

## 1. System Requirements

### *Operating Systems:*

- **Windows:**

Windows 10 and above (for x64)

- **macOS:**

macOS 12, 13, 14 (for x64 and AArch64)

- **Linux:**

SuSE Leap 15 and above (for x64), Ubuntu Linux 24.0S (for x64), Debian (x64),

### *Hardware Requirements:*

- Minimum 4GB of RAM
- 500MB of free disk space
- Intel i3 or equivalent processor

### *Software Requirements:*

- Java JRE: Minimum JRE 23

## 2. Deployment Strategy Summary

The **File Catalog** application is designed for local use, as a standalone Java Swing application packaged in a JAR file with Maven-managed dependencies.

### *Deployment Artifacts:*

1. **Executable JAR file ([file-catalog.jar](#))**

contains the application, including all dependencies bundled via Maven.

2. **Installer Scripts:**

Cross-platform installation scripts for setting up the environment.

- *install.sh*: Shell script for Linux/macOS.
- *install.bat*: Batch file for Windows.

### *Distribution Kit:*

All components will be packaged in a ZIP file containing:

- `file-catalog.jar`
- Installation and configuration scripts
- `README.md` with instructions

## **3. Installation Package Contents**

### *3.1 Required Source or Compiled Files*

- `file-catalog.jar`:

Main executable file with compiled source code and resources.

### *3.2 Required Third-party Components*

- Java JRE 23

### *3.4 Required Graphical Assets, Configuration, and Other Non-program Files*

- *Configuration:*  
`config.properties` for user-defined settings, located in `src/resources/config` generated by `install.sh`.

### *3.5 Documentation Files to be Provided*

- `README.md`: Instructions for installation, basic usage, and system requirements.

### *3.6 Development Files and Components to Exclude*

- *Source Code:*

All source code files (e.g., `src/`) excluded from final distribution.

- *IDE Files:*

Exclude IDE-specific files (e.g., `.idea` directory, any temporary build files).

- *Testing Artifacts:*

Exclude files generated in `target/` or `out/..`

## 4. Additional Code Required for Deployment

Shell Script for Linux/macOS ([install.sh](#)):

- Sets up the required environment locally, checks for Java and Maven installations, installs any missing dependencies, and configures paths.

Batch Script for Windows ([install.bat](#)):

- Provides the same installation process for Windows users, ensuring a consistent installation experience.

## 5. Deployment Tasks

*Prepare the project for packaging:*

- ☐ **Verify Code Quality:**  
Run tests to ensure no issues are present in the code.
- ☐ **Commit All Necessary Code:**  
Ensure the latest code is in the repository.
- ☐ **Organize Resources:** Ensure resources like images, configuration files, and scripts are correctly organized.
- ☐ **Update [pom.xml](#) for Build Settings:**  
Double-check for dependencies and ensure plugins (like the Maven Shade Plugin) are correctly configured.
- ☐ **Set Version Numbers:**  
Ensure the project version is updated in [pom.xml](#) if necessary.

*Set Up the Maven Build Process:*

- ☐ **Maven Shade Plugin Configuration:**  
Ensure the plugin is set up to package dependencies in an executable JAR.
- ☐ **Build Verification:**  
Run [mvn clean package](#) to confirm that the build process completes without errors.
- ☐ **Testing the JAR:**  
Run [java -jar target/file-catalog.jar](#) to verify it works as a standalone JAR.
- ☐ **Error Logging Verification:**  
Check that logging works as expected within the packaged JAR to capture errors.

*Create Distribution Kit:*

- ☐ **Assemble Required Files:**  
Collect [file-catalog.jar](#), [README.md](#), and other necessary documentation and scripts.
- ☐ **File Structure Check:**  
Ensure the directory structure in the distribution kit matches what's described in the README.
- ☐ **Configuration Documentation:**  
Ensure [config.properties](#) includes example values and detailed instructions for setup.

### Create Installer Scripts:

- ☐ Write platform-specific installer scripts (**install.sh**, **install.bat**).
- ☐ Verify both scripts work correctly by testing them on the respective operating systems.
- ☐ Account for platform differences, especially in file paths and permissions on Linux/macOS.
- ☐ Include sample values in **config.properties** for users to customize.

### Package Distribution kit

- ☐ **Compression Test:**  
Ensure the ZIP file includes the correct folder structure and all required files.

### Testing and deployment:

- ☐ **Cross-Platform Testing:**  
Execute on Windows, MacOS, and Linux systems as part of acceptance testing.
- ☐ **Update Documentation:**  
Document any required adjustments in **README.md** and other guides based on testing feedback.
- ☐ Deploy distribution file

## 6. Deployment Test Plan

Test Case	Test step	Expected Result
Verify Installation Script Execution	Run the installation script ( <b>install.sh</b> on Linux/MacOS, <b>install.bat</b> on Windows)	Script sets up file in correct directory and install necessary dependencies
Script Error Handling	Run installation script without dependencies like Java installed	Script should detect missing dependencies, display error messages, and guide user to install required dependencies.
Check Java JRE 23 Installation	Run <b>java -version</b>	Output shows Java version 23
Platform-Specific Settings Verification:	Check <b>JAVA_HOME</b> and advanced settings on Windows	<b>JAVA_HOME</b> and other environment variables should be configured correctly.
Validate Application Directory Structure	Look at application directory to see all files	Application directory contains <b>file-catalog.jar</b> , <b>config/config.properties</b> , and scripts folder with config scripts

Configuration File Validation	Enter invalid values in <code>config.properties</code>	Script detects incorrect values, prompts the user to correct them, and does not proceed with invalid configuration.
Basic Application Functionality	Verify UI interaction, file cataloging, and other functions	Application responds correctly to all interactions, updates the catalog, and handles search inputs as expected.
Test Dependency Isolation (JAR Packaging)	Verify that all dependencies are packaged within <code>file-catalog.jar</code>	All dependencies are included, ensuring the application does not require external dependencies other than Java.
Documentation and Help Command	Verify the clarity of instructions in <code>README.md</code>	Documentation is clear, concise, and covers all installation, configuration, and usage steps comprehensively.
Uninstallation Process	Remove the application files	No residual files or dependencies remain after uninstallation, ensuring a clean system state.