



CXBanking

OptiVLM Invoice Validation 10.0

Installation Guide

Build 7014
Revised March 2023

Copyright and Trademark Information

The products described in this document are copyrighted works of NCR Corporation.

NCR and APTRA are trademarks of NCR Corporation.

Microsoft and Windows are registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

All other trademarks are the property of their respective owners.

It is the policy of NCR Corporation (NCR) to improve products as new technology, components, software, and firmware become available. NCR, therefore, reserves the right to change specifications without prior notice.

All features, functions, and operations described herein may not be marketed by NCR in all parts of the world. In some instances, photographs are of equipment prototypes. Therefore, before using this document, consult with your NCR representative or NCR office for information that is applicable and current.

© 2020 NCR Corporation

Atlanta

Georgia

USA

www.ncr.com

All Rights Reserved

Revision Record

Date	Page No.	Description of Change
March 2023	ALL	Formatting Changes

Contents

Copyright and Trademark Information	2
Revision Record.....	3
Contents.....	4
1 Introduction	6
2 Application Distribution.....	7
2.1 Application Component Checklist	7
3 Oracle Setup.....	8
3.1 Configuration	8
3.2 Tablespaces	8
3.3 Schema User	9
3.4 Schema Definition	9
4 SQL Server Setup	10
4.1 Configuration	10
4.2 Schema User	10
4.3 Schema Definition	10
5 JDK.....	12
5.1 Deployment Scenarios.....	12
5.1.1 Single Application / Database Server Scenario	12
5.1.2 Split Application / Database Servers Scenario	12
6 Application Server.....	13
6.1 General	13
7 OptiVLM Invoice Validation Deployment (Application Server)	14
7.1 WAR File Deployment.....	14
OptiVLM Invoice Validation Deployment (Using WebLogic Server)	15
WAR File Deployment.....	15
OptiVLM Invoice Validation System Configuration	16
8 OptiVLM Invoice Validation Licensing.....	24
9 OptiVLM Invoice Validation Database Upgrade.....	25
10 OptiVLM Invoice Validation Customization	26
10.1 Making Changes to the Language File	26
10.2 Making Changes to the Styles and Logos	27
11 Redeploying the Oracle Schema.....	28

12	Batch Processes.....	29
13	IMPORTANT NOTE	32
14	EPSS Integration	33

1 Introduction

The purpose of this document is to provide basic installation instructions for the OptiVLM Invoice Validation application. The particular architecture and environment of the installation may require additional configuration.

Although the installation media provides components for both Windows and UNIX, this document will provide samples and syntax based on the Windows operating system. The main modification for UNIX will be the file path name syntax.

Invoice Validation requires an active installation of OptiCash and OptiVault, it is assumed that the database setup steps have already been performed for these applications. All that will be needed is to provide the Invoice Validation application with login credentials to these databases.

Invoice Validation requires a base version 8 of Java Development Kit (JDK). Most Application Servers already come with the required JDK (e.g. IBM WebSphere or Apache Tomcat).

A clear understanding of Oracle and Application Server technology is required on the part of the user performing the installation.

2 Application Distribution

2.1 Application Component Checklist

Depending upon the client environment, the Application Server and Oracle Database Server could reside on the same physical machine or different machines. The exact nature of this configuration should be agreed upon between NCR Cash Management and the client prior to installation. It is required that JDBC access is available between the Application Server and Oracle Database Server (as defined by the JDBC URL, which typically runs through port 1521).

In a split-server example, the Application Server (e.g. IBM WebSphere or Apache Tomcat) would house the Web Components (e.g. Invoice Validation WAR file) on one machine, and a different machine will house the Oracle components.

The following are the required components for the installation:

- **Oracle 12.2 or 19c and the latest patches relevant to the applicable O/S:** It is the responsibility of the client to ensure the Oracle database is running correctly and readily accessible PRIOR to the on-site installation.
- Running versions of OptiCash and OptiVault applications
- **Java Application servers, such as IBM WebSphere or Apache Tomcat:** It is the responsibility of the client to ensure the Application Server is running correctly and readily accessible PRIOR to the on-site installation.

Note: Java 8 JDK is required.

- **VLM Invoice Validation WAR File:** NCR Cash Management will provide the WAR file for deployment.
- **License File:** NCR Cash Management will provide a license SQL file. Certain technical information regarding the installation is required to create the license file, so a common scenario is to install without a license and then attempt to log in. The license failure message will provide the information i.e., to be sent to NCR Cash Management for license creation.

3 Oracle Setup

Note: It is the client's responsibility to have Oracle installed and running correctly and readily accessible prior to the on-site installation performed by NCR Cash Management.

Additionally, it is the client's responsibility to understand and agree with NCR Cash Management on the information contained within the Technical Overview (a separate document). That shall serve as a basis for architectural consideration.

3.1 Configuration

Verify the Oracle memory settings are correctly defined and do not fall below the minimum memory requirements necessary for running OptiVLM Invoice Validation (refer to the Oracle Installation document for more information on minimum memory requirements). kindly coordinate with Oracle System Administrator to ensure that memory settings are also taken into consideration with other databases used in the same Oracle environment.

3.2 Tablespaces

The following assumes and steps are performed using the '**sqlplus**' command line interface. Similar actions may be performed with the tool and interface per DBA choice.

VLM Invoice Validation defines the following default tablespace name for data and index respectively, it is recommended to use these names exactly:

- Data tablespace: OPTIVLM_IV_DAT
- Index tablespace: OPTIVLM_IV_IDX

The following steps detail how to create the above tablespace:

From the Windows command prompt, type '**sqlplus**' and enter username/password as prompted.

Modify the following example to fit your needs:

```
CREATE BIGFILE TABLESPACE OPTIVLM_IV_DAT DATAFILE
'C:\ORACLE\APP\MYCOMPUTER\ORADATA\ORCL\OPTIVLM_IV_DAT01.DBF' SIZE 100M REUSE
AUTOEXTEND ON NEXT 100M MAXSIZE 5060M LOGGING EXTENT MANAGEMENT LOCAL SEGMENT
SPACE MANAGEMENT AUTO;

CREATE BIGFILE TABLESPACE OPTIVLM_IV_IDX DATAFILE
'C:\ORACLE\APP\MYCOMPUTER\ORADATA\ORCL\OPTIVLM_IV_IDX01.DBF' SIZE 100M REUSE
AUTOEXTEND ON NEXT 100M MAXSIZE 5060M LOGGING EXTENT MANAGEMENT LOCAL SEGMENT
SPACE MANAGEMENT AUTO;
```

Where

- 'C:\ORACLE\APP\MYCOMPUTER\ORADATA\ORCL\OPTIVLM_IV_DAT01.DBF' is your directory and filename.
- "OPTIVLM_IV_DAT" and "OPTIVLM_IV_IDX" are your tablespace names.

Note: The OptiVLM Invoice Validation Schema Definition Script (DDL) will have tables and constraints defined such that:

1. Tables, Foreign Keys, and Views are defined in the Schema User's default tablespace. (e.g. OPTIVLM_IV_DAT).

Primary Keys and Indexes are defined in the OPTIVLM_IV_IDX tablespace.

Users may modify the DDL prior to execution as needed.

3.3 Schema User

Create a **"Schema User"** using commands similar to the example below:

```
CREATE USER "OPTIVLMIV" PROFILE "DEFAULT" IDENTIFIED BY "OPTIVLMIV" DEFAULT
TABLESPACE "OPTIVLM_IV_DAT" TEMPORARY TABLESPACE "TEMP" ACCOUNT UNLOCK;
GRANT UNLIMITED TABLESPACE TO "OPTIVLMIV";
GRANT "CONNECT" TO "OPTIVLMIV";
GRANT "RESOURCE" TO "OPTIVLMIV";
GRANT CREATE ANY VIEW TO "OPTIVLMIV";
```

3.4 Schema Definition

Oracle schema should be created using the **'Schema User'** created in the prior section, using one of two sources:

- Oracle Schema Data Dump as provided by NCR Cash Management.
 - e.g. <client name>.dmp
- DDL script to define tables, indexes, and default data records.
 - e.g.: OptiVLM-IV-<build_number>.sql

Create the data schema in one of the following ways, respective to the items from above:

Run a data pump import, which will load the contents of the provided data pump file into the target tablespaces and schema name.

It is strongly recommended to analyze the tables and indexes for the newly imported data immediately after import.

4 SQL Server Setup

4.1 Configuration

Verify the SQL Server memory settings are correctly defined and do not fall below the minimum memory requirements necessary for running OptiSuite (refer to *SQL Server Installation* document for more information on minimum memory requirements). Kindly coordinate with System Administrator to ensure that memory settings also take into consideration other databases used by the bank in the SQL Server environment.

4.2 Schema User

1. Choose a name similar to the schema you want to create, e.g. InvoiceValidation, client name, etc. and password accordingly in the General tab

Select “**public**” and “**dbcreator**” from the **Role list** for the user. The user will need these roles to connect to the database and access database functions.

Select the respective database in the user mapping tab for particular users. Select **Grant permission** to connect to the database engine and login enabled in the Status tab.

4.3 Schema Definition

The following files may be used to create a new schema based on the new user created in the previous step:

- Minimum of 2 DDL scripts to define tables, indexes, and default data records.
 - e.g.: invoice-validation_schema/data.sql

NCR Cash Management will provide these files to the client. Make sure to save these files in the same directory.

Users will have two options to create the data schema:

1. Run the provided DDL SQL files.
 - a. You will first need to create the schema objects, per the invoice-validation_schema.sql script. This script can be modified as needed to reflect the desired tables, etc.
 - b. Once completed, verify that there were no errors in the process, and the appropriate objects are created within the user schema. followed by running the invoice-validation_data.sql script to add default data records to the created tables and invoice-validation_storedprocedures.sql for stored procedures.

It is strongly recommended to analyze the tables and indexes for the newly imported data immediately after import.

or:

DDL SQL script comes from the **Database Upgrade** process as described later in this document. Skip past this step for now and run the DDL scripts after hooking up the OptiVLM InvoiceValidation application – including authentication – and attempting to log in one time (the system generates the script at that point).

5 JDK

Most Application Servers (e.g. IBM WebSphere, Apache Tomcat) already come with the JDK required for running the application server. The OptiVLM Invoice Validation application requires version 8 JDK.

Note: The Runtime equivalent (JRE) is not sufficient since runtime compiling is required by the web application.

This document does not detail the installation of the JDK for a batch server, or database server nor the installation of the Application Server (e.g. WebSphere), since these are identified as a client's direct responsibility.

The following provides a brief overview of specific scenarios that might be used:

5.1 Deployment Scenarios

5.1.1 Single Application / Database Server Scenario

The critical thing in this scenario is simply to ensure an appropriate Application Server & Oracle/SQL Server have been installed on the machine and are ready for use.

5.1.2 Split Application / Database Servers Scenario

In many client installations, the Application Server (e.g. WebSphere) and Database Server reside on different machines. In this example, WebSphere will house the Web Component (e.g. OptiVLM Invoice Validation WAR file) on one machine and another machine would house the database.

6 Application Server

6.1 General

Note: It is the responsibility of the client to ensure the Application Server is running correctly and readily accessible PRIOR to the on-site installation to be performed by NCR Cash Management.

Additionally, the Application Server should be installed in a directory structure without spaces, e.g. C:\IBM\WebSphere.

It is also strongly recommended that the application server be deployed in the root directory (or close to the root directory).

7 OptiVLM Invoice Validation Deployment (Application Server)

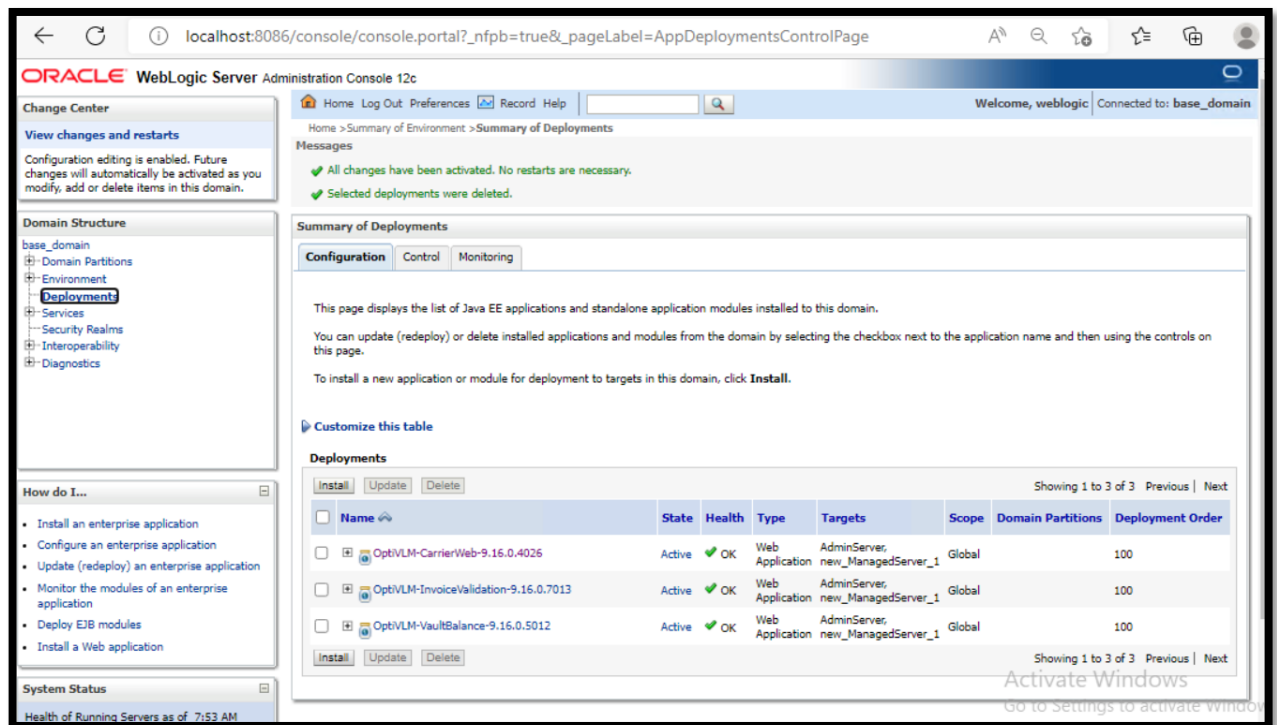
7.1 WAR File Deployment

1. WAR files are readily deployable web application containers, complete with supporting jars.
2. The default deployment of OptiVLM Invoice Validation will be to a web application and URL called **"OptiVLM-InvoiceValidation"**. Many Application Servers (WebSphere, etc.) provide an application assembly tool to allow you to change the default context prior to deployment. Refer to the specifics of your Application Server.
3. The precise deployment technique for the WAR file installation depends on the Application Server chosen, e.g. WebSphere, Sun Java, etc.
 - 3.1 If deploying on WebSphere, it is strongly recommended to enable the option **"Show me all installation options and parameters"**, unless the user is comfortable using a custom deployment script specific to their institution.
 - 3.2 If deploying on WebSphere, the global security option **"Enable Application Security"** is required to be turned on. If turning this on for the first time, a WebSphere restart is required.
4. OptiVLM Invoice Validation contains security roles within the WAR file (in web.xml file), which permits you to match authenticated users to application access.
 - 4.1 ROLE_USER (the typical user who logs into the OptiVLM Invoice Validation application)
 - 4.2 ROLE_ADMIN (the user with full control of the OptiVLM Invoice Validation application)
5. You may choose to give any user access to these URLs of the application, and if so simply enable the applicable options in accordance with your application server.
 - 5.1 **Example:** In the case of WebSphere, this is handled in the **"Map security roles to users or groups"** step of WAR deployment, where you can Look Up Users or Groups based on your applicable access directory plug-in. You can also grant these two roles to **"Everyone"**.
6. If applicable, make sure to save the configuration post-deployment.
7. If applicable, make sure to regenerate the plug-in configuration for proper communication between the **Web Server** and the **Application Server**.

OptiVLM Invoice Validation Deployment (Using WebLogic Server)

WAR File Deployment

- After logging in to the WebLogic console using valid credentials which were set at the time of installation in the left side pane under **Domain structure** click on **Deployments**
- Click on the **Install** button under the **Configuration** tab an **Install Application Assistant** will be displayed using this the required .war file can be imported into WebLogic
- Select the installation type as an **application** on the next page of Install Application Assistant
- Select the available target types like admin server and managed servers (if any) depending on the requirement i.e in how many servers the application should be deployed
- Click on the **Finish** button to finish the setup, WebLogic will automatically display the deployed application in the dashboard with the status of the deployment like **Active** for successful and **Errors** if the deployment failed
- Below is the dashboard view of WebLogic



OptiVLM Invoice Validation System Configuration

After the WAR file is deployed following the instructions in the previous steps, the following will indicate additional configurations that need to be defined.

1. **Data source installation:** OptiVLM Invoice Validation supports two types of data source connection: JDBC data source connection and JNDI connection.
 - JDBC is the default and is used for a simple connection.
 - JNDI is the more secure and recommended method. The following section will cover each separately (section 1.1 for JDBC data source connection configuration and section 1.2 for JNDI data source connection configuration):
- 1.1. JDBC connection is the default data source connection method. It requires configuration updates to multiple files under the OptiVLM Invoice Validation deployment folder:
 - 1.1.1. OptiVLM Invoice Validation uses an InvoiceValidation.properties file located at <application-path>\WEB-INF\classes to specify a connection to the database. You will need to update the **"URL"**, **"username"**, **"schema"** and **"password"** of each to match your environment.

Here is an oracle example:

```
database.driverClassName=oracle.jdbc.driver.OracleDriver
database.IV.url=jdbc\:oracle\:thin\:@server\:1521\:sid
database.IV.username=(username)
database.IV.password=(password)
database.IV.schema=(schemaname)
database.OC.url=jdbc\:oracle\:thin\:@server\:1521\:sid
database.OC.username=(username)
database.OC.password=(password)
database.OC.schema=(schemaname)
database.OV.url=jdbc\:oracle\:thin\:@server\:1521\:sid
database.OV.username=(username)
database.OV.password=(password)
database.OC.schema=(schemaname)
```

Here is an example of an SQL server:

```
database.driverClassName= com.microsoft.sqlserver.jdbc.SQLServerDriver
database.IV.url=jdbc:sqlserver://@server;databaseName=@schemaName
database.IV.username=(username)
database.IV.password=(password)
database.OC.url= jdbc:sqlserver://@server;databaseName=@schemaName
database.OC.username=(username)
database.OC.password=(password)
database.OV.url= jdbc:sqlserver://@server;databaseName=@schemaName
database.OV.username=(username)
database.OV.password=(password)
database.IV.schema=(schemaname)
```



```
database.OC.schema=(schemaname)
database.OV.schema=(schemaname)
```

The file contains three sets of login credentials, where the first corresponds to the OptiVLM Invoice Validation schema, the second corresponds to the OptiCash schema, and the third corresponds to the OptiVault schema. Only one set of database properties can be defined in the properties file either Oracle or SQL server. When oracle is being used SQL server properties must be commented on and vice versa.

1.1.2. Note: If the current application server has been configured to use JNDI data sources (for other applications, or prior use with OptiVLM Invoice Validation), then it is possible to have a conflict with JDBC. Check applicationContext.xml (under <application-path>\WEB-INF\classes\META-INF\spring) and be sure that the "jee:jndi-lookup" element's "jndi-name" does NOT match your JNDI data source names.

1.2. To run OptiVLM Invoice Validation with a JNDI connection the configuration could vary depending on what kind of application server will run the application and also relying on the application server environment setup. The following section gives an example of JNDI data source configuration on Websphere 7 or 8, but the actual configuration may vary:

1.2.1. Log on to Websphere Application Server Console as administrator, go to - >Security->Global Security on the left panel and expand "**Java Authentication and Authorization Service**"(JAAS) on the right panel, click "**J2C authentication data**". Click "**new**" and type in the value to each required field explained as follows to create a new JAAS object.

Alias:	The name for the current JAAS connection appears as a reference on the Application Server scope (recommend using any string that may suggest the schema name, i.e. "profile_IV")
Username:	The username of the database schema
Password:	The password of the database schema

Then click "**OK**" to save, and repeat the above process for each database schema access (Invoice Validation, OptiCash and OptiVault).

1.2.2. On the **Websphere Application Console** home page, go to *Resource->JDBC->JDBC provider* on the left panel, click "**New..**" and create a new Oracle JDBC provider, please specify the proper scope, classpath of the ojdbc driver file and proper implementation class name ("**oracle.jdbc.pool.OracleConnectionPoolDataSource**" is recommended but not absolute, depends on production environment configuration). Click "**OK**" to save the JDBC provider.

1.2.3. Still, on the “**JDBC Provider**” page, click the “**Data sources**” link and click “**New..**” to create the data source for each schema:

1.2.3.1. On the next page, specify the JNDI name.

Note: The JNDI name defined here **MUST MATCH** the exact name specified in <application-path>\WEB-INF\classes\InvoiceValidation.properties in the field “database.IV.jndi.datasourceName”.

1.2.3.2. Fill in the “**URL**” field with the current database schema, i.e.
jdbc:oracle:thin:@server:1521:serverdb

1.2.3.3. On the next page, select the proper JAAS alias which has been setup on 1.2.1 for “**Component-managed authentication alias**” and choose “**WSLogin**” for “**Mapping-configuration alias**”, then click “**Next**” and “**Finish**”

1.2.3.4. Repeat 1.2.3.1 to 1.2.3.3 to create JNDI data source for OptiVLM Invoice Validation, OptiCash, and OptiVault respectively. Go back to the “**Data Sources**” page, click every data source created and click the “**Test Connection**” button. If everything is properly configured, there should be a test successful message.

Note: WebSphere restart may be required for these changes to take effect.

2. Update the <application-path>/WEB-INF/classes/log4j.properties file to point the OptiVLM-Val.log to the desired location. Log4j is a Java logging utility that logs the usage activity in the application.

Note: This should be the full path to the log file.

2.1. There are two provided methods for logging in to the log4j.properties file:

2.1.1. logging to the standard out file known as the **Console Appender** or **stdout**, and

2.1.2. logging to a separate log file known as a **Rolling File Appender** or **ROL**.

2.2. Typically, most installations will use the second option (VLM logging to a separate log file as opposed to the application server default out log). The first option is the default, which results in a standard log4j.properties file that can run “out-of-the-box” by logging into the standard out. The default is for WARN (warning-level) logging to the standard out of the JVM. Typically, this would be the default application server (e.g. WebSphere) log file. The second option is for logging into a separate log file.

Note: The file and directory must exist and be accessible for writing from the application. You can use either or both options, although it is recommended to use the second option (ROL) to have a separate log file for VLM not muddled with other Application Server messages.

2.3. Simply comment out the appropriate fields as desired with a preceding hash (#) symbol to comment out a line.

Note: The file name and full path to the log file should be used, assuming a ROL log file.

2.4. **Note:** Each time changes are made to any of the property files users need to restart OptiVLM Invoice Validation within the Application Server so that the changes are applied. (Reconfigure the plug-in, as well, between your App Server and Web Server, if necessary.)

3. Users may need to update the licensing function to reflect your institution's name. The institution's name must match the name listed in the license.

Note: Not necessary if already included in a provided seed database.

- 3.1. The institution name is saved in the OptiVLM Invoice Validation system setting table. kindly use the following SQL query to update the institution name:

update application setting set value = '{INSTITUTION_NAME}' where name='licensing_institution_name';

Here {INSTITUTION_NAME} is the placeholder for desired institution name.

4. Authentication can be configured in one of two possible methods: Server-based authentication, and OptiCore authentication.

4.1. **Server-based authentication** leaves the authentication process to the application server. This authentication mode is recommended by NCR because the application server provides efficient and powerful security support.

4.2. **OptiCore authentication** is an in-house Single Sign On (SSO) solution from OptiSuite and OptiVLM so that all applications authenticate the user via the OptiCore authentication library. This mode requires customized java code to meet customers' specific requirements. User credentials should be provided by the client's system/source

* To comment XML configuration file, use "<!--" and "-->" to include the section desired, uncomment is to remove the "<!--" and "-->"

Authentication Method	Process to install
Server-based authentication	<ol style="list-style-type: none"> 1. Map roles "ROLE_USER" and "ROLE_ADMIN" in the app server for the users who will log into Invoice Validation respectively. 2. In <application-path>/WEB-INF/classes/InvoiceValidation.properties file, set authentication.type=server 3. In <application-path>/WEB-INF/web.xml file, ensure that the section under "<!-- Container-managed authentication -->" is not commented out
OptiCore authentication	<ol style="list-style-type: none"> 1. Configure external user profile and access defined by OptiCore.

	<ol style="list-style-type: none"> In <application-path>/WEB-INF/classes/InvoiceValidation.properties file, set authentication.type=opticore In <application-path>/WEB-INF/web.xml file, comment out the section under “<!-- Container-managed authentication -->” In <application-path>/WEB-INF/classes/InvoiceValidation.properties, define the mapping between the external role name and InvoiceValidation internal role name(ROLE_USER, ROLE_ADMIN), also assign the correct value for the external role and InvoiceValidation user group name mapping. See the following item “Explanation of major parameters under <application-path>/WEB-INF/classes/InvoiceValidation.properties”
DBAuthorizer variant to OptiCore authentication	<ol style="list-style-type: none"> Apply the above steps for OptiCore authentication. Ensure that “transoft.samples.DBAuthorizer” is the value used in InvoiceValidation.properties for opticore.auth.authorizer.class Set opticore.auth.init.params in InvoiceValidation.properties to point to DBAuthorizer’s additional properties file. By default, this is ../WEB-INF/classes/DBAuthorizer-OC.properties. This default file may be a useful example, even if you choose to use another file elsewhere. Inside the DBAuthorizer properties file, set the following as appropriate for your environment. Note: You may see some lines in the file which are not listed here (query definitions, etc) – these should be kept as provided by default. <p>dbAuthorizer.database.dsn: JNDI database connection name. Similar to section 1.2 above, this is a database connection. If desiring to use a JDBC connection instead, then this field should be empty.</p> <p>dbAuthorizer.database.url: URL to authorization database, example: jdbc:oracle:thin:@server:1521:serverdb</p> <p>dbAuthorizer.database.username, dbAuthorizer.database.password: Schema username and password if using a JDBC connection. Can be blank if using a JNDI connection.</p> <p>dbAuthorizer.digest.length: Length of password after encryption. Longer is more secure, but cannot exceed the maximum size of your target database’s password field.</p> <p>dbAuthorizer.digest.algorithm: Algorithm to be used when encrypting a password.</p> <p>dbAuthorizer.digest.seed: Character string used to seed the encryption algorithm. It is recommended to change this away from the default.</p> <p>dbAuthorizer.option.quiet: “true” or “false”. Quiet mode allows a password that was unencrypted to be encrypted without the user having to change that password. If false, then users are forced to change the password in that</p>

	<p>situation. This can be relevant when migrating from another auth method or in a case where the administrator previously reset a user's password.</p> <p>Note: If you plan to use DBAuthorizer from multiple OptiSuite applications and point to the same single database for user management, then the digest "length", "algorithm", and "seed" settings above must be the same between all OptiSuite applications.</p> <p>Note: If switching from one form of password encryption to another, then you will need to manually set at least one administrator user's password temporarily. All users' passwords will need to be reset, but doing only one initially allows that user to change for others inside OptiVLM Vault Balance application. Update as follows using SQL where THE_PASSWORD and THE_USER are replaced with your values:</p> <p><i>update users set password='THE_PASSWORD' where username='THE_USER';</i></p>
--	--

5. Update <application-path>/WEB-INF/classes/InvoiceValidation.properties for various settings.

Property name	Comments
database.upgrade.outputFile	Filename where generated database upgrade script will appear. Example: C:\Users\OptiSuite\InvoiceValidation\db-upgrade.sql
database.IV.tablespace.index database.IV.tablespace.data	These define the index tablespace name and data tablespace name respectively. Both are included when generating database upgrade scripts. If not found, the system uses the schema name to create a script without tablespace names (which is not recommended).
systemDate	For demo or test environments only, define static system date.
pageSizes defaultPageSize	Defines the options of pagination on every listing page in InvoiceValidation. i.e. pageSizes=5,10,25,100 defaultPageSize=10
authentication.type	Define what authentication method and API is used to run Invoice Validation "server": Currently Invoice Validation is set with application server base authentication.

	<p>"opticore": Currently Invoice Validation is set with opticore authentication</p>
externalRole.{EXTERNAL_ROLENAME}	<p>Only for OptiCore authentication. {EXTERNAL_ROLENAME} is a placeholder for a role name from an external system (usually a customer authentication system, i.e. Microsoft Active Directory). The value of this properties should be one or more Invoice Validation internal ROLES, i.e.</p> <pre>externalRole.SystemAdmin=ROLE_USER,ROLE_ADMIN externalRole.User=ROLE_USER externalRole.UserAdmin=ROLE_ADMIN</pre>
<p>opticore.auth.authorizer.class=transsoft.samples.FileAuthorizer</p> <p>opticore.auth.app.name=InvoiceValidation</p> <p>opticore.auth.init.params=FileAuthorizer</p> <p>opticore.auth.bundle.file.name=InvoiceValidation</p> <p>or</p> <p>opticore.auth.authorizer.class=transsoft.samples.DBAuthorizer</p> <p>opticore.auth.app.name=IV</p> <p>opticore.auth.init.params=DBAuthorizer-OC</p> <p>opticore.auth.bundle.file.name=InvoiceValidation</p> <p>or customized.</p>	<p>Only for OptiCore authentication.</p> <p>These 4 properties defined the values for the placeholders in contextSecurity-opticore.xml</p> <p>opticore.auth.authorizer.class: Defines OptiCore authentication authorizer class name</p> <p>opticore.auth.app.name: Defines the current application name which may be used by the authorizer, in case there is a specific logic process regarding the current application.</p> <p>opticore.auth.init.params: Defines initial parameters for an authentication process. It is a list of items in String format</p> <p>opticore.auth.bundle.file.name: Defines the name of the properties file that is holding external information. i.e. InvoiceValidation.properties</p>
<p>available.locales</p> <p>default.locale</p>	<p>For language settings. available.locales define the list of locales that are available on the UI for the user to choose, default.locale is the default language once Invoice Validation starts</p>

	<p>i.e. <i>available.locales=es,en,th_TH</i> (there are three available locales in current settings)</p> <p><i>default.locale=en</i> (English is the default locale)</p>
user.need.sync	<p>Only for OptiCore authentication. Defines whether Invoice Validation should synchronize user profile (cashpoints, username, etc.) after optiCore authentication is complete.</p>
defaultCsvDelimiter	<p>The character is used as a delimiter during the Invoice Import function. Example: <i>defaultCsvDelimiter=,</i></p>
logs.path	<p>Location on the server where log files are expected to be found.</p> <p>(Note: May be same as directory in section 2 above.)</p> <p>This will be used by the user interface page which provides links for the user to read log files.</p>

8 OptiVLM Invoice Validation Licensing

After the initial installation, the application will show an '**exception**' screen if the license is not installed.

Copy the exception message and provide it to NCR Cash Management for license generation. In response, NCR Cash Management will send back an SQL query containing the license for the application. This query needs to be run on the OptiVLM Invoice Validation database.

Here's a sample of what you might see:

```
License not available: Today=2011-09-30 iName=demo dbURL=jdbc:oracle:thin:@jdnb37-  
pc:1521:orcl?user=vlm appPath=c:\java...
```

▲ **Warning:** License application should only be carried out by a system administrator or database administrator as incorrect use of the SQL application could cause corruption of the data.

9 OptiVLM Invoice Validation Database Upgrade

If OptiVLM Invoice Validation requires a database schema change, there will be an SQL upgrade script produced in the database upgrade directory (defined earlier in <application-path>/WEB-INF/classes/InvoiceValidation.properties file). This will be in the form of SQL queries. kindly run all queries in the file using SQLplus or an SQL tool of your choice. Remember to commit changes.

For the first time to install Invoice Validation with the SQL script generated, kindly add the following query and run it before executing any queries from the script:

```
SET define off;
```

▲ Warning: This should only be carried out by a system administrator or database administrator as incorrect use of the SQL application could cause corruption of the data.

10 OptiVLM Invoice Validation Customization

10.1 Making Changes to the Language File

Language files may be updated for translation purposes, or to simply modify text within OptiVLM Invoice Validation to better suit client institution preferences. These are simply displayed text changes and do not modify system behaviour.

The following files are involved for English. For other languages, similar files are used, but the “_en” suffix will instead reflect that specific language.

- <application-path>/WEB-INF/i18n/custom_en.properties
 - <application-path>/WEB-INF/i18n/messages_en.properties
 - <application-path>/WEB-INF/i18n/application_en.properties
1. Customized elements are recommended to go in the “**custom_**.properties**” file. This allows to override system text without deleting (and thus easily reversible if desired). Elements not found in the “**custom_**.properties**” file will be sourced from the system-provided files instead.
 2. To change a text element, copy it from the relevant system file (“**messages_**.properties**” or “**application_**.properties**”) into “**custom_**.properties**”.

Example:

```
menu_item_invoice_generate_label=Generate
```

can change to

```
menu_item_invoice_generate_label=Create
```

This effectively changes the text “**Generate**” that would appear in Invoice Validation to “**Create**” instead.

Note: Changes should be made only to the right of the “=” character and never to the left of “=”.

3. After changing the file, save the file and restart the OptiVLM Invoice Validation application.



Customized language files are the client’s responsibility to maintain. Before editing, save the original file (i.e. messages.orig.properties). In the future, with each OptiVLM Invoice Validation upgrade, the edited version will need to be saved PRIOR to installing a new WAR file. Once the WAR file has been deployed, then restore the edited version.

It is the client's responsibility to maintain non-English language files. Some non-English sample language files are included with OptiVLM-InvoiceValidation – these are often incomplete and are not checked for the correctness of the translation. The translation should be based on the provided English version.

10.2 Making Changes to the Styles and Logos

Virtually all images are stored in the <application-path>/images directory.

1. The main OptiVLM Invoice Validation style sheet is found at <application-path>/styles/standard.css

Customized styles and logos are the client's responsibility to maintain. Before editing, save the original style sheet and image files. In the future, with each OptiVLM Invoice Validation upgrade, the edited version will need to be saved PRIOR to installing a new WAR file. Once the WAR file has been deployed, then restore the edited version.

11 Redeploying the Oracle Schema


It is recommended to have nightly backups of the OptiVLM Invoice Validation Oracle schema.

Users may back up the data using replication, a simple Oracle EXP or EXPDP command, or other tools of choice for the DBA.

You would re-load that data following the schema user created and import methods outlined earlier in the installation guide.

12 Batch Processes

The purpose of batch process execution is to provide time-efficient execution of regular OptiVLM processes. The jobs can be scheduled to run overnight.

	<p>Note: NCR Cash Management does not provide ongoing support for the batch files due to integration requirements to existing customer systems, specific customer procedures in relation to the data file interfaces, a variety of scheduling capabilities and security concerns. Therefore, it is the responsibility of the client to review, understand and support these batch files.</p>
---	---

The file “setup-OptiVLM-Val-Jobs.exe” provided by NCR Cash Management is a self-extracting archive, which contains the “**jobs**” directory. Inside that directory, there is a JAR file “OptiVLM-Val-jobs.jar”. It will need to be configured for your database details. Use a JAR file editor (such as 7zip) to update the usual files inside the JAR:

- InvoiceValidation.properties (same as earlier, especially database connection)
- log4j.properties (same as earlier, setup logging)

Once those are updated, there are provided command files to run the three jobs, **InvoiceImportJob**, **InvoiceGenerateJob** and **ReconciliationJob**. For each job, there are three command files, one that just displays help for parameters, another to run a limited test, and a third that you can use as a template for a production job.

Note: These are very simple command scripts. Here is the production import template:

(InvoiceImportJob.cmd) contents:

```
set JAVA_HOME=C:\java\jdk1.6.0_24
%JAVA_HOME%\bin\java -jar OptiVLM-Val-jobs.jar InvoiceImportJob T01F01
samples\BFAVV1103.txt
pause
```

Here, the first line explicitly selects the JDK. This is not necessary but highly recommended. Without this, you will use whichever JDK was last installed. The second line executes the job using the explicitly selected JDK. The last line just pauses, so you can see the results when you run it by double-clicking the file from Windows Explorer. The minimum command line is actually:

```
java -jar OptiVLM-Val-jobs.jar InvoiceImportJob T01F01
samples\BFAVV1103.txt
```

For convenience, here are the parameter definitions for each job. Square brackets ([]) indicate optional parameters.

java -jar OptiVLM-Val-jobs.jar InvoiceImportJob formatCode fileName [firstDate [lastDate]]

```
InvoiceImportJob = Literal value (case sensitive).
formatCode = The invoice format code for batch.
fileName = The import source filename.
firstDate = (YYYY-MM-DD) First date to import.
lastDate = (YYYY-MM-DD) Last date to import.
```

java -jar OptiVLM-Val-jobs.jar InvoiceGenerateJob contractCode [firstDate [lastDate]]

```
InvoiceGenerateJob = Literal value (case sensitive).
contractCode = The contract rate code for batch.
firstDate = (YYYY-MM-DD) First date to import.
lastDate = (YYYY-MM-DD) Last date to import.
```

java -jar OptiVLM-Val-jobs.jar ReconciliationJob generatedInvoiceId importedInvoiceId validationType [itemType [cashpointType [fundingSourceType [firstDate [lastDate [strongMatchOption [strongMatchPercentage[strongMatchRate]

```
ReconciliationJob = Literal value (case sensitive).
generatedInvoiceId = ID of the generated invoice.
importedInvoiceId = ID of the imported invoice.
validationType = IGNORE, RECONCILESERVICES, RECONCILEFEES
itemType = ANY, TRANSPORTITEM, PROCESSINGITEM, HOLDINGITEM
cashpointType = ANY, BRANCH, VAULT, COMMERCIALCUSTOMER
fundingSourceType = ANY, CENTRALBANK, VAULT, OTHERSOURCE,
EXTERNALSOURCE
firstDate = (YYYY-MM-DD) First date to reconcile.
lastDate = (YYYY-MM-DD) Last date to reconcile.
strongMatchOption = PERCENTAGE, RATE
strongMatchPercentage = valid number between 0 and 100 for Strong
Match Allowable Variance
strongMatchRate = Valid number for Strong Match Allowable Amount
```

java -jar OptiVLM-Val-jobs.jar MonthlyJob contractCode formatCode fileName validationType [itemType [cashpointType [fundingSourceType [firstDate [lastDate [strongMatchOption [strongMatchPercentage[strongMatchRate]

```
contract = The contract rate code for batch.
format = The invoice format code for batch.
importFile = The import source filename.
csvDelimiter = The symbol used to separate values in the imported
file.
```

```

validationType = IGNORE, RECONCILESERVICES, RECONCILEFEES
itemType = ANYITEM, TRANSPORTITEM, PROCESSINGITEM, HOLDINGITEM
cashpointType = ANY, BRANCH, VAULT, COMMERCIALCUSTOMER
fundingSourceType = ANY, CENTRALBANK, VAULT, OTHERSOURCE,
EXTERNALSOURCE
firstDate = (YYYY-MM-DD) First date to import, generate and reconcile.
lastDate = (YYYY-MM-DD) Last date to import, generate and reconcile.
strongMatchOption = PERCENTAGE, RATE
strongMatchPercentage = valid number between 0 and 100 for Strong
Match Allowable Variance
strongMatchRate = Valid number for Strong Match Allowable Amount

```

Each batch job serves a different purpose, as described below.

Job	Description
InvoiceGenerateJob	Generates an invoice based on the orders in OptiCash/OptiVault databases and the input from the user.
InvoiceImportJob	Imports a carrier invoice into the system. Receives a CSV file containing invoice data as input.
MonthlyJob	Generates an invoice for the previous month, imports an invoice supplied by the user, and finally, reconciles the two invoices.
ReconciliationJob	Reconciles two invoices (generated and imported). Requires the user to provide invoice IDs.

13IMPORTANT NOTE

To upgrade the application version to 9.16 with SQL Server, below are the steps.

1. Customers with older versions (<9.16) must be upgraded to 9.16 without changing the DB server (Oracle).
 - a. This step is taken care by the application if 9.16 war with oracle properties is deployed into the webserver.
 - b. After deployment, the user is prompted to upgrade DB Schema. This is also specific to OC and OV. For VLM, products the upgrade is taken care by Liquibase.
2. Create Database and schema in SQL Server as mentioned in installation.
3. Migrate data from Oracle to SQLServer(taken care by NCR Team).
4. Update Oracle DB properties with SQLServer Details in 9.16 war and deploy.

14EPSS Integration

- After 10.x, this application must be integrated with EPSS for authentication and authorization.
- User Groups for (ROLE_ADMIN, ROLE_USER) have to be created in EPSS.
- Please refer to *CM apps Installation and EPSS Integration guide10.0.pdf* for detailed steps.

CXBanking, OptiVLM Invoice Validation, Installation Guide
Revised March 2023

www.ncr.com

