



May 7, 2015

---

# INDIGO ELN INSTALLATION GUIDE

---

**VERSION 1.2**

---

## GLOBAL HEADQUARTERS

41 University Drive, Suite 202  
Newtown, PA 18940 USA

## EU HEADQUARTERS

Corvin Offices I. Futó Street 47-53  
Budapest, H-1082, Hungary

## CIS HEADQUARTERS

9th Radialnaya Street, Building 2  
Moscow, 115404, Russia

CONFIDENTIAL

## CONTENTS

<b>1. INTRODUCTION .....</b>	<b>1</b>
<b>2. HARDWARE AND SOFTWARE REQUIREMENTS .....</b>	<b>1</b>
2.1. Database Tier .....	1
2.1.1. Database Server Hardware Requirements .....	1
2.1.2. Database Server Software Requirements .....	2
2.2. Application Server .....	2
2.2.1. Application Server Hardware Requirements .....	2
2.2.2. Application Server Software Requirements .....	3
2.3. Client Tier .....	3
2.3.1. Client Tier Hardware Requirements .....	3
2.3.2. Client Tier Software Requirements .....	4
2.4. Development Environment Requirements .....	4
<b>3. INSTALLATION OVERVIEW .....</b>	<b>4</b>
3.1. Download Indigo ELN Delivery Package .....	5
3.2. Delivery Structure .....	5
3.2.1. File Structure for Indigo ELN Database Scripts .....	5
3.2.2. File Structure for Indigo ELN Source Code .....	5
3.2.3. File Structure for UsersToolForIndigoELN Source Code .....	7
<b>4. INSTALLING INDIGO ELN DATABASE .....</b>	<b>7</b>
4.1. Bingo Cartridge Installation .....	7
4.2. Database Users and Schema Creation .....	7
4.3. User Management in Indigo ELN .....	8
<b>5. CONFIGURING AND BUILDING INDIGO ELN .....</b>	<b>8</b>
5.1. Code Overview .....	9
5.1.1. Code Structure .....	9
5.1.2. Build Procedure .....	9
5.2. Quick Start with Default Settings .....	9
5.3. Integration of Indigo ELN with External Services .....	10
5.3.1. External Service Interfaces Used in Indigo ELN .....	10
5.3.2. Mock Implementations of Service Interfaces .....	11
5.3.3. Custom Implementations of Service Interfaces .....	11
5.4. Adding Custom External Service to Indigo ELN .....	11
5.4.1. Building New Project with Service Implementation .....	12
5.5. Integration of Indigo ELN with Epam services .....	12
5.5.1. Bingo Compound Registration Search (CRS) service .....	12
5.5.2. Indigo Signature Service .....	13
<b>6. DEPLOYING AND CONFIGURING INDIGO ELN APPLICATION SERVER .....</b>	<b>13</b>
6.1. Editing Indigo ELN Configuration Files .....	13
6.2. Simple Application Security .....	14
<b>7. INDIGO ELN USER INTERFACE CONSTANTS .....</b>	<b>14</b>
<b>8. RUNNING INDIGO ELN OPEN SOURCE ON CLIENT MACHINE .....</b>	<b>17</b>

## 1. INTRODUCTION

Indigo ELN 1.2, an open-source Chemistry Electronic Lab Notebook, was initially developed within Pfizer, Inc. as CeN Chemistry Electronic Lab Notebook and is widely used by Pfizer scientists.

Pfizer licensed the software code of Indigo ELN to EPAM Life Sciences (former GGA) in order to create and deliver an external, open source version of the Chemistry Electronic Lab Notebook.

Pfizer granted a license to EPAM Life Sciences in order to:

- Install, use, modify, and create derivative works of CeN.
- Allow its open source community to use CeN itself and the versions originating from this open-source code.
- Distribute Indigo ELN to its users on a standalone basis and/or in conjunction with EPAM products and services.
- Support Indigo ELN

This document describes the process of deploying the Indigo ELN database and application server at your site. This document also explains how to install the Indigo ELN source code and customize it to use with your custom external services.

## 2. HARDWARE AND SOFTWARE REQUIREMENTS

### 2.1. DATABASE TIER

#### 2.1.1. Database Server Hardware Requirements

Indigo ELN requires the following hardware:

Component	Linux Computer: Supported Configuration	Windows Computer: Supported Configuration
Processor	Intel IA-32, Intel EM64T (Xeon), or AMD64 (Opteron) 1 GHz or higher <b>Tip:</b> A faster processor delivers better performance.	Intel or Intel-compatible computer Pentium III or better 2 GHz or higher <b>Tip:</b> A faster processor delivers better performance.
RAM	1 GB or higher <b>Tip:</b> More RAM provides better performance.	1 GB or higher <b>Tip:</b> More RAM provides better performance.
Hard disk space	500 MB	500 MB
Network	Ethernet	Ethernet
Component	Linux Computer: Supported Configuration	Windows Computer: Supported Configuration

### 2.1.2. Database Server Software Requirements

The Indigo ELN database tier supports the following operating systems:

Linux Computer: Supported Configuration	Windows Computer: Supported Configuration
Red Hat Enterprise Linux ES 5.0 and higher	Any one of the following operating systems: <ul style="list-style-type: none"> <li>• Microsoft Windows Enterprise Server 2008</li> <li>• Microsoft Windows Standard Server 2003 SP2, 2003 R2 SP2, or 2008</li> <li>• Microsoft Windows Advanced Server 2003 SP2 or 2003 R2 SP2</li> <li>• Microsoft Windows Data Center Server 2003 SP2 or 2003 R2 SP2</li> </ul>

One of the following RDBMS is **required** on the database tier:

Software	Product	Vendor	Supported Versions
RDBMS	Oracle	Oracle	11 R2 or higher
RDBMS	Oracle XE	Oracle	11 R2 or higher
RDBMS	PostgreSQL	PostgreSQL	9.2

The following software is **required** to operate Indigo ELN properly on the database tier:

Software	Product	Vendor	Supported Versions
Molecule Search Cartridge	Bingo	EPAM	1.7.9+

#### Note

The database can be installed under both Windows and Linux. The current version of the database installation scripts requires admin access to database. SQL command line tools should be available in PATH (sqlplus and sqlldr for Oracle; psql for PostgreSQL).

## 2.2. APPLICATION SERVER

### 2.2.1. Application Server Hardware Requirements

The Indigo ELN application server requires the following hardware:

Component	Hardware
Processor	Intel or Intel-compatible computer Pentium III or better, 2 GHz or higher <b>Tip:</b> A faster processor delivers faster performance.
RAM	1 GB or higher (2 GB is recommended) <b>Tip:</b> More RAM provides faster performance.
Hard disk space	500 MB
Network	Ethernet

### 2.2.2. Application Server Software Requirements

The Indigo ELN application server supports the following operating systems:

Linux Computer: Supported Configuration	Windows Computer: Supported Configuration
Red Hat Enterprise Linux ES 5.0 and higher	Any one of the following operating systems: <ul style="list-style-type: none"> <li>• Microsoft Windows Enterprise Server 2008</li> <li>• Microsoft Windows Standard Server 2003 SP2, 2003 R2 SP2, or 2008</li> <li>• Microsoft Windows Advanced Server 2003 SP2 or 2003 R2 SP2</li> <li>• Microsoft Windows Data Center Server 2003 SP2 or 2003 R2 SP2</li> </ul>

The following software must be installed on the computer on which the Indigo ELN application server is installed and on which further configuration occurs:

Software	Product	Vendor	Supported Versions
Application server	Tomcat	Apache	7+
Environment for Tomcat and Maven	Java JDK	Oracle	1.6.* (not later version)

**Note:**

*Prior to starting the installation of the Indigo ELN application server, you must download and install the Apache Tomcat 7 (EPAM recommends installing the latest versions of JDK and Tomcat). Tomcat uses the environment variables to locate the Java runtime environment. OpenJDK 6b27 or later can be used on Linux computers instead of Oracle JDK.*

## 2.3. CLIENT TIER

### 2.3.1. Client Tier Hardware Requirements

The Indigo ELN client computer requires the following hardware:

Component	Supported Configuration
Processor	Intel or Intel-compatible computer Pentium III or better, 2 GHz or higher <b>Tip:</b> A faster processor delivers better performance.
RAM	1 GB or higher <b>Tip:</b> More RAM provides better performance.
Hard disk space	500 MB
Network	Ethernet or Wi-Fi

### 2.3.2. Client Tier Software Requirements

The Indigo ELN client supports the following operating systems:

	Supported Configuration
Supported Operating Systems	<ul style="list-style-type: none"><li>• Microsoft Windows XP Professional</li><li>• Microsoft Windows 7</li><li>• Mac OS X</li><li>• GNU/Linux with GNOME Desktop Environment</li></ul>

The following software is required to operate Indigo ELN properly on the client tier:

Software	Vendor	Supported Versions	Comments
Java JRE	Oracle	1.6+	
ISIS/Draw	Accelrys	2.5	Required to edit molecular structures

The following software is optional on the client tier:

Software	Vendor	Supported Versions	Comments
ISIS/Draw	Accelrys	2.5	Used for editing molecular structures

**Note**

*OpenJDK 6b27 or later can be used on Linux computers instead of Oracle JRE.*

### 2.4. DEVELOPMENT ENVIRONMENT REQUIREMENTS

The following software is required to customize Indigo ELN properly in the development environment:

Software	Vendor	Supported Versions	Comments
Maven	Apache	3.2.* (not later version)	Required to build the deployment units
Java JDK	Oracle	1.6+	Environment for Tomcat and Maven

**Note**

*OpenJDK 6b27 or later can be used on Linux computers instead of Oracle JDK.*

## 3. INSTALLATION OVERVIEW

To install and run Indigo ELN at your site:

1. Download the Indigo ELN sources.
2. Install the Indigo ELN database on the database server.
3. (Optional) Customize Indigo ELN source code to use custom services.

**Note**

*You will not be able to use the full functionality of Indigo ELN until you compile and build it with your custom service implementations.*

4. Deploy and configure the Indigo ELN application server.
5. Run Indigo ELN on the client computers.

The following sections explain these steps in detail.

### 3.1. DOWNLOAD INDIGO ELN DELIVERY PACKAGE

The Indigo ELN sources are available to download from the EPAM Life Sciences open source web site. To access the file, open <http://lifescience.opensource.epam.com/indigo/eln/index.html>, navigate to the download link for this product, and download the files to a folder on your server.

The Indigo ELN sources include:

- **/database** - contains the database scripts.
- **/IndigoELN** - contains the IndigoELN source files.
- **/UsersToolForIndigoELN** - contains the source files of the user tool for Indigo ELN.

### 3.2. DELIVERY STRUCTURE

This section lists the files included in the Indigo ELN sources package.

#### 3.2.1. File Structure for Indigo ELN Database Scripts

Table 1 lists the contents of the **/database** folder. These files create the Indigo ELN database objects on the database tier.

**Note**

*This section uses the following notations:*

- **SANDBOX** - to refer to the **/database** folder

Table 1

Folder or file	Description
SANDBOX\oracle	Contains SQL scripts to create Indigo ELN Oracle schema
SANDBOX\postgresql	Contains SQL scripts to create Indigo ELN PostgreSQL schema
SANDBOX\xml	Contains the initial XML data for Indigo ELN and reagents
SANDBOX\temp	Temp folder for Oracle installation scripts
SANDBOX\README.md	Simple README file

#### 3.2.2. File Structure for Indigo ELN Source Code

The Indigo ELN sources located in the **/IndigoELN** folder consist of a set of projects. Table 2 lists these projects and their short descriptions.

**Note**

*This section uses the following notation:*

- **SANDBOX** - to refer to the **/IndigoELN** folder.

Table 2

Project Name	Description
AnalyticalService	Classes for integration with the service for storing analytical files content (like WebDav)
ChemistryService	Classes for integration with EPAM's universal organic chemistry toolkit (Indigo)

Project Name	Description
CloracnegenService	Classes for integration with “cloracnegen service” and for compound search
CodeTableService	The service that provides application-wide constants
Common	Common classes used by the client and server parts
CompoundService	Classes for integration with “compound search service” and for compound search
DesignService	Classes for integration with the services that provide “design synthesis plans” to Indigo ELN
DesktopClient	Client application
EpamServices	The project containing the implementations of some external interfaces which integrate the Indigo ELN with services provided by EPAM
ExternalCollaborator	The project responsible for converting experiments from the external collaborator to the form accepted by Indigo ELN
ExternalServices	The project that contains all required interfaces to external services and their clean implementations which do nothing and return zeros or empty strings
JobScheduler	The utility project for running periodical tasks
LoggingService	The service for logging some Indigo ELN events and log messages
ReagentService	The service for reagent management, that is reagent search, personal reagent list management
RegistrationManager	Classes for integration with the registration service
RegistrationService	
ReportService	The utility project used for generating printable representation of Indigo ELN experiments
ServiceLocator	The utility project used for initially locating Indigo ELN services
SessionService	Utility projects used for creating Indigo ELN user sessions
SessionTokenService	
SignatureService	Classes for integration with the service of electronic signing and archiving experiments
StorageService	The service for saving Indigo ELN data to the database
UsersToolForIndigoELN	The project with the tool for Indigo ELN user management
WebApplication	The project responsible for the build procedure



### 3.2.3. File Structure for UsersToolForIndigoELN Source Code

The source code in the `/UsersToolForIndigoELN` folder contains a single project with the tool for Indigo ELN users management.

## 4. INSTALLING INDIGO ELN DATABASE

This section explains the deployment of the Indigo ELN database scripts from the `/database` folder into a target environment.

For Oracle, the installation path must contain the path to `sqlplus.exe` and `sqlldr.exe` (included into Oracle Client). For PostgreSQL, the installation path must include the path to `psql.exe` (available after the installation of PGAdmin).

To install the Indigo ELN database:

1. Install the Bingo cartridge (See Section 4.1 for details).
2. Extract the database scripts from the delivery package.
3. Edit the database properties file.
4. Set URL properties for services.
5. Run the scripts to create the database schema and users.

### 4.1. BINGO CARTRIDGE INSTALLATION

In order to allow users to search among Indigo ELN experiments by a compound structure or a reaction scheme, it is required to install the Bingo cartridge into Indigo ELN DB. You can find the information about the Bingo cartridge installation [here](#).

Bingo must be installed before the Indigo ELN database installation. It is required for Indigo ELN to have correct permissions to access the Bingo schema.

### 4.2. DATABASE USERS AND SCHEMA CREATION

To create the database users and schema:

1. Open the Command prompt window.
2. Navigate to the `<SANDBOX>` folder.
3. Execute the following commands (Change parameters according to your configuration):
  - a. For Oracle:

```
sqlplus sys/admin@//localhost:1521/xe as sysdba
@oracle/schema_install.sql

sqlldr indigo_owner/indigo_owner@//localhost:1521/xe
oracle/xml_metadata.ctl
```

- b. For PostgreSQL:

```
psql -h localhost -p 5432 -d postgres -U postgres -f
postgresql/schema_install.sql
```

4. Review the output for errors.

#### Note

To uninstall the Indigo ELN DB Schema, execute the following commands.

**WARNING!** This removes all Indigo ELN data from the database:

- a. For Oracle:

```
sqlplus sys/admin@//localhost:1521/xe as sysdba
@oracle/schema_uninstall.sql
```

- b. For PostgreSQL:

```
psql -h localhost -p 5432 -d postgres -U postgres -f  
postgresql/schema_uninstall.sql
```

### 4.3. USER MANAGEMENT IN INDIGO ELN

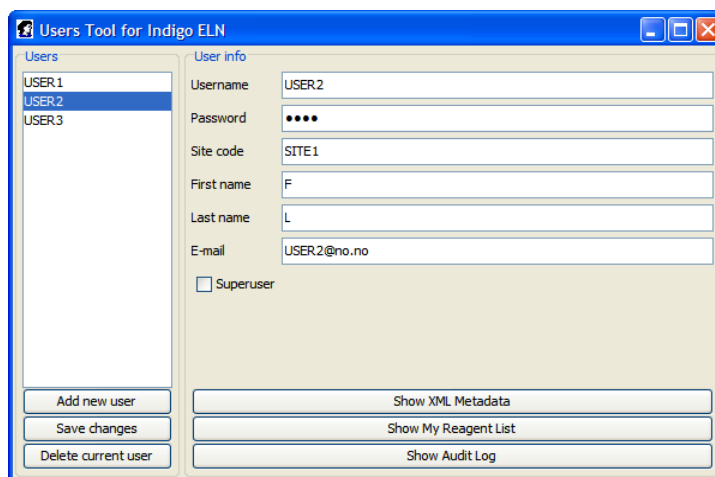
The User Tool from the `/UsersToolForIndigoELN` folder allows adding/editing users in Indigo ELN.

To set the tool to operate with the Indigo ELN database, you should open the **connection.properties** file in the `src\main\resources\` folder and add the JDBC address, the connection string for the database, the name of the table with users in the database, the name of the user, and the password to access the database.

The project contains the maven **pom.xml** file. To build the tool: execute "mvn clean package".

Run **UsersToolForIndigoELN.jar** with Java:

The following window opens:



The left pane displays the current list of users. The right pane displays the properties of a particular user.

To change the properties of an existing user:

1. Select the user in the list.
2. Make necessary changes in the right pane.
3. Click **Save changes**. The new properties are saved.

#### Note

*If the Password field is left blank, the current password of the user remains unchanged.*

To add a new user:

1. Click **Add new user**.
2. In the right pane, fill in **Username**, **Password**, and the other fields.
3. Click **Save changes**. The properties of the new user are saved.

To remove a user:

1. Select the user in the list.
2. Click **Delete current user**. The user is removed from the database.

There are other three buttons to view the properties changed by Indigo ELN itself: **Show XML Metadata**, **Show My Reagent List**, and **Show Audit Log**.

## 5. CONFIGURING AND BUILDING INDIGO ELN

This section explains how to organize the Indigo ELN workspace so that it can use real services instead of mock ones.

For a successful operation and in order to provide the full range of functions, Indigo ELN requires numerous services, like search or registration services. Each service contains specific operations. Thus, the Indigo ELN code defines interfaces of all the required services. When all required service implementations are used, Indigo ELN becomes a powerful and easy-to-use tool for creating and handling chemical experiments.

The following sections explain:

- The structure of the source code, all required services, and their empty/EPAM implementations supplied with the source code;
- How to define the service implementations correctly and to make them work;
- How to organize the Indigo ELN workspace so that it can use real services instead of mock ones.

#### Note

*This section explains how to build the Indigo ELN code in the development environment. Ensure that the following software is installed in this environment:*

- Maven 3.2.\* or later
- Java JRE 1.6.\* or later

*This section assumes that you have extracted the Indigo ELN source code to a folder on the development computer at your site and further refers to this folder path as SANDBOX.*

## 5.1. CODE OVERVIEW

### 5.1.1. Code Structure

The delivery package provides the source code for the ExternalServices project. This project contains all required interfaces to the external services. The source code of the ExternalServices project is available in the SANDBOX folder.

The delivery package with the full sources provides the code for all Indigo ELN internal projects. You can find their list and short descriptions in Section 5.5.

### 5.1.2. Build Procedure

The project contains the maven `pom.xml` file. The script creates `WebApplication\target\indigoeln.war` deployment unit. The `indigoeln.war` file contains both the client and server parts.

## 5.2. QUICK START WITH DEFAULT SETTINGS

By default, Indigo ELN is configured to use all services on a local host. The following are the simple steps to build and run Indigo ELN on a local machine:

1. Get the source code of the required applications: IndigoELN, CRS (optional), IndigoSignatureService (optional).
2. Build IndigoELN with Maven:
  - a. For use without CRS and IndigoSignatureService: `mvn clean package`.
  - b. For use with CRS and IndigoSignatureService: `mvn clean package -P epam`.
3. (Optional) build CRS (run `mvn clean package`).
4. (Optional) build IndigoSignatureService (run `mvn clean package`).
5. Place deployment units into the Tomcat Webapps folder:
  - a. For use without CRS and IndigoSignatureService:
    - i. `IndigoELN/WebApplication/indigoeln.war`
  - b. For use with CRS and IndigoSignatureService:
    - i. `IndigoELN/WebApplication/indigoeln.war`
    - ii. `CRS/target/crs.war`

- iii. IndigoSignatureService/target/signatureservice.war
6. Run Tomcat.
7. Open <http://localhost:8080/indigoeln> and start Indigo ELN Desktop Client.

### 5.3. INTEGRATION OF INDIGO ELN WITH EXTERNAL SERVICES

#### 5.3.1. External Service Interfaces Used in Indigo ELN

External services are not part of Indigo ELN. To enable communication between the external services and Indigo ELN, you need special adapters. Each adapter implements an interface that is a part of Indigo ELN. This implementation performs requests to the service.

All external service interfaces are located in the **ExternalServices** project. Each service has its own package **com.chemistry.enotebook.<servicename>** where the service interface, factory class, required classes, and mock implementation are placed.

The **service\_\*\*\*.properties** configuration files from the **ExternalServices\conf** subfolder define the relationship between **<service interface class>** and **<service adapter implementation>**. Based on the **service\_\*\*\*.properties** file, Indigo ELN chooses the implementation to use. The maven profiles specifies which **service\_\*\*\*.properties** file to use. By default, the “clean” profile is active.

For each service, the **service\_implementation.properties** configuration file contains a variable named after the corresponding interface title (including the full path). The value of each variable is used for loading the class that implements the corresponding interface. The value should contain the full path to the real class. The service with an incorrect path to the implementation class fails with a class loader error.

The following table presents the names of service interfaces in the same way as they exist in the **service\_implementation.properties** file (the **com.chemistry.enotebook.** prefix is omitted).

Service Name	Description
compoundaggregation. CompoundAggregationService	This service is used when compounds are submitted for screening.  <b>Note:</b> Compounds can be in the form of batches (non-plated batches) or plates (plated batches).
compoundmanagement. CompoundManagementService	Provides Indigo ELN with access to compounds, containers, and plates. Performs container registration functions.
security.SecurityService	Performs user identification and authentication.
analyticalservice.AnalyticalService	Links product batches to analytical files in the analytical service.
compoundregistration. CompoundRegistrationService	Submits compounds to the registration service, performs verification and validation of chemical structures.
vnv.VnVService	Tests the conformity of chemical structures with stereo isomer code detection. Verifies the uniqueness of structures.
extcol.ExternalCollaboratorService	Searches for the outsourced experiments through an external collaborator system and handles outsourced chemical syntheses. This service is not a part of ExternalServices because it is tightly coupled with the Indigo ELN internal structure (it should build an Indigo

Service Name	Description
	ELN experiment from an external format). Thus, the adapter for this service should be a part of Indigo ELN.
chloracnegen.ChloracnegenService	Detects a chloracnegen class for a structure.
purificationservice.CeNPurificationServiceSubmissionService	Submits compounds to the purification service, the system for managing the auto-purification of lab submissions and processes.
person.IPersonService	Searches users by parameters like the first name, last name, and location.
signature.ESignatureService	Performs operations for communication with the IP repository of electronically signed and archived experiments. Creates requests for electronic signatures when the experiment is completed.
search.SearchService	Searches for registered structures by compound details.
publisher.PublisherService	Publishes structures and reactions to external systems.
hazard.HazardService	Calculates hazard information for compounds.
vcr.VCRService	Performs registration of virtual compounds (prefix VC).

### 5.3.2. Mock Implementations of Service Interfaces

To enable the operation of Indigo ELN without any services, the source package provides a set of mock implementations for the service interfaces.

If you build Indigo ELN with the default parameters (mvn clean package), you can use the basic application functionality. In this case, `service_implementation.properties` contains only mock implementations of service interfaces. Empty implementations facilitate the process of creating new service implementations. It is simpler to integrate one new service implementation after another instead of trying to integrate the whole set simultaneously.

Example from `service_implementation.properties`:

```
com.chemistry.enotebook.signature.ESignatureService=com.chemistry.enotebook.signature.fake.ESignatureServiceFakeImpl
com.chemistry.enotebook.publisher.PublisherService=com.chemistry.enotebook.publisher.fake.PublisherServiceFakeImpl
```

### 5.3.3. Custom Implementations of Service Interfaces

Each service implementation is created by the factory which gets the class name of the implementation from the `service_***.properties` file in the ExternalServices project. In order to make the custom implementation work, ensure that the class name of this implementation is in the `service_***.properties` file as `<service interface class> = <custom implementation class>` and that this custom implementation class is in classpath.

#### Note

*If the custom implementation is part of the ExternalServices project, then it appears in the classpath automatically.*

## 5.4. ADDING CUSTOM EXTERNAL SERVICE TO INDIGO ELN

To customize the Indigo ELN source code (for example, to add a new service, see Section 5.3.1):

1. Apply required modifications in the ExternalServices project.
  - If you have the service implemented and running in your environment:
    - a. Implement the service adapter.
    - b. Add the adapter as a package to the ExternalServices project or as a separate project.
  - If you do not have the service implemented:
    - a. Implement the service as a separate project.
    - b. Add the corresponding jar file to the **indigoeln.war/WEB-INF/lib** subfolder.
2. Add the name of the class that implements the service interface to the **service\_\*\*\*.properties** file (the file to use for project building). For example:

```
com.chemistry.enotebook.signature.ESignatureService=<class name of the custom implementation>
```
3. (Optional) If you have added the service/adapter as a separate project, follow the instructions in Section 5.4.1 to build the project.
4. Run mvn clean package from the project folder. This script rebuilds the whole application and creates the **ExternalServices.jar** file.
5. If you updated only the ExternalServices project then you may just substitute the existing jar files in the deployment unit with the new **ExternalServices.jar** file: **indigoeln.war\WEB-INF\lib\ExternalServices-1.2.jar**.

#### 5.4.1. Building New Project with Service Implementation

##### Note

*You must perform this step if you have added the adapter or service implementation as a separate project.*

Maven 3.2+ is used for building the deployment units and Java archives. To rebuild the project, you must run a Maven script as a **pom.xml** file that compiles the source code and builds the jar file.

To build the new project:

1. Compile the source code.
2. Build the jar file using the class files created in the previous step.

##### Note:

*All files and libraries used in the external service implementation should be added to the file set of the jar file.*

3. Move the jar file created in the previous step to the same location as **ExternalServices.jar**: **indigoeln.war\WEB-INF\lib\NewProject.jar**.

To create your customized maven script, you can use the **pom.xml** file from the ExternalServices project as an example.

## 5.5. INTEGRATION OF INDIGO ELN WITH EPAM SERVICES

At present, there are two EPAM services that can be used in Indigo ELN:

- Bingo Compound Registration Search
- Indigo Signature Service

### 5.5.1. Bingo Compound Registration Search (CRS) service

This service provides the functionality of compounds registration and search among the registered compounds. You can download the code and installation instruction at [lifescience.opensource.epam.com](http://lifescience.opensource.epam.com), in the Indigo ELN downloads section.

This paragraph presumes that Bingo CRS has been successfully installed and its services are available at <CRS service URL>.

EPAM provides the search and registration adapters for the Bingo service, you can find them in the EpamServices project.

To integrate Indigo ELN with the CRS process with the following steps:

1. Open `EpamServices/conf/registration.properties`.
2. Change `SERVICE_URL` to the actual CRS URL.

### 5.5.2. Indigo Signature Service

This service provides the experiment reports signing functionality. You can download the code and installation instruction at [lifescience.opensource.epam.com](http://lifescience.opensource.epam.com), in the Indigo ELN downloads section.

It is necessary to create user accounts in Signature Service for users who should be able to sign ELN experiments. To integrate Indigo ELN with Indigo Signature Service:

1. Open `EpamServices/conf/signature.properties`.
2. Change `SERVICE_URL` to the actual Indigo Signature Service URL.

## 6. DEPLOYING AND CONFIGURING INDIGO ELN APPLICATION SERVER

This section explains how to install, configure, and launch the Indigo ELN application under the Tomcat server.

The following are the steps to install and configure the application server:

1. Ensure that all required software is installed and configured.

See Sections 2.2 and 2.4 for the list of required software for the Indigo ELN application server and development environment.

### Note

*You might want to establish the development environment on the same computer as the application server, or on a separate machine. This section assumes that the development environment computer and the Indigo ELN application server are two separate computers.*

2. Extract the Indigo ELN source files to the development environment as Section 3.2.2 explains.
3. Edit the Indigo ELN configuration files as Section 6.1 explains.
4. Configure the Tomcat Server.

The next sections explain these steps in detail.

### 6.1. EDITING INDIGO ELN CONFIGURATION FILES

To edit the Indigo ELN configuration files:

1. Navigate to the folder where you saved the Indigo ELN source code.
2. Edit `ServiceLocator/src/main/resources/application.properties`:
  - a. `SERVICE_URL` - Application URL (usually `http://<server_host>:<server_port>/indigoeln`).
  - b. `SERVICE_USERNAME` - Username for application security.
  - c. `SERVICE_PASSWORD` - Password for application security.
  - d. `DATASOURCE` - Name of the default IndigoELN datasource (default datasource configured in `WebApplication/src/main/resources/database/xml/elN-datasource-context.xml`).
3. Configure the logging in `WebApplication/src/main/resources/log4j.properties` and `DesktopClient/src/main/resources/log4j.properties`.



## 6.2. SIMPLE APPLICATION SECURITY

Indigo ELN supports basic application security. This means that all application services (like Storage Service or Report Service) are password-protected.

### Note

*The only restriction is that users should enter the application username/password when downloading the application via JNLP, because application jars contain application username/password to access secured services.*

To enable application security, just comment the line `<security:anonymous granted-authority="ROLE_SERVICE" enabled="true"/>` in `WebApplication/src/main/webapp/WEB-INF/security-context.xml` and change the default application password in `ServiceLocator/src/main/resources/application.properties`.

## 7. INDIGO ELN USER INTERFACE CONSTANTS

Indigo ELN provides possibilities for different customers to see different values depending on their company internal structure, names, and values they got used to. To meet potential requirements, Indigo ELN provides the possibility to change these constants.

The constants are kept in csv files in the `Chem eNotebook/server/APP-INF/classes/data/` folder (or in the deployment unit `enotebook_server.ear/properties/data/`).

The following is a short description of the constants and how these constants are represented in the user interface:

1. Therapeutic Area code
  - a. Location in the UI: **Experiments** tab -> **Therapeutic Area** dropdown list.
  - b. CSV file name: `COMPOUND_REGISTRATION_TA_CDT.csv`.
  - c. Required values:
    - i. `TA_CODE` - how it is represented in the database.
    - ii. `TA_DESC` - how the therapeutic area is represented in UI in the **Therapeutic Area** field, users do not see `TA_CODE` itself.
  - d. Example: "GE", "General".
2. Project code
  - a. Location in the UI: **Experiments** tab -> **Project Code** dropdown list.
  - b. CSV file name: `COMPOUND_REGISTRATION_PROJECT_CDT.csv`.
  - c. Required values:
    - i. `PROJECT_CODE` - how it is represented in the database.
    - ii. `PROJECT_NAME_DESC` - values in the **Project Code & Name** field, that is "PROJECT\_CODE - PROJECT\_NAME\_DESC", for example "ABCDE12345 - Project\_1".
    - iii. `TA_CODE` - the code of a therapeutic area to which the project code belongs. If users select TA, then in the **Project Code & Name** dropdown list, users see only the project codes with the corresponding `TA_CODE`.
  - d. Example: "ABCDE12345", "Project\_1", "GE".
3. Site code
  - a. Location in the UI: in the panel of the Indigo ELN window, **All Sites** lists all available sites. Every user is assigned to some site code. If the user's site code is in the CSV file then this user's name is present under the corresponding site on the **All Sites** tab.
  - b. CSV file name: `COMPOUND_MANAGEMENT_SITE_CDT.csv`.
  - c. Required values:
    - i. `SITE_CODE` - how it is represented in the database.
    - ii. `LABEL` - how it is represented in the UI.
4. Salt code
  - a. Location in the UI:



- i. **Experiments** tab -> **Stoichiometry Table** pane -> **Salt Code** column.
    - ii. **Batches** tab -> **Product Nbk Batch Summary** pane -> **Salt Code** column.
    - iii. **Batches** tab -> **Product Batch Details** pane -> **Salt Code & Name** dropdown list.
  - b. CSV file name: **COMPOUND\_MANAGEMENT\_SALT\_CDT.csv**.
  - c. Required values:
    - i. **SALT\_CODE** - presented the same was as in the database.
    - ii. **SALT\_CODE** - **SALT\_DESC** - presented the same was as in the UI.
    - iii. **SALT\_WEIGHT** - used for the compound weight calculation.
    - iv. **SALT\_FORMULA** - used for the compound formula building.
  - d. Example: "44","MORPHOLINE","87.1224","C4H9NO".
5. Source
  - a. Location in the UI:
    - i. **Experiments** tab -> **Stoichiometry Table** pane -> **Source** column.
    - ii. **Batches** tab -> **Product Nbk Batch Summary** pane -> **Source** column.
    - iii. **Batches** tab -> **Product Batch Details** pane -> **Source** dropdown list.
  - b. CSV file name: **COMPOUND\_REGISTRATION\_SOURCE\_CDT.csv**.
  - c. Required values:
    - i. **SOURCE\_CODE** - presented the same was as in the database.
    - ii. **SOURCE\_DESC** - presented the same was as in the UI.
  - d. Example: ("HSS","PGRD High Speed Synthesis"); ("EXTERNL","External").
6. Source Details
  - a. Location in the UI:
    - i. **Experiments** tab -> **Stoichiometry Table** pane -> **Source Details** column.
    - ii. **Batches** tab -> **Product Nbk Batch Summary** pane -> **Source Details** column.
    - iii. **Batches** tab -> **Product Batch Details** pane -> **Source Details** dropdown list.
  - b. CSV file name: **COMPOUND\_REGISTRATION\_SOURCE\_DETAIL\_CDT.csv**.
  - c. Required values:
    - i. **SOURCE\_DETAIL\_CODE** - presented the same was as in the database.
    - ii. **SOURCE\_DETAIL\_DESC** - presented the same was as in the UI.
    - iii. **SOURCE\_CODE** - the source code that the source detail belongs to. If a user selected Source, then the **Source Details** dropdown list displays only the source details with the corresponding **SOURCE\_CODE**.
  - d. Example: ("NAG\_CT","Nagoya - Chemical Technologies Parallel Synthesis","HSS").
7. External Supplier
  - a. Location in the UI:
    - i. **Batches** tab -> **Product Nbk Batch Summary** pane -> **Ext Supplier** column.
    - ii. **Batches** tab -> **Product Batch Details** pane -> **Ext Supplier** dropdown list.
  - b. CSV file name: **COMPOUND\_MANAGEMENT\_SUPPLIER\_CDT.csv**.
  - c. Required values:
    - i. **SUPPLIER\_CODE**.
    - ii. **SUPPLIER\_DESC** - if it is empty, then only the code is displayed.
8. Solvent codes
  - a. Location in the UI:
    - i. **Batches** tab -> **Product Batch Details** pane -> **Vials/Tubes Containing this Batch** table -> **Solvent** column.
  - b. CSV file name: **COMPOUND\_MANAGEMENT\_SOLVENT\_CDT.csv**.
  - c. Required values:
    - i. **SOLVENT\_CODE** - presented the same was as in the database.
    - ii. **SOLVENT\_DESC** - presented the same was as in the UI.
9. Stereoisomer codes
  - a. Location in the UI:
    - i. **Batches** tab -> **Product Batch Details** pane -> **Stereoisomer Code** dropdown list.
    - ii. **Batches** tab -> **Product Nbk Batch Summary** pane -> **Stereoisomer** column.

- b. CSV file name: **COMPOUND\_MANAGEMENT\_STEREOISOMER\_CDT.csv**.
  - c. Required values:
    - i. STEREOISOMER\_CODE - presented the same was as in the database.
    - ii. STEREOISOMER\_DESC - presented the same was as in the UI.
10. Analytical purity code
- a. Location in the UI:
    - i. **Batches** tab -> **Product Batch Details** pane -> **Purity** -> **Edit** button.
    - ii. **Batches** tab -> **Product Nbk Batch Summary** pane -> **Purity** column.
  - b. CSV file name: **COMPOUND\_REGISTRATION\_ANALYTIC\_PURITY\_CDT.csv**.
  - c. Required values:
    - i. ANALYTICAL\_PURITY\_CODE - presented the same was as in the database.
    - ii. ANALYTICAL\_PURITY\_DESC - presented the same was as in the UI.
11. Protection code
- a. Location in the UI:
    - i. Representation: PROTECTION\_CODE - PROTECTION\_DESC.
    - ii. **Batches** tab -> **Product Batch Details** pane -> **Compound protection** dropdown list.
    - iii. **Batches** tab -> **Product Nbk Batch Summary** pane -> **Compound Protection** column.
  - b. CSV file name: **COMPOUND\_REGISTRATION\_PROTECTION\_CDT.csv**.
  - c. Required values:
    - i. PROTECTION\_CODE - presented the same was as in the database.
    - ii. PROTECTION\_DESC - presented the same was as in the UI.
12. Qualitative codes
- a. Location in the UI:
    - i. **Batches** tab -> **Product Batch Details** pane -> **Solubility in Solvents** -> **Edit** button -> **Qualitative** button.
  - b. CSV file name: **COMPOUND\_REGISTRATION\_QUAL\_SOLUBILITY\_CDT.csv**.
  - c. Required values:
    - i. QUALITATIVE\_CODE - presented the same was as in the database.
    - ii. QUALITATIVE\_DESC - presented the same was as in the UI.
13. Residual Solvent codes
- a. Location in the UI:
    - i. **Batches** tab -> **Product Batch Details** pane -> **Residual Solvents** -> **Edit** button.
  - b. CSV file name: **COMPOUND\_REGISTRATION\_RESIDUAL\_SOLVENT\_CDT.csv**.
  - c. Required values:
    - i. RESIDUAL\_SOLVENT\_CODE - presented the same was as in the database.
    - ii. RESIDUAL\_SOLVENT\_DESC - presented the same was as in the UI.
    - iii. RESIDUAL\_SOLVENT\_WEIGHT.
14. Solubility solvent codes
- a. Location in the UI:
    - i. **Batches** tab -> **Product Batch Details** pane -> **Solubility in Solvents** -> **Edit** button -> **Solubility Solvent Code & Name** column.
  - b. CSV file name: **COMPOUND\_REGISTRATION\_SOLUBILITY\_SOLVENT\_CDT.csv**.
  - c. Required values:
    - i. SOLUBILITY\_SOLVENT\_CODE - presented the same was as in the database.
    - ii. SOLUBILITY\_SOLVENT\_DESC - presented the same was as in the UI.
15. Compound state codes
- a. Location in the UI:
    - i. **Batches** tab -> **Product Batch Details** pane -> **Compound State** dropdown list.
    - ii. **Batches** tab -> **Product Nbk Batch Summary** pane -> **Compound State** column.
  - b. CSV file name: **COMPOUND\_REGISTRATION\_STATE\_CDT.csv**.
  - c. Required values:

- i. COMPOUND\_STATE\_CODE - presented the same was as in the database.
- ii. COMPOUND\_STATE\_DESC - presented the same was as in the UI.

## 8. RUNNING INDIGO ELN OPEN SOURCE ON CLIENT MACHINE

To run the Indigo ELN client:

1. Open the following URL in a Web browser on the client computer:

<Indigo ELN service URL>:<Indigo ELN service port>/indigoeln/

2. Click the Indigo ELN Logo in the center of the page.  
Indigo ELN Client starts.