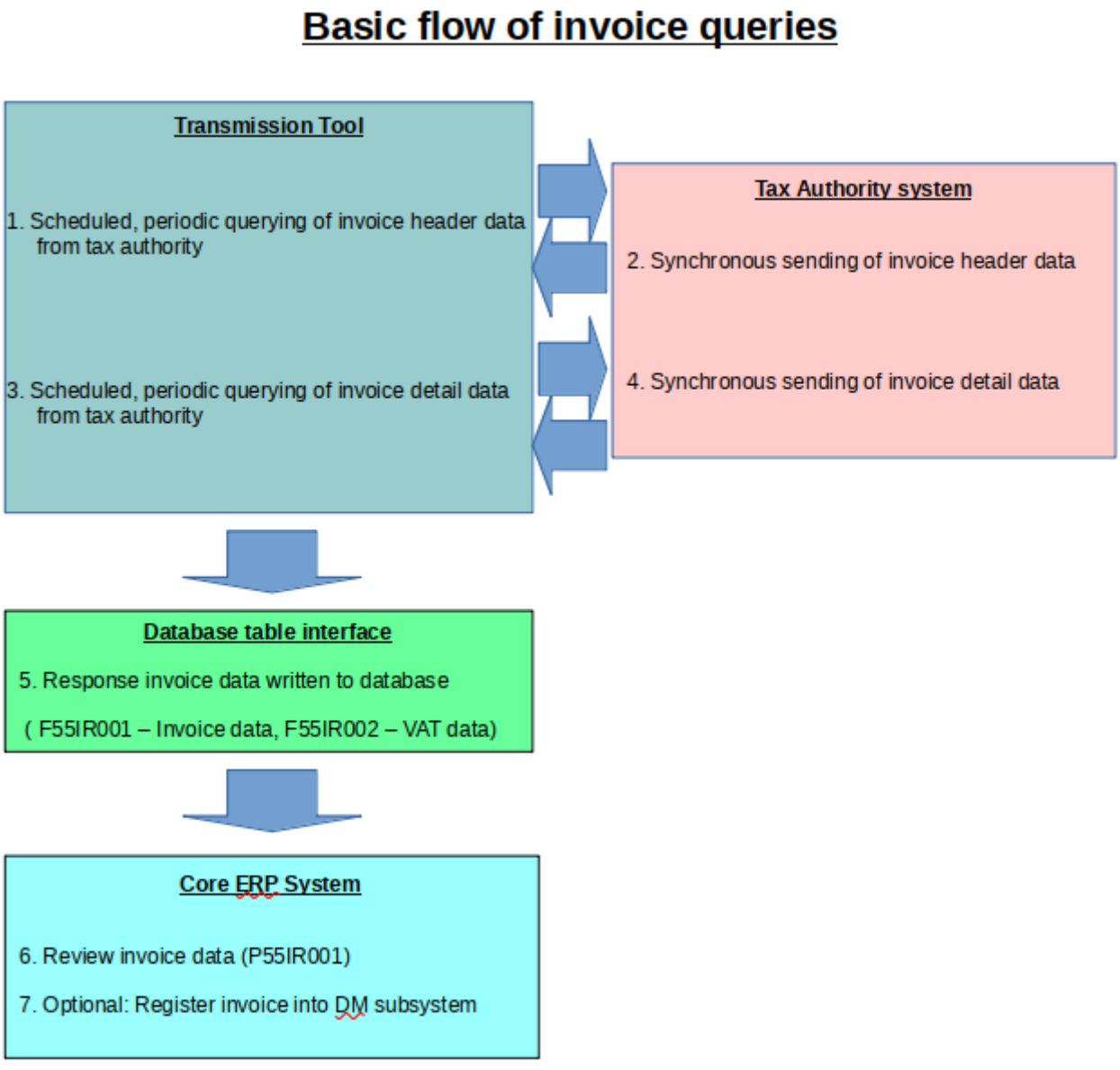
Along with theV1.1 protocol change the NAV is forcing TLSv1.2 HTTPS submission since 29th April 2019. [1] TLSv1.2 has Java version prerequisites – please check chapter "*III.2.1 Required Java version since 29th April 2019*". In case the system's default version is not supported alternate Java runtime required to be installed. In case of connection issues please refer to chapter "*VIII.2 "SSLHandshakeException: Received fatal alert: handshake failure" during transmission*".

By 2020 new NAV API & Data layer protocols will be introduced as Version3. Program version V3 is to support these changes. The former V2 protocol is expected not to be available after 1st April 2021.

1 Please refer to TLSv1.2 changes to: https://onlineszamla.nav.gov.hu/home / News 2019-03-12

I.1 Overview of basic flow of information

The following figure shows how the basic flow and communication steps between different systems and components during the querying activity. The InvoiceReceiver application takes apart in the flow referred as the Transmission Tool.



II. InvoiceReceiver working modes

The InvoiceReceiver application is an asynchronous polling-querying-receiving application.

The application has different modes (Single Invoice QueryMode, WebService Mode, Token-testing[2]) – this document describes the basic working mode.

In basic mode the application starts and terminates. This is useful for checking settings and database connectivity.

In WebServer mode the application does the two following main activities:
- Opens a REST webservice port to provide simple APIs
- Runs a periodic scheduler to query NAV data

In SingleInvoiceQueryMode mode the application does the two following main activities:
- Next sheduled interval is queries from NAV InvoiceDigest API (once)
- If invoices found on previously quieried interval NAV InvoiceDetail API is queried for each

The program is recommended to start as a part from the OS bootup sequence (Windows scheduler / Unix init, systemctl or supervisor) or optionally from OS scheduler (Windows scheduler or Unix Cron).

2 For Token-testing mode please refer to chapter: *IV.6 Testing of configuration*

III. Installation

III.1 Distribution format

The InvoiceReceiver application is a Java JAR application packed with supplemental directories and files into a ZIP archive file[3].

III.2 Prerequisites

- Java JRE 1.9 (preferred OpenJDK 13 or later)[4]
- Windows or Linux environment
- Read/Write access to directories (./logs)
- Read access to directories (./resources)
- Internet connectivity / Java SSL configuration
- Optionally: OS Scheduler/OS Startup subsystems (Windows scheduler or Linux startup subsystem.)
- Hungarian tax authority registration / Nav Online szamla technical user credentials

III.2.1 Required Java version

NAV data transmission protocol required SHA3-512 hashing on V2 protocol communication.

Java versions prior to Java9 are lack of the required crypto.

As a requirement a supported, up-to-date Java Runtime (JRE) required to be installed for running the application.

3 The detailed description of the distribution package is available at: Appendix B.)

4 In case multiple Java versions installed PATH and JAVA_HOME environment variables can be set accordingly in runme.bat or runme.sh.

III.3 Installation steps

1. Unzip distribution archive to a specific location.
2. Provide RW access to required directories.
3. Provide Execute access to shell scripts/cmd files.
4. Configure mandatory credential parameters (./resources/InvoiceReceiver.properties)
5. Optional: Configure further parameters (eg: proxy, non-standard directories, logging etc., scheduler)
6. Optional: Configure NAV production system serviceurl
7. Test configuration

III.3.1 Accessing NAV production system

The application distribution package is configured for NAV's Test environment. To switch to NAV Production system please apply the following changes:

- ./resources/InvoiceReceiver.properties – the serviceurl parameter needs to be changed:
serviceurl=https://api.onlineszamla.nav.gov.hu/
- ./resources/InvoiceReceiver.properties – Make sure that local XML validation is turned off.
...

IV. Application configuration

IV.1 Credential configuration

Before run credential information needs to be supplied into ./resources/InvoiceReceiver.properties file. The following data is required from NAV authentication:

- Hungarain tax registration number of the company (only first 8 characters)
- NAV side generated technical username
- NAV side provided technical password
- NAV side generated XMLSignKey
- NAV side generated Exchange key

Please refer to *Appendix D.) Sample of InvoiceReceiver.properties config* for further information.

IV.2 Optional configuration parameters

Besides the mandatory credential configuration the following group of parameters can be tuned in ./resources/InvoiceReceiver.properties file:

- HTTP REST endpoint (NAV test / NAV production system REST endpoints)
- HTTP Proxy configuration
- Internal database configuration
- Invoice query scheduling parameters
- JDE number divider/multiplicatior
- Query direction (INBOUND (default) / OUTBOUND)
- WebServer port (Defaults: 8099)

Please refer to *Appendix D.) Sample of InvoiceReceiver.properties config* for further information.

IV.3 Java SSL configuration

To protect transmission channel NAV applied SSL Layer on the REST interface. The application can use the OS's default or an application configured Java keystore to accept NAV's certificate.

On ./runme.sh & ./runme.bat files Java parameters are configured to use ./resources/cacerts.jks keystore (currently holding NAV Test System certificate and NAV Production System certificates).

Alternatively NAV SSL certificates can be added to the keystore of the operation system – please refer to OpSys & Java user documentations.

Upon NAV certificates are changing the update of keystore entries might be nescessary. Example scripts provided as ./resources/1_downloadcert_teszt.sh and ./resources/2_downloadcert_prod.sh where the required steps can be found upon NAV certificate changes and update required on keystore. (Certificate is expected to change during NAV software lifecycle.)

IV.3.1 Run time Java SSL configuration variables

To succesfully connect an SSL endpoints the following Java runtime parameters might be configured:

Parameter	**Description**
javax.net.ssl.trustStore	*Default*: ./resources/cacerts.jks Points to Java trustSrore file holding all remote server certificates.
javax.net.ssl.trustStorePassword	*Default*: changeit Cleartext password protecting keystore file. Provided at keystore generatrion.
javax.net.ssl.trustStoreType	*Default*: PKCS12 [5] Type of trustStrore. Possible values: JKS, PKCS12

Runtime parameters can be setup in execution command line in like the following example:

```
#!/bin/bash
java -jar \
      -Djavax.net.ssl.trustStore=./resources/cacerts.jks \
      -Djavax.net.ssl.trustStorePassword=changeit \
      -Djavax.net.ssl.trustStoreType=PKCS12 \
      -Dlog4j.configurationFile=./resources/log4j2.xml \
      -jar InvoiceReceiver.jar -kt 2 $*
```

5 Default trusStoreType was changed by Oracle from JKS → PKCS12 by Java version 9.

IV.3.2 External tools to handling of SSL certificates

OpenSource *openssl* and Java *keytool* programs are usefull to generate and manage trustStore certificates. The file resources/1_downloadcert-cmdlineparamserver.sh provides the following steps to regenrate trusStore

```
#!/bin/bash

echo -n | openssl s_client -connect $1:443 | sed -ne '/-BEGIN CERTIFICATE-/,/-
END CERTIFICATE-/p' > ./$1.pem


openssl x509 -in ./$1.pem -text -noout | grep "Subject:"


rm ./cacerts.jks


keytool -import -trustcacerts -keystore ./cacerts.jks -storepass changeit -
noprompt -alias $1.cert -file ./$1.pem
keytool -list -keystore ./cacerts.jks -storepass changeit -noprompt
```

IV.4 Command line parameters

Command line parameters of the program can be invoked by issuing:

```
java -jar InvoiceReceiver.jar -h
```

The available command line parameters are the following:

```
 -c,--config <arg>               Properties file location (Default:
                                 ./resources/InvoiceReceiver.properties)
 -d,--disablepropertiesfile      Disable load of InvoiceReceiver.properties
 -db,--database <arg>            Database JDBC location. Default: empty.
                                 Possible 'h2:filename'
 -dbdc,--dbtabledontcreate       DB disbale automatic [REQUESTS] table
                                 creation. True if specified. Defaults:
                                 false
 -dbpw,--dbpwd <arg>             Database password. Default: ''
 -dbq,--dbautoquery <arg>        DB automatic query runs. [true, false]
                                 Requires db. Default: true
 -dbu,--dbuser <arg>             Database user. Default: sa
 -h,--help                       Command line help
 -k,--lockfile <arg>             Lockfile location (Default
                                 ./resources/InvoiceReceiver.lock)
 -kt,--locktimeout <arg>         Lockfile timeout (Default: none)
 -p,--httpuseproxy               Use http proxy
 -q,--maxqueryattempts <arg>     Maximum number of query attempts.
                                 Default: 300 (eg: 25 hours * 12 queries/hour)
 -ph,--httpproxyhost <arg>       Proxy host. Default: localhost
 -pp,--httpproxyport <arg>       Proxy port. Default: 3128
 -pt,--httpproxyprotocol <arg>   Proxy protocol. Default: http Possible:
```

```
 -S,--singlequerymode             Single InvoiceReceiver query mode

 -t,--tokentest                   Test mode. Get NAV Exchange token.
 -u,--serviceurl <arg>            Service URL (Default:
                                  https://api-test.onlineszamla.nav.gov.hu/
 -v,--version                     Version info
 -W,--webservermode               Activate webserver mode.
 -x,--httpuseproxyuser            Use http with proxy user (requires -p
                                  also)
 -xp,--httpproxypassword <arg>    Proxy user password
 -xu,--httpproxyuser <arg>        Proxy user
```

IV.5 Configure JDE Connectivity

The application is designed to support JD Edwards solution.

When JDE interfacing is required the data link between JDE and InvoiceReceiver are provided through specific interface tables. The tables are JDE tables onwhere the InvoiceReceiver saves information through specific views.

To setup link between JDE and InvocieReceiver the following steps need to be configured:

1. JDE InvoiceReceiver application needs to be deployed
2. JDE InvoiceReceiver tables (F55IR001, F55IR002) have to be generated under JDE[6]
3. As the application accessing the tables through specific views and requires DB sequences to insert SQL scripts needs to be run for specific setting on the database. Please refer to: *Appendix F.) Oracle SQL Script for JDE interfacing*
4. DB access and grants needs to be provided to the application - JDBC connection string needs to be created.
5. InvoiceReceiver DB specific access parameters needs to be setup in ./resources/InvoiceReceiver.properties.

IV.6 Testing of configuration

Testing of setup and credential configuration is possible via the -t command line parameter.
A successful check should give similar output:

```
[2020-08-24 12:38:20:437] [INFO] [InvoiceReceiver:647] - InvoiceReceiver 2020. V0.1.0
[2020-08-24 12:38:20:438] [INFO] [InvoiceReceiver:226] - Forcing https.protocols=TLSv1.2
```

6 As the InvoiceReceiver is writing data to two of the tables it is recommended to provide and “interfacing namespace” for the F55IR001 and F55IR002 tables. under JDE and provide application specific access to the tables. Tables can be mapped to specific namespaces in JDE under OCM.

```
[2020-08-24 12:38:20:642] [INFO] [InvoiceReceiver:445] - use -h or --help for usage
[2020-08-24 12:38:20:642] [INFO] [InvoiceReceiver:490] - Token test mode selected.
[2020-08-24 12:38:20:645] [INFO] [InvoiceReceiver:302] - Don't move on technical
exception: false
[2020-08-24 12:38:20:649] [INFO] [InvoiceReceiver:327] - QueryScheduleParameters:
Epoc:=1592438400, Step:=864000, Offset:=30, Schedule=*/10 * * * * ?
[2020-08-24 12:38:20:649] [INFO] [InvoiceReceiver:328] - InterfaceDirection := INBOUND
[2020-08-24 12:38:20:649] [INFO] [InvoiceReceiver:329] - DisableInvoiceQuery := false,
DisableInvoiceDetails := false
[2020-08-24 12:38:20:650] [INFO] [InvoiceReceiver:330] - JDENumberMultiplicator:= 100.0
[2020-08-24 12:38:21:275] [INFO] [DatabaseHandler:204] - DB connected:
oracle:thin:@10.92.10.161:1521:JD92?useUnicode=false&characterEncoding=iso88592
[2020-08-24 12:38:21:276] [INFO] [InvoiceReceiver:731] - Acquire NAV exchange token...
[2020-08-24 12:38:21:292] [WARN] [NavRestHandler:244] - Java version is obsolate.
Fallback to 3rdParty SHA3-512 hashing. Please update Java >= 9!
[2020-08-24 12:38:21:854] [INFO] [InvoiceReceiver:735] - NAV Exchangetoken succesfully
acquired: ad7871f3-c843-4ecc-aae0-3d2c6fa0080a32U47B5LPESU
[2020-08-24 12:38:21:877] [INFO] [InvoiceReceiver:749] - Application terminates
```

V. Running of application

V.1 Starting the application

Running of application is possible as standard runable jar files. The ./runme.bat and ./runme.sh files are provided to help issuing the application.

Example distributed runme.sh:

```
#!/bin/bash

java -Djavax.net.ssl.trustStore=./resources/cacerts.jks \
  -Dlog4j.configurationFile=./resources/log4j2.xml -jar InvoiceReceiver.jar $*
```

The following Java configuration defines recommended:

- javax.net.ssl.trustStore – Define default SSL keystore for the application
- log4j.configurationFile - Define default Log4j2 configuration file

V.2 Application locking

During the program run the application writes ./resources/InvoiceReceiver.lock file (path can be overwritten from command-line parameter -k) to prevent concurrent execution.

In case of abnormal program termination lock file needs to be removed manually before next program invocation!

V.2.1 Lock timeout feature

A lock timeout feature can be switched on to prevent long-time stale locks hanging out operation. With switch the feature on “–locktimeout <integer_secs>” (or “-kt <integer_secs>”) lock timeout value can be specified. If the feature is switched on than if lock file exists the program run examines the last modification time of the lockfile – in case if it’s older than the specified time the lockfile is ignored and “taken over”.

VI. Housekeeping

VI.1 Backup and restore

Application binary and all related application data can be backed up and restored by archiving the root directory of deploy.

Please make sure that during backup and restore operations the application is not running and scheduler is switched off.

VI.2 Internal database maintenance

By default the application is using an embedded H2 Database engine to track and query NAV transmissions. By default the database resides in a single file – if not exists it is created automatically. The database file location can be specified in ./resources/InvoiceReceiver.properties file – it is recommended to change the filename annually.

The application is using standard JDBC connector and syntax – connecting to different databases might be possible but out of scope of this document.

VI.3 Purging of obsolete data

To purge obsolete data client purging/archiving rules should apply on following points:

./logs folder – Contains application and result logs

Purging old data is out of scope of application run.

VI.4 Maintenance of Java SSL certificates

NAV HTTPS certificates can be store on operating system level Java keystores or program level keystore (./resources/cacerts.jks). If NAV changes certificates and program terminates due to SSL connection issues update of keys might be required. Local certificate maintenance is out of scope of program tasks and belongs locale IT operations.

VII. Log files

The log files are created by applying the standard Log4j2 subsystem – fine tuning is possible as Log4j2 allows it. Please refer to Log4j documentation[7].

Both InvoiceReceiver log files are configured to be daily-rolling, compressed archive logs.

VII.1 Application logs

Application logs can be found in ./logs/InvoiceReceiver.log files.
Example of application log:

```
[2020-08-24 12:23:29:917] [INFO] [InvoiceReceiver:647] - InvoiceReceiver 2020. V0.1.0
[2020-08-24 12:23:29:918] [INFO] [InvoiceReceiver:226] - Forcing https.protocols=TLSv1.2
[2020-08-24 12:23:30:129] [INFO] [InvoiceReceiver:445] - use -h or --help for usage
[2020-08-24 12:23:30:132] [INFO] [InvoiceReceiver:302] - Don't move on technical exception: false
[2020-08-24 12:23:30:137] [INFO] [InvoiceReceiver:327] - QueryScheduleParameters: Epoc:=1592438400,
Step:=864000, Offset:=30, Schedule=*/10 * * * * ?
[2020-08-24 12:23:30:137] [INFO] [InvoiceReceiver:328] - InterfaceDirection := INBOUND
[2020-08-24 12:23:30:138] [INFO] [InvoiceReceiver:329] - DisableInvoiceQuery := false,
DisableInvoiceDetails := false
[2020-08-24 12:23:30:138] [INFO] [InvoiceReceiver:330] - JDENumberMultiplicator:= 100.0
[2020-08-24 12:23:30:941] [INFO] [DatabaseHandler:204] - DB connected:
oracle:thin:@10.92.10.161:1521:JD92?useUnicode=false&characterEncoding=iso88592
[2020-08-24 12:23:30:942] [DEBUG] [InvoiceReceiver:414] - Application run is locked with
filelock: ./resources/InvoiceReceiver.lock
[2020-08-24 12:23:30:989] [INFO] [InvoiceReceiver:749] - Application terminates
```

7 https://logging.apache.org/log4j/2.x/manual/index.html

VIII. Known issues

VIII.1 Transmission: “NoSuchAlgorithmException: SHA3-512 MessageDigest not available”

As from NAV protocol version V2 (mandatory in production system since 1st April 2020) SHA3-512 hashing algorithms is required on NAV API protocol during transmissions.

Java version since Java 9 are supporting SHA3-512

Please upgrade running Java runtime to a supported version.
(Minimum version: Java9, recommended version: OpenJDK 13.)
If upgrade is not possible (eg: JD Application server is using system default Java) alternative Java runtime needs to be deployed and set up.

Sample steps for deploying alternative JDK/JRE:

1. Download supported JDK/JRE from OracleJDK: https://jdk.java.net/13/
   (eg: openjdk-13.0.2_linux-x64_bin.tar.gz)
2. Extract downloaded JDK/JRE to a system directory. Eg: *//usr/lib/jvm/<your_java_version>/ (eg: /usr/lib/jvm/jdk-13.0.2/)*
3. Setup runme.bat or runme.sh to run the appropriate java from the new path.

```
#!/bin/bash
export JAVA_HOME=/usr/lib/jvm/jdk-13.0.2/
/usr/lib/jvm/jdk-13.0.2/bin/java -Djavax.net.ssl.trustStore=./resources/cacerts.jks -
Dlog4j.configurationFile=./resources/log4j2.xml -jar InvoiceReceiver.jar $*
```

VIII.2 Multiple failures during connect

In some cases – specially for OpenJDK – cacert.jks is generated in invalid format.

Use cacert.jks shipped with the package or regenerate it by Oracle Java keytool.

IX. Deploying upgrades

IX.1 V3.x.x → V3.0.6

Simple overwrite InvoiceReceiver.jar file.

IX.2 V3.x.x → V3.0.7

- Recommended: Backup database

- Execute SQL script to add new column to INVOICEQUERIES table.
 –> docs/modify_ora_2021-10-07.sql

- Replace InvocieReceiver.jar file.

Appendix A.) ChangeLog of current version

```
# Changelog
All notable changes to this project will be documented in this file.

The format is based on [Keep a Changelog](http://keepachangelog.com/en/1.0.0/)
and this project adheres to [Semantic Versioning](http://semver.org/spec/v2.0.0.html).

## [1.0.0] - 2020-08-24
- Initial version

## [1.0.1] - 2020-10-01
- Fix crash due to missing XML data like paymentMethod
- Add jaxb API to JAR

## [1.0.2] - 2020-10-02
- Adding customerGroupTaxNumber

## [1.0.3] - 2020-10-06
- Adding SingleInvoiceQuery mode

## [1.0.5] - 2020-10-29
- Adding SmallBusinessIndicator

## [1.1.0] - 2020-11-05
- Handling of InvoiceLines (F55IR003)

## [3.0.0] - 2020-12-08
- NAV API V3
- timeoutconnection timeoutsocket properties

## [3.0.1] - 2020-12-14
- NAV API V3 SHA-512 CryptoType change

## [3.0.2] - 2021-01-06
- TaxNumber Query fixes

## [3.0.3] - 2021-01-07
- TaxNumber Query fixes
- Special Invoice Query Mode

## [3.0.4] - 2021-02-14
- TaxNumber Query minor fix

## [3.0.5] - 2021-09-10
- Error handling dummyresponse fix on missing info

## [3.0.6] - 2021-10-05
- InvoiceReceiver trunk supplierName & customerName to 100 chars
- customerGroupTaxNumber path change

## [3.0.7] - 2021-10-07
- Adding taxno to INVOICEQUERIES

## [3.0.8] - 2021-10-26
- Adding featureUPMJ

## [3.0.9] - 2021-10-28
- Fixing null inserts
```

```
## [3.0.10] - 2021-11-12
- QueryTime From +1 ms

## [3.0.11] - 2021-12-11
- Outbound query multiple TaxNumber enhancement
```

Appendix B.) Distributed files and their descriptions

File	Description
./resources	Directory containing configuration files
./logs	Directory containing logs
./runme.bat	Example batch file for running under Windows environment
./runme.sh	Example batch file for running under Unix environment
./CHANGELOG.md	InvoiceReceiver application ChangeLog
./README.md	InvoiceReceiver application basic Readme file
./InvoiceReceiver.jar	InvoiceReceiver JAR application binary
./resources/ 1_downloadcert_teszt.sh	Unix shell script example to update keystore – NAV Test system
./resources/ 2_downloadcert_prod.sh	Unix shell script example to update keystore – NAV Production system
./resources/api-test.pem	PEM certificate of NAV test system
./resources/cacerts.jks	Java keystore holding NAV test system certificate
./resources/ invoiceApi_20180111.xsd	NAV issued API XSD
./resources/ invoiceData_20180601.xsd	NAV issued Data XSD
./resources/ InvoiceReceiver.properties	InvoiceReceiver main configuration file
./resources/log4j2.xml	InvoiceReceiver log4j configuration file
./docs/ InvoiceReceiver_manual_V1.0.0 _20200824.pdf	InvoiceReceiver program documentation.
./docs/	InvoiceReceiver enduser documentation.

Appendix C.) Used OpenSource components and licensing

Component name	License	Further info
Log4j2	Apache License, Version 2.0	https://logging.apache.org/log4j/2.x/license.html
Apache commons	Apache License, Version 2.0	http://www.apache.org/licenses/
H2 Database engine	MPL2.0 / EPL 1.0	http://www.h2database.com/html/license.html
Junit	EPL 1.0	https://junit.org/junit4/license.html
Apache Directory Studio	Apache License, Version 2.0	http://www.apache.org/licenses/
Apache HTTP Components	Apache License, Version 2.0	http://www.apache.org/licenses/

Appendix D.) Sample of InvoiceReceiver.properties config

```
# REST API listen port
#webserverport=8099

# TCP/IP Connection parameters (msec)
# timeoutconnection=30000
# timeoutsocket=90000

# ------------------------------------------------------------ #
#
#  Change this section as of NAV registration data
#
taxnumber=4884______
servicelogin=HzZnb_____________
servicepassword=Password______
xmlsignkey=d9-c012-0349293bcd_________________
exchangekey=8f9212__________

# INBOUND (default) / OUTBOUND
#direction=OUTBOUND

# ------------------------------------------------------------ #
#
#  Change this section to setup invoice query parameters

disableinvoicequery=false
disableinvoicedetails=false

# 2020-01-01T00:00:00Z
queryepoch=2020-09-01T00:00:00Z

# 3600*24*10 = 10 days = 864000 secs
querystep=864000

# Seconds for query negative offset
queryoffset=30

# http://www.quartz-scheduler.org/documentation/quartz-2.3.0/tutorials/crontrigger.html
# Sec, Min, Hour, DayOfMonth, DayOfWeek, Year
queryschedule=0,10,20,30,40,50 * * * * ?


# ------------------------------------------------------------ #
#
#  Change this section to modify DB properties

# JDBC database access specification.
# H2 DB JDBC driver included. Pls. refer: http://www.h2database.com
# Missing entry means disabled DB functionality! (Also AutoQuery!)
database:oracle:thin:@1.2.3.4:1521:JD92?useUnicode=false&characterEncoding=iso88592
dbuser=DVIFACE
dbpwd=DVIFACE
dbtabledontcreate=true


# JDE Number conversion factor
jdenumbermultiplicator=100
```

```
# ------------------------------------------------------------ #
#
#  Change this section in case HTTP Proxy required
#

httpuseproxy=false
httpproxyprotocol=http
httpproxyhost=localhost
httpproxyport=3128
# Basic Authentication is supported
httpuseproxyuser=true
httpproxyuser=user
httpproxypassword=Password123


# ------------------------------------------------------------ #
#
# Custom Interface identification - on the begin of RequestID
#
# Interface ID - 2 chars string
interfaceid=EB
# ------------------------------------------------------------ #

#Nav Teszt URL
#serviceurl=https://api.onlineszamla.nav.gov.hu/
serviceurl=https://api-test.onlineszamla.nav.gov.hu/
#serviceurl=https://localhost:4430/
jtoberling@jtoberling-mint:~/workspa
```

Appendix E.) In case of program errors

If program misbehavior occurs please do the following steps:

- Check is application lock exists. Remove it if necessary. (Please refer to: Application locking)
- Check network connectivity to NAV system.
- Perform if application can get token (Please refer to: Testing of configuration)
- Check application logs. Increase log level to DEBUG in ./resources/log4j2.xml if necessary. (Please refer to: Application logs)
- In case internal H2 database is used H2 file can be monitored with h2 jar.
  (Hint: http://h2database.com/html/quickstart.html)
- Collect last referring application logs / submit to support.

Appendix F.) Oracle SQL Script for JDE interfacing

```
-- CREATE USER InvoiceReceiver IDENTIFIED BY "PASSWORD";

-- GRANT  CREATE SESSION, ALTER SESSION, CREATE DATABASE LINK, -
--  CREATE MATERIALIZED VIEW, CREATE PROCEDURE, CREATE PUBLIC SYNONYM, -
--  CREATE ROLE, CREATE SEQUENCE, CREATE SYNONYM, CREATE TABLE, -
--  CREATE TRIGGER, CREATE TYPE, CREATE VIEW, UNLIMITED TABLESPACE -
--  to InvoiceReceiver;

-- GRANT  CREATE MATERIALIZED VIEW, CREATE PROCEDURE, CREATE PUBLIC SYNONYM, -
--  CREATE SEQUENCE, CREATE SYNONYM, CREATE TABLE, -
--  CREATE TRIGGER, CREATE TYPE, CREATE VIEW, UNLIMITED TABLESPACE -
--  ON DVIFACE TO InvoiceReceiver;

-------------- Using DVIFACE

-- GRANT CREATE ANY SEQUENCE, ALTER ANY SEQUENCE, DROP ANY SEQUENCE, SELECT ANY
--    SEQUENCE TO DVIFACE;
-- GRANT CREATE TRIGGER TO DVIFACE;

--              DISCONNECT / RECONNECT HERE    !!!!!!!!


DROP TABLE INVOICEQUERIES;

CREATE TABLE INVOICEQUERIES (
                  id NUMBER(19,0),
                  starttime_ts NUMBER(19,0),
                  starttime nvarchar2(50),
                  querystep NUMBER(19,0),
                  endtime_ts NUMBER(19,0),
                  endtime nvarchar2(50),
                  querytime_ts NUMBER(19,0),
                  querytime nvarchar2(50),
                  newinvoices integer default 0,
                  requestId nvarchar2(50)
);

ALTER TABLE INVOICEQUERIES ADD ( CONSTRAINT INVOICEQUERIES_pk PRIMARY KEY (ID));
CREATE SEQUENCE INVOICEQUERIES_SEQ START WITH 1;
CREATE INDEX "REQINV_I1" ON "INVOICEQUERIES" ("STARTTIME_TS" DESC);
CREATE INDEX "REQINV_I2" ON "INVOICEQUERIES" (requestId DESC, id);
CREATE OR REPLACE TRIGGER INVOICERECEIVER_BIR
                  BEFORE INSERT ON INVOICEQUERIES
                  FOR EACH ROW

                  BEGIN
                    SELECT INVOICEQUERIES_SEQ.NEXTVAL
                    INTO   :new.id
                    FROM   dual;
                  END;
                  /
```

```
CREATE OR REPLACE VIEW invoices AS SELECT
  IVUKID id,
  IVGEN10 direction,
  IVL74MINV invoiceNumber,
  IVXREFID originalInvoiceNumber,
  IVBNDLSTS queryStatus,
  IVWAPRN invoiceOperation,
  IVC9CCAT invoiceCategory,
  IVFDAT invoiceIssueDate,
  IVBCGF supplierTaxNumber,
  IVR74ADL1U supplierName,
  IVBCGT customerTaxNumber,
  IVBCGC customerGroupTaxNumber,
  IVR74ADL2U customerName,
  IVPJRST05 paymentMethod,
  IVPAYD paymentDate,
  IVCRECTYPE invoiceAppearance,
  IVSSYS "SOURCE",
  IVPMDA invoiceDeliveryDate,
  IVCURC currency,
  IV74UNT invoiceNetAmount,
  IV74UFNT invoiceNetAmountHUF,
  IVTXAM invoiceVatAmount,
  IVVSTTA invoiceVatAmountHUF,
  IVC75TRNO transactionId,
  IVNDET "INDEX",
  IVBUGDTTM insDate,
  IVB76CIB cashAccountingIndicator,
  IVHT01 individualExemption,
  IVICBI smallBusinessIndicator,
  IVYCRR exchangeRate,
  IVACTIVID detailQueryUUID,
  IVODSEQTM detailQueryTS
FROM F55IR001;



CREATE SEQUENCE INVOICES_SEQ START WITH 1;
CREATE OR REPLACE TRIGGER INVOICES_BIR
                  BEFORE INSERT ON F55IR001
                  FOR EACH ROW

                  BEGIN
                    SELECT INVOICES_SEQ.NEXTVAL
                    INTO   :new.IVUKID
                    FROM   dual;
                  END;
                  /


CREATE OR REPLACE VIEW invoicevat AS SELECT
  VAUKID id,
  VAPUKID invoiceid,
  VANLIN vatlineid,
  VACMXTT vattype,
  VAACPE vatpct,
  VAATXA vatRateNetAmount,
```

```
  VATXAM vatRateVatAmount,
  VATXAG vatRateGrossAmount
FROM F55IR002 ;



CREATE SEQUENCE INVOICEVAT_SEQ START WITH 1;
CREATE OR REPLACE TRIGGER INVOICEVAT_BIR
                  BEFORE INSERT ON F55IR002
                  FOR EACH ROW

                  BEGIN
                    SELECT INVOICEVAT_SEQ.NEXTVAL
                    INTO   :new.VAUKID
                    FROM   dual;
                  END;
                  /


CREATE OR REPLACE VIEW invoicelines AS SELECT
  ILUKID id,
  ILPUKID pid,
  ILALIN lineNumber,
  ILD200 lineDescription,
  ILQTY quantity,
  ILPUOMDES unitOfMeasure,
  ILLIST unitPrice,
  IL74UNT lineNetAmount,
  IL74UFNT lineNetAmountHUF,
  ILTXAM lineVatAmount,
  ILVSTTA lineVatAmountHUF,
  IL74UCG lineGrossAmount,
  IL74UFCG lineGrossAmountHUF,
  ILADV     advanceIndicator,
  ILCMXTT vatPercentage
FROM F55IR003;

CREATE SEQUENCE INVOICELINE_SEQ START WITH 1;
CREATE OR REPLACE TRIGGER INVOICELINE_BIR
                  BEFORE INSERT ON F55IR003
                  FOR EACH ROW

                  BEGIN
                    SELECT INVOICELINE_SEQ.NEXTVAL
                    INTO   :new.ILUKID
                    FROM   dual;
                  END;
                  /

COMMIT;
```