**Lab 6: Liberty - Migration Toolkit**

**Software Analyzer and Binary Scanner**

Contents

[Lab 6 Liberty Migration Toolkit 3](#_Toc75505030)

[0.1 The lab environment 4](#_Toc75505031)

[**0.1.1 Login to the "Liberty vPOT … Desktop" VM and Get Started** 5](#_Toc75505032)

[1.1 Import the Sample Day Trader Application 9](#_Toc75505033)

[1.2 Scan the DayTrader Sample application 14](#_Toc75505034)

[1.2.1 Review the software analysis results 19](#_Toc75505035)

[1.3 Create Scan Report for WebSphere to Liberty 24](#_Toc75505036)

[1.4 Migration Toolkit for Application Binaries 29](#_Toc75505037)

[1.5 Summary 35](#_Toc75505038)

# Lab 6 **Liberty Migration Toolkit**

In this lab we will learn to use the following migration tools, which are a subset of the WebSphere Migration Toolkit suite of tools.

The **Liberty Migration Toolkit** determines the suitability of migrating your applications from WebSphere Application Server, or other third-party Java EE servers, to WebSphere Liberty. To analyze an application for migration suitability, the application must be imported into your Eclipse-based IDE.

The **Migration Toolkit Eclipse plug-in** includes several rule sets to scan Java EE applications for the use of vendor specific deployment descriptors, JSP files with proprietary APIs, and Java code with proprietary APIs. The rules scan applications for the use of Java technologies and APIs that are not supported in WebSphere Liberty.

The **Migration Toolkit for Application Binaries** is a stand-alone tool that can scan your application binaries, without source code, to give you a report of the programming models used by your application, and where they will run. It also offers an option to generate a detailed report about which line of your application may need to be changed.

In this lab exercise, you will learn how to:

* Import the Sample Application onto Eclipse
* Scan the imported application
* View Migration Results and Generate Report.
* Create additional scan reports and compare.

You will also learn how to:

* Install the Migration Toolkit for Application Binaries
* Use the toolkit to scan an application
* Generate an evaluation report
* Generate a detailed report

|  |
| --- |
| **Notice**   * You don’t need to be an Eclipse power-user. But, a basic level of familiarity with Eclipse is beneficial. * Efforts have been made to ensure the screenshots in this lab are current and accurate. However, there may be subtle variations in what you see. The differences may be attributed to the Eclipse version used. |

## WebSphere developer and Migration tools in the Eclipse Marketplace

For your reference, below are some links and remarks about the various WebSphere developer tools and migration tools, all of which can be downloaded from the Eclipse marketplace.

**Note:** The tools required for this lab have already been downloaded an installed in the lab environment.

**1) WebSphere Application Server Developer Tools** (this link is for v.8.5.5, but 9.x contains the same function for v9) <https://marketplace.eclipse.org/content/ibm-websphere-application-server-v85x-developer-tools>

An Eclipse IDE for building and deploying Java EE, OSGi and Web 2.0 applications to WebSphere Application Server V8.5x.

For more information about installing and downloading the tools, including archived versions of the tools for older versions of Eclipse, see [WebSphere Application Server Developer Tools Releases](https://developer.ibm.com/wasdev/docs/websphere-developer-tools-releases/)

**2) Liberty Developer Tools** <https://marketplace.eclipse.org/content/ibm-liberty-developer-tools>

An Eclipse IDE for building and deploying Java EE, OSGi and Web 2.0 applications to Open Liberty and WebSphere Application Server Liberty.

For more information about installing and downloading the tools, including archived versions of the tools for older versions of Eclipse, see [WebSphere Application Server Developer Tools Releases](https://developer.ibm.com/wasdev/docs/websphere-developer-tools-releases/). Releases from 2021-03 onwards have some known issues and workarounds which can be found [here.](https://github.com/OpenLiberty/open-liberty-tools/wiki/Liberty-Tools-known-issues)

**3) WebSphere Application Server Migration Toolkit** <https://marketplace.eclipse.org/content/ibm-websphere-application-server-migration-toolkit>

The IBM® WebSphere® Application Server Migration Toolkit is a suite of tools and knowledge collections that enables your organization to migrate quickly and cost-effectively to WAS Liberty and WebSphere Application Server V8.5.5 or V9, whether from a previous version of WebSphere Application Server or competitive application servers including Apache Tomcat Server, JBoss Application Server, Oracle Application Server, and Oracle® WebLogic Server. It features cloud migration rules for running your application on Liberty for Java on IBM Cloud and other third-party PaaS environments such as Cloud Foundry.

Go to the [WebSphere migration tools download page](http://ibm.biz/DownloadWASMigTools) to download this tool and the command line binary scanner.

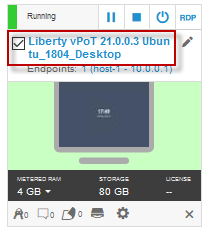
**4) IBM WebSphere Application Server Migration Toolkit - WAS Liberty** <https://marketplace.eclipse.org/content/ibm-websphere-application-server-migration-toolkit-was-liberty>

The IBM® WebSphere Application Server Migration Toolkit helps you move traditional WebSphere Application Server applications to Liberty, which can be running inside or outside of the cloud.

Go to the [WebSphere migration tools download page](http://ibm.biz/DownloadWASMigTools) to download this tool and the command line binary scanner.

## The lab environment

One (1) Linux VM has been provided for this lab.



The **“Liberty vPOT … Desktop”** VM has the following software available:

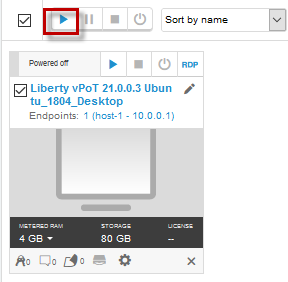
* Application Project with Liberty
* Maven 3.6.0
* The login credentials for the **Liberty vPOT … Desktop”** VM are:

User ID: **ibmdemo**

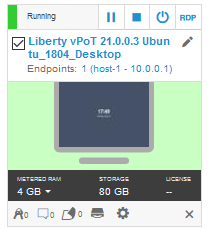
Password: **passw0rd** (That is a numeric zero in passw0rd)

### **0.1.1** **Login to the "Liberty vPOT … Desktop" VM and Get Started**

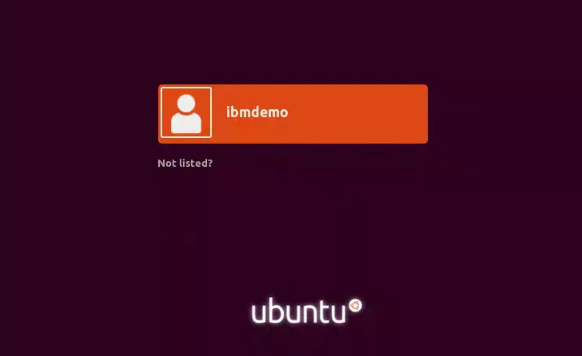
1. If the VM is **not** already started, start it by clicking the **Play** button.



1. After the VM is started, click the **“Liberty vPOT … Desktop”** VM icon to access it.



1. Login with ibmdemo ID.
   1. Click on the “ibmdemo” icon on the Ubuntu screen.



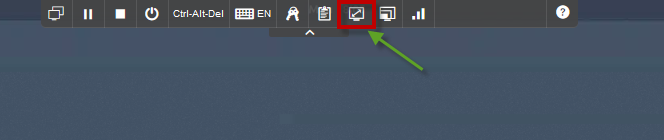
* 1. When prompted for the password for “ibmdemo” user, enter “passw0rd” as the password:

Password: passw0rd (lowercase with a zero instead of the o)



1. Resize the Skytap environment window for a larger viewing area while doing the lab.

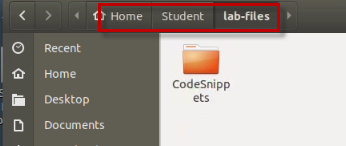
From the Skytap menu bar, click on the “**Fit to Size**”  icon. This will enlarge the viewing area to fit the size of your browser window.



|  |  |
| --- | --- |
| **sign-info** | **TIP:** Liberty is pre-installed on the VM environment provided. **{LAB\_HOME}** refers to: **/home/ibmdemo/Student/WLP\_21.0.0.3** |

**TIP:** To reduce typing or copy & past of commands, you can find the related code snippets or commands in the VMWare image in the directory:

**/home/ibmdemo/Student/lab-files/CodeSnippets/LibertyBootcamp\_Lab6\_LibertyMigrationToolkit\_CodeSnippets.txt**



|  |  |
| --- | --- |
| sign-info | **Important:**  **Click CANCEL**…. If, at any time during the lab, you get a pop-up asking to install updated software onto the Ubuntu VM.  The one we experience is an update available for VS Code.  **CLICK CANCEL!** |

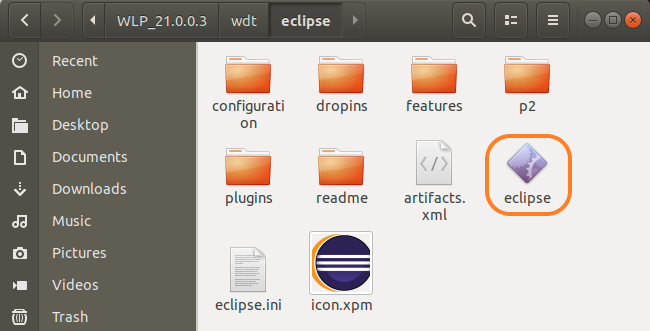
## Import the Sample Day Trader Application

To analyze an application for migration suitability, the application **source code** must be imported into your Eclipse-based IDE. If the application is not already in Eclipse, an easy way to import the application and organize it in projects that reflect their structure as EAR, WAR, and EJB files is by using the Eclipse import function as illustrated in the steps below.

1. If Eclipse is not already started, start it now
   1. Use the File Explorer to navigate to the directory

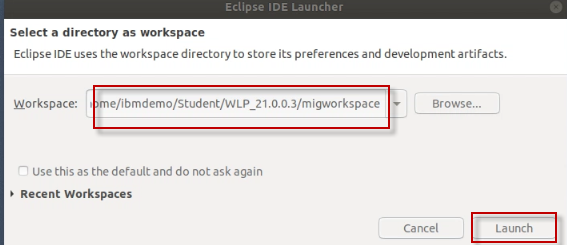
Home > Student > WLP\_21.0.0.3 > wdt > eclipse

* 1. Double-click on the **eclipse** executable to start Eclipse.

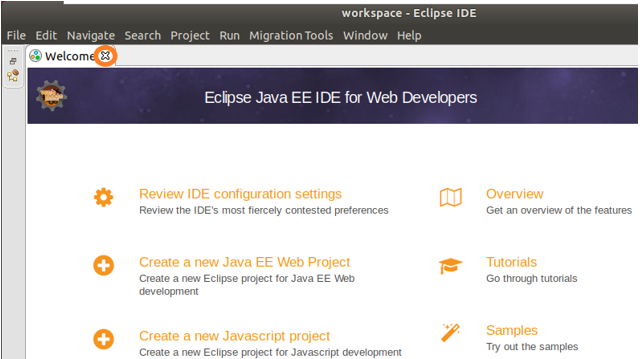


* 1. When the Eclipse launcher prompts you to select a workspace, enter the following directory. Then click the **Launch** button.

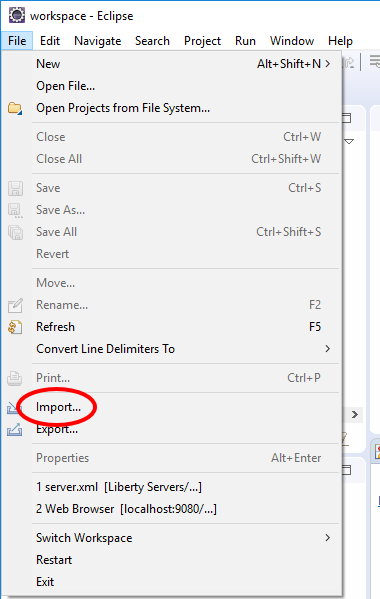
|  |
| --- |
| /home/ibmdemo/Student/WLP\_21.0.0.3/migworkspace |



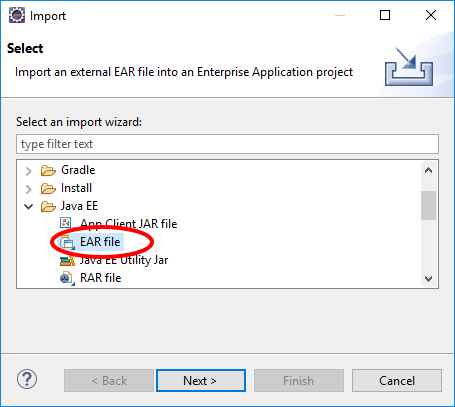
* 1. Close the **welcome page** by clicking on the **‘X**’ icon.



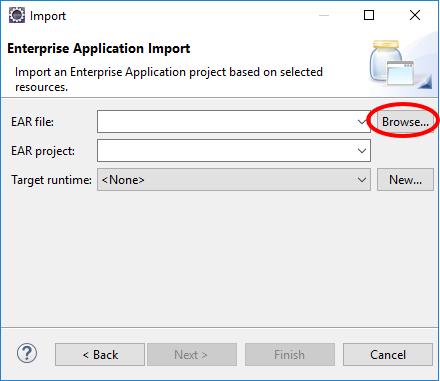
1. Import the sample application source code that is provided with the lab
   1. From the Eclipse main menu, Select **File > Import** menu option.



* 1. From the context menu. expand **Java EE**, select **EAR** File, then click **Next**



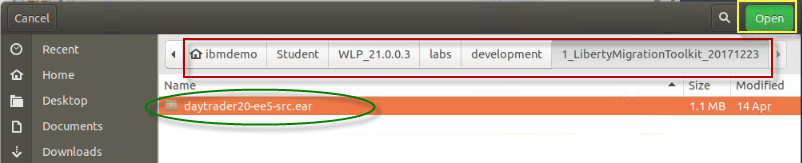
* 1. Click the **Browse** button to navigate to the location of the day trader sample application EAR file used in this lab



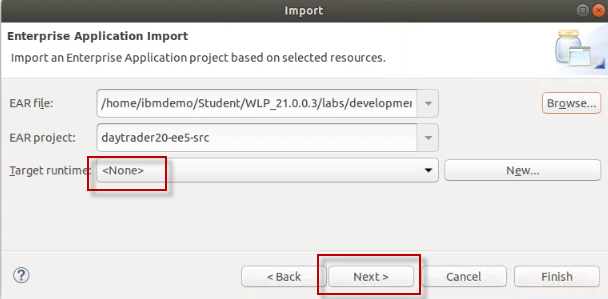
* 1. Navigate to the following directory:

/home/ibmdemo/Student/WLP\_21.0.0.3/labs/development/1\_LibertyMigrationToolkit\_20171223

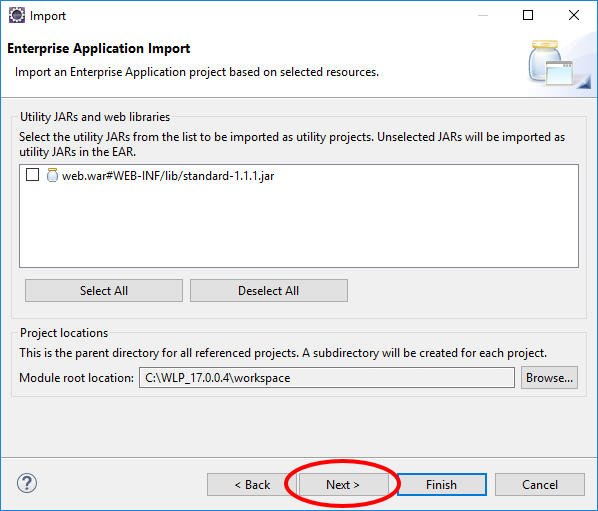
* 1. Select **daytrader20-ee5-src.ear**. Click **Open**



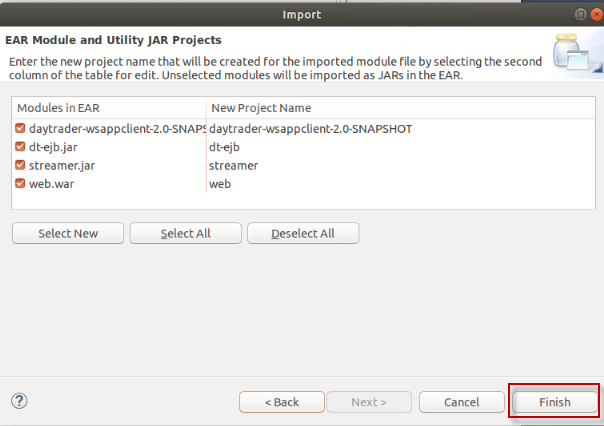
* 1. Set **Target runtime** to **None**. Click **Next**



* 1. On the “**Utility JARs and web libraries**” page, accept the default values, click **Next**



* 1. The next panel shows different additional projects that will be created. Later we will configure the migration toolkit to scan these projects. Click **Finish**.



The daytrader application will be imported into the Eclipse workspace.

* 1. Click **Yes,** if prompted to switch to the Java EE perspective

|  |  |
| --- | --- |
| **sign-info** | **IMPORTANT TIP:**  The daytrader application source code that was imported into the Eclipse workspace will show ERRORS in the projects.  That is OK and expected since the workspace has not been setup with the proper Java Class Paths for the application to successfully compile  The application does **NOT** need to compile successfully for the migration toolkit to scan the application and report possible application migration issues in the source code. |

## Scan the DayTrader Sample application

One the application is imported into the Eclipse workspace that has the migration toolkit plugin installed, you are ready to use the software analyzer to scan the application for application possible migration issues.

The daytrader application that you will scan is an application that currently runs on WebSphere Traditional Application Server, version 7 environment. The goal of this exercise is to determine what application code changes are necessary to migrate the daytrader application form WebSphere Traditional (Version 7) to Liberty.

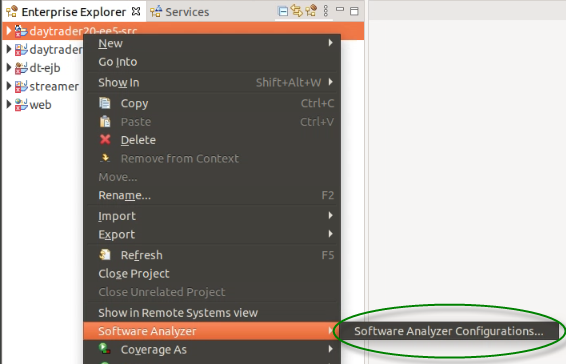
The migration tools take into consideration, the JEE version and Java version differences between the source and target runtime environments. It also considers differences between WebSphere versions and editions.

|  |  |
| --- | --- |
| 1. **sign-info** | 1. **Information:** 2. The Eclipse environment used for this lab has the Migration Tools Eclipse Plugin installed. The plugins are available from the Eclipse Marketplace, which can be accessed from within the Eclipse workspace via **Help > Eclipse** **Marketplace** option from the main menu. |

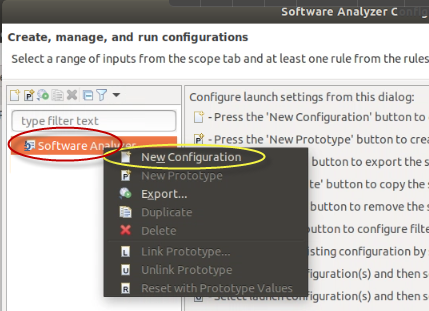
To begin, launch the **Software Analyzer** tool in Eclipse, and set the configuration options based on the migration source and target runtime environment preferences.

These preferences determine the **migration rules** that must be applied when the Software Analyzer scans the application for potential migration issues from the source to the target runtime environments.

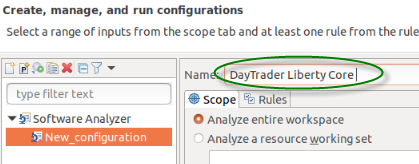
1. Launch the Software Analyzer tool: Right-click on the **daytrader20-ee5-src** sample application that you imported, and navigate to **Software Analyzer 🡪 Software Analyzer Configurations**



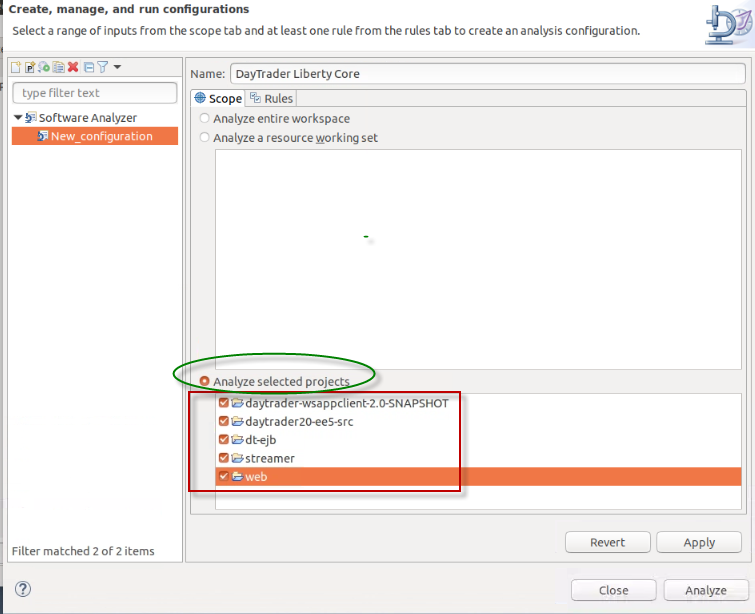
1. Create a new Software Analyzer configuration
   1. Right-mouse click on **Software Analyzer**, and select “**New Configuration**” form the context menu



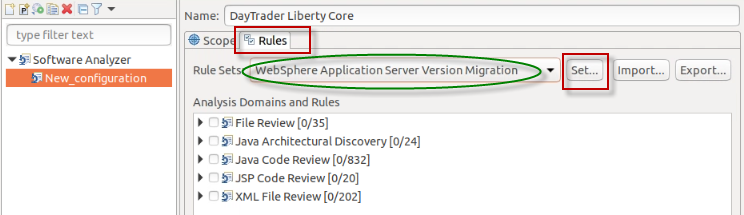
* 1. Enter **DayTrader Liberty Core** as the Name of the new configuration.



* 1. Click **Analyze selected projects** radio button. and choose all five of the Day Trader application related projects to be analyzed.

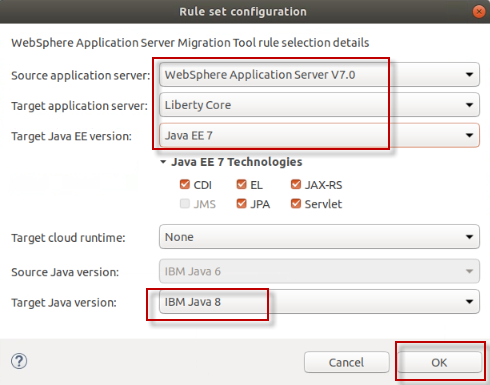


1. Set the **Rule Set** for the software analyzer configuration
   1. Select the **Rules** tab
   2. Select **WebSphere Application Server Version Migration** from the **Rule Sets** drop-down.
   3. Click the **Set** button



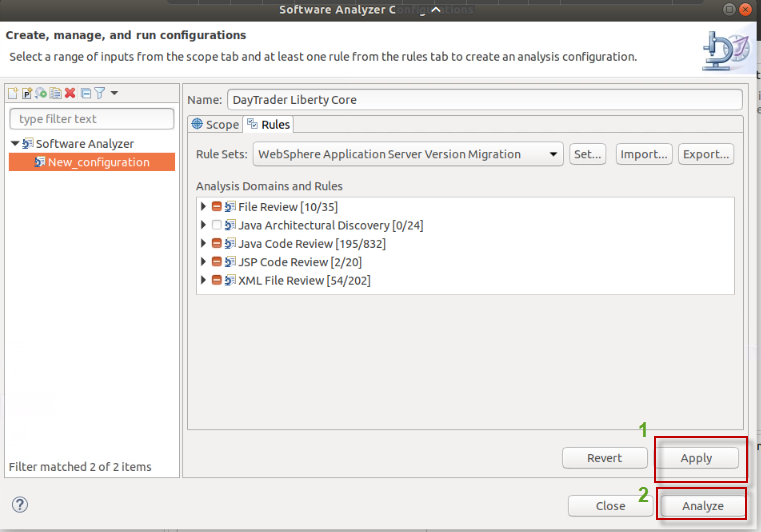
1. Complete the Rule Set configuration by selecting the source and target environment configurations
   1. Configure the rule set as shown below: Click **OK** button when complete

* Source Application server: **WebSphere Application Server V7.0**
* Target Application server: **Liberty Core**
* Target Java EE version: **Java EE 7**
* Target Java version: **IBM Java 8**



* 1. Click the **OK** button to continue

* 1. From the New Configuration page, Click **Apply** and then **Analyze**



At this point, the software analyzer scans the application projects. When complete, the results are displayed in the **Software Analyzer Results** tab at the bottom pf the Eclipse perspective.

### Review the software analysis results

In this section of the lab, you will review the results of the migration analysis.

The results are presented in groups, based on the types of rules that are applied. At the highest-level grouping, the analysis results to be reviewed are arranged by Files, JSP pages, Java Code, and XML files.

Under each of those hi-level groupings, further categorization of flagged rules is displayed based on the differences between versions of Java EE and Java between the Source and Target runtimes configured defined in the rule set.

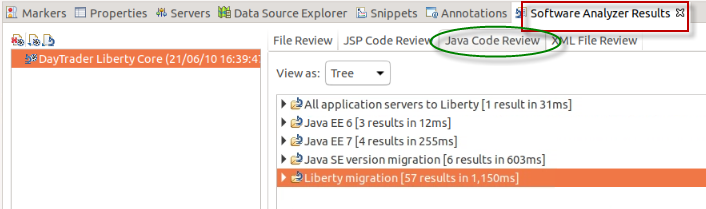
In the steps to follow, you will review the analysis results for the **Java Code** from the daytrader application projects. You will learn how the migration tools help the developer understand where the application may require changes during migration, based on the differences between the source and target environments, and how to get additional contextual help and insights to make the necessary code changes during the migration.

|  |  |
| --- | --- |
| sign-info | **TIP:**  Remember, in this case, the rules flagged are based on migrating the daytarder application from traditional WebSphere Application Server V7 to Liberty Core.  If you change the rule set to specify different source and target environments, the results will be different |

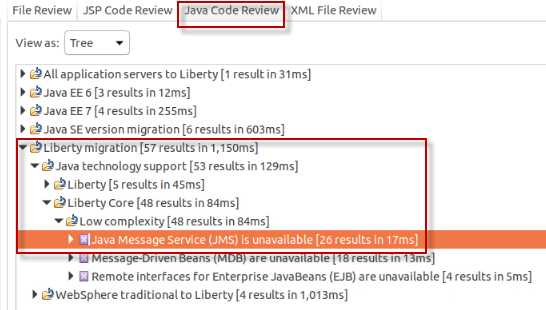
1. Locate the Java code review results
   1. Click **Software Analyzer Results** tab, located at the bottom of the Eclipse workspace.

On the left is the scan history. There is currently only one scan for **DayTrader Liberty Core**

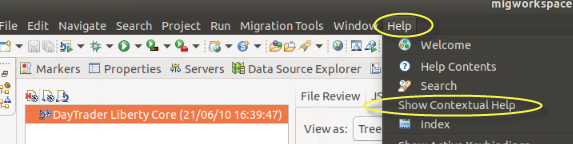
* 1. Click the **Java Code review** tab to inspect the results of scanning Java code



1. Choose one of the flagged items under the **Liberty Migration** section to view a specific issue
   1. Double click on the **Software Analyzer** tab to bring into full screen mode (for a larger viewing area)
   2. Click and expand the **Liberty migration > Java technology Report > Liberty Core > Low complexity > Java Message Service (JMS)**



1. Display the **contextual help** for the selected item
   1. From the main menu, click **Help > Show contextual help**
   2. Click on the **Java Message Service (JMS)** to highlight the line item.

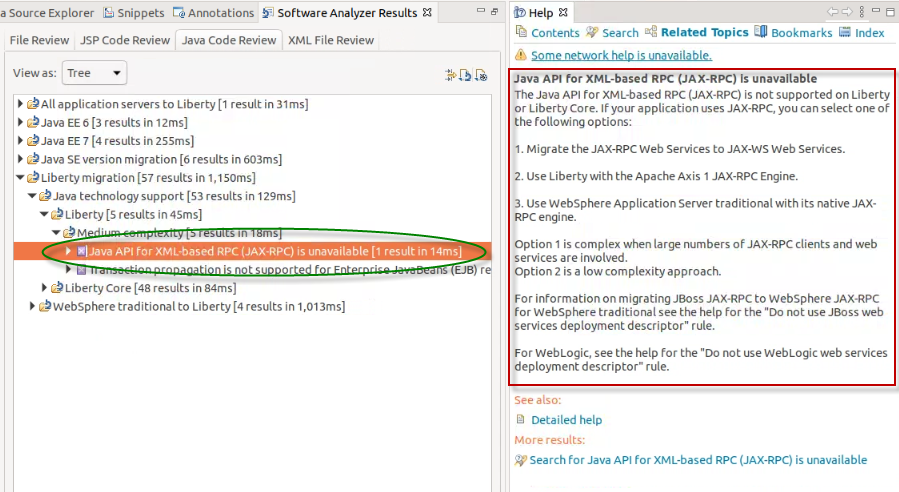


* 1. The **contextual help** is displayed for the rule that was flagged for this item.

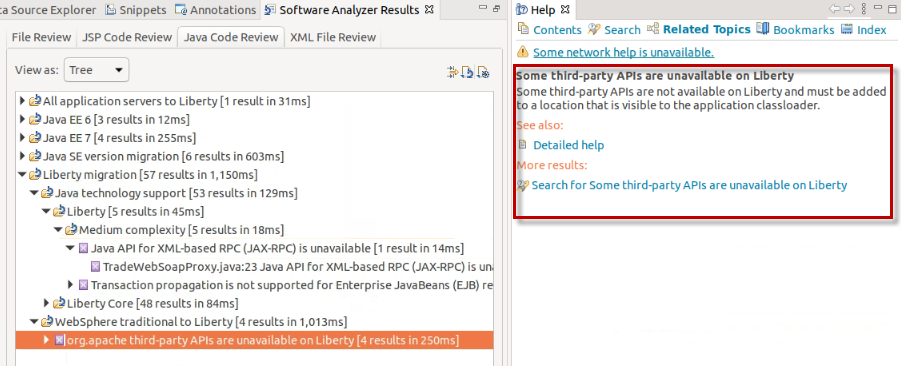
Note that the help message explains that even though JMS is not supported for Liberty core edition, it is available in other editions via other features. The same restriction applies to MDB and remote EJBs that are flagged.



1. Expand **Liberty** to examine additional scan results. You will see that the deprecated JAX-RPC API is not available in Liberty. In the HELP, Options are provided for the developer.
   1. Expand **Liberty migration > Java technology support > Liberty > Medium complexity**
   2. Under **Medium complexity**, click on **JAX-RPC** and read the **Help** text

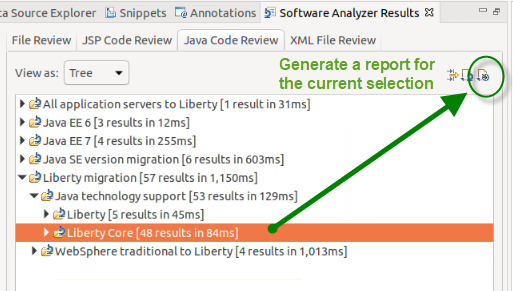


1. Let’s review a flagged issue under the category of **WebSphere traditional to Liberty**. These issues relate specifically to differences between WebSphere traditional and Liberty. The previous issues were related to differences in Java technologies.
   1. Under **WebSphere traditional to Liberty** section, examine which third party APIs are not available in Liberty.

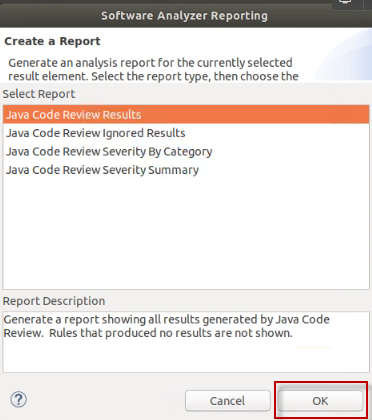


In addition to reviewing the migration analysis using the Eclipse IDE, you can also generate reports that can be shared with your development team.

1. Generate a report for the Liberty Core section of the migration analysis
   1. Select **Liberty Core** and click the **Generate Report** button

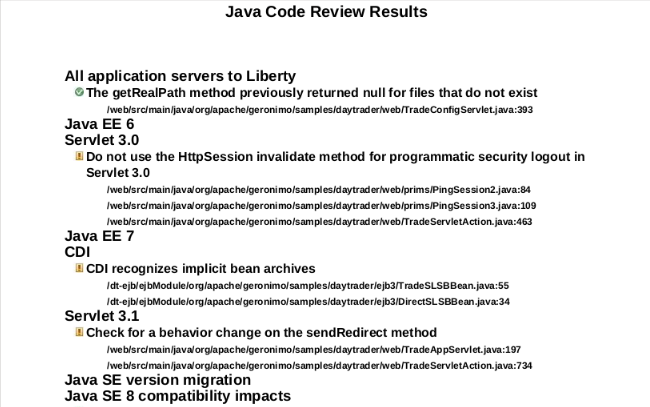


* 1. Select **Java Code Review Results** from the Report dialog box. Click the **OK** button



The result will appear in a different pane and will look like the one below:

* 1. **Close** the PDF report when you finish reviewing it



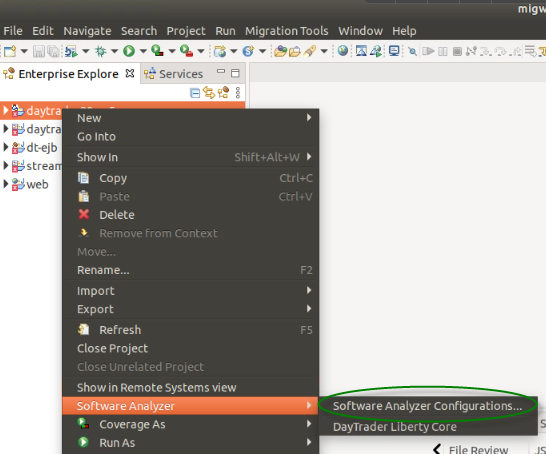
## Create Scan Report for WebSphere to Liberty

The scan report and analysis in the previous section was a result of the migration rule set from WebSphere Traditional V7 to Liberty Core.

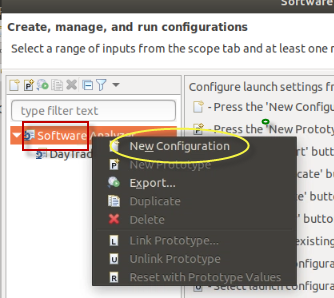
Because Liberty Core does NOT support the full range of Java EE API specifications, in which WebSphere Traditional does, you found there were migration issues that were flagged because the APIs are not available in Liberty Core. However, Liberty (or Liberty Base) does support the full set of Java EE APIs.

So, lets configure a migration scan that focuses on migration from WebSphere Traditional V7 to Liberty and review the results.

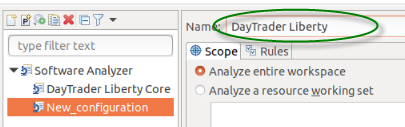
1. If the Software Analyzer Results is still in **full screen** mode, **double click** on it to revert to the normal view, and you can see the daytrader projects in the Enterprise Explorer view.
2. Setup a new **Software Analyzer configuration** for the new migration analysis.
   1. Right click on any **daytrader2—ee5-src** project and navigate to **Software Analyzer** 🡪 **Software Analyzer Configurations…**



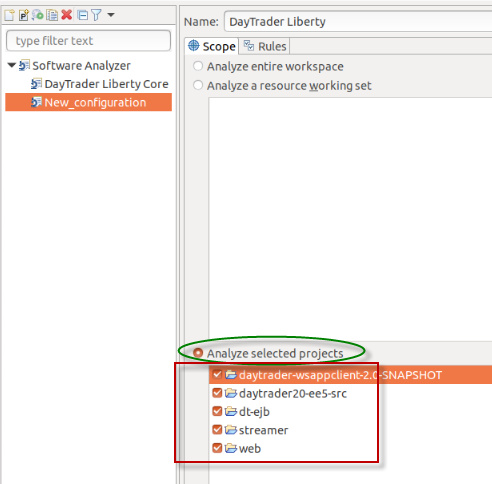
* 1. Right-mouse click on **Software Analyzer**, and select “**New Configuration**” form the context menu



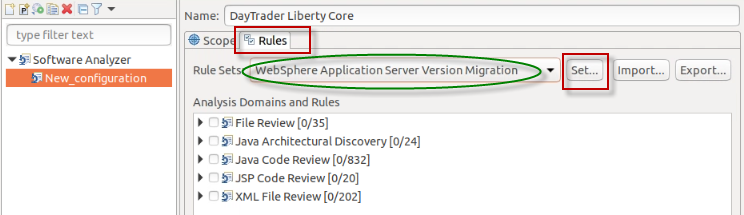
* 1. Enter **DayTrader Liberty** as the Name of the new configuration.



* 1. Click **Analyze selected projects** radio button. and choose all five of the Day Trader application related projects to be analyzed.

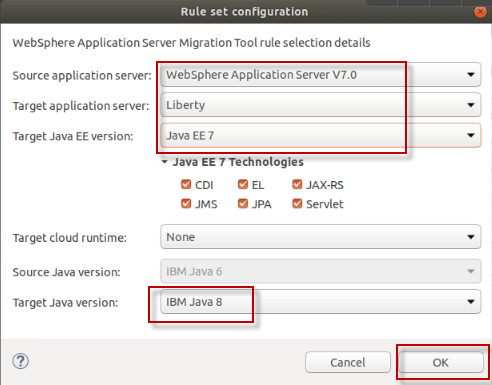


1. Set the **Rule Set** for the software analyzer configuration
   1. Select the **Rules** tab
   2. Select **WebSphere Application Server Version Migration** from the **Rule Sets** drop-down.
   3. Click the **Set** button



1. Complete the Rule Set configuration by selecting the source and target environment configurations
   1. Configure the rule set as shown below: Click **OK** button when complete

* Source Application server: **WebSphere Application Server V7.0**
* Target Application server: **Liberty**
* Target Java EE version: **Java EE 7**
* Target Java version: **IBM Java 8**



* 1. Click the **OK** button to continue

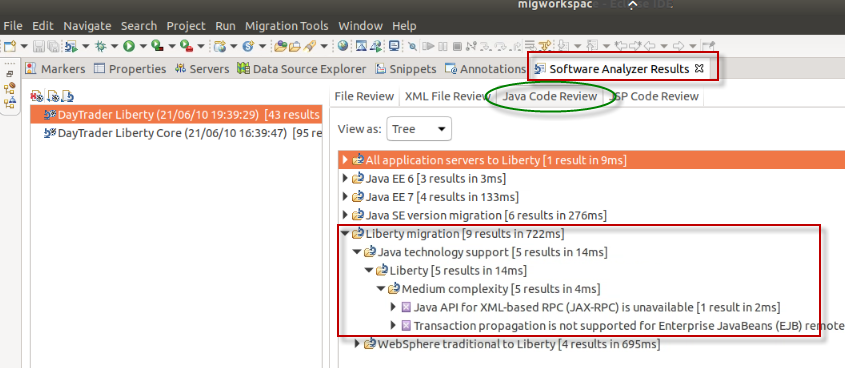
* 1. From the New Configuration page, Click **Apply** and then **Analyze**



At this point, the software analyzer scans the application projects. When complete, the results are displayed in the **Software Analyzer Results** tab at the bottom pf the Eclipse perspective.

1. Note there are now two runs in the scan history pane. The most recent is the “**DayTrader Liberty**” scan.
   1. Select the **DayTrader Liberty**” scan
   2. Select the **Java Code Review** tab
   3. Drill down to **Liberty migration > Java Technology support > Liberty**

Note that compared to the previous run, **no JMS**, **MDB, or EJB** issues are reported, because they only apply to the **Liberty Core Edition. Liberty (base) edition supports these Java EE APIs.**



1. **Exit** Eclipse

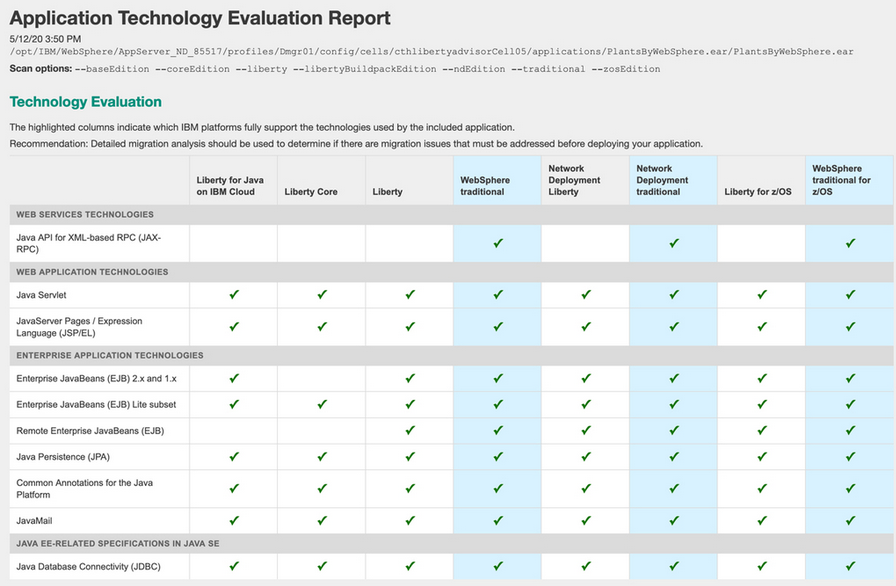
## Migration Toolkit for Application Binaries

The Migration Toolkit for Application Binaries provides a command line tool that quickly evaluates application binaries for rapid deployment on newer versions of WebSphere Application Server traditional or Liberty.

This command-line tool enables administrators and developers to evaluate, inventory and analyze applications in minutes without accessing the source code. The tool can generate individual reports or a consolidated application migration report containing the following reports:

* Application Evaluation Report
* Application Inventory Report
* Detailed Migration Analysis Report
* Liberty Configuration

The tool can either display an HTML report or save the report as an HTML or JSON file.



Learn more and see samples of the reports here: <https://www.ibm.com/support/pages/node/318851#binaryscanner>

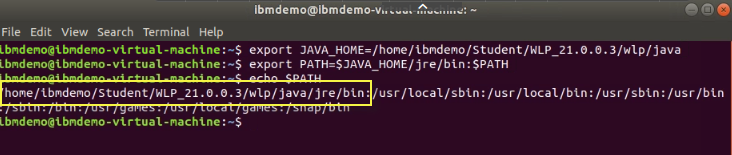
The migration tools and Binary scanner can be downloaded from here: <https://www.ibm.com/support/pages/websphere-migration-knowledge-collection-downloads>

.

The **Migration Toolkit for Application Binaries** has already been downloaded on the VM used for the lab. Installing the Binary Scanner is simply a matter of extracting the Java archive file. You will do that in the steps below.

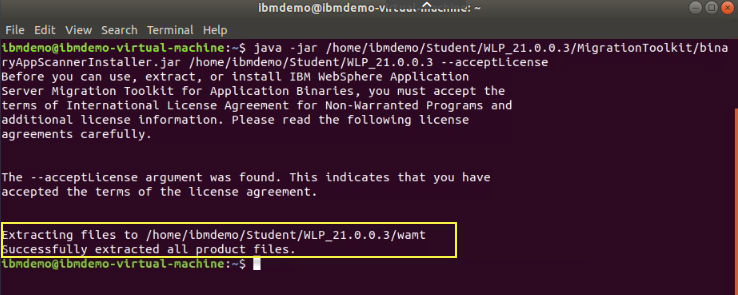
1. Install the migration toolkit for application binaries.
   1. Open a Terminal window
   2. Set the JAVA\_HOME variable and add it to the path

|  |
| --- |
| export JAVA\_HOME=/home/ibmdemo/Student/WLP\_21.0.0.3/wlp/java  export PATH=$JAVA\_HOME/jre/bin:$PATH  echo $PATH |



1. Install the binary scanner from the java archive provided

|  |
| --- |
| java -jar /home/ibmdemo/Student/WLP\_21.0.0.3/MigrationToolkit/binaryAppScannerInstaller.jar /home/ibmdemo/Student/WLP\_21.0.0.3 --acceptLicense |

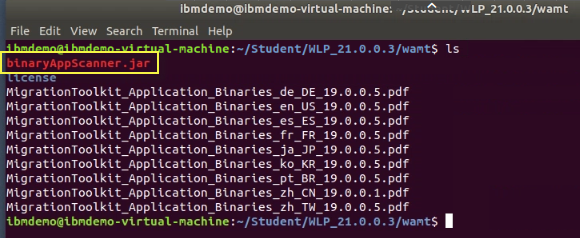


This will create the “**wamt**” directory under a directory under {LAB\_HOME.

1. View the newly created directory.

|  |
| --- |
| cd /home/ibmdemo/Student/WLP\_21.0.0.3/wamt  ls |

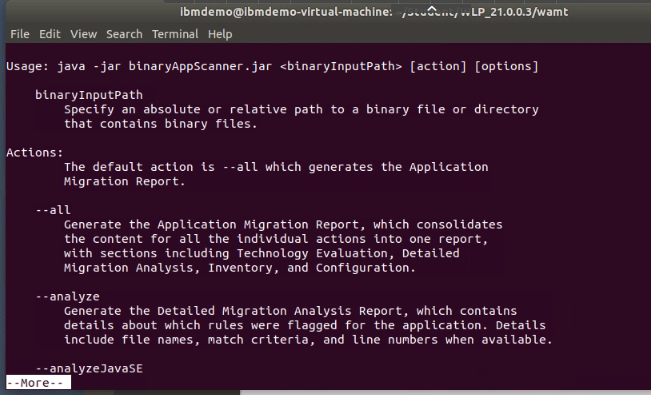
The executable is “**binaryAppScanner.jar**”. The PDF files is the documentation, with examples, in several languages.



1. To see a summary of the available command line options, run the binary scanner with the **--help** option.

|  |
| --- |
| java -jar ./binaryAppScanner.jar --help | more |

* 1. Use the “**Space bar**” to scroll through the output from the help, as it was piped to the “**more**” command.



|  |
| --- |
| **Notice**   * It is a best practice to identify your custom application class packages with the  --includePackages option. By doing so, you avoid scanning Java EE and third-party packages which should not affect your migration effort. If no --includePackages or --excludePackages options are explicitly specified, the tool excludes Java EE and some third-party packages by default. These packages are identified as scan options near the beginning of the report. |

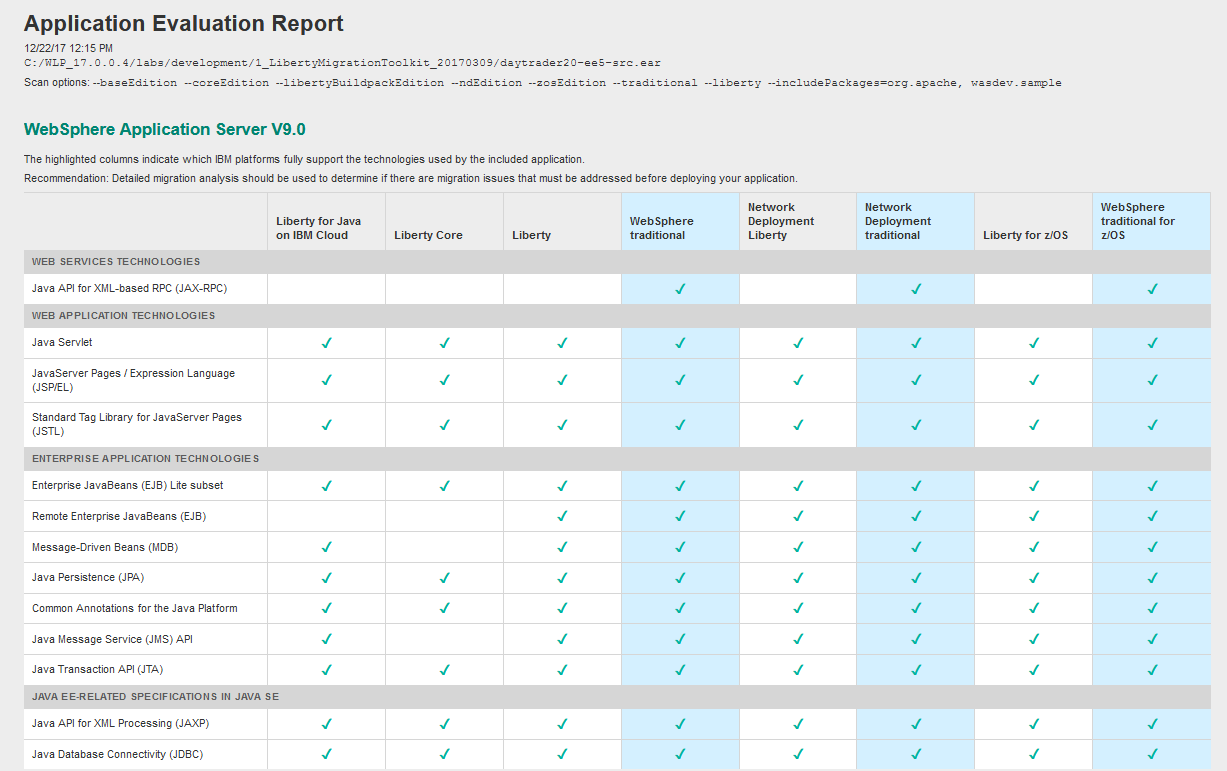
In the next step, you will use the binary scanner to scan the same **daytrader** application that was used earlier in the lab. With the binary scanner’s “**technology report**”, you will easily see which Java EE APIS are used in the application and supported by the various editions of WebSphere and Liberty.

1. Run the command to generate an “**Evaluation Technology report**” for the daytrader application

|  |
| --- |
| java -jar ./binaryAppScanner.jar /home/ibmdemo/Student/WLP\_21.0.0.3/labs/development/1\_LibertyMigrationToolkit\_20171223/daytrader20-ee5-src.ear --evaluate --includePackages=org.apache,wasdev.sample |

A browser window will open showing a report about the programming models used by the application and where they will run.

Review the report to see what programming models are used in the day trader application. Note the Java EE API used in the DayTrader app that are NOT supported in Liberty and/or Liberty Core. Note that the APIs are supported in the Traditional WebSphere editions.

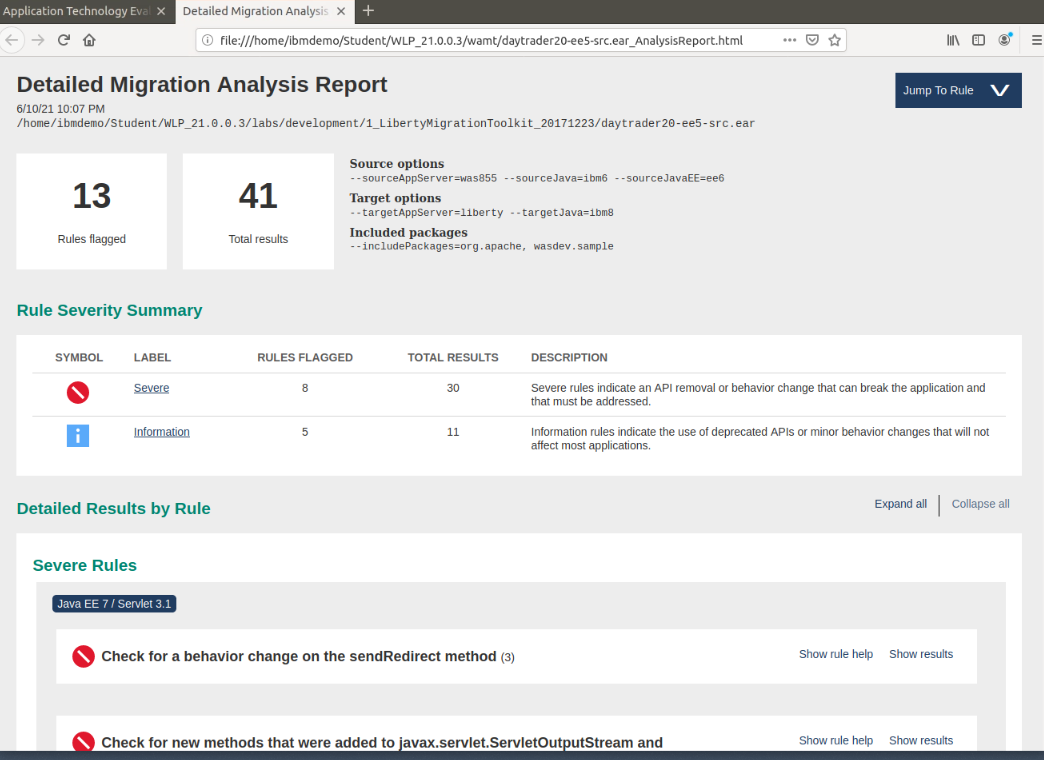


1. Re-run the command, but this time with **--analyze** option.

Specifying the **“--analyze**” option on the AppScanner generates the **Detailed Migration Analysis Report**, which contains details about which rules were flagged for the application. Details include file names, match criteria, and line numbers when available.

|  |
| --- |
| java -jar ./binaryAppScanner.jar /home/ibmdemo/Student/WLP\_21.0.0.3/labs/development/1\_LibertyMigrationToolkit\_20171223/daytrader20-ee5-src.ear --analyze --includePackages=org.apache,wasdev.sample |

* 1. A browser window will open showing detailed information about which code may have to be changed: Review the report



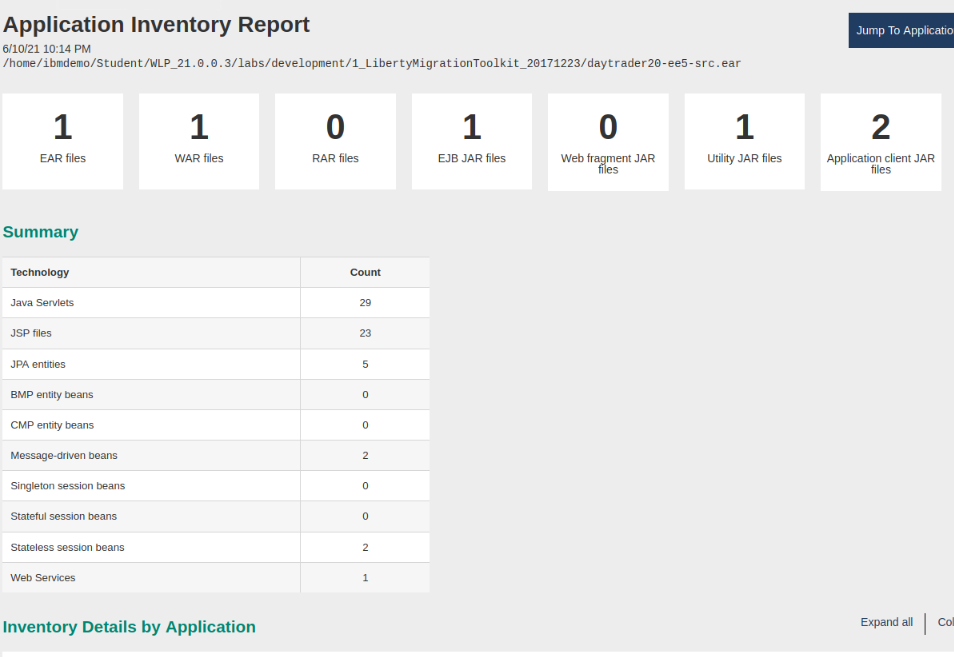
1. Re-run the command, but this time with **--inventory** option

Specifying the **“--inventory**” option on the AppScanner generates the **Application Inventory Report**, which contains a high-level inventory of the content and structure of each application.

Potential deployment problems and performance considerations are also included.

|  |
| --- |
| java -jar ./binaryAppScanner.jar /home/ibmdemo/Student/WLP\_21.0.0.3/labs/development/1\_LibertyMigrationToolkit\_20171223/daytrader20-ee5-src.ear --inventory --includePackages=org.apache,wasdev.sample |

* 1. A browser window will open showing detailed information about the structure of your application, providing information that helps understand the relative complexity of the application being considered for migration to Liberty.



## Summary

In this lab you have learned:

* How to install and use Liberty Migration Toolkit on Eclipse
* How to install and use Migration Toolkit for Application Binaries

|  |  |
| --- | --- |
| **sign-info** | **TIP:**  The Migration Toolkit for Application Binaries is the technology that underpins the reports in the **IBM Transformation Advisor** Tool that is included in the WebSphere Hybrid Edition product.  Refer to the link below to learn more about Transformation Advisor and how accelerates Java application Modernization.  <https://www.ibm.com/garage/method/practices/learn/ibm-transformation-advisor> |

**=== END OF LAB ===**